

VacationPy

Note

- Keep an eye on your API usage. Use <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 [9]: # Dependencies and Setup
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import requests
import gmaps
import os
import json

# Import API key
from api_keys import g_key

# Configure gmaps
gmaps.configure(api_key=g_key)
```

Store Part I results into DataFrame

- Load the csv exported in Part I to a DataFrame

```
In [10]: Cities_data = pd.read_csv('../WeatherPy/weather_data.csv')
Cities_data
```

Out[10]:

	City	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country	Date
0	Le Moule	16.33	-61.35	75.20	88	100	2.24	GP	1603956416
1	Palmer	42.16	-72.33	46.99	100	90	1.48	US	1603956416
2	Yanam	16.73	82.22	89.58	40	5	7.36	IN	1603956418
3	Souillac	-20.52	57.52	80.01	57	17	8.99	MU	1603956418
4	Nizhniy Baskunchak	48.22	46.83	50.67	47	66	16.91	RU	1603956419
...
394	Saint-Pierre	-21.34	55.48	75.20	73	75	28.86	RE	1603955843
395	Port Alfred	-33.59	26.89	69.58	76	100	8.08	ZA	1603956052
396	Ushuaia	-54.80	-68.30	42.80	81	40	36.91	AR	1603955915
397	Cherskiy	68.75	161.30	5.18	92	80	3.49	RU	1603956447
398	Castro	-24.79	-50.01	58.50	50	0	1.92	BR	1603956077

399 rows × 9 columns

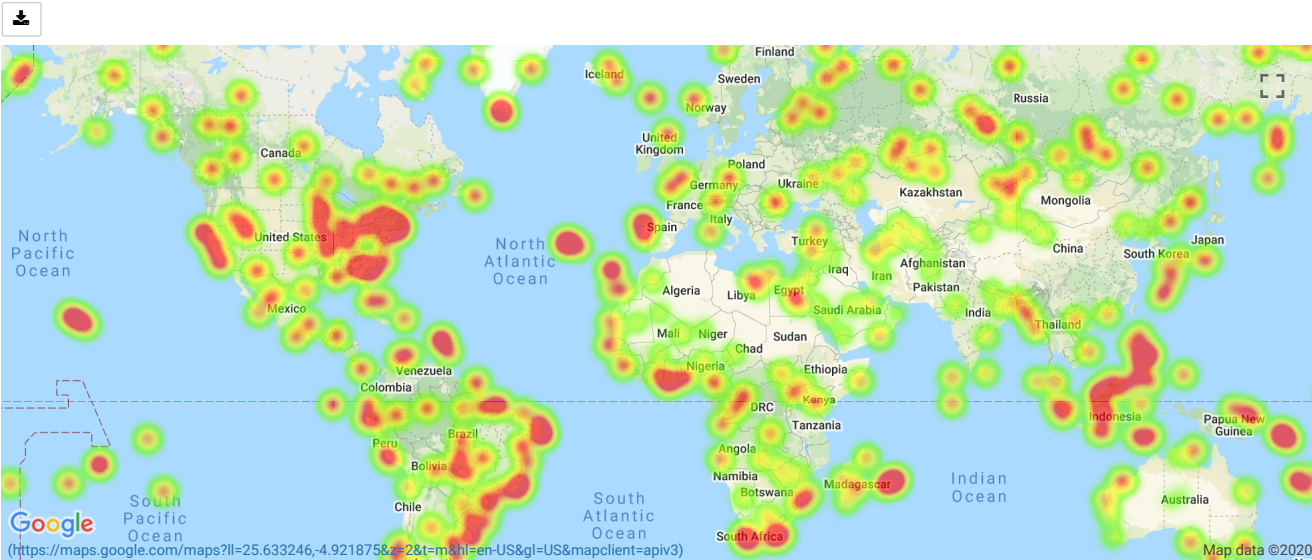
Humidity Heatmap

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

```
In [11]: # Store Latitude and Longitude in Locations
locations = Cities_data[["Lat", "Lng"]]
humidity = Cities_data["Humidity"].astype(float)
max_humidity = humidity.max()
```

```
In [12]: fig = gmaps.figure()
# Assign the heat layer to a variable
heat_layer = gmaps.heatmap_layer(locations, weights=humidity,
                                dissipating=False, max_intensity=max_humidity,
                                point_radius=4)

# Add the Layer to the map
fig.add_layer(heat_layer)
fig
```



Create new DataFrame fitting weather criteria

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

```
In [13]: Filter_Cities_data = Cities_data.loc[(Cities_data['Max Temp'] < 80) & (Cities_data['Max Temp'] > 70) &
(Cities_data['Cloudiness'] == 0) & (Cities_data['Wind Speed'] < 10)]

Filter_Cities_data = Filter_Cities_data.dropna(how='any')

Filter_Cities_data = Filter_Cities_data.reset_index()

del Filter_Cities_data['index']

Filter_Cities_data
```

Out[13]:

	City	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country	Date
0	Arlit	18.74	7.39	78.85	16	0	5.95	NE	1603956100
1	Bäglung	28.27	83.59	78.96	25	0	5.30	NP	1603956438
2	Esna	25.29	32.55	75.20	69	0	2.24	EG	1603956443
3	Arraial do Cabo	-22.97	-42.03	73.40	100	0	2.24	BR	1603955849
4	Gao	16.64	1.64	78.75	16	0	8.68	ML	1603956470
5	São João da Barra	-21.64	-41.05	74.71	93	0	8.01	BR	1603956092
6	João Câmara	-5.54	-35.82	73.40	88	0	4.70	BR	1603956475
7	Najrân	17.49	44.13	75.20	21	0	4.70	SA	1603956493
8	Cayenne	4.93	-52.33	77.00	100	0	8.88	GF	1603955830
9	Riyadh	24.69	46.72	75.20	29	0	2.98	SA	1603956064
10	Mutare	-18.97	32.67	76.15	32	0	9.44	ZW	1603956504
11	Fereydūnkenār	36.69	52.52	71.60	68	0	2.24	IR	1603956524
12	Kidal	18.44	1.41	76.59	17	0	8.21	ML	1603956536
13	Kaduna	10.52	7.44	72.81	37	0	5.75	NG	1603956542

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 [15]: Hotel_list = []
drop_list= []

for list in range(len(Filter_Cities_data)):
    lat = Filter_Cities_data.loc[list]['Lat']
    lng = Filter_Cities_data.loc[list]['Lng']

    params = {
        "location": f"{lat},{lng}",
        "radius": 5000,
        "types" : "hotel",
        "key": g_key
    }

    base_url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json"
    requested = requests.get(base_url, params=params)
    data = requested.json()
    try:
        Hotel_list.append(data['results'][0]['name'])
    except:
        Hotel_list.append("")

Filter_Cities_data["Hotel Name"] = Hotel_list

index_names = Filter_Cities_data[ (Filter_Cities_data['Hotel Name'] == "")].index

Filter_Cities_update = Filter_Cities_data.drop(index_names)
Filter_Cities_update
```

Out[15]:

	City	Lat	Lng	Max Temp	Humidity	Cloudiness	Wind Speed	Country	Date	Hotel Name
0	Arlit	18.74	7.39	78.85	16	0	5.95	NE	1603956100	Arlit
1	Bāglung	28.27	83.59	78.96	25	0	5.30	NP	1603956438	Baglung
2	Esna	25.29	32.55	75.20	69	0	2.24	EG	1603956443	Esna
3	Arraial do Cabo	-22.97	-42.03	73.40	100	0	2.24	BR	1603955849	Cabo Frio
5	São João da Barra	-21.64	-41.05	74.71	93	0	8.01	BR	1603956092	Grussaí
6	João Câmara	-5.54	-35.82	73.40	88	0	4.70	BR	1603956475	João Câmara, Rio Grande do Norte
7	Najrān	17.49	44.13	75.20	21	0	4.70	SA	1603956493	Najran
8	Cayenne	4.93	-52.33	77.00	100	0	8.88	GF	1603955830	Cayenne
9	Riyadh	24.69	46.72	75.20	29	0	2.98	SA	1603956064	Riyadh
10	Mutare	-18.97	32.67	76.15	32	0	9.44	ZW	1603956504	Mutare
11	Fereydūnkenār	36.69	52.52	71.60	68	0	2.24	IR	1603956524	Fereydunkenar
12	Kidal	18.44	1.41	76.59	17	0	8.21	ML	1603956536	Kidal
13	Kaduna	10.52	7.44	72.81	37	0	5.75	NG	1603956542	Kaduna

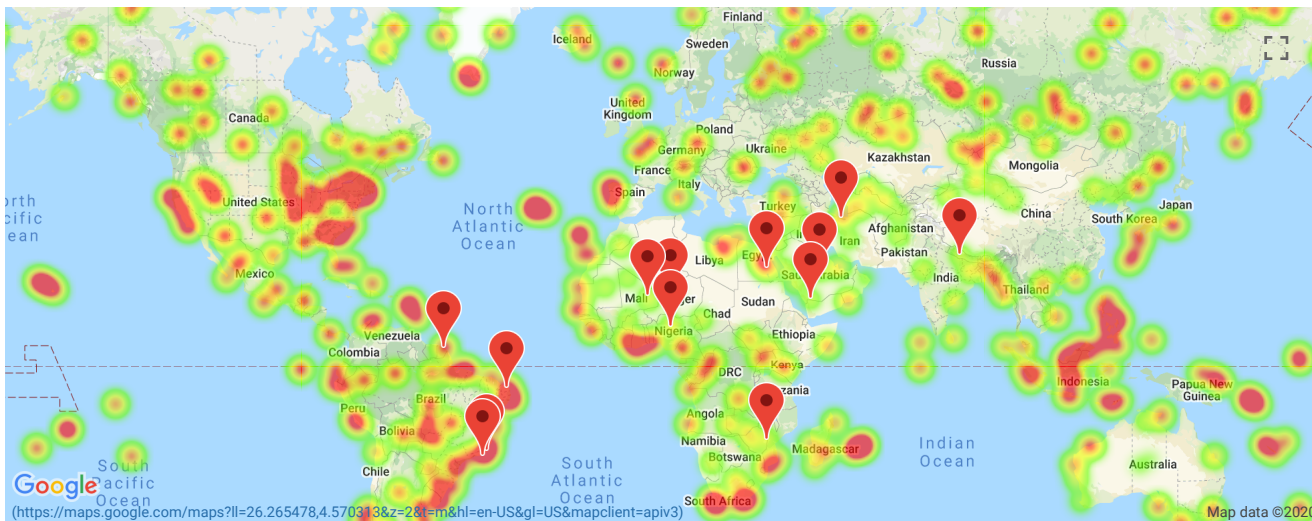
```
In [16]: # NOTE: Do not change any of the code in this cell

# 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 Filter_Cities_update.iterrows()]
locations = Filter_Cities_update[["Lat", "Lng"]]
```

```
In [17]: # Add marker Layer ontop of heat map
markers = gmaps.marker_layer(locations)

# Display figure
fig.add_layer(markers)
fig
```



```
In [ ]:
```

(data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAA9gAAAGCCAYAAAAboGvJAAAgAEIEQVR4Xuy9Z4wkS1Y2/GRGpCnXfvzMNeuAZVIYvH0B4T1C2B8IAYsVTICB8PxAWIGEFwDifc
(data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAA9gAAAGCCAYAAAAboGvJAAAgAEIEQVR4Xuy9d4wkSVY//smltFXfvzMzrzHlc5vPceDnMgOCRACAQChJBAwhsJISEQEkL8wYC