VacationPy

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

In [1]:
```
# 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")

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

	City	City.1	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country
0	0	st. john's	47.5649	-52.7093	42.98	93	100	6.91	CA
1	1	adamstown	-25.0660	-130.1015	72.09	89	100	18.92	PN
2	2	chail	25.4333	81.6333	69.48	66	0	2.10	IN
3	3	port-aux-francais	-49.3500	70.2167	41.29	84	83	46.30	TF
4	4	pervomaysk	48.0443	30.8507	32.65	79	100	11.27	UA

In [5]:
```
city_data_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 469 entries, 0 to 468
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   City        469 non-null    int64  
 1   City.1      469 non-null    object 
 2   Lat         469 non-null    float64
 3   Lng         469 non-null    float64
 4   Max Temp    469 non-null    float64
 5   Humidity    469 non-null    int64  
 6   Cloudiness  469 non-null    int64  
 7   Wind Speed  469 non-null    float64
 8   Country     465 non-null    object 
 9   Date        469 non-null    int64  
dtypes: float64(4), int64(4), object(2)
memory usage: 36.8+ KB
```

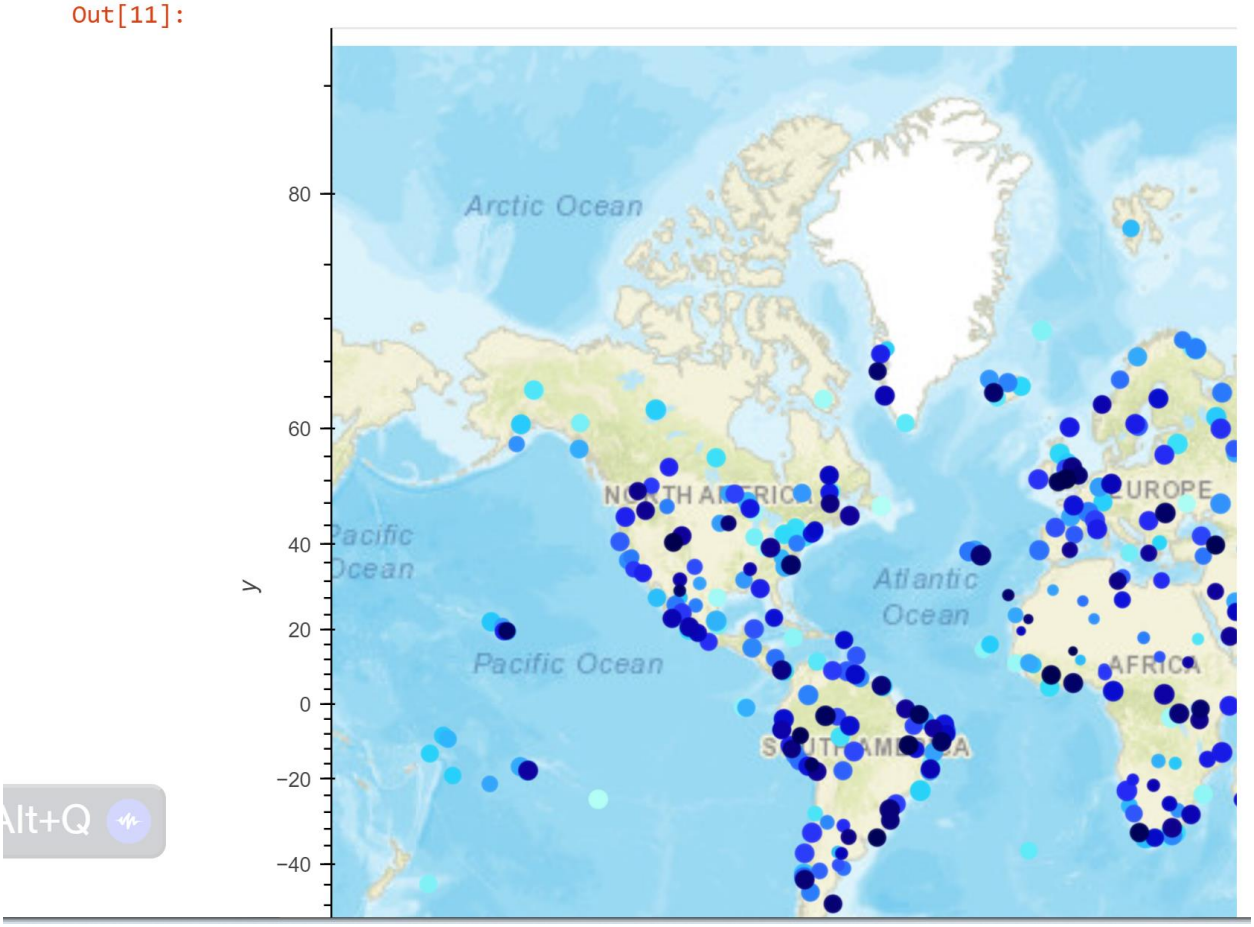

Step 1: Create a map that displays a point for every city in the `city_data_df` DataFrame. The size of the point should be the humidity in each city.

In [11]:
```
%%capture --no-display

# Configure the map plot
map_plot = city_data_df.hvplot.points(
    "Lng",
    "Lat",
    geo = True,
    tiles = "EsriStreet",
    frame_width = 800,
    frame_height = 600,
    size = "Humidity",
    color = "City"
)


# Display the map
map_plot
```

Out[11]:



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

In [12]:
```
city_data_df.head()
```

Out[12]:

	City	City.1	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country
0	0	st. john's	47.5649	-52.7093	42.98	93	100	6.91	CA
1	1	adamstown	-25.0660	-130.1015	72.09	89	100	18.92	PN
2	2	chail	25.4333	81.6333	69.48	66	0	2.10	IN
3	3	port-aux-francais	-49.3500	70.2167	41.29	84	83	46.30	TF
4	4	pervomaysk	48.0443	30.8507	32.65	79	100	11.27	UA

In [16]:
```
# Narrow down cities that fit criteria and drop any results with null va
mask = (city_data_df["Max Temp"] >= 60) & (city_data_df["Max Temp"] < 80
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[16]:

	City	City.1	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country	
2	2	chail	25.4333	81.6333	69.48	66	0	2.10	IN	170
7	7	dera bugti	29.0307	69.1510	60.21	38	0	2.89	PK	170
39	39	tura	25.5198	90.2201	70.27	81	0	1.57	IN	170
133	133	ciudad madero	22.2667	-97.8333	76.05	100	0	9.22	MX	170
154	154	tanumah	27.1000	44.1333	68.38	73	0	9.22	SA	170

In [17]:
```
df_sub.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 21 entries, 2 to 427
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   City        21 non-null     int64
 1   City.1      21 non-null     object
 2   Lat         21 non-null     float64
 3   Lng         21 non-null     float64
 4   Max Temp    21 non-null     float64
 5   Humidity    21 non-null     int64
 6   Cloudiness  21 non-null     int64
 7   Wind Speed  21 non-null     float64
 8   Country     21 non-null     object
 9   Date        21 non-null     int64
dtypes: float64(4), int64(4), object(2)
memory usage: 1.8+ KB
```

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

In [18]:
```
# Use the Pandas copy function to create DataFrame called hotel_df to st
hotel_df = df_sub.loc[:, ["City", "Country", "Lat", "Lng", "Humidity", "

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


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

Out[18]:

	City	Country	Lat	Lng	Humidity	Max Temp	Cloudiness	Hotel Name
2	2	IN	25.4333	81.6333	66	69.48	0	
7	7	PK	29.0307	69.1510	38	60.21	0	
39	39	IN	25.5198	90.2201	81	70.27	0	
133	133	MX	22.2667	-97.8333	100	76.05	0	
154	154	SA	27.1000	44.1333	73	68.38	0	

Alt+Q

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

In [20]:
```
# 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 an
    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 an API request using the params dictionaty
```

Alt+Q

```
    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 an 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 ho
    try:
        hotel_df.loc[index, "Hotel Name"] = name_address["features"][0][
    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

# Display sample data
hotel_df
```

Alt+Q

```
Starting hotel search
2 - nearest hotel: No hotel found
7 - nearest hotel: No hotel found
39 - nearest hotel: No hotel found
133 - nearest hotel: Hotel Posada Casa Blanca
154 - nearest hotel: No hotel found
158 - nearest hotel: Hotel Kasbah Azalay
159 - nearest hotel: No hotel found
196 - nearest hotel: No hotel found
205 - nearest hotel: No hotel found
252 - nearest hotel: No hotel found
256 - nearest hotel: No hotel found
285 - nearest hotel: Kanyum cottage village
295 - nearest hotel: No hotel found
315 - nearest hotel: No hotel found
323 - nearest hotel: Holiday Inn Express
328 - nearest hotel: Le Terminus
342 - nearest hotel: Zhonghan Hotel
350 - nearest hotel: No hotel found
417 - nearest hotel: Colón
420 - nearest hotel: Muscat International Hotel
427 - nearest hotel: Hotel Castelo
```

Out[20]:

	City	Country	Lat	Lng	Humidity	Max Temp	Cloudiness	Hotel Name
2	2	IN	25.4333	81.6333	66	69.48	0	No hotel found
7	7	PK	29.0307	69.1510	38	60.21	0	No hotel found
39	39	IN	25.5198	90.2201	81	70.27	0	No hotel found
133	133	MX	22.2667	-97.8333	100	76.05	0	Hotel Posada Casa Blanca
154	154	SA	27.1000	44.1333	73	68.38	0	No hotel found
158	158	MA	29.8200	-5.7200	26	60.78	0	Hotel Kasbah Azalay
159	159	US	31.3974	-102.3501	38	64.27	0	No hotel found
196	196	IN	19.8667	72.7000	59	79.25	0	No hotel found
205	205	MX	26.1167	-103.4500	44	73.53	0	No hotel found
252	252	CN	23.0964	109.6092	48	70.23	0	No hotel found
256	256	LY	32.8817	13.3506	55	64.20	0	No hotel found
285	285	NP	26.8418	88.0763	60	64.20	0	Kanyum cottage village
295	295	MX	24.2556	-107.1828	86	72.90	0	No hotel found
315	315	CN	23.7000	109.2667	47	70.18	0	No hotel found
323	323	US	33.7456	-117.8678	75	66.04	0	Holiday Inn Express

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

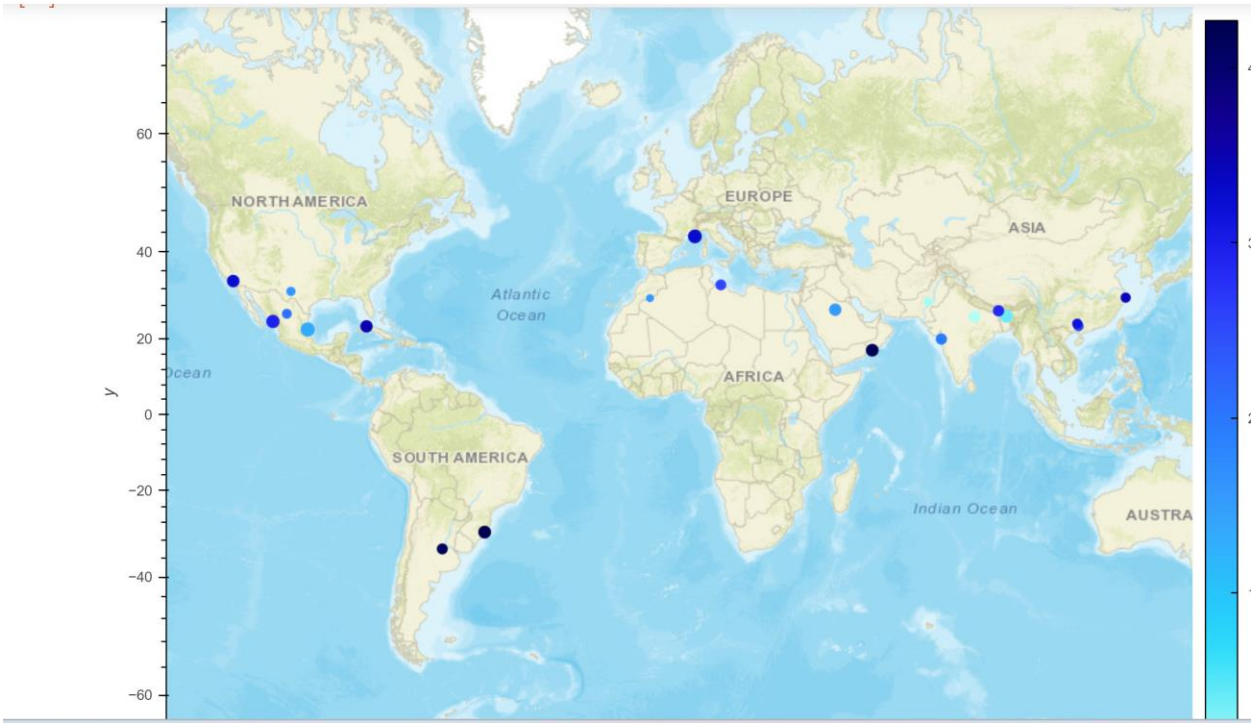
In [22]:

```
%%capture --no-display

# Configure the map plot
map_plot2 = hotel_df.hvplot.points(
    "Lng",
    "Lat",
    geo = True,
    tiles = "EsriStreet",
    frame_width = 800,
    frame_height = 600,
    size = "Humidity",
    color = "City",
    hover_cols=["Lat", "Lng", "City", "Country", "Hotel Name", "Max Temp
)

# Display the map
map_plot2
```

Out[22]:


60
40
20
0
−20
−40
−60
y
NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
Atlantic
Ocean
Indian Ocean
AUSTRA