VacationPy

Note

- Keep an eye on your API usage. Use https://developers.google.com/maps/reporting/gmp-reporting (https://developers.google.com/maps/reporting/gmp-reporting) as reference for how to monitor your usage and billing.
- 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 [3]:
```
# 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 [2]:
```
#!jupyter nbextension enable --py widgetsnbextension
```

```
Enabling notebook extension jupyter-js-widgets/extension...
      - Validating: ok
```

Store Part I results into DataFrame

- Load the csv exported in Part I to a DataFrame

In [4]:
```
airport_df = pd.read_csv('..//WeatherPy//output_data//cities.csv')
```

Humidity Heatmap

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

In [5]:

```
gmaps.configure(api_key=g_key)

locations = airport_df[["Latitude", "Longitude"]]
humidity = airport_df['Humidity']
airport_df
```

Out[5]:

	Unnamed: 0	City	Latitude	Longitude	Max Temperature	Humidity	Cloudiness	Wind Speed	Country
0	0	nyurba	63.2842	118.3319	25.65	81	93	11.56	RU
1	1	upernavik	72.7868	-56.1549	12.47	62	86	4.43	GL
2	2	chipinge	-20.1883	32.6236	60.76	81	41	5.59	ZW
3	3	airai	-8.9266	125.4092	59.43	87	75	3.24	TL
4	4	dikson	73.5069	80.5464	8.73	97	35	9.86	RU
...	...	...	...	...	...	...	...	...	...
575	575	santa maria	-29.6842	-53.8069	60.06	98	100	2.57	BR
576	576	harper	4.3750	-7.7169	84.56	73	99	6.49	LR
577	577	kamaishi	39.2667	141.8833	46.11	95	100	3.60	JP
578	578	tazovskiy	67.4667	78.7000	18.91	93	58	9.51	RU
579	579	trinidad	-14.8333	-64.9000	87.78	66	75	6.91	BO

580 rows × 10 columns

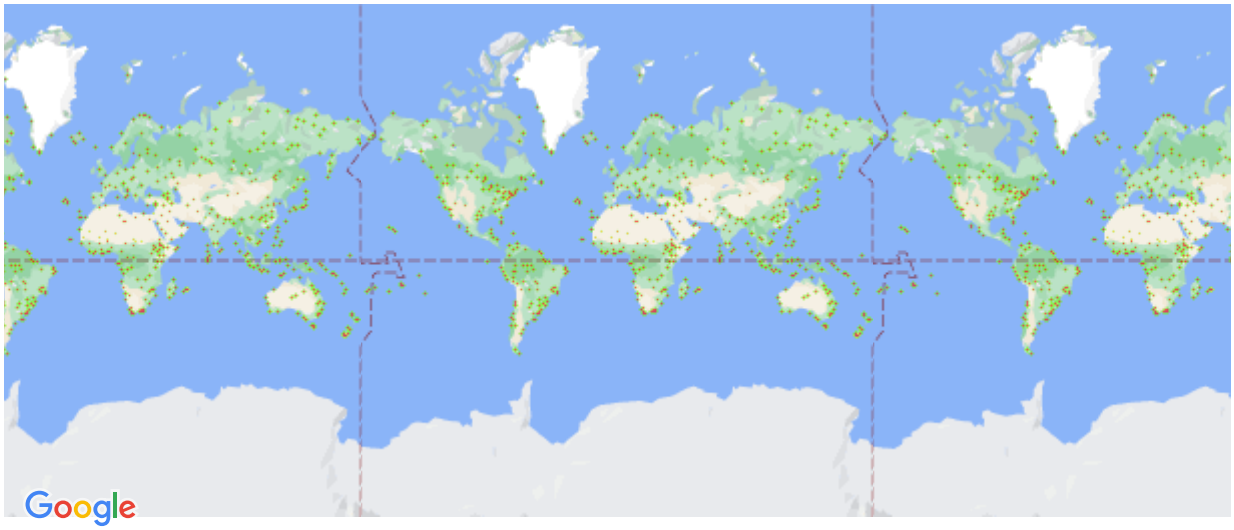
In [6]:

```
# Plot Heatmap
fig = gmaps.figure()

# Create heat layer
heat_layer = gmaps.heatmap_layer(locations, weights=humidity,
                                 dissipating=False, max_intensity=10,
                                 point_radius=1)

# Add layer
fig.add_layer(heat_layer)

# Display figure
fig
```



(https://maps.google.com/maps?ll=51.37038,0&z=0&t=m&hl=en-US&gl=US&mapclient=apiv3) Map data ©2022

Create new DataFrame fitting weather criteria

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

In [7]:

```
#airport_df = pd.DataFrame()
airport_df.dropna()
airport_df['Hotel Name'] = ''
airport_df
```

Out[7]:

	Unnamed: 0	City	Latitude	Longitude	Max Temperature	Humidity	Cloudiness	Wind Speed	Country
0	0	nyurba	63.2842	118.3319	25.65	81	93	11.56	RU
1	1	upernavik	72.7868	-56.1549	12.47	62	86	4.43	GL
2	2	chipinge	-20.1883	32.6236	60.76	81	41	5.59	ZW
3	3	airai	-8.9266	125.4092	59.43	87	75	3.24	TL
4	4	dikson	73.5069	80.5464	8.73	97	35	9.86	RU
...	...	...	...	...	...	...	...	...	...
575	575	santa maria	-29.6842	-53.8069	60.06	98	100	2.57	BR
576	576	harper	4.3750	-7.7169	84.56	73	99	6.49	LR
577	577	kamaishi	39.2667	141.8833	46.11	95	100	3.60	JP
578	578	tazovskiy	67.4667	78.7000	18.91	93	58	9.51	RU
579	579	trinidad	-14.8333	-64.9000	87.78	66	75	6.91	BO

580 rows × 11 columns

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 [8]:

```
# geocoordinates
target_radius = 5000
target_type = "hotel"

# set up a parameters dictionary
params = {
    "radius": 5000,
    "types": "lodging",
    "key": g_key
}

for index, row in airport_df.iterrows():
    lat = row["Latitude"]
    lng = row["Longitude"]

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

    # base url
    base_url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json

    # run a request using our params dictionary
    response = requests.get(base_url, params=params)

    response = response.json()
#     print(response['results'])

    try:
        airport_df.loc[index, "Hotel Name"] = response["results"][0]["name"]

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

airport_df
```

```
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
```

```
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
Missing field/result... skipping.
```

Out[8]:

	Unnamed: 0	City	Latitude	Longitude	Max Temperature	Humidity	Cloudiness	Wind Speed	Cou
0	0	nyurba	63.2842	118.3319	25.65	81	93	11.56	
1	1	upernavik	72.7868	-56.1549	12.47	62	86	4.43	
2	2	chipinge	-20.1883	32.6236	60.76	81	41	5.59	
3	3	airai	-8.9266	125.4092	59.43	87	75	3.24	
4	4	dikson	73.5069	80.5464	8.73	97	35	9.86	
...	...	...	...	...	...	...	...	...	
575	575	santa maria	-29.6842	-53.8069	60.06	98	100	2.57	
576	576	harper	4.3750	-7.7169	84.56	73	99	6.49	

	Unnamed: 0	City	Latitude	Longitude	Max Temperature	Humidity	Cloudiness	Wind Speed	Cou
577	577	kamaishi	39.2667	141.8833	46.11	95	100	3.60	
578	578	tazovskiy	67.4667	78.7000	18.91	93	58	9.51	
579	579	trinidad	-14.8333	-64.9000	87.78	66	75	6.91	

580 rows × 11 columns

In [9]:

```
# 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 airport_df.i
locations = airport_df[["Latitude", "Longitude"]]
```

In [10]:

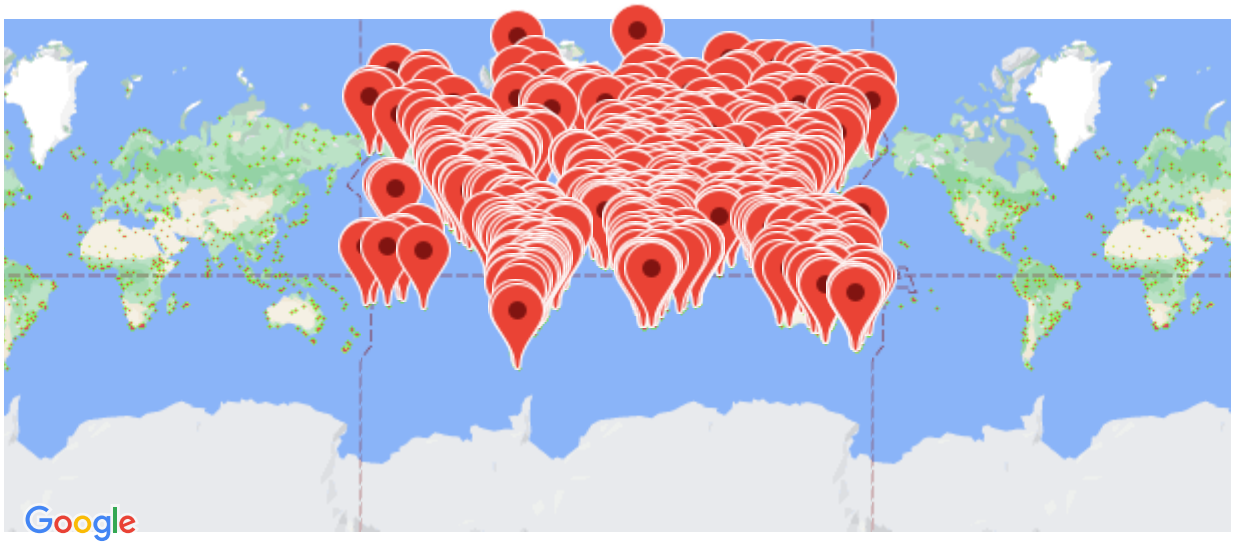
```
# Add marker layer ontop of heat map
# Create a combined map
heat_layer = gmaps.marker_layer(locations, info_box_content=hotel_info)

fig.add_layer(heat_layer)

fig

# Display figure

```



(https://maps.google.com/maps?ll=51.37038,0&z=0&t=m&hl=en-US&gl=US&mapclient=apiv3) Map data ©2022

In [ ]:

```
1
```