Introduction to Python, ISA, ULisboa

Instructor: Manuel Campagnolo

To be presented in class on October 27, 2023

Group assignment (Problem P3)

Geographical coordinates are usually represented in Python by a tuple lon,lat, which are the longitude and latitude expressed in decimal degrees. A collection of labeled points can be represented by a dictionary, where the keys are the point labels and the values are tuples lon,lat that represent the locations, e.g. {'a': (-9.12395, 38.76993), 'b': (-9.13469, 38.7644)}.

One simplified approach to compute the distance between two points is to consider that the Earth's shape is a sphere. Then, for a small region of interest (ROI), differences of latitudes can be converted to meters over the surface by multiplying by a given constant (~111000 m), and differences of longitudes can also be converted with a distinct constant (on the Equator, both constants are the same).

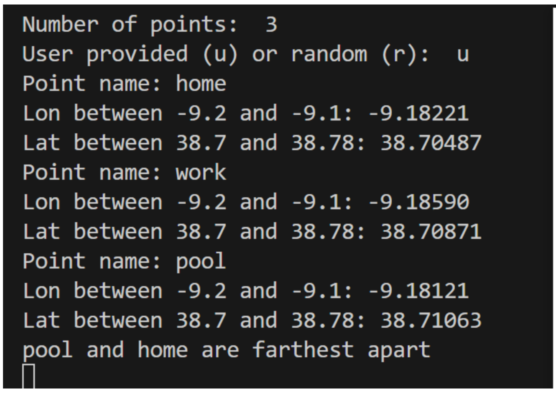
The program should request the following information from the user:

1. The number of points, between 2 and 10;
2. If the points should be generated randomly within the ROI (option "r") or should be provided by the user (option "u");
3. If the points are generated randomly, the user just has to provide a name for each point; if the points are provided by the user, the user has to provide the name and the coordinates for each point.

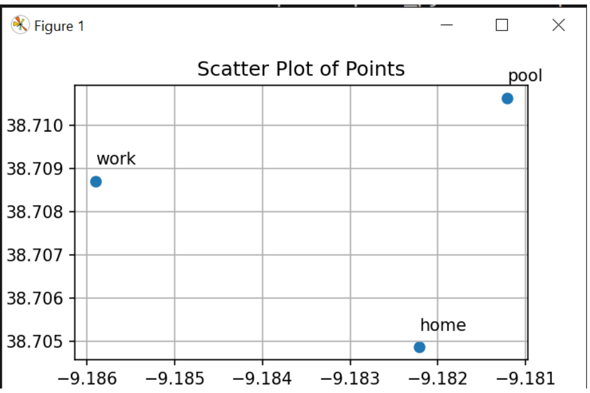
To create coordinates randomly for the ROI, you can use function "uniform" from the "random" module. The square root function "sqrt" is available in module "math.

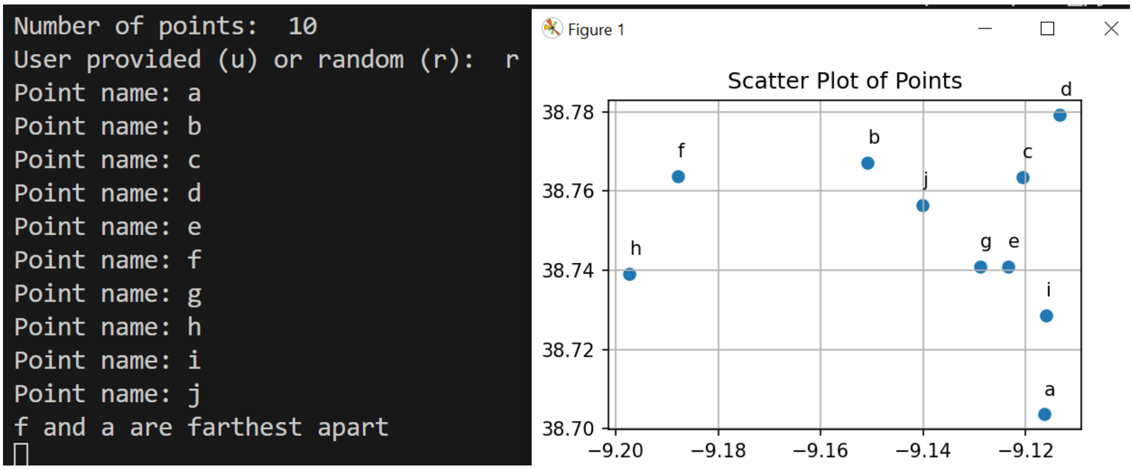
The program will then create a dictionary of points, and will determine which pair of points is farthest apart (according to the distance over the surface). Finally, a scatter plot will be created to plot the points and the respective labels.

You can find below two possible inputs and outputs, for option "u" and for option "r". Note that the distances are to be computed in meters using coef_lon and coef_lat for the region of interest.



```
Number of points:  3
User provided (u) or random (r):  u
Point name: home
Lon between -9.2 and -9.1: -9.18221
Lat between 38.7 and 38.78: 38.70487
Point name: work
Lon between -9.2 and -9.1: -9.18590
Lat between 38.7 and 38.78: 38.70871
Point name: pool
Lon between -9.2 and -9.1: -9.18121
Lat between 38.7 and 38.78: 38.71063
pool and home are farthest apart
```





```
Number of points:  10
User provided (u) or random (r):  r
Point name: a
Point name: b
Point name: c
Point name: d
Point name: e
Point name: f
Point name: g
Point name: h
Point name: i
Point name: j
f and a are farthest apart
```

Complete the code available at https://github.com/isa-ulisboa/greends-ipython-exercises/blob/main/distances_incomplete.py and create a file **distances.py** that runs as expected. Test each function separately, and then the whole code.