Homework 4

Due: May 8, 2024 (No Extensions)

This homework covers use of Pandas and Geopandas for analysis of motor vehicle accidents in NYC, and use of Numpy, Pandas, and user-defined code for problems requiring interpolation, least squares, and integration using Trapezoid and Simpson's Rule, Gauss Quadrature and Romberg Integration.

Question 1: 20 points. Motor vehicle accidents in New York City are the leading cause of death for the city residents. Statistics indicate that close to one in four accidents results in someone being injured or losing their life. The problem is particularly acute for accidents involving high-speed, and/or when accidents involve pedestrians, bicyclists, or motor cycles. Root causes for this situation can be traced back to the city being flat and very walkable, as well as a significant biking culture.

The purpose of this question is to take a first step toward formally analyzing data on motor vehicle accidents and, specifically, understand where and when vehicle accidents occur? We will not investigate the particulars of who has been injured and associated details on the vehicles involved.

Accident Data: The folder `python-code.d/data/cities/nyc/` contains data files that can be used in the analysis of motor vehicle accidents in NYC. The main file, Motor-Vehicle-Collisions.csv, comprises 2.06 million motor vehicle accidents recorded across the five boroughs of NYC and for about a decade.

The data is organized into 29 columns:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2062758 entries, 0 to 2062757
Data columns (total 29 columns):
 #   Column                          Dtype
---  ------                          -----
 0   CRASH DATE                      object
 1   CRASH TIME                      object
 2   BOROUGH                         object
 3   ZIP CODE                        object
 4   LATITUDE                        float64
 5   LONGITUDE                       float64
 6   LOCATION                        object
 7   ON STREET NAME                  object
 8   CROSS STREET NAME               object
 9   OFF STREET NAME                 object
 10  NUMBER OF PERSONS INJURED       float64
 11  NUMBER OF PERSONS KILLED        float64
 12  NUMBER OF PEDESTRIANS INJURED   int64
 13  NUMBER OF PEDESTRIANS KILLED    int64
```

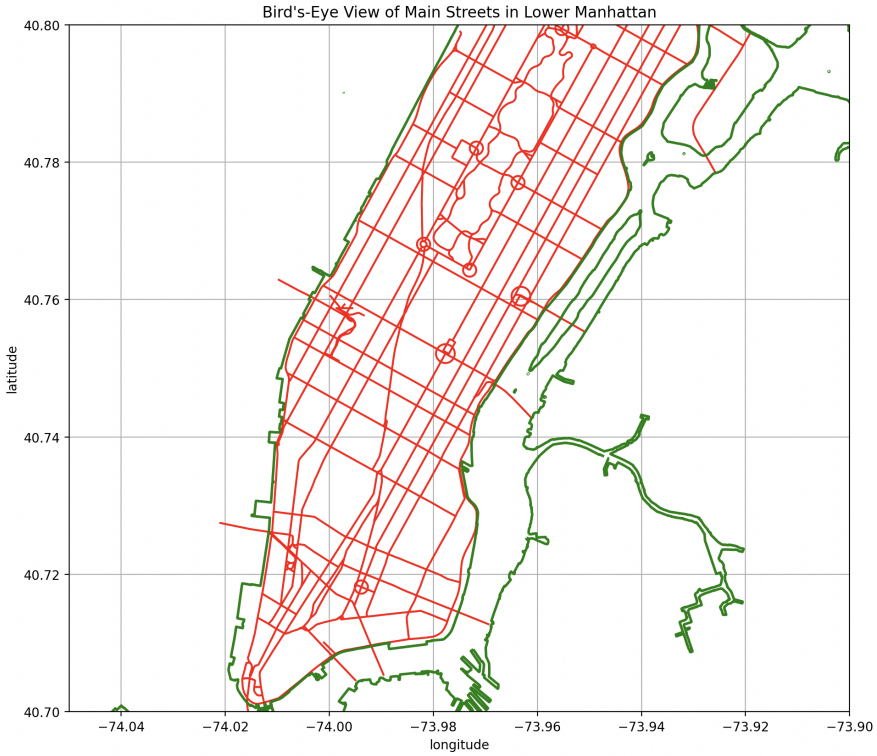

Figure 1: Spatial view: Main streets in Lower Manhattan.

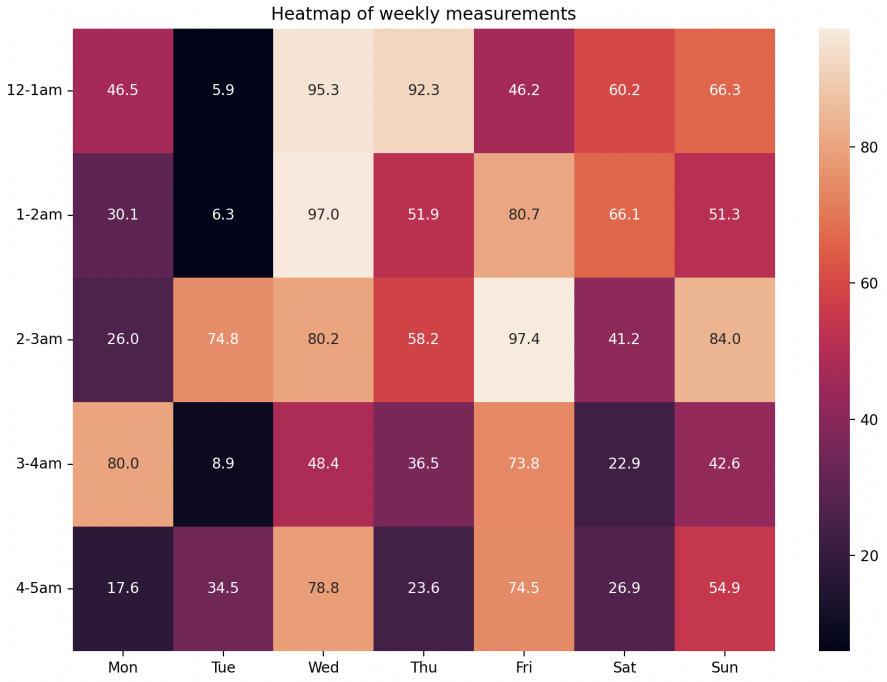

Figure 2: Temporal view: Heatmap of vehicle accidents organized into weekly measurements.

```
 14  NUMBER OF CYCLIST INJURED      int64
 15  NUMBER OF CYCLIST KILLED       int64
 16  NUMBER OF MOTORIST INJURED     int64
 17  NUMBER OF MOTORIST KILLED      int64
 18  CONTRIBUTING FACTOR VEHICLE 1  object
 19  CONTRIBUTING FACTOR VEHICLE 2  object
 20  CONTRIBUTING FACTOR VEHICLE 3  object
 21  CONTRIBUTING FACTOR VEHICLE 4  object
 22  CONTRIBUTING FACTOR VEHICLE 5  object
 23  COLLISION_ID                   int64
 24  VEHICLE TYPE CODE 1            object
 25  VEHICLE TYPE CODE 2            object
 26  VEHICLE TYPE CODE 3            object
 27  VEHICLE TYPE CODE 4            object
 28  VEHICLE TYPE CODE 5            object
dtypes: float64(4), int64(7), object(18)
memory usage: 456.4+ MB
None
(2062758, 29)
```

Geospatial Data: The data files `dcm-nyc-major-street.csv` and `nyc-shoreline.csv` contain geospatial data on the main streets and shoreline in the NYC area.

Things to do: Write a Python program that will read the motor vehicle accidents and geospatial data files, and systematically filter and transform the data into spatial and temporal views of accident events in Lower Manhattan. Figure 1 shows a spatial view of streets in Lower Manhattan. Figure 2 shows a heatmap/temporal view of accidents, organized along the dimensions of time-of-the-day and day of the week. Thus, for this question, columns 0 – 5 of the accident data are most relevant.

For the spatial view:

1. Filter the accident data to only keep accidents occuring in Manhattan – this operation will reduce the number of accidents from 2 million to approximately 318,000. Then, remove from consideration accidents that do not have a (lat,long) coordinates – there are about 10,000 of them.

1. Add locations of remaining accidents to Figure 1. Small blue dots might work, but there are over 300,000 of them ...

For the temporal view:

2. For each accident date, extract the day of the week and time of day. Map this data to rows and columns in the heatmap/temporal view, then display. Again, there are over 300,000 data points, so a good strategy might be to display the data as percentages?

Question 2: 20 points. Figure 3 is a three-dimensional view of a 2 by 2 km site that is believed to overlay a thick layer of mineral deposits.

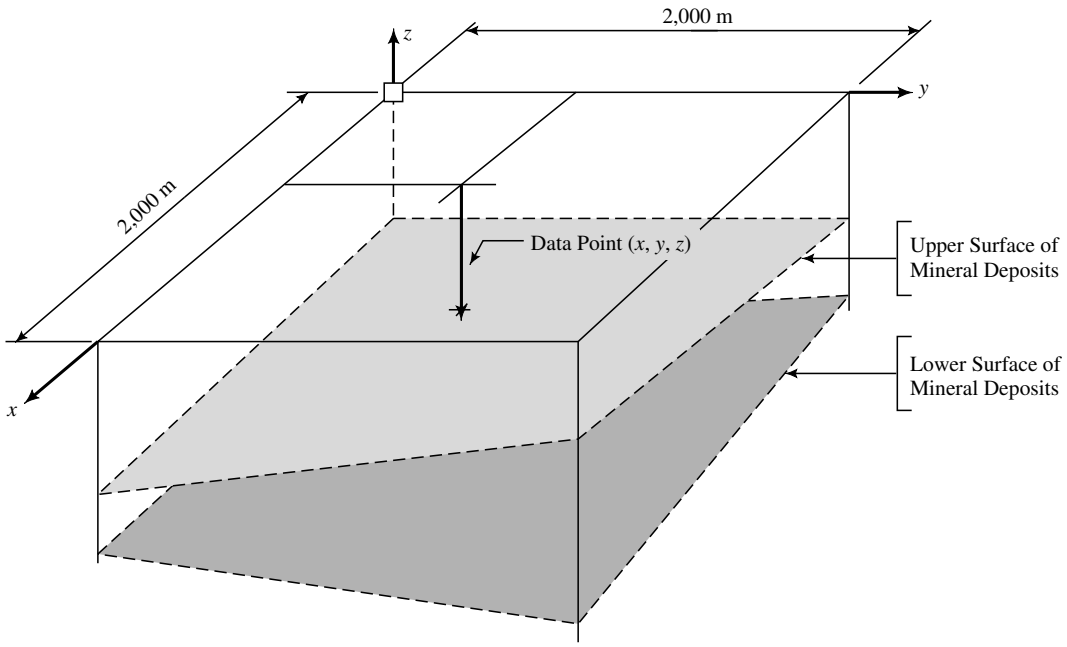

Figure 3: Three-dimensional view of mineral deposits.

To create a model of the mineral deposit profile and establish the economic viability of mining the site, a preliminary subsurface exploration consisting of 16 bore holes is conducted. Each bore hole is drilled to approximately 45 m, with the upper and lower boundaries of mineral deposits being recorded. The bore hole data is as follows:

Borehole	X (m)	Y(m)	Upper Surface (m)	Lower Surface (m)
1	10.0	10.0	-28.5	-42.5
2	750.0	10.0	-27.0	-41.8
3	1250.0	10.0	-26.0	-41.3
4	1990.0	10.0	-24.6	-40.5
5	10.0	750.0	-32.2	-43.4
6	750.0	750.0	-30.8	-42.6
7	1250.0	750.0	-29.8	-42.1
8	1990.0	750.0	-28.3	-41.4
9	10.0	1250.0	-34.7	-44.0
10	750.0	1250.0	-33.2	-43.2
11	1250.0	1250.0	-32.2	-42.7
12	1990.0	1250.0	-30.8	-42.0
13	10.0	1990.0	-38.4	-44.8
14	750.0	1990.0	-37.0	-44.1
15	1250.0	1990.0	-36.0	-43.6
16	1990.0	1990.0	-34.5	-43.9

With the bore hole data collected, the next step is to create a simplified three-dimensional computer model of the site and subsurface mineral deposits. The mineral deposits will be modeled as a single six-sided object. The four vertical sides are simply defined by the boundaries of the site. The upper and lower sides are to be defined by a three-dimensional plane

$$z(x, y) = a_o + a_1 \cdot x + a_2 \cdot y \tag{1}$$

where coefficients a_o, a_1, and a_2 correspond to minimum values of

$$S\left(a_o,\ a_1,\ a_2\right) = \sum_{i=1}^{N} \left[z_i - z(x_i, y_i)\right]^2 \tag{2}$$

Things to do:

1. Show that minimum value of $S(a_o, a_1, a_2)$ corresponds to the solution of the matrix equations

$$\begin{bmatrix} N & \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \cdot y_i \\ \sum_{i=1}^{N} y_i & \sum_{i=1}^{N} x_i \cdot y_i & \sum_{i=1}^{N} y_i^2 \end{bmatrix} \cdot \begin{bmatrix} a_o \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} z_i \\ \sum_{i=1}^{N} x_i \cdot z_i \\ \sum_{i=1}^{N} y_i \cdot z_i \end{bmatrix} \tag{3}$$

2. Create a comma-separated datafile (e.g., borehole-data.csv) for the geological surface data.
3. Write a Python program to read the borehole csv datafile, and then create a three-dimensional plot of the geological borehole data at the lower and upper surfaces.
4. Set up and solve the matrix equations derived in part 1 for the upper and lower mineral planes. Compute and print the average depth and volume of mineral deposits enclosed within the site.

Note. The least squares solution corresponds to the minimum value of function $S(a_o, a_1, a_2)$. At the minimum function value, we will have

$$\frac{\partial S}{\partial a_o} = \frac{\partial S}{\partial a_1} = \frac{\partial S}{\partial a_2} = 0 \tag{4}$$

Matrix Equation 3 is simply the three equations 4 written in matrix form. You should find that the equation of the upper surface is close to `z(x,y) = -28.5 + x/500 - y/200` and the lower surface close is to `z(x,y) = -42.5 + x/1000 - y/850`.

Question 3: 10 points. It is well known that first derivative of $f(x) = sin(x)$ is $cos(x)$. Given the double angle formulae,

$$sin(a+b) = sin(a)cos(b) + cos(a)sin(b) \tag{5}$$

and

$$sin(a-b) = sin(a)cos(b) - cos(a)sin(b) \tag{6}$$

write a Python program that will estimate forward and central finite difference approximations. For each approximation, plot the error in the derivative estimate versus h about the point x=0. Plots covering the interval h = -$\pi/4$ to h = $\pi/4$ might be reasonable. Repeat for x=$\pi/2$.

Question 4: 10 points. Write a Python program that uses the methods of divided differences and Lagrange interpolation to fit the data set:

x	-1	2	4	5	6
f(x)	2	7	10	3	4

Use the Lagrange interpolation formula to approximate the functional value at x = 3.0. Create plots of the dataset coordinates and interpolated polynomial curves.

Question 5: 10 points. Consider the integral

$$I = \int_0^{10} 3x^2 + 4x^3 + 5x^4 dx = 111,000. \tag{7}$$

Write a Python program to compute numerical approximations to equation 7 using: (1) the Trapezoid rule, (2) Simpson's rule, and (3) two-point Gauss Quadrature. For cases 1 and 2, use only three data ordinates. Compute and print the absolute and relative errors for each numerical procedure.

Question 6: 10 points. Write a Python program that uses Romberg Integration to show:

$$\int_0^1 \left[\frac{1+x^2}{1+x^4}\right] dx = \frac{\pi \cdot \sqrt{2}}{4}. \tag{8}$$

Start off by evaluating the function at 0, $\frac{1}{4}$, $\frac{1}{2}$, $\frac{3}{4}$, and 1. Compute and print the absolute and relative errors.