ENCE 201 Engineering Information Processing, Spring Semester, 2024

Homework 2

(Due: March 8, 2024)

The purposes of this homework are to give you experience in working with Pandas and Geopandas, and Python's builtin functions. Submit to gradescope a pdf writeup containing the problem description for each problem, your python code, and approprite textual and graphical output.

Question 1: 10 points. An experiment is conducted on 36 specimens to determine the tensile yield strength of `A36` steel. The experimental results are as follows:

```
42.3      42.1      41.8      42.4      47.7      41.4
40.5      38.7      40.8      39.6      42.4      37.5
39.9      45.3      41.6      36.8      45.4      44.8
39.2      40.7      38.5      40.1      42.8      42.5
43.1      36.2      46.2      41.5      38.3      40.2
41.9      40.4      39.1      38.6      46.3      39.5
```

Write a Python program that will:

1. Read the experimental test results from a file `steelA36.csv` into a Pandas dataframe.
2. Compute and print the maximum, minimum, and average tensile strengths.
3. Use the Pandas cut function (i.e., google `pd.cut()`) to organize the data into intervals covering the ranges: 36-38, 38-40, 40-42, 42-44, 44-46, 46-48.
4. Generate a histogram of "observations" versus "tensile yield stress."
5. Construct a stair-step graph of "cumulative frequency" versus "yield stress."

Note. The average value of the experimental results can be computed using Python's buildin functions. The "cumulative frequency" versus "yield stress" is given by

$$\text{Cumulative frequency(y)} = \int_0^y p(x)dx \tag{1}$$

where $p(x)$ is the probability distribution of tensile yield strengths. The matplotlib functions `plt.hist(.)` and `plt.step(.)` create histogram and stair-step graphs.

Question 2: 10 points. Let dx be a floating point number whose magnitude is very small compared to 1. Write a Python program that will systematically evaluate the expression

$$f(dx) = \sqrt{1+dx} \; - \; \sqrt{1-dx} \tag{2}$$

for $|dx| \rightarrow 0$. Demonstrate that errors due to subtractive cancellation can be avoided by rewriting equation 2 as

$$g(dx) = \left[\frac{2dx}{\sqrt{1+dx} + \sqrt{1-dx}} \right]. \tag{3}$$

You should use math function `math.sqrt()` for the square root evaluations.

Note. I suggest that you set dx = 1, and then evaluate equations 2 and 3. Then, decrease dx by a factor of 10 and repeat the experiment. You should find that equations 2 and 3 will evaluate to the same value until the limits of double precision floating point storage are reached. And then, whereas evaluation of equation 2 will truncate to zero, equation 3 will evaluate to dx. It is relatively straight forward to show via a Taylor series expansion that dx is correct.

Question 3: 10 points. Figure 1 is a schematic of public-use airports located in Maryland.

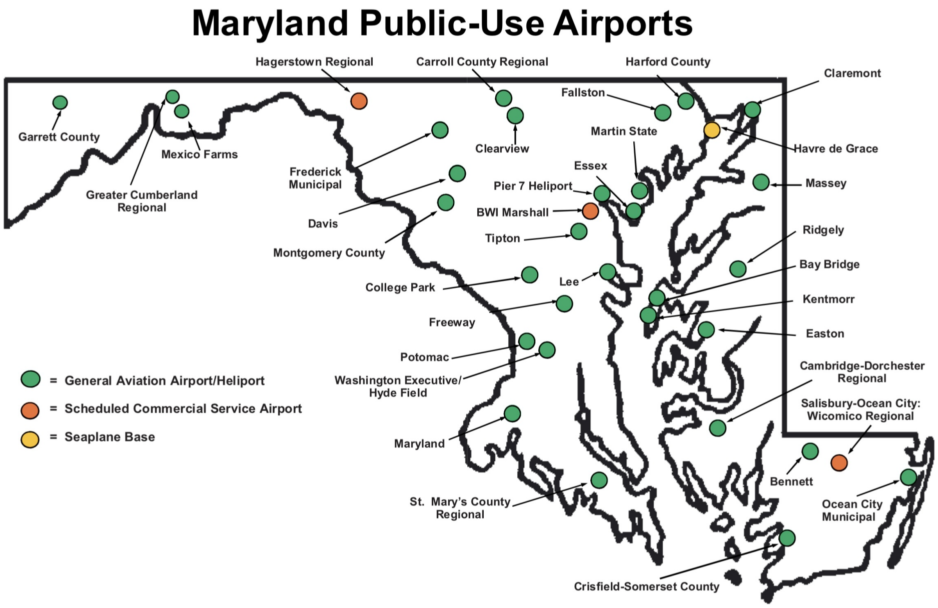

Figure 1: Maryland public-use airports (source: MD Airport Administration).

Write a Python program to assemble a **similar graphic** by pulling together – reading, filtering – and visualizing data sources for: (1) airports in the US, (2) boundary data for the state of MD, and (3) boundary data for the coastline.

You should read and store the airport data in a Pandas dataframe; then, employ a filter to isolate airports located in Maryland. You should use Pandas to store the airport data. The MD boundary and coastline data can be read directly into GeoPandas.

Note: The data file `python-code.d/data/airports-small.csv` contains a listing of 3,300 airports in the US. See `python-code.d/data/geography/maryland/` for boundary data for MD and the Chesapeake Bay. Finally, see the test examples in `python-code.d/geopandas/` for guidance on structuring your Python code.

Question 4: 10 points. On April 12, 1912, the RMS Titanic (a mail and passenger vessel) commenced her maiden voyage across the Atlantic, departing from Ireland and headed to New York. Three days later (April 15) the Titanic struck an iceberg and sank. Of the 2,224 passengers and crew aboard, approximately 1,500 died, making it the deadliest sinking of a ship at that time.

See Wikipedia for a nice writeup on the Titanic and the numerous deficiencies that lead to this disaster. In more recent times, the sinking of the Titanic has inspired numerous artistic works, including the 1997 romantic disaster film Titanic (directed by James Cameron). Like many movies, Titanic places an emphasis on romance and drama – a viewer might wonder, how much of this story is true? Who survived, and why? What does the data say?

Problem Statement. The data file `python-code.d/data/titanic.csv` contains information on 887 of the passengers and their attributes, including:

```
--- Survived: 1 means passenger survived; 0 for victims.
--- Pclass:   1, 2 and 3 for first, second and third class.
--- Name:     Master/miss first name, family name.
--- Sex:      male or female.
--- Age:      covers the range 0 to 80.
--- Siblings/Spouses Aboard
--- Parents/Children Aboard
--- Fare:     First class (1) tickets are the most expensive.
```

Write a Python program that will read `titanic.csv` into a Pandas dataframe, and then systematically analyze the content from a variety of perspectives. As noted above, the goal is to understand: Who survived, and why?

Things to do:

1. Read `titanic.csv` into a Pandas dataframe.

2. Separate the data into two categories: passengers that survived, passengers that drowned. For each category compute the relevant statistics (e.g., how many people, ratio of males and females, number of passengers in each passenger class).

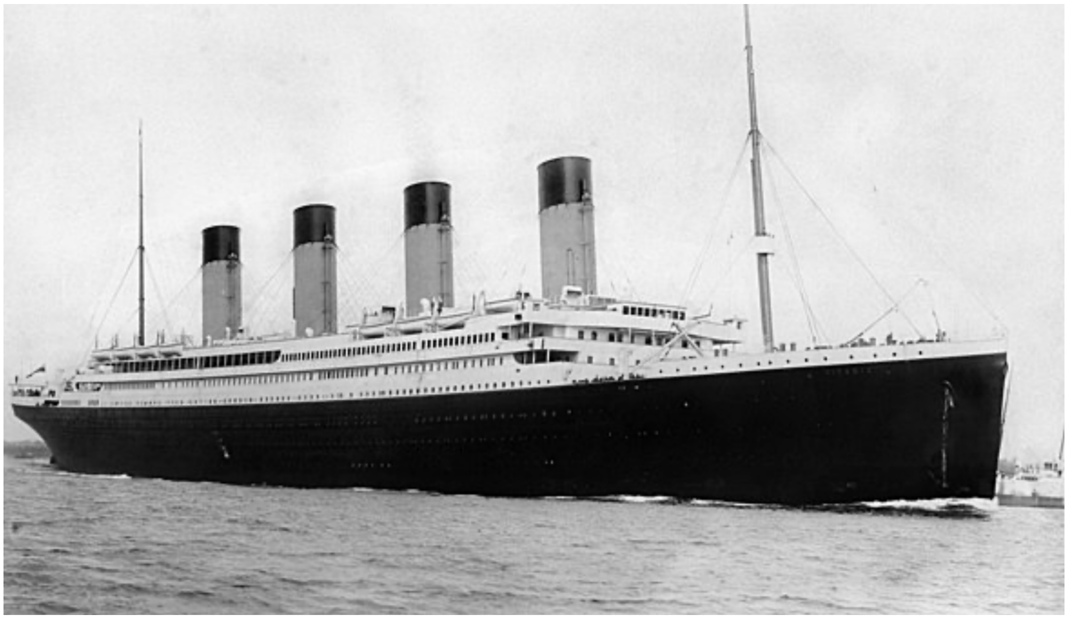

Figure 2: Titanic departing Southampton on April 10, 1912.

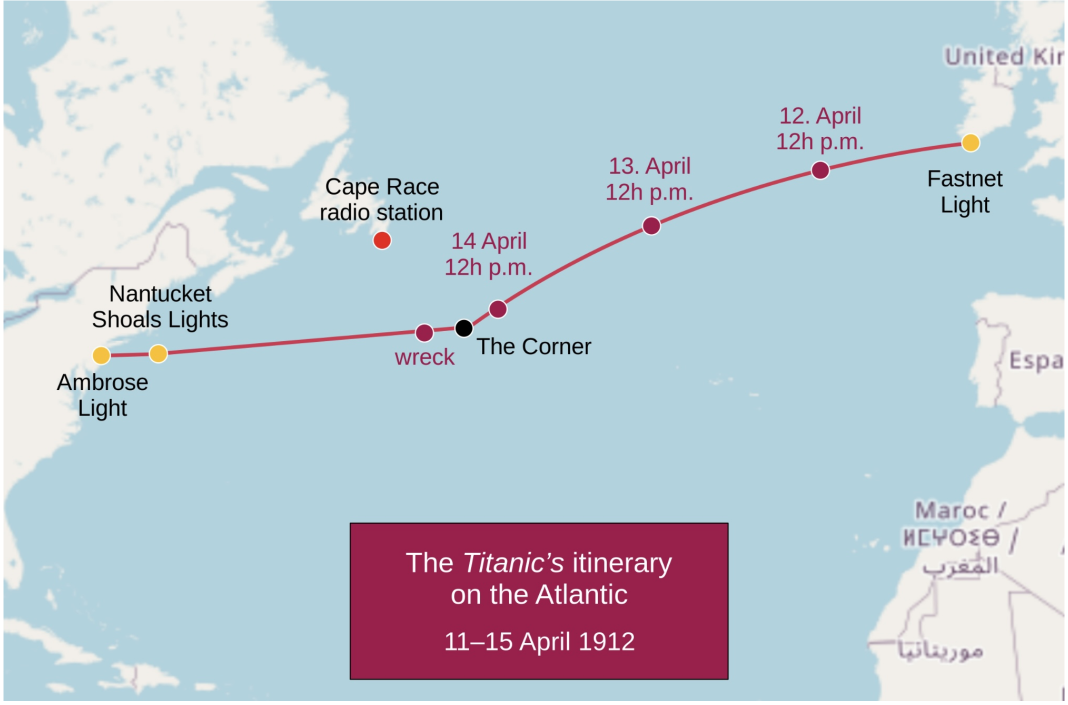

Figure 3: Trans-Atlantic route for the Titanic, from Ireland to New York.

3. Generate histograms for the distribution of age among the survivors and victims.

In Cameron's movie, women and children were given priority to board a lifeboat, and hence survived.

4. Is this part of the story supported by the `titanic.csv` data, or not?

5. Is there any evidence in the data that first class passengers (class 1) were given priority in boarding a lifeboat?