Class Exercises

Multimedia and Computer Graphics

Escola Superior de Tecnologia e Gestão
Politécnico de Coimbra

3D data visualization

Create a 3D graph using the plot_trisurf method from the Matplotlib library. The data that should be used for the X, Y and Z axis correspond to the existing columns in the fine denominated "data.csv". This file is available in the course support material. The output is represented in Figure 1.

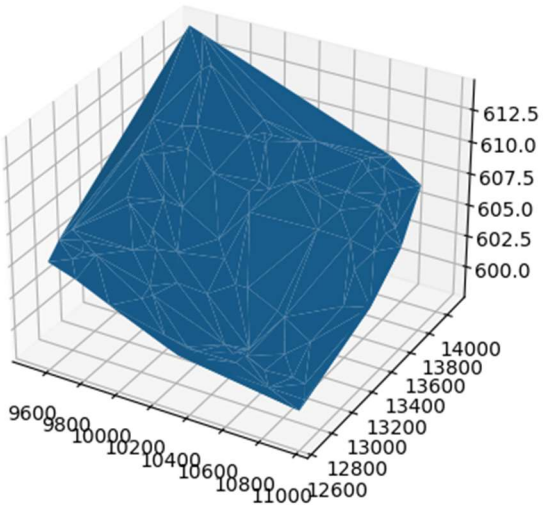
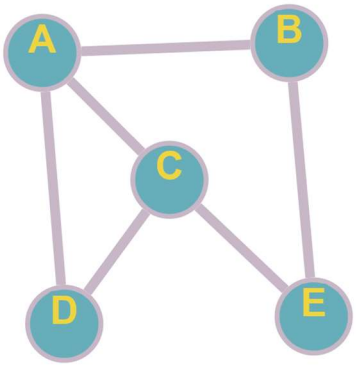


Figure 1 - 3D graph

Graph theory:

A graph can be represented in a file by listing once link per line, with each link represented by a pair of nodes. For example, the graph below is represented by the file on the right. Notice that, personalization preferences will be valorized in the score of the exercise.



graph.txt

A C
A B
A D
B E
C D
C E

Figure 2 – Graph and corresponding txt file.

a) Write a function that reads the file and creates the graph. The user should have the possibility to decide if the graph is directed or undirected. This function should load any file that respect the format of Figure 2.
b) Print the adjacency matrix for the graph.
c) Compute the basic information about the graph: number of nodes and edges.
d) Calculate the average degree of the graph.
e) Calculate the density of the graph.
f) Compute planarity and adjacency matrix.
g) Compute the shorted path for the input nodes entered by the user.
h) Create a method so that the user can update the weight and add other edges to this graph dynamically.

Data and Statistics with Python

USGS earthquake data is available in many formats, the simplest of which is a tabular file called CSV, short for "comma-separated values. CSV files contain one row of text per line, with columns separated by commas. A CSV file containing data about all the earthquakes in the past 30 days is available at:

http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_month.csv

a) Count the number of missing values in this dataset and remove the lines with missing data.
b) Compute the descriptive statistics of the data.
c) Using matplotlib create a 2D graph with the longitude data as the x values and the latitude data as the y values. Hint: use a scatter plot to represent the data. The output is represented in Figure 3.

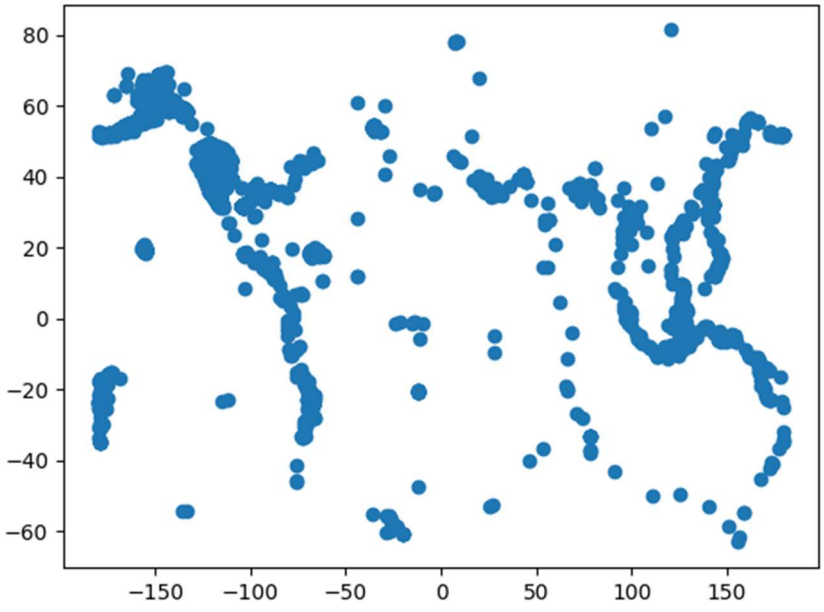


Figure 3 - Output of exercise c)

d) Modify the exercise c) to use a different color according to the depth of the earthquake. Shallow (< 10 km deep) earthquakes will be yellow, medium depth (>10 and <50 km deep) earthquakes will be red and deep earthquakes (>50 km) will be blue. The output is represented in Figure 4.

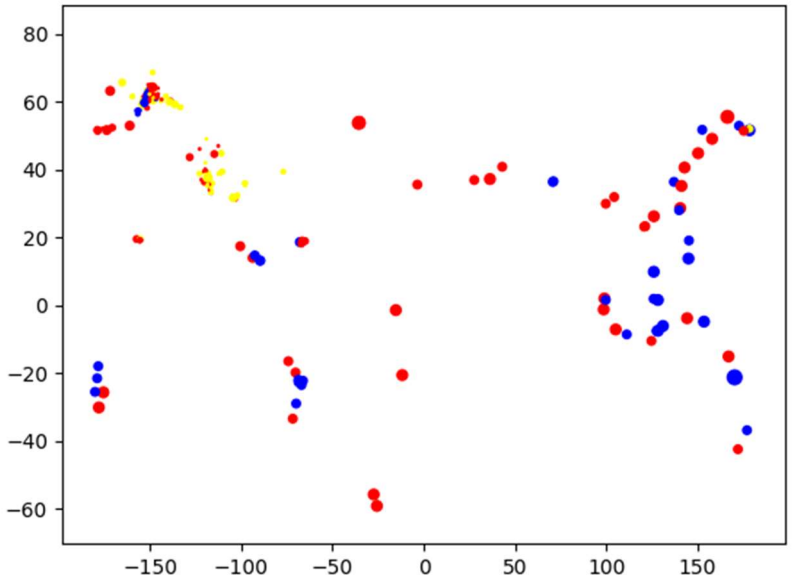


Figure 4 - Output of exercise d)

e) Modify the exercise d) to read the magnitude and draws larger circles for higher magnitude earthquakes. The size of each point should be the square of the magnitude of the corresponding earthquake. The output is represented in Figure 5.

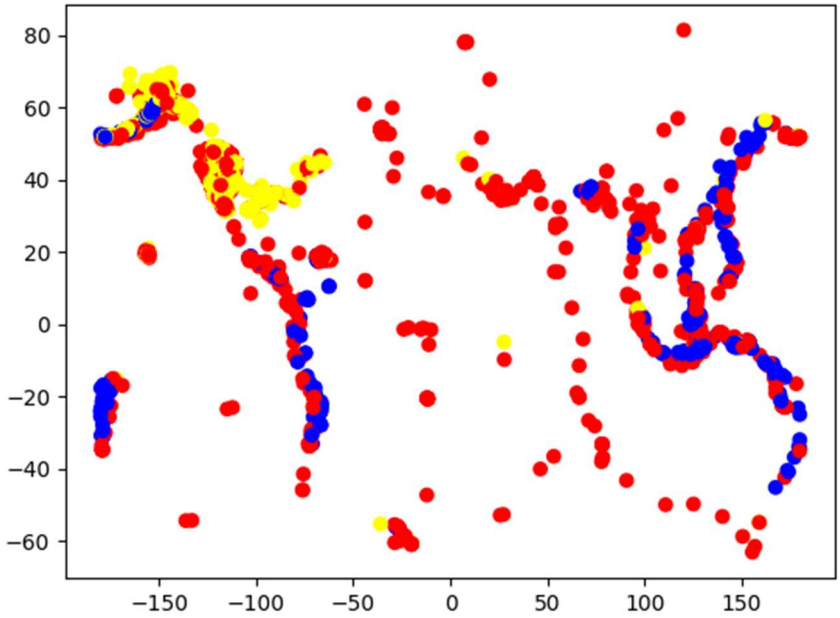


Figure 5 - Output of exercise e)

Notice that, exercises c), d) and e) will not show a map background. However, if you would like to add it, look into the Basemap class from the mpl_toolkits.basemap module. Kindly check Figure 6.

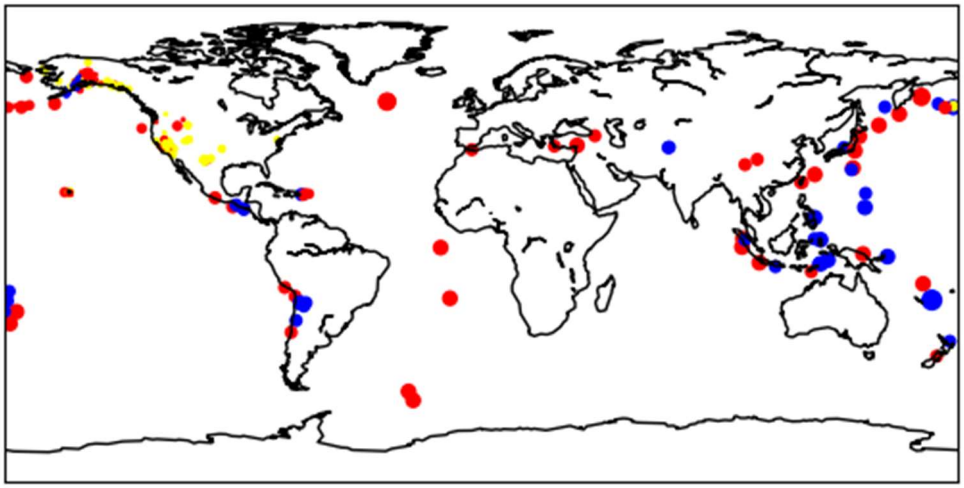


Figure 6 - Output with map background