VacationPy

Starter Code to Import Libraries and Load the Weather and Coordinates Data

In [3]:
```
# Dependencies and Setup
import hvplot.pandas
import pandas as pd
import requests

# Import API key
from api_keys import geoapify_key
```

In [4]:
```
# Load the CSV file created in Part 1 into a Pandas DataFrame
city_data_df = pd.read_csv("output_data/cities.csv")

# Display sample data
city_data_df.head()
```

Out[4]:

	City_ID	City	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country	Da
0	0	albany	42.6001	-73.9662	43.21	60	27	3.00	US	17011164
1	1	waitangi	-43.9535	-176.5597	53.82	90	100	3.00	NZ	17011165
2	2	puerto ayacucho	5.6639	-67.6236	87.06	68	49	2.10	VE	17011165
3	3	ilulissat	69.2167	-51.1000	26.62	80	75	12.66	GL	17011165
4	4	andovoranto	-18.9500	49.1000	74.46	89	13	6.04	MG	17011165

Step 1: Create a map that displays a point for every city in the `city_data_df` DataFrame. The size of the point should be the humidity in each city.

In [6]:
```
# Configure the map
map_plot = city_data_df.hvplot.points(
    "Lng",
    "Lat",
    geo = True,
    tiles = "CartoDark",
    frame_width = 800,
    frame_height = 600,
    size = "Humidity",
```

```
    color = "City"
)

# Display the map plot
map_plot
```

Out[6]:

Step 2: Narrow down the `city_data_df` DataFrame to find your ideal weather condition

In [7]:
```
city_data_df.head()
```

Out[7]:

	City_ID	City	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country	Da
0	0	albany	42.6001	-73.9662	43.21	60	27	3.00	US	17011164
1	1	waitangi	-43.9535	-176.5597	53.82	90	100	3.00	NZ	17011165
2	2	puerto ayacucho	5.6639	-67.6236	87.06	68	49	2.10	VE	17011165
3	3	ilulissat	69.2167	-51.1000	26.62	80	75	12.66	GL	17011165
4	4	andovoranto	-18.9500	49.1000	74.46	89	13	6.04	MG	17011165

In [8]:
```
# Narrow down cities that fit criteria and drop any results with null values
# temp>= 70, <80
#wind <10 MPH
#cloudiness=0

mask=(city_data_df["Max Temp"]>=70) & (city_data_df["Max Temp"]<80) & (city_data_df["C
df_sub=city_data_df.loc[mask]

# Drop any rows with null values
df_sub=df_sub.dropna(how="any")

# Display sample data
df_sub.head()
```

Out[8]:

	City_ID	City	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country	Da
159	159	inhambane	-23.8650	35.3833	73.67	86	0	7.43	MZ	17011166
172	172	porbandar	21.6422	69.6093	72.63	70	0	11.54	IN	17011166
181	181	pisco	-13.7000	-76.2167	75.25	64	0	19.57	PE	17011166
182	182	lompoc	34.6391	-120.4579	70.93	26	0	11.50	US	17011166
188	188	ankazoabo	-22.2833	44.5167	77.05	46	0	8.25	MG	17011166

Step 3: Create a new DataFrame called `hotel_df`.

In [12]:
```
# Use the Pandas copy function to create DataFrame called hotel_df to store the city,
hotel_df=df_sub.loc[:, ["City","Country","Lat","Lng","Humidity", "Max Temp", "Cloudine

# Add an empty column, "Hotel Name," to the DataFrame so you can store the hotel found
hotel_df["Hotel Name"]=""

# Display sample data
hotel_df.head()
```

Out[12]:

	City	Country	Lat	Lng	Humidity	Max Temp	Cloudiness	Hotel Name
159	inhambane	MZ	-23.8650	35.3833	86	73.67	0	
172	porbandar	IN	21.6422	69.6093	70	72.63	0	
181	pisco	PE	-13.7000	-76.2167	64	75.25	0	
182	lompoc	US	34.6391	-120.4579	26	70.93	0	
188	ankazoabo	MG	-22.2833	44.5167	46	77.05	0	

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

In [14]:
```
# Set parameters to search for a hotel
categories = "accommodation.hotel"
radius = 10000
limit=20
params = {
    "categories":categories,
    "limit":limit,
    "apiKey":geoapify_key
}

# Print a message to follow up the hotel search
print("Starting hotel search")

# Iterate through the hotel_df DataFrame
for index, row in hotel_df.iterrows():
```

```
    # get latitude, longitude from the DataFrame
    latitude=row.Lat
    longitude=row.Lng

    # Add filter and bias parameters with the current city's latitude and longitude to
    params["filter"] = f"circle:{longitude},{latitude},{radius}"
    params["bias"] = f"proximity:{longitude},{latitude}"

    # Set base URL
    base_url = "https://api.geoapify.com/v2/places"


    # Make and API request using the params dictionaty
    response = requests.get(base_url, params=params)

    # Convert the API response to JSON format
    name_address = response.json()

    # Grab the first hotel from the results and store the name in the hotel_df DataFra
    try:
        hotel_df.loc[index, "Hotel Name"] = name_address["features"][0]["properties"][
    except (KeyError, IndexError):
        # If no hotel is found, set the hotel name as "No hotel found".
        hotel_df.loc[index, "Hotel Name"] = "No hotel found"

    # Log the search results
    print(f"{hotel_df.loc[index, 'City']} - nearest hotel: {hotel_df.loc[index, 'Hotel

# Display sample data
hotel_df
```

```
Starting hotel search
inhambane - nearest hotel: Hotel de Inhambane
porbandar - nearest hotel: Toran Tourist Bungalow
pisco - nearest hotel: La Portada
lompoc - nearest hotel: Embassy Suites by Hilton Lompoc Central Coast
ankazoabo - nearest hotel: No hotel found
saint-pierre - nearest hotel: Tropic Hotel
timbuktu - nearest hotel: Hotel La Maison
ormara - nearest hotel: No hotel found
yung shue wan - nearest hotel: 浪濤軒酒店 Concerto Inn
```

Out[14]:

	City	Country	Lat	Lng	Humidity	Max Temp	Cloudiness	Hotel Name
159	inhambane	MZ	-23.8650	35.3833	86	73.67	0	Hotel de Inhambane
172	porbandar	IN	21.6422	69.6093	70	72.63	0	Toran Tourist Bungalow
181	pisco	PE	-13.7000	-76.2167	64	75.25	0	La Portada
182	lompoc	US	34.6391	-120.4579	26	70.93	0	Embassy Suites by Hilton Lompoc Central Coast
188	ankazoabo	MG	-22.2833	44.5167	46	77.05	0	No hotel found
370	saint-pierre	RE	-21.3393	55.4781	73	73.08	0	Tropic Hotel
451	timbuktu	ML	16.7735	-3.0074	14	76.95	0	Hotel La Maison
452	ormara	PK	25.2088	64.6357	52	75.76	0	No hotel found
502	yung shue wan	HK	22.2333	114.1167	68	71.67	0	浪濤軒酒店 Concerto Inn

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

In [15]:
```
# Configure the map
map_plot_2 = hotel_df.hvplot.points(
    "Lng",
    "Lat",
    geo = True,
    tiles = "CartoDark",
    frame_width = 800,
    frame_height = 600,
    size = "Humidity",
    color = "City",
)

# Display the map plot
map_plot_2
```

Out[15]:

In [ ]:
```

```