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 json
from pprint import pprint

# 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]:
```
# load output_data_file from WeatherPy located in the folder output data
output_data_file = 'output_data/cities.csv'
```

In [3]:
```
#Create a data frame for the cities
cities_df = pd.read_csv(output_data_file)

weather_data_frame = cities_df.drop(columns=["Unnamed: 0"])
weather_data_frame
```

Out[3]:

	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wind Speed
0	jamestown	90	US	1579998908	97	42.10	-79.24	36.00	5.82
1	ulladulla	29	AU	1579998970	49	-35.35	150.47	89.60	8.05
2	batagay-alyta	0	RU	1579998962	54	67.80	130.41	-34.38	4.38
3	puerto ayora	75	EC	1579999060	78	-0.74	-90.35	78.80	8.05
4	ancud	100	CL	1579998934	68	-41.87	-73.82	66.20	6.93
...	...	...	...	...	...	...	...	...	...
556	port-gentil	40	GA	1579998986	88	-0.72	8.78	82.40	5.82
557	tuatapere	78	NZ	1579998957	47	-46.13	167.68	72.34	3.47
558	chipinge	10	ZW	1579998923	72	-20.19	32.62	64.89	3.47
559	abastumani	0	GE	1579999129	71	41.71	42.85	32.00	4.70
560	stornoway	100	GB	1579998939	93	58.21	-6.39	46.40	33.33

561 rows × 9 columns

Humidity Heatmap

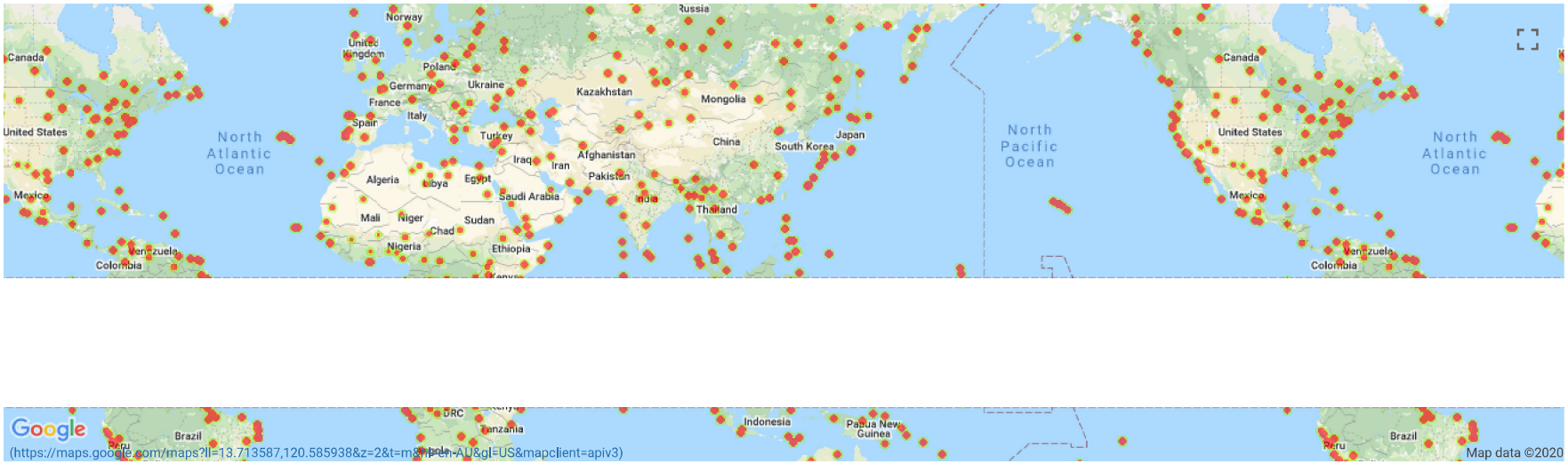
- Configure gmaps.
- Use the Lat and Lng as locations and Humidity as the weight.
- Add Heatmap layer to map.

In [4]:

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

```

In [5]:

```
# Store latitude and longitude in locations
locations = weather_data_frame[["Lat", "Lng"]]

# Used humidity as the weight
# Fill NaN values and convert to float
humidity = weather_data_frame["Humidity"].astype(float)

# Figure Layout
figure_layout={
    'width': '400px',
    'height': '300px',
    'border': '1px solid black',
    'padding': '1px'
}

# Plot the Heatmap
fig = gmaps.figure()

# create de heat layer
heatmap_layer = gmaps.heatmap_layer(locations, weights=humidity,
                                    dissipating=False, max_intensity=5,
                                    point_radius=1)

# Add the layer to the map
fig.add_layer(heatmap_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 [6]:

```
# Narrow down the DataFrame to find your ideal weather condition. For example:
# 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

# Create ideal cities data frame with the desirable variables
ideal_cities_df = pd.DataFrame(weather_data_frame, columns = ["City", "Cloudiness", "Max Temp", "Wind Speed"])

max_temp = (ideal_cities_df["Max Temp"] <= 80) & (ideal_cities_df["Max Temp"] > 70)
wind_speed = ideal_cities_df["Wind Speed"] < 10
cloudiness = ideal_cities_df["Cloudiness"] == 0

# Ideals Cities data frame with max_temp, wind_speed and cloudiness]
ideal_cities_df[cloudiness & max_temp & wind_speed]
```

Out[6]:

	City	Cloudiness	Max Temp	Wind Speed
46	arraial do cabo	0	75.79	9.40
210	ampanihy	0	71.40	9.01
319	presidencia roque saenz pena	0	79.27	4.27

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]:
```
# Store into variable named hotel_df .
hotel_df = pd.DataFrame(weather_data_frame, columns=["City", "Country", "Lat", "Lng"])

# Add a "Hotel Name" column to the weather_data_frame.
hotel_df["Hotel Name"] = ""

# see table
hotel_df
```

Out[7]:

	City	Country	Lat	Lng	Hotel Name
0	jamestown	US	42.10	-79.24	
1	ulladulla	AU	-35.35	150.47	
2	batagay-alyta	RU	67.80	130.41	
3	puerto ayora	EC	-0.74	-90.35	
4	ancud	CL	-41.87	-73.82	
...	...	...	...	...	...
556	port-gentil	GA	-0.72	8.78	
557	tuatapere	NZ	-46.13	167.68	
558	chipinge	ZW	-20.19	32.62	
559	abastumani	GE	41.71	42.85	
560	stornoway	GB	58.21	-6.39	

561 rows × 5 columns

In [8]:
```
# Set parameters to search for hotels with 5000 meters.
coordinates = f"{hotel_df['Lat'][0]},{hotel_df['Lng'][0]}"
search = "hotel"
radius = 5000
target_type = "hotel"

# set up a parameters dictionary
params = {
    "location": coordinates,
    "keyword": search,
    "radius": radius,
    "type": target_type,
    "key": g_key
}

# Build URL using the Google Maps API
# 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)

cities_ideal = response.json()

pprint(response.json(), depth=1)
```

```
{'html_attributions': [], 'results': [...], 'status': 'OK'}
```

In [9]:
```
hotel_df.iloc[0][0]
```

Out[9]:
```
'jamestown'
```

```
# Using Google Places API to find the first hotel for each city located within 5000 meters of your coordinates

```

In [10]:

```
# Define a list to storage the hotel names
hotel_name = []

# Assign Geo Coordinates
radious = '5000'
keyword = 'hotel'
types = 'hotel'

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

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

    # change location each iteration while leaving original params in place
    params["location"] = f"{lat},{lng}"

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

    # make request and print url
    name_address = requests.get(base_url, params=params)

    # convert to json
    name_address = name_address.json()

    # print(json.dumps(name_address, indent=4, sort_keys=True))
    try:
        hotel_name.append(name_address['results'][0]['name'])
    except IndexError:
        hotel_name.append(np.nan)
```

In [11]:

```
hotel_name
```

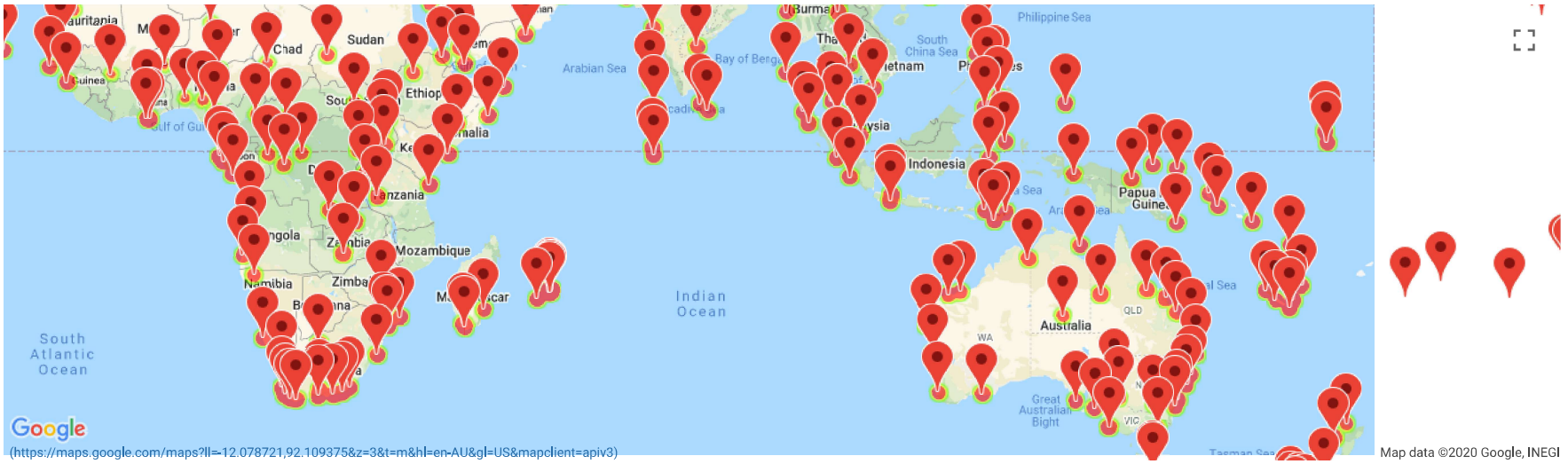
```
'The Explorer Hotel',
'HOTEL MADRID PLAZA',
'Golden Tulip Porto Vitória',
'Harbour House Hotel',
'Hotel Olafsvik',
'Hotel RIKMAN Continental',
'Kauai Shores Hotel',
'Falls Hotel',
nan,
'Mini-hotel Meduza',
'The Tusks Lodge',
'Inn Inubosaki hotel of superb view',
'Hotel Santos Pina',
'MyPond Hotel',
'Hotel Internacional',
nan,
'Hotel Adras',
'Red Hotel',
'Hotel Miramar',
```

In [16]:
```
# export file
hotel_df['Hotel Name']= hotel_name
hotel_df.dropna()
hotel_df.to_csv('output_data/hotel_export.csv')
```

```
# Plot the hotels on top of the humidity heatmap with each pin containing the **Hotel Name**, **City**, and **Country**.
```

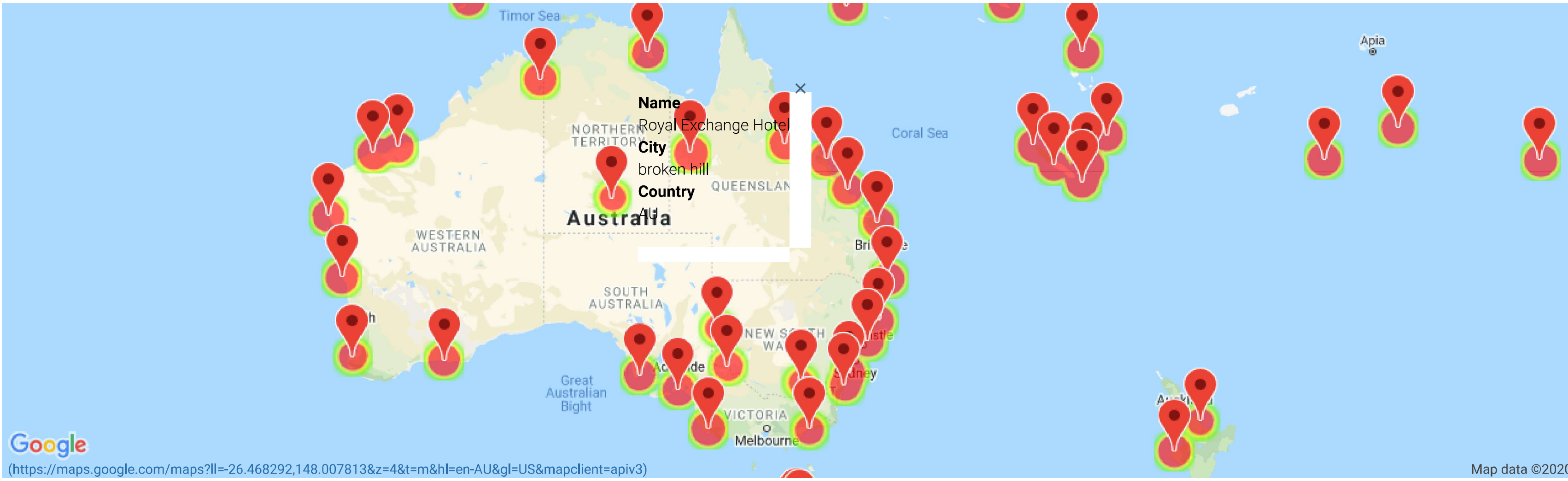
In [13]:
```
# NOTE: Do not change any of the code in this cell

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

```

In [14]:
```
# Add marker layer ontop of heat map
markers = gmaps.marker_layer(locations)

# Add the layer to the map
fig.add_layer(markers)


# Display Map
fig

```



In [15]:

```
# Add marker layer ontop of heat map
markers_layer = gmaps.marker_layer(locations, info_box_content=hotel_info)

# Add the layer to the map
fig.add_layer(markers_layer)

# Display Map
fig
```




```
References:
Pascal Bugnion. gmaps documentation. Jan 13, 2019. URL: https://buildmedia.readthedocs.org/media/pdf/jupyter-gmaps/latest/jupyter-gmaps.pdf

API documentation. Pascal Bugnion. 2016. URL: https://jupyter-gmaps.readthedocs.io/en/latest/api.html

gmaps. Pascal Bugnion. 2016. URL: https://jupyter-gmaps.readthedocs.io/en/latest/tutorial.html

Note:
In the last figure, the some name couldn't not show when I clicked, I assume is for long of the Hotel Name.
```

(