**Instructions**

This activity is broken down into two deliverables, WeatherPy and VacationPy.

**Part 1: WeatherPy**

In this deliverable, you'll create a Python script to visualize the weather of over 500 cities of varying distances from the equator. You'll use the [citipy Python library](https://pypi.python.org/pypi/citipy)

[Links to an external site.](https://pypi.python.org/pypi/citipy), the [OpenWeatherMap API](https://openweathermap.org/api)

[Links to an external site.](https://openweathermap.org/api), and your problem-solving skills to create a representative model of weather across cities.

For this part, you'll use the WeatherPy.ipynb Jupyter notebook provided in the starter code ZIP file. The starter code will guide you through the process of using your Python coding skills to develop a solution to address the required functionalities.

To get started, the code required to generate random geographic coordinates and the nearest city to each latitude and longitude combination is provided.

**Requirement 1: Create Plots to Showcase the Relationship Between Weather Variables and Latitude**

To fulfill the first requirement, you'll use the OpenWeatherMap API to retrieve weather data from the cities list generated in the starter code. Next, you'll create a series of scatter plots to showcase the following relationships:

* Latitude vs. Temperature
* Latitude vs. Humidity
* Latitude vs. Cloudiness
* Latitude vs. Wind Speed

**Requirement 2: Compute Linear Regression for Each Relationship**

To fulfill the second requirement, compute the linear regression for each relationship. Separate the plots into Northern Hemisphere (greater than or equal to 0 degrees latitude) and Southern Hemisphere (less than 0 degrees latitude). You may find it helpful to define a function in order to create the linear regression plots.

Next, create a series of scatter plots. Be sure to include the linear regression line, the model's formula, and the r values as you can see in the following image

A graph with numbers and dots

Description automatically generated

You should create the following plots:

* Northern Hemisphere: Temperature vs. Latitude
* Southern Hemisphere: Temperature vs. Latitude
* Northern Hemisphere: Humidity vs. Latitude
* Southern Hemisphere: Humidity vs. Latitude
* Northern Hemisphere: Cloudiness vs. Latitude
* Southern Hemisphere: Cloudiness vs. Latitude
* Northern Hemisphere: Wind Speed vs. Latitude
* Southern Hemisphere: Wind Speed vs. Latitude

After each pair of plots, explain what the linear regression is modeling. Describe any relationships that you notice and any other findings you may uncover.

**Part 2: VacationPy**

In this deliverable, you'll use your weather data skills to plan future vacations. Also, you'll use Jupyter notebooks, the geoViews Python library, and the Geoapify API.

The code needed to import the required libraries and load the CSV file with the weather and coordinates data for each city created in Part 1 is provided to help you get started.

Your main tasks will be to use the Geoapify API and the geoViews Python library and employ your Python skills to create map visualizations.

To succeed on this deliverable of the assignment, open the VacationPy.ipynb starter code and complete the following steps:

1. Create a map that displays a point for every city in the city\_data\_df DataFrame as shown in the following image. The size of the point should be the humidity in each city.

A map of the world with different colored circles

Description automatically generated

2. Narrow down the city\_data\_df DataFrame to find your ideal weather condition. For example:

* A max temperature lower than 27 degrees but higher than 21
* Wind speed less than 4.5 m/s
* Zero cloudiness

**note**

Feel free to adjust your specifications but make sure to set a reasonable limit to the number of rows returned by your API requests.

3. Create a new DataFrame called hotel\_df to store the city, country, coordinates, and humidity.

4. For each city, use the Geoapify API to find the first hotel located within 10,000 meters of your coordinates.

5. Add the hotel name and the country as additional information in the hover message for each city on the map as in the following image:

A map of the world with a white text

Description automatically generated