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 [162]:

```
# 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
```

In [163]:

```
os.getcwd()
```

Out[163]:

```
'/Users/huiyingzheng/Desktop/GT Databootcamp/python-api-challenge/We
atherPy'
```

Store Part I results into DataFrame

- Load the csv exported in Part I to a DataFrame

In [164]:

```
# Define the path to get the excel data
filepath = os.path.join("..", "output_data", "weather_data.csv")
map_weather = pd.read_csv(filepath)
map_weather.head()
```

Out[164]:

	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wind Speed
0	Byron Bay	75	AU	1587265703	65	-28.65	153.62	298.15	3.10
1	Albany	1	US	1587265932	44	42.60	-73.97	277.59	0.45
2	Snåase	99	NO	1587265997	85	64.25	12.38	276.19	1.49
3	Thompson	5	CA	1587265933	40	55.74	-97.86	263.15	3.10
4	Bandarbeyla	73	SO	1587266208	69	9.49	50.81	300.31	2.68

Humidity Heatmap

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

In [165]:

```
# Create coordinates list
locations = map_weather[["Lat", "Lng"]]

figure_layout = {
    'width': '400px',
    'height': '300px',
    'border': '1px solid black',
    'padding': '1px',
    'margin': '0 auto 0 auto'
}

weight_factor = map_weather["Humidity"]

fig = gmaps.figure(map_type="HYBRID")


# Create heat layer
heat_layer = gmaps.heatmap_layer(locations, weights=weight_factor,
                                 dissipating=False, max_intensity=weight_factor.
max(),
                                 point_radius=1.3)
# Add layer
fig.add_layer(heat_layer)

# Display figure
fig
```

Create new DataFrame fitting weather criteria

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

In [166]:

```
# A max temperature lower than 80 degrees but higher than 70.
# Wind speed less than 10 mph.
# Zero cloudiness.
# Drop any rows that don't contain all three conditions. You want to be sure the
weather is ideal.

# Because open weather website uses Kelvin unit for temperature, need to create
 fahrenheit
map_weather["Max Temp (F)"] = (map_weather["Max Temp"] - 273.15) * 9/5 + 32

map_weather_rd = map_weather.loc[(map_weather["Max Temp (F)"] >= 70) & (map_weat
her["Max Temp (F)"] <= 80) &
                                 (map_weather["Wind Speed"]<=10) & (map_weather[
"Cloudiness"] == 0)]
map_weather_rd.head()
map_weather_rd
```

Out[166]:

	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wind Speed	M Te
125	Sabha	0	LY	1587266081	19	27.04	14.43	295.36	6.83	71.9
187	Sechura	0	PE	1587265964	76	-5.56	-80.82	296.29	5.68	73.6
436	Riohacha	0	CO	1587266270	73	11.54	-72.91	299.50	4.35	79.4
453	Samaná	0	DO	1587266273	86	19.21	-69.34	296.65	2.73	74.3
526	Alta Floresta	0	BR	1587266281	89	-9.88	-56.09	294.59	1.55	70.5

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 [167]:

```
# get city name for each row
hotel_city = list(map_weather_rd['City'])[0]

# Build the endpoint URL
target_url = ('https://maps.googleapis.com/maps/api/geocode/json?'
    'address={0}&key={1}').format(hotel_city, g_key)

geo_data = requests.get(target_url).json()
city_lat = geo_data['results'][0]['geometry']['location']['lat']
city_lng = geo_data['results'][0]['geometry']['location']['lng']

params['location'] = f"{city_lat},{city_lng}"

# add keyword to params dict
# params['keyword'] = restr_type

# assemble url and make API request
print(f"Retrieving Results for Index {index}: {hotel_city}.")
response = requests.get(base_url, params=params).json()

# extract results
results = response['results']
```

```
Retrieving Results for Index 526: Sabha.
```

In [168]:

```
hotel_df = pd.DataFrame(map_weather_rd[['City','Lat','Lng']])
hotel_df['Hotel Name'] = ""
hotel_df['City Lat'] = ""
hotel_df['City Lng'] = ""

# use iterrows to iterate through pandas dataframe
for index, row in hotel_df.iterrows():

    # get city name for each row
    hotel_city = row['City']

    # Build the endpoint URL
    target_url = ('https://maps.googleapis.com/maps/api/geocode/json?'
        'address={0}&key={1}').format(hotel_city, g_key)

    geo_data = requests.get(target_url).json()
    city_lat = geo_data['results'][0]['geometry']['location']['lat']
    city_lng = geo_data['results'][0]['geometry']['location']['lng']

    params['location'] = f"{city_lat},{city_lng}"

    # add keyword to params dict
    # params['keyword'] = restr_type

    # assemble url and make API request
    print(f"Retrieving Results for Index {index}: {hotel_city}.")
    response = requests.get(base_url, params=params).json()

    # extract results
    results = response['results']

    try:
        print(f"Closest {hotel_city} hotel is {results[0]['name']}.")

        hotel_df.loc[index, 'Hotel Name'] = hotel_city
        hotel_df.loc[index, 'City Lat'] = city_lat
        hotel_df.loc[index, 'City Lng'] = city_lng

    except (KeyError, IndexError):
        print("Missing field/result... skipping.")

    print("------------")
```

```
Retrieving Results for Index 125: Sabha.
Closest Sabha hotel is مركز الفؤاد للقلب والشرايين.
------------
Retrieving Results for Index 187: Sechura.
Closest Sechura hotel is Restaurant El Club.
------------
Retrieving Results for Index 436: Riohacha.
Closest Riohacha hotel is FEPASDE Productos & Servicios S.A..
------------
Retrieving Results for Index 453: Samaná.
Closest Samaná hotel is Kelly Copy.
------------
Retrieving Results for Index 526: Alta Floresta.
Closest Alta Floresta hotel is Stanichesch's House.
------------
```

In [169]:

```
hotel_df.head()
```

Out[169]:

	City	Lat	Lng	Hotel Name	City Lat	City Lng
125	Sabha	27.04	14.43	Sabha	27.0365	14.429
187	Sechura	-5.56	-80.82	Sechura	-5.56224	-80.8188
436	Riohacha	11.54	-72.91	Riohacha	11.5384	-72.9168
453	Samaná	19.21	-69.34	Samaná	19.2058	-69.3363
526	Alta Floresta	-9.88	-56.09	Alta Floresta	-9.86722	-56.087

In [170]:

```
# Create heat layer

locations2 = hotel_df[["City Lat", "City Lng"]]
cities = hotel_df["City"]

markers = gmaps.marker_layer(locations2,
                             info_box_content=[f"Next vocation hotel candidat
e: {city}" for city in cities])


fig.add_layer(markers)
fig
```

In [ ]:

```
# 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 narrowed_city_d
f.iterrows()]
locations = hotel_df[["Lat", "Lng"]]
```

In [ ]:

```
# Add marker layer ontop of heat map


# Display Map
```