For this project, start by creating a new repository called "python-api-challenge" and clone it to your local machine. Inside this repository, establish a folder corresponding to the challenge, such as "WeatherPy," and add the essential files, including "api_keys.py," "WeatherPy.ipynb," and "VacationPy.ipynb" from the provided starter code ZIP file. Don't forget to include a .gitignore file to safeguard your API key. In Part 1, named "WeatherPy," you'll use Python to visualize weather data for over 500 cities, retrieved from the OpenWeatherMap API. Your tasks involve creating scatter plots to illustrate the relationships between latitude and temperature, humidity, cloudiness, and wind speed. Additionally, you'll compute linear regressions for these relationships, dividing them into Northern and Southern Hemispheres.

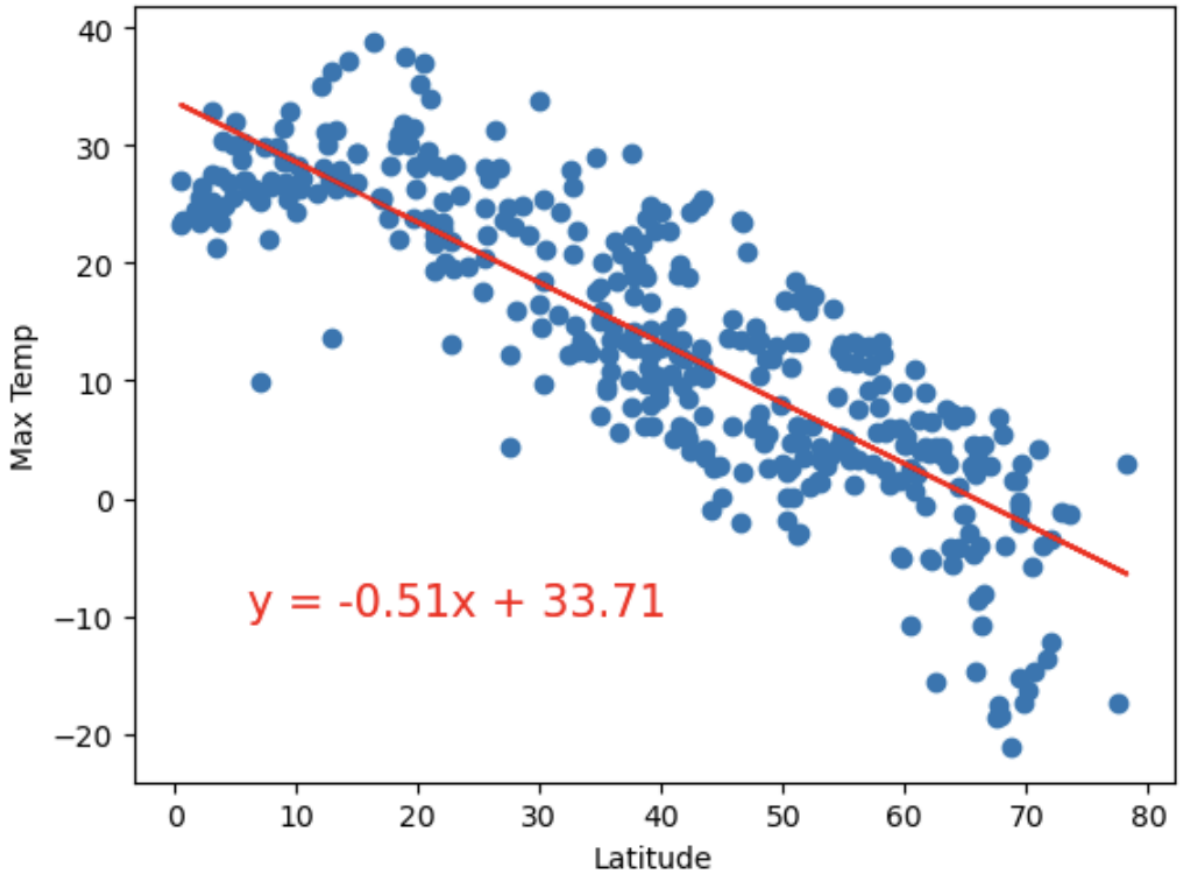
In Part 2, "VacationPy," you'll employ your weather data skills to plan future vacations. This involves creating maps displaying cities and filtering for ideal weather conditions. You'll also use the Geoapify API to locate hotels near chosen locations and enhance map markers with hotel names and countries as hover information. The project outlines specific point allocations for various tasks in both parts.

Part 1: WeatherPy

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

```
The r-value is: 0.7161472181434118
```



Part 2: VacationPy

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.

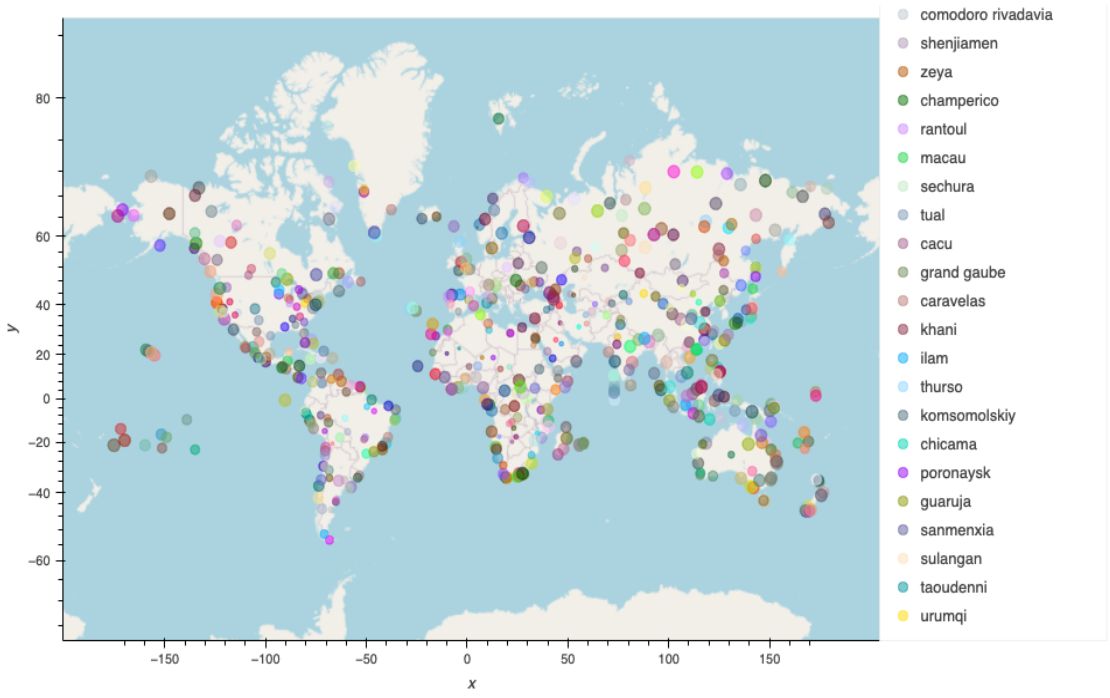
2: Narrow down the `city_data_df` DataFrame to find your ideal weather condition. For example:

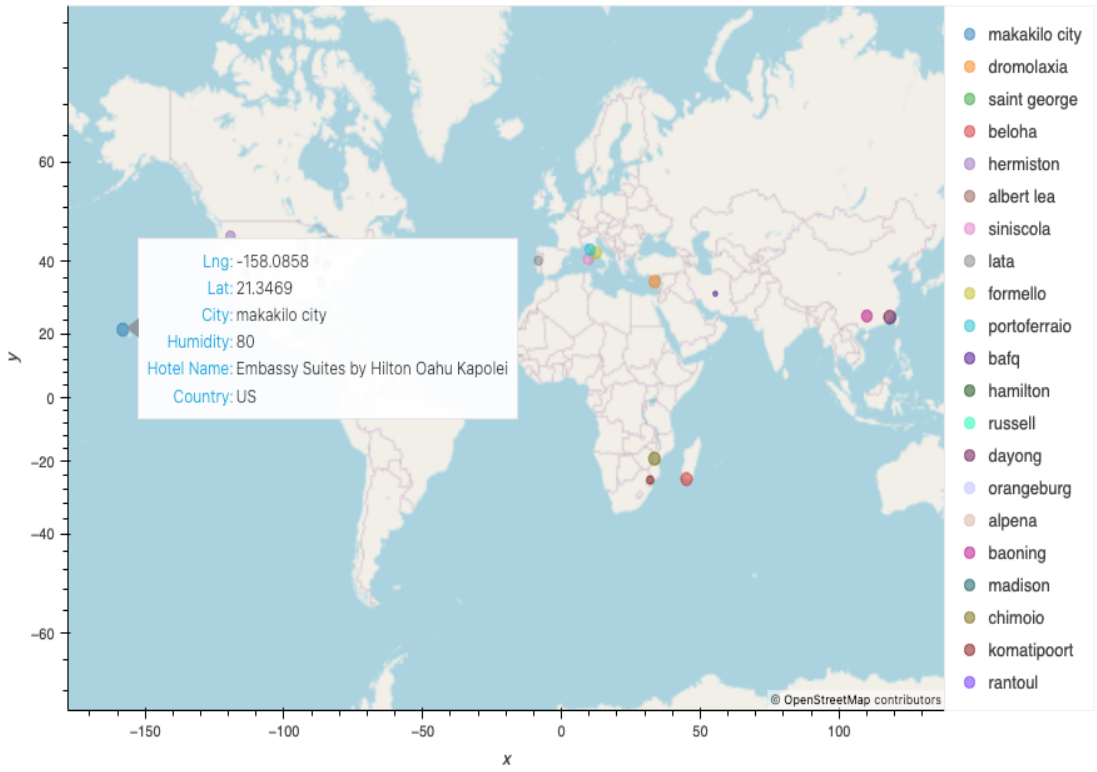
- A max temperature lower than 27 degrees but higher than 21
- Wind speed less than 4.5 m/s
- Zero cloudiness

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:





Please follow the Requirements

The requirements for "Part 1: WeatherPy" are the following

Create Plots to Showcase the Relationship Between Weather Variables and Latitude (30 points)

- Use the OpenWeatherMap API to retrieve weather data from the cities list generated in the started code (10 points)
- Create a scatter plot to showcase the relationship between Latitude vs. Temperature (5 points)
- Create a scatter plot to showcase the relationship between Latitude vs. Humidity (5 points)
- Create a scatter plot to showcase the relationship between Latitude vs. Cloudiness (5 points)

- Create a scatter plot to showcase the relationship between Latitude vs. Wind Speed (5 points)

Compute Linear Regression for Each Relationship (40 points)

- Linear regression scatter plot for Northern Hemisphere: Temperature (C) vs. Latitude (5 points)
- Linear regression scatter plot for Southern Hemisphere: Temperature (C) vs. Latitude (5 points)
- Linear regression scatter plot for Northern Hemisphere: Humidity (%) vs. Latitude (5 points)
- Linear regression scatter plot for Southern Hemisphere: Humidity (%) vs. Latitude (5 points)
- Linear regression scatter plot for Northern Hemisphere: Cloudiness (%) vs. Latitude (5 points)
- Linear regression scatter plot for Southern Hemisphere: Cloudiness (%) vs. Latitude (5 points)
- Linear regression scatter plot for Northern Hemisphere: Wind Speed (m/s) vs. Latitude (5 points)
- Linear regression scatter plot for Southern Hemisphere: Wind Speed (m/s) vs. Latitude (5 points)

The requirements for "Part 2: VacationPy" are the following (30 points)

- Create a map that displays a point for every city in the `city_data_df` DataFrame (5 points)
- Narrow down the `city_data_df` DataFrame to find your ideal weather condition (5 points)
- For each city in the `hotel_df` DataFrame, use the Geoapify API to find the first hotel located within 10,000 meters of your coordinates (10 points)
- Add the hotel name and the country as additional information in the hover message for each city in the map. (10 points)