Introduction to programming: Final project

Part 3: Visualize the data

In this part, your task will be to visualize the data. Put all the functions that you will create in this part to graph.py.

Task 1. Create a function that draws a barplot for the earthquake data. Earthquakes are divided into categories by magnitude: magnitude 4.0-4.9, 5.0-5.9, 6.0-6.9, 7.0-7.9, 8.0 or over. The function takes four arguments: time_start, time_end (as datetime objects), list_of_regions, show, and creates a barplot for the earthquakes that occurred between time_start and time_end in one of the regions in list_of_regions. If show is True, the function should show the graph, otherwise it should simply save it into the folder ../graphs with a unique identifier.

Task 2. Create a function that draws a piechart of strong earthquakes (magnitude > 5, depth < 70 km). The function should take the following arguments: time_start, time_end (as datetime objects), show. Labels of the piechart are the regions, count only the number of earthquakes between time_start and time_end. If show is True, the function should show the graph, otherwise it should simply save it into the folder ../graphs with a unique identifier.

Task 3. Create a function that draws a 2D map of earthquakes. The function should take the following arguments: an integer k, time_start, time_end (as datetime objects), show. It should draw the k strongest earthquakes between time_start and time_end. Use different colors for different regions. To convert from longitude, latitude to x, y use the following formula:

x = (MAP_WIDTH/360.0) * (180 + longitude)
y = (MAP_HEIGHT/180.0) * (latitude – 90)

where MAP_WIDTH is the width of your figure and MAP_HEIGHT is the height of your figure.

Hint: Matplot has a built-in function to create scatterplots called *scatter()*.

Task 4. Recent devastating earthquakes in Haiti, Chile and China, as well as magnitude 7+ earthquakes in Indonesia and California, might give the impression that earthquake activity is increasing. Let us take a look at earthquake statistics to understand if this is the case. Create a function that takes a region and a boolean variable *show* as arguments, and return an array with the number of strong earthquakes that happened in this region per month (add this function to statistics.py).
Then create a function that draws a plot graph of this data and save it into the folder ../graphs with a unique identifier (add this function to a graph.py).

Part 4: Risk prevision

Task 1: Imagine that you are going on a research mission and your university wants to check the seismic activity around the place where you are going to ensure your safety. Create a function (add it to statistics.py) city_is_safe to help your university. The function should take the following arguments: name of a city, and length of your stay (number of days). It should return the probability not to have a strong earthquake during your stay with decimal precision of 6 digits. The probability is computed as follows:

λ = average number of strong earthquakes (magnitude > 5, depth < 70 km) in t days (where t is the number of days of your mission) in at most 100 km from the city.

Probability = $e^{-\lambda}$

(We use Poisson distribution here, take a look at https://en.wikipedia.org/wiki/Poisson_distribution if you are curious.)

Task 2. Create a file main.py. This is the file from which all the functions will be called from. Your task is to create a text user interface that will give access to all the functions that we have created above. In more detail, when you launch main.py, we should see a line like, for example,

“Hello! Earthquake project is there. What would you like to learn?”

Then the program should display available options (statistics / graphs / prediction), which a user can choose by typing an option (or its number). When an option is selected, show the user a list of functions with docstrings. In docstrings, explain what are the possible values for the arguments. Don't forget to add commands to allow switching options / terminate the program.