

In [23]: ▶

```
1 #Import Dependencies
2 import csv
3 import pprint
4 import json
5 import requests
6 import pandas as pd
7 import numpy as np
8 import gmaps
9 from config import gkey
10 from sklearn import datasets
11 import ipywidgets as widgets
12 #access gmaps with API key
13 gmaps.configure(api_key=gkey)
14 import gmaps.datasets
```

In [2]: ▶

```
1 #Pull List from WeatherPY
2 %store -r place_location
3 %store -r place
```

```
In [3]: 1 #Rename random Locations retrieved from WeatherPY
        2 place_df=pd.DataFrame(place)
        3 renamed_df=place_df.rename(columns={0:"Name",1:"Latitude",2:"Longitude",3:"Humidity",4:"Max_Temp",5:"Windspeed",6:"Cloudiness"})
        4 renamed_df
```

Out[3]:

	Name	Latitude	Longitude	Humidity	Max_Temp	Windspeed	Cloudiness
0	Lamut	16.65	121.22	81	296.23	1.38	95
1	St Helens	53.45	-2.73	87	289.15	4.10	40
2	Heliconia	6.21	-75.73	57	297.15	0.42	75
3	Camargo	27.67	-105.17	18	309.26	5.80	30
4	Zaragoza	41.66	-0.88	50	297.15	7.70	66
5	Teontepec	18.50	-97.52	69	293.71	4.47	39
6	Dhamtari	20.71	81.55	90	297.10	2.71	87
7	Columbus	39.96	-83.00	65	300.15	4.10	75
8	Benbrook	32.67	-97.46	26	309.82	4.10	20
9	Tongi	23.89	90.41	92	298.59	1.67	97
10	Bassenge	50.76	5.61	68	291.15	5.10	100
11	Orleans	30.03	-89.95	46	307.15	2.60	1
12	Mangga	15.25	120.91	92	299.26	0.56	100
13	Shatki	55.19	44.12	87	284.48	0.93	21
14	Lourinhã	39.23	-9.32	86	292.59	3.13	0
15	Harboøre	56.62	8.18	91	289.26	0.45	17
16	Grădinari	44.39	25.82	60	299.82	1.00	0
17	Krasnozatonskiy	61.68	50.98	100	285.15	1.37	75
18	Bollène	44.28	4.75	41	298.15	6.70	5
19	Mazkeret Batya	31.85	34.84	74	300.15	2.60	75
20	Poço Verde	-10.71	-38.18	82	292.95	2.93	9
21	Bănsi	27.18	82.93	91	298.20	4.21	12
22	Nambucca Heads	-30.65	153.00	70	280.15	3.10	0
23	Thames	-37.13	175.53	95	289.26	0.89	5
24	Rodolfo Sánchez Taboada	31.72	-116.57	79	300.93	1.79	1
25	Ágios Athanásios	41.07	24.25	69	300.37	1.50	0
26	Ishioka	36.18	140.27	94	298.15	2.10	75
27	Hope Mills	34.97	-78.95	69	301.48	4.10	75
28	Tāybād	34.74	60.78	17	298.28	3.73	0
29	Pregassona	46.02	8.97	69	296.15	1.50	20
30	Jarinu	-23.10	-46.73	71	287.15	10.30	20

	Name	Latitude	Longitude	Humidity	Max_Temp	Windspeed	Cloudiness
31	Kalabagh	32.97	71.56	53	306.03	1.25	25
32	Chichica	8.49	-82.16	83	300.15	1.50	75
33	Siay	7.70	122.86	86	298.40	1.10	48
34	Vera Cruz	-12.63	-41.03	54	300.15	2.60	13
35	Flekkefjord	58.30	6.66	85	289.26	3.37	44
36	Rakamaz	48.13	21.47	83	294.15	1.00	0
37	Eppingen	49.14	8.91	65	294.26	2.62	12
38	Sonta	45.59	19.10	77	294.15	2.10	0
39	Lawrenceville	33.96	-83.99	66	302.59	3.10	75
40	Pennsauken	39.96	-75.06	62	303.71	4.10	1
41	Rio Claro	-22.41	-47.56	63	291.15	5.70	81
42	Säkylä	61.03	22.33	93	292.15	4.10	75
43	Montenegro	42.50	19.30	57	299.15	1.00	0
44	Rājāpur	25.38	81.15	85	299.05	3.06	4
45	Cahabón	15.57	-89.82	66	303.37	0.12	53
46	Guisa	20.25	-76.54	59	306.15	4.10	75
47	Kyaka	-1.25	31.42	83	294.15	0.91	20
48	Haliyal	15.33	74.77	96	294.62	2.70	100
49	Kirya	-3.92	37.48	71	293.31	3.74	51

```
In [4]: ▶ 1 #Heatmap of all random locations weighted by max temp
2 locations=renamed_df[["Latitude","Longitude"]]
3 weights=renamed_df["Max_Temp"]
4 fig = gmaps.figure()
5 heat_layer=gmaps.heatmap_layer(locations,weights=weights)
6 fig.add_layer(heat_layer)
7 fig
```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

```
In [5]: 1 #Define travel requirements
2 renamed_df=renamed_df.loc[renamed_df["Windspeed"]<10,:]
3 renamed_df=renamed_df.loc[renamed_df["Cloudiness"]==0,:]
4 renamed_df=renamed_df.loc[renamed_df["Max_Temp"]>290,:]
5 renamed_df=renamed_df.loc[renamed_df["Max_Temp"]<298,:]
6 renamed_df
```

Out[5]:

	Name	Latitude	Longitude	Humidity	Max_Temp	Windspeed	Cloudiness
14	Lourinhã	39.23	-9.32	86	292.59	3.13	0
36	Rakamaz	48.13	21.47	83	294.15	1.00	0
38	Sonta	45.59	19.10	77	294.15	2.10	0

```
In [6]: 1 locations=renamed_df[["Latitude","Longitude"]]
2 location_list=locations.values.tolist()
3 location_list
```

Out[6]: [[39.23, -9.32], [48.13, 21.47], [45.59, 19.1]]

```
In [103]: 1 #Hotels within 5000km of the random locations identified to meet my travel
2 search_results=[]
3 url="https://maps.googleapis.com/maps/api/place/textsearch/json?query=location"
4 name=request["results"][0]["name"]
5 for row in location_list:
6     #print(row)
7     request=requests.get(url+"&location="+str(row)+"&radius=5000&key="+google_api_key)
8     search_results.append(request.json())
9 search_results_df=pd.DataFrame(search_results[0]["results"])
10 search_results_df["name"]
```

Out[103]: 0 US Hotel Ventures Co
1 Homely
2 Lodging Partners
3 The Lodge at Ballantyne
4 Lodging
5 Evon Allen Hospitality Management
6 Charlotte Kennels Actual
7 Holiday Inn
8 Mike Morgan
9 Hoiday Inn Express
10 Brookshire Inn
11 Fairfield Inn
12 my bf
13 300 Briar Ridge Dr.
14 Mictotel
15 Woodland hollow
16 Auto zone 4858
17 Kim's home
18 Comunidad Monroe 8 a 1
19 Wade Arscott Lodging
Name: name, dtype: object

```
In [ ]: 1
```

