

# VacationPy

## Data Analyst

Vidal Garcia Martinez

### 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 warnings
warnings.filterwarnings("ignore")

# Import API key
from api_keys import geoapify_key
```

```
In [3]: # 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[3]:
```

	City_ID	City	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country	Date
0	0	chokurdakh	70.6333	147.9167	-31.94	96	100	2.44	RU	02/08/2023 22:42:43
1	1	verkhoyansk	67.5447	133.3850	-43.10	95	75	0.33	RU	02/08/2023 22:46:23
2	2	albany	42.6001	-73.9662	0.15	80	81	1.23	US	02/08/2023 22:41:57
3	3	cidreira	-30.1811	-50.2056	24.49	83	24	8.46	BR	02/08/2023 22:40:27
4	4	vaini	-21.2000	-175.2000	29.09	79	75	2.06	TO	02/08/2023 22:41:55

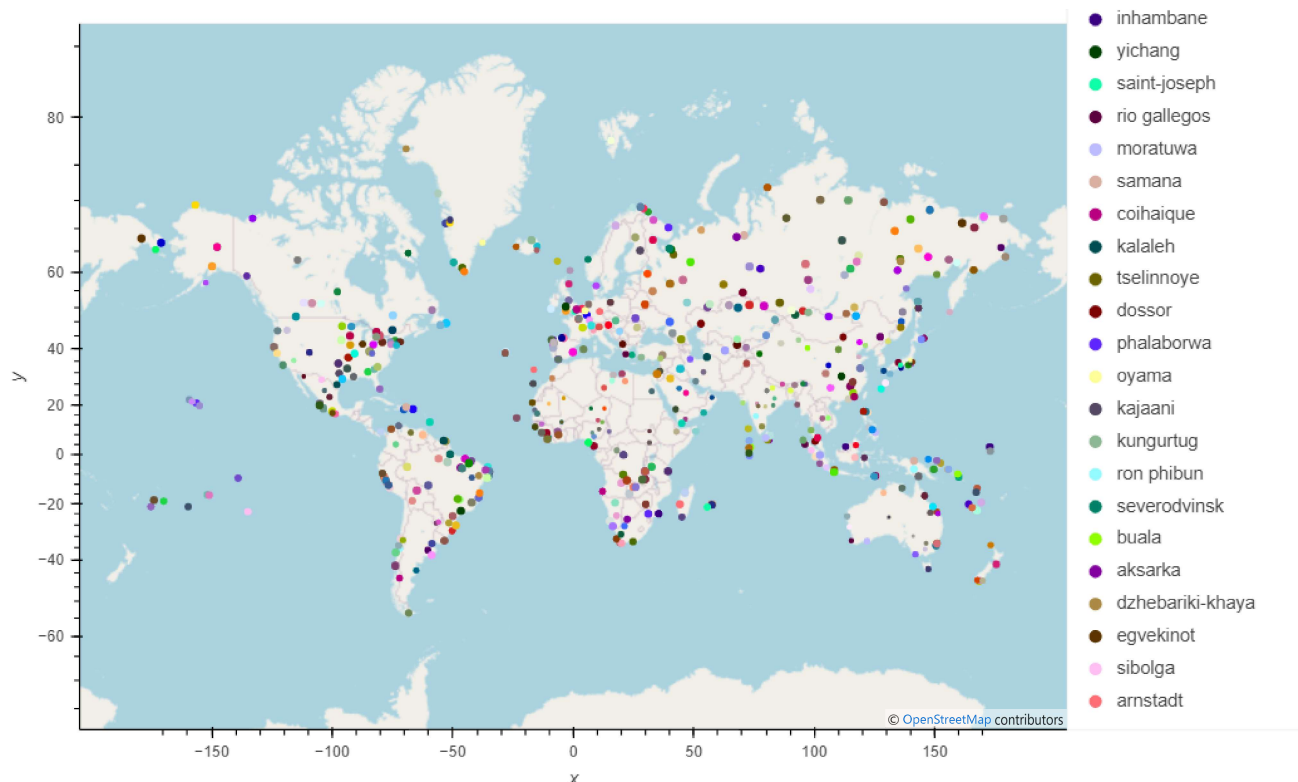
**Step 1: Create a map that displays a point for every city in the `city_data_df` DataFrame. The size of the point should be the humidity in each city.**

```
In [4]: %%capture --no-display

# Configure the map plot
map_plot_hum = city_data_df.hvplot.points("Lng", "Lat", geo = True, tiles = "OSM",
                                           frame_width = 700, frame_height = 500,
                                           size = "Humidity", color = "City", scale=0.5)

# Display the map
map_plot_hum
```

Out[4]:



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

```
In [5]: # Narrow down cities that fit criteria and drop any results with null values
ideal_weather = city_data_df[(city_data_df['Max Temp'] < 30) & (city_data_df['Max Temp'] > 20)
& (city_data_df['Humidity'] < 50) & (city_data_df['Humidity'] > 30)]

# Drop any rows with null values
ideal_weather = ideal_weather.dropna()

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

Out[5]:

	City_ID	City	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country	Date
13	13	busselton	-33.6500	115.3333	27.94	35	0	1.30	AU	02/08/2023 22:41:54
56	56	tezu	27.9167	96.1667	22.99	33	71	1.44	IN	02/08/2023 22:46:43
71	71	ayolas	-27.4000	-56.9000	27.50	32	82	3.36	PY	02/08/2023 22:46:47
99	99	tura	25.5198	90.2201	22.98	39	48	1.31	IN	02/08/2023 22:42:15
155	155	trelew	-43.2490	-65.3051	22.82	41	52	11.70	AR	02/08/2023 22:47:08

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

```
In [6]: # Use the Pandas copy function to create DataFrame called hotel_df to store the city, country, coordinates, and humidity
hotel_df = ideal_weather.filter(['City', 'Country', 'Lat', 'Lng', 'Humidity'], axis=1)

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

# Display sample data
hotel_df
```

Out[6]:

	City	Country	Lat	Lng	Humidity	Hotel Name
13	busselton	AU	-33.6500	115.3333	35	
56	tezu	IN	27.9167	96.1667	33	
71	ayolas	PY	-27.4000	-56.9000	32	
99	tura	IN	25.5198	90.2201	39	
155	trelew	AR	-43.2490	-65.3051	41	
211	yaan	NG	7.3833	8.5667	37	
213	remanso	BR	-9.6179	-42.0831	46	
253	fram	PY	-26.9833	-55.8833	37	
263	banda aceh	IN	25.4833	80.3333	32	
320	veraval	IN	20.9000	70.3667	33	
366	youkounkoun	GN	12.5333	-13.1333	35	
376	pasighat	IN	28.0667	95.3333	40	
424	akyab	MM	20.1500	92.9000	41	
483	santiago del estero	CL	-33.4569	-70.6483	49	
507	tha bo	LA	14.9604	105.7212	36	
536	ron phibun	IN	15.6667	75.7333	42	

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

In [7]:

```
# Set parameters to search for a hotel
radius = 10000
params = {
    "radius": radius,
    "categories": "accommodation.hotel",
    "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():
    lat = row["Lat"]
    lng = row["Lng"]

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

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

    # Make an API request using the params dictionary
    name_address = requests.get(base_url, params=params)

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

    # Grab the first hotel from the results and store the name in the hotel_df DataFrame
    try:
        hotel_df.loc[index, "Hotel Name"] = name_address["features"][0]["properties"]["name"]
    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 Name']}")

# Display sample data
hotel_df
```

Starting hotel search  
 busselton - nearest hotel: Broadwater Beach Resort  
 tezu - nearest hotel: No hotel found  
 ayolas - nearest hotel: Hotel Nacional de Turismo Ayolas  
 tura - nearest hotel: No hotel found  
 trelew - nearest hotel: Patagonia Suites & Apart  
 yaan - nearest hotel: No hotel found  
 remanso - nearest hotel: No hotel found  
 fram - nearest hotel: No hotel found  
 banda aceh - nearest hotel: #acnindiafy21  
 veraval - nearest hotel: Shyam  
 youkounkoun - nearest hotel: No hotel found  
 pasighat - nearest hotel: Aane Hotel  
 akyab - nearest hotel: Yuzana Aung Motel 1  
 santiago del estero - nearest hotel: apart arturo prat  
 tha bo - nearest hotel: No hotel found  
 ron phibun - nearest hotel: No hotel found

Out[7]:

	City	Country	Lat	Lng	Humidity	Hotel Name
13	busselton	AU	-33.6500	115.3333	35	Broadwater Beach Resort
56	tezu	IN	27.9167	96.1667	33	No hotel found
71	ayolas	PY	-27.4000	-56.9000	32	Hotel Nacional de Turismo Ayolas
99	tura	IN	25.5198	90.2201	39	No hotel found
155	trelew	AR	-43.2490	-65.3051	41	Patagonia Suites & Apart
211	yaan	NG	7.3833	8.5667	37	No hotel found
213	remanso	BR	-9.6179	-42.0831	46	No hotel found
253	fram	PY	-26.9833	-55.8833	37	No hotel found
263	banda aceh	IN	25.4833	80.3333	32	#acnindiafy21
320	veraval	IN	20.9000	70.3667	33	Shyam
366	youkounkoun	GN	12.5333	-13.1333	35	No hotel found
376	pasighat	IN	28.0667	95.3333	40	Aane Hotel
424	akyab	MM	20.1500	92.9000	41	Yuzana Aung Motel 1
483	santiago del estero	CL	-33.4569	-70.6483	49	apart arturo prat
507	tha bo	LA	14.9604	105.7212	36	No hotel found
536	ron phibun	IN	15.6667	75.7333	42	No hotel found

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

In [8]:

```
%%capture --no-display

# Configure the map plot
map_plot_hotel = hotel_df.hvplot.points("Lng", "Lat", geo = True, tiles = "OSM",
                                         frame_width = 700, frame_height = 500,
                                         size = "Humidity", color = "City", scale=1.5,
                                         hover_cols=['Hotel Name', 'Country'])

# Display the map
map_plot_hotel
```

Out[8]:

