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 [2]:

```
# 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 [3]:

```
df = pd.read_csv("cities.csv")
df.set_index("City_ID",inplace =True)
df.head()
```

Out[3]:

	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wi Spe
City_ID									
0	ostrovnoy	2	RU	1558378754	72	68.05	39.51	37.50	7.
1	mahebourg	75	MU	1558378503	74	-20.41	57.70	78.80	11.
2	qaanaaq	25	GL	1558378755	73	77.48	-69.36	22.20	2.
3	zhuhai	0	CN	1558378755	39	40.71	112.04	44.60	4.
4	cape town	20	ZA	1558378755	76	-33.93	18.42	55.99	8.

Humidity Heatmap

- Configure gmaps.
- Use the Lat and Lng as locations and Humidity as the weight

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

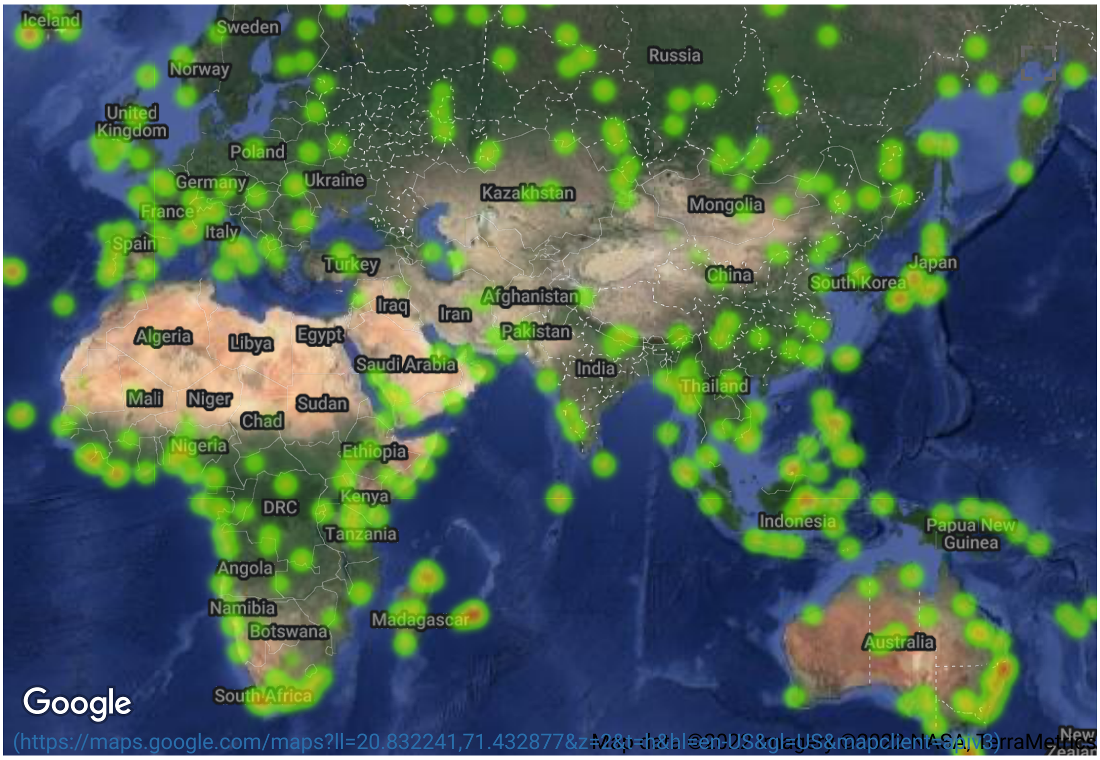
In [4]:

```
gmaps.configure(api_key = g_key)
fig = gmaps.figure(map_type = "HYBRID")
locations = df[["Lat","Lng"]]
weight = df["Humidity"]
#heatmap_layer = gmaps.heatmap_layer(locations)
#fig.add_layer(heatmap_layer)

fig.add_layer(gmaps.heatmap_layer(locations,weights=weight))
```

In [5]:

```
fig
```



Create new DataFrame fitting weather criteria

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

In [6]:

```



df = df.dropna()
```

In [ ]:

```

```

Hotel Map

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 [7]:

```
hotel_df = df.copy()
hotel_df.head()
```

Out[7]:

City_ID	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wi Spe
0	ostrovnoy	2	RU	1558378754	72	68.05	39.51	37.50	7.
1	mahebourg	75	MU	1558378503	74	-20.41	57.70	78.80	11.
2	qaanaaq	25	GL	1558378755	73	77.48	-69.36	22.20	2.
3	zhuhai	0	CN	1558378755	39	40.71	112.04	44.60	4.
4	cape town	20	ZA	1558378755	76	-33.93	18.42	55.99	8.

In [8]:

```
hotel_df = hotel_df[hotel_df["Country"] =="US"]
hotel_df["Hotel Name"] = hotel_df["City"]
```

In [ ]:

```

```

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 hot
locations = hotel_df[["Lat", "Lng"]]
```

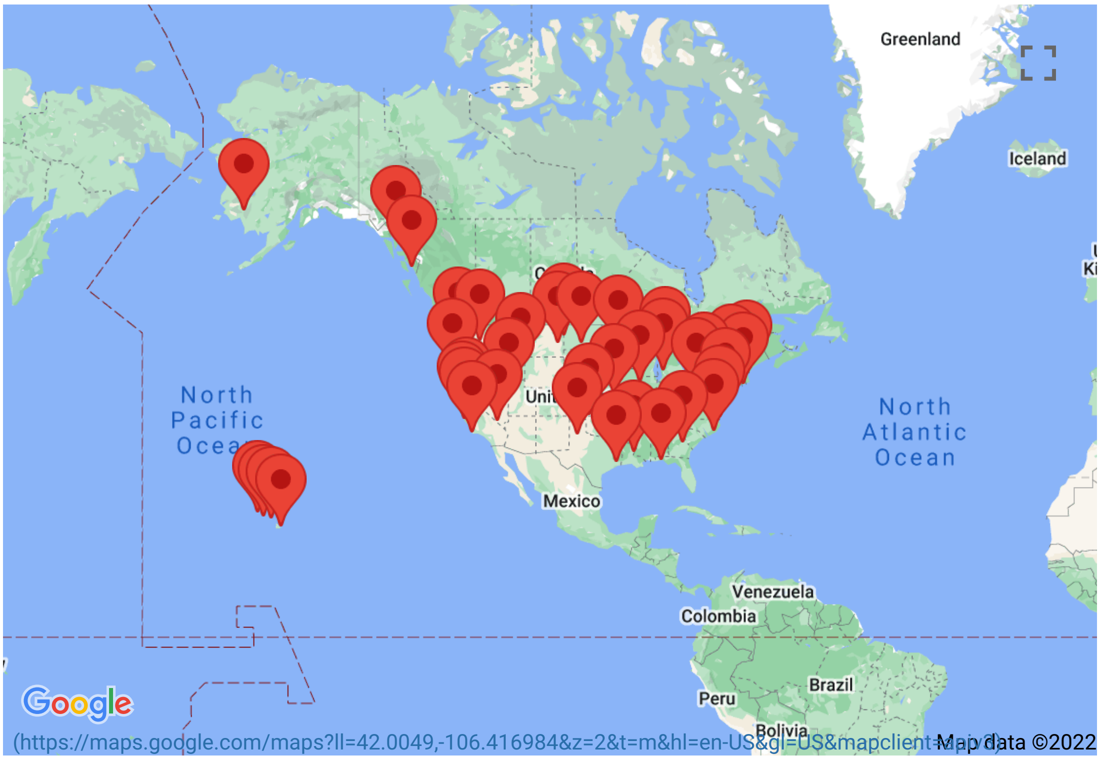
In [ ]:

```
# Add marker layer ontop of heat map


# Display figure
```

In [10]:

```
marker_layer = gmaps.marker_layer(locations, info_box_content=hotel_
fig = gmaps.figure()
fig.add_layer(marker_layer)
fig
```


Greenland
Iceland
North Pacific Ocean
North Atlantic Ocean
Mexico
Venezuela
Colombia
Peru
Brazil
Bolivia
Google
(https://maps.google.com/maps?ll=42.0049,-106.416984&z=2&t=m&hl=en-US&gl=US&mapclient=apiv3)
Map data ©2022