PROJECT: MONITORING FLIGHT INFORMATION

Introduction

This project is concerned with a large dataset that consists of flight information in 2023. The end product will be a dashboard in which various information on flights can be monitored.

Part 1: Getting acquainted with the data

In the first part, you will use only one of the tables in the data, namely `airports.csv`. Use `pandas` to read the `.csv` into Python and save it as a `DataFrame`. This dataset is part of a larger database on all flights departing New York City in 2023 and contains information on various airports. Use this dataset to perform the following tasks. It is advised to use the functions such as `scatter_geo` from the library `plotly.express`.

- Set up a map of the world with points on it indicating the airports in the set.
- Identify the airports outside of the US and in addition, make a new map of only the US.
  Extra: Color code the airports by their altitude
- Make a function that takes the FAA-abbreviation of the name of an airport as input and then plots a map of the world and add a line from NYC to the airport on that map.
  Extra: Specify the function to make a map of only the US if the airport is located in the US
- Extend the previous function to accept a list of FAA-abbreviations and plot a line for each of the multiple lines for each of the airports.
- Look op the position of John F. Kennedy airport in New York City and compute the Euclidean distance $\sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}$ for each airport and visualize the distribution of the distances in a suitable figure.
- Since the earth is not flat, it makes much mor sense to compute the geodesic distance between two airports. i.e. the length of a circular arc connecting them. For two locations with difference $\Delta\lambda$ in longitude and $\Delta\phi$ in latitude, this distance is given by

$$R\sqrt{\left(2\sin\left(\frac{\Delta\phi}{2}\right)\cos\left(\frac{\Delta\lambda}{2}\right)\right)^2 + \left(2\cos(\phi_m)\sin\left(\frac{\Delta\lambda}{2}\right)\right)^2},$$

  where $\phi_m = \frac{\phi_1+\phi_2}{2}$ is the midpoint of the two latitudes and R denotes the radius of the earth. Look up this radius and repeat the previous task for this distance.
- Analyse the different time zones; Make a graphical representation of the time zones of the airports that represents the relative amount of flights to them.

Use your own creativity to discover any other features about this dataset you find interesting or noteworthy! Look for relations between variables of create attractive visualizations of the data. Points will be awarded for creative insights or features of the final product!

Create one `.py`-file which, upon execution produces all required figures

Part 2: Setting up a GitHub repository

The final product needs proper version control. Therefore everyone in the group will need an account on GitHub. Set up a private repository with your group and add a `.py`-file to it containing the code used for Part 1. Make sure to add a `README.md` file to it.

The `README.md`-file is very important as it contains information for users of the software you create. It should therefore serve as a guide for a use to use your library. `.md`-files can be opened and edited in any text editor, including Visual Studio Code.

Initialize `git` Take turns within your group and add small sections to the `README.md`-file each time. Between these turns `commit` and `push` the changes until you are satisfied with the file.Make sure to keep updating the `README` throughout the project, as this file will also be judged at the end.