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 time
import json

# 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]:

```
citiesDf = pd.read_csv("cities.csv")
citiesDf
```

Out[2]:

	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wind Speed
0	Jiazi	3	CN	1595003225	88	22.88	116.07	82.60	10.63
1	Tiksi	99	RU	1595003225	78	71.69	128.87	43.48	17.49
2	Pangnirtung	90	CA	1595003225	49	66.15	-65.71	68.00	12.75
3	Emba	25	KZ	1595003226	17	48.83	58.14	90.14	7.54
4	Aitape	100	PG	1595003226	85	-3.14	142.35	77.58	5.10
...	...	...	...	...	...	...	...	...	...
571	At-Bashi	100	KG	1595003323	70	41.17	75.81	57.87	3.89
572	Brigantine	90	US	1595003324	78	39.41	-74.36	80.60	8.05
573	Shahr-e Bābak	75	IR	1595003324	28	30.12	55.12	78.76	2.06
574	Faya	20	SA	1595003324	49	18.39	42.45	75.20	11.41
575	Chara	100	RU	1595003324	85	56.91	118.26	59.27	1.03

576 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 [3]:

```
gmaps.configure(api_key = g_key)

locations = citiesDf[["Lat", "Lng"]]

humidity = citiesDf["Humidity"]
```

In [4]:

```
# Heatmap
fig = gmaps.figure(center=(46.0, -5.0), zoom_level=2)
# heat layer
heat_layer = gmaps.heatmap_layer(locations, weights = humidity, dissipating = False, max_intensity=np.max(humidity), poin
t_radius=3)
#add layer
fig.add_layer(heat_layer)

#display
fig
```

Create new DataFrame fitting weather criteria

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

In [5]:

```
narrowedCityDf = citiesDf.loc[(citiesDf["Wind Speed"] <= 10) & (citiesDf["Cloudiness"] == 0) & \
                              (citiesDf["Max Temp"] >= 70) & (citiesDf["Max Temp"] <= 80)].dropna()

narrowedCityDf
```

Out[5]:

	City	Cloudiness	Country	Date	Humidity	Lat	Lng	Max Temp	Wind Speed
113	Kisanga	0	TZ	1595003246	49	-7.44	37.70	70.72	5.21
153	Goderich	0	CA	1595003253	93	43.75	-81.72	78.01	8.55
362	Marquette	0	US	1595003287	64	46.53	-87.63	78.01	3.00
389	Dergachi	0	RU	1595003293	51	51.23	48.77	76.39	9.10
437	Zabul Province	0	AF	1595003301	19	32.25	67.25	75.81	6.80
452	Portsmouth	0	GB	1595003304	66	50.80	-1.09	75.00	1.99
455	Abalak	0	RU	1595003304	68	58.13	68.59	75.81	5.88
457	Waingapu	0	ID	1595003305	85	-9.66	120.26	74.03	5.88

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 [6]:

```
hotelDf = narrowedCityDf.loc[:,["City","Country","Lat","Lng"]]

hotelDf["Hotel Name"] = ""

hotelDf

#display results
```

Out[6]:

	City	Country	Lat	Lng	Hotel Name
113	Kisanga	TZ	-7.44	37.70	
153	Goderich	CA	43.75	-81.72	
362	Marquette	US	46.53	-87.63	
389	Dergachi	RU	51.23	48.77	
437	Zabul Province	AF	32.25	67.25	
452	Portsmouth	GB	50.80	-1.09	
455	Abalak	RU	58.13	68.59	
457	Waingapu	ID	-9.66	120.26	

In [7]:

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

params = {"type" : "hotel",
          "keyword" : "hotel",
          "radius" : 5000,
          "key" : g_key}
```

In [8]:

```
for index, row in hotelDf.iterrows():
    lat = row["Lat"]
    lng = row["Lng"]
    city_name = row["City"]
    params["location"] = f"{lat},{lng}"
    print(f"Retrieving Results for Index {index}: {city_name}.")
    response = requests.get(baseUrl, params=params).json()

    results = response['results']

    try:
        print(f"Closest hotel in {city_name} is {results[0]['name']}.")
        hotelDf.loc[index, "Hotel Name"] = results[0]['name']

    except (KeyError, IndexError):
        print("Missing something - Skipping")

    print("------------------")
    time.sleep(1)
print("End of search -------------")
```

```
Retrieving Results for Index 113: Kisanga.
Missing something - Skipping
------------------
Retrieving Results for Index 153: Goderich.
Closest hotel in Goderich is Samuels Hotel.
------------------
Retrieving Results for Index 362: Marquette.
Closest hotel in Marquette is Magnuson Hotel Country Inn.
------------------
Retrieving Results for Index 389: Dergachi.
Missing something - Skipping
------------------
Retrieving Results for Index 437: Zabul Province.
Missing something - Skipping
------------------
Retrieving Results for Index 452: Portsmouth.
Closest hotel in Portsmouth is Best Western Royal Beach Hotel.
------------------
Retrieving Results for Index 455: Abalak.
Closest hotel in Abalak is Туристический комплекс Абалак.
------------------
Retrieving Results for Index 457: Waingapu.
Closest hotel in Waingapu is Padadita Beach Hotel.
------------------
End of search -------------
```

In [9]:

```
hotelDf
```

Out[9]:

	City	Country	Lat	Lng	Hotel Name
113	Kisanga	TZ	-7.44	37.70	
153	Goderich	CA	43.75	-81.72	Samuels Hotel
362	Marquette	US	46.53	-87.63	Magnuson Hotel Country Inn
389	Dergachi	RU	51.23	48.77	
437	Zabul Province	AF	32.25	67.25	
452	Portsmouth	GB	50.80	-1.09	Best Western Royal Beach Hotel
455	Abalak	RU	58.13	68.59	Туристический комплекс Абалак
457	Waingapu	ID	-9.66	120.26	Padadita Beach Hotel

In [10]:

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

In [11]:

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

fig.add_layer(markers)

# Display figure
fig
```

In [ ]:

```

```