VacationPy

Note

- Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

In [1]:

```
# Dependencies and Setup
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import requests
import gmaps
import os

# Import API key
from api_keys import g_key
```

Store Part I results into DataFrame

- Load the csv exported in Part I to a DataFrame

In [2]:

```
cities_df = pd.read_csv('../WeatherPy/output_data/cities.csv')

cities_df.head()
```

Out[2]:

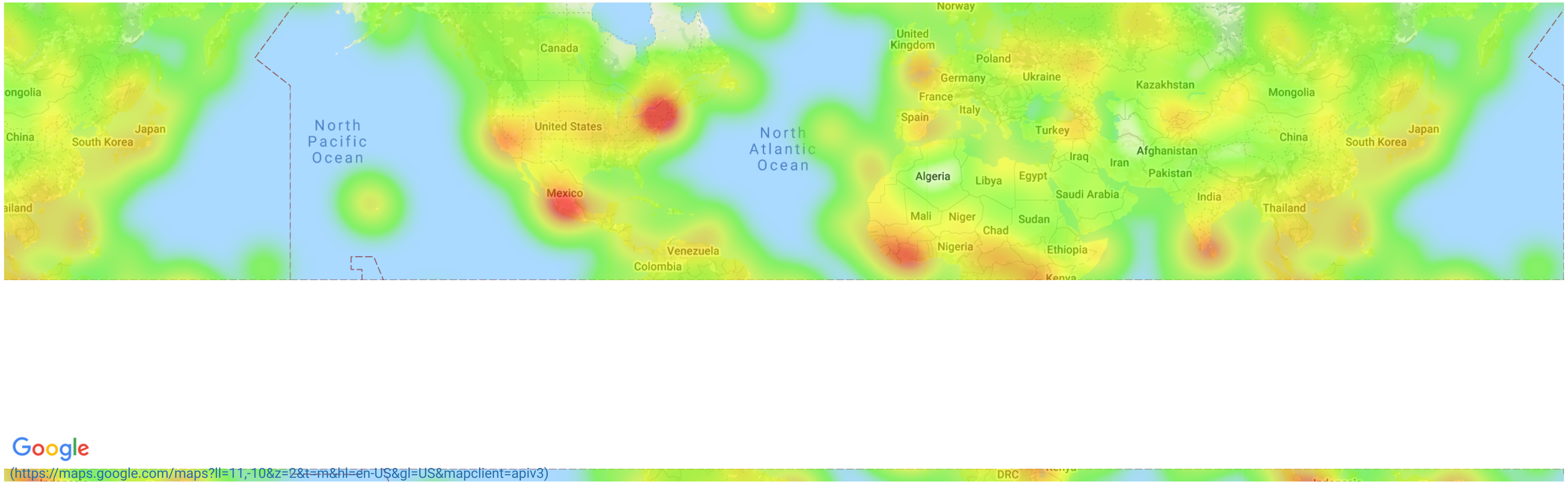
	Unnamed: 0	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wind Speed
0	0	cherskiy	0	RU	1583667319	76	68.75	161.30	-10.21	9.66
1	1	jamestown	90	US	1583667319	74	42.10	-79.24	41.00	10.29
2	2	bang saphan	2	TH	1583667319	74	11.21	99.51	87.80	4.70
3	3	vaini	20	TO	1583667319	94	-21.20	-175.20	71.60	3.36
4	4	ushuaia	75	AR	1583667106	81	-54.80	-68.30	44.60	32.21

Humidity Heatmap

- Configure gmaps.
- Use the Lat and Lng as locations and Humidity as the weight.
- Add Heatmap layer to map.

In [3]:
```
gmaps.configure(api_key=g_key)
# Store latitude and longitude in locations
locations = cities_df[["Lat", "Lng"]]
```

In [4]:
```
# Plot Heatmap
fig = gmaps.figure()
# Create heat layer
heat_layer = gmaps.heatmap_layer(locations,dissipating=False, max_intensity=8)

# Display figure

fig = gmaps.figure(center=(11.0, -10.0), zoom_level=2)

# Add layer
fig.add_layer(heat_layer)
fig
```



(https://maps.google.com/maps?ll=11,-10&z=2&t=m&hl=en-US&gl=US&mapclient=apiv3)

Map data ©2020

Create new DataFrame fitting weather criteria

- Narrow down the cities to fit weather conditions.
- Drop any rows will null values.

In [5]:

```
# cities_of_interest selection
cities_of_interest_df = cities_df.loc[(cities_df["Max Temp"] <= 78) &
                                      (cities_df["Max Temp"] >= 70) &
                                      (cities_df["Humidity"] <= 60) &
                                      (cities_df["Wind Speed"] <= 10.5) ]

cities_of_interest_df = cities_of_interest_df.dropna()

cities_of_interest_df
```

Out[5]:

	Unnamed: 0	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wind Speed
14	14	cabo san lucas	90	MX	1583667320	40	22.89	-109.91	70.00	6.93
180	180	tura	4	IN	1583667331	41	25.52	90.22	75.47	4.76
214	214	erzin	1	TR	1583667333	53	36.96	36.20	75.20	2.24
227	227	kahului	90	US	1583667103	60	20.89	-156.47	71.60	10.29
240	240	pisco	33	PE	1583667335	30	-13.70	-76.22	71.60	3.36

Hotel Map

- Store into variable named `hotel_df` .
- Add a "Hotel Name" column to the DataFrame.
- Set parameters to search for hotels with 5000 meters.
- Hit the Google Places API for each city's coordinates.
- Store the first Hotel result into the DataFrame.
- Plot markers on top of the heatmap.

In [6]:

```
params = {
    "radius": 5000,
    "types": "hotel",
    "keyword": "hotels",
    "key": g_key
}

for index, row in cities_of_interest_df.iterrows():
    # get lat, lng from df
    lat = row["Lat"]
    lng = row["Lng"]

    params["location"] = f"{lat},{lng}"

    # Use the nearbysearch Google Place API
    base_url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json"
    name_address = requests.get(base_url, params=params)
    # convert to json
    name_address = name_address.json()

    try:
        cities_of_interest_df.loc[index, "Hotel Name"] = name_address["results"][0]["name"]
    except (KeyError, IndexError):
        pass

cities_of_interest_df.reset_index(drop = True)
```

Out[6]:

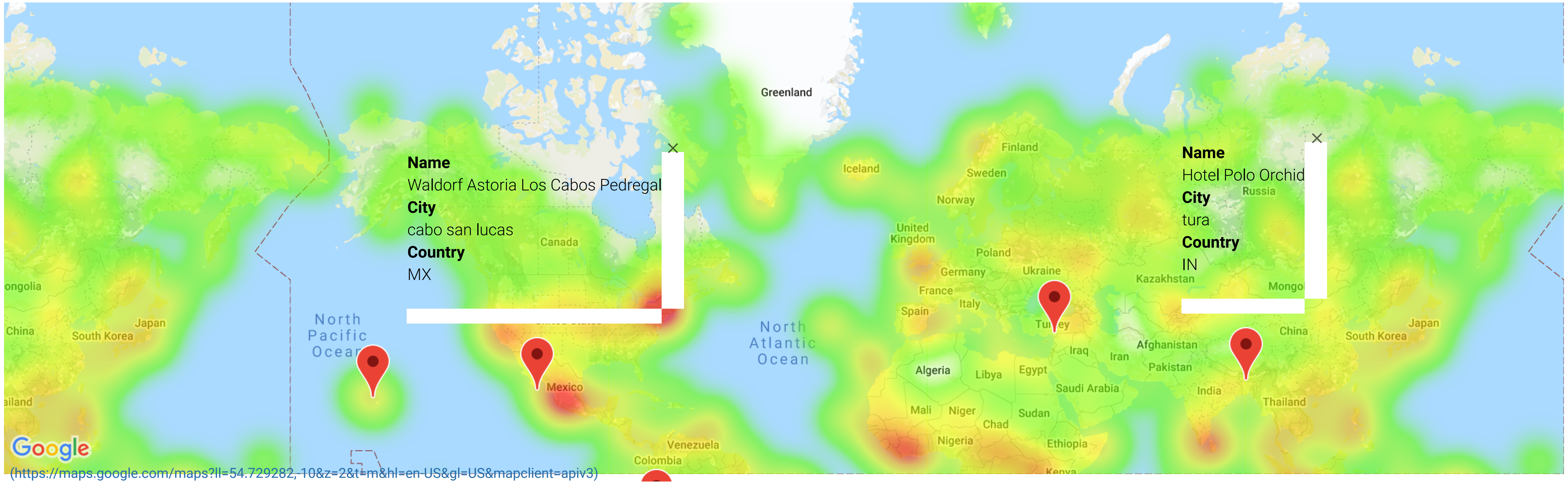
	Unnamed: 0	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wind Speed	Hotel Name
0	14	cabo san lucas	90	MX	1583667320	40	22.89	-109.91	70.00	6.93	Waldorf Astoria Los Cabos Pedregal
1	180	tura	4	IN	1583667331	41	25.52	90.22	75.47	4.76	Hotel Polo Orchid
2	214	erzin	1	TR	1583667333	53	36.96	36.20	75.20	2.24	Hattusa Vacation Thermal Club Erzin
3	227	kahului	90	US	1583667103	60	20.89	-156.47	71.60	10.29	Maui Seaside Hotel
4	240	pisco	33	PE	1583667335	30	-13.70	-76.22	71.60	3.36	Embassy Beach

In [7]:

```
# NOTE: Do not change any of the code in this cell

# Using the template add the hotel marks to the heatmap
info_box_template = """
<dl>
<dt>Name</dt><dd>{Hotel Name}</dd>
<dt>City</dt><dd>{City}</dd>
<dt>Country</dt><dd>{Country}</dd>
</dl>
"""
# Store the DataFrame Row
# NOTE: be sure to update with your DataFrame name
hotel_info = [info_box_template.format(**row) for index, row in cities_of_interest_df.iterrows()]
locations_of_interest = cities_of_interest_df[["Lat", "Lng"]]
```

In [9]:

```
# Add marker layer ontop of heat map

# Plot Heatmap
fig = gmaps.figure()

# Create heat layer and markers
heat_layer = gmaps.heatmap_layer(locations,dissipating=False, max_intensity=8)
markers = gmaps.marker_layer(locations_of_interest,info_box_content =hotel_info)

# Display figure

fig = gmaps.figure(center=(11.0, -10.0), zoom_level=2)

# Add layer
fig.add_layer(heat_layer)
fig.add_layer(markers)

# Display Map
fig
```



Name
Waldorf Astoria Los Cabos Pedregal
City
cabo san lucas
Country
MX

Name
Hotel Polo Orchid
City
tura
Country
IN

(https://maps.google.com/maps?ll=54.729282,-10&z=2&t=m&hl=en-US&gl=US&mapclient=apiv3)

Map data ©2020

In [ ]: