VacationPy

Introduction

After analyzing the weather of more than 500 cities in ***WeatherPy***, I presented my findings to my group of friends. Looking at the list, we wondered: ***"How nice it would be to find a perfect vacation spot within this list".*** "We can use Google to search" one of my friends chimed in. We all agreed that Google is awesome but manually searching through the world wide web was out of question. I volunteered to create a script that would use ***gmaps*** and ***places*** API from Google and automate the job. Here is the problem I decided to investigate:

Figure 1. Illustration by balabolka @graphicriver.net

Problem Statement:

How to find a perfect vacation spot based on weather conditions?

Target Audience:

Apart from my friends, this report will be interesting to ***(a) vacation planners (b) travel agents*** and ***(c) globetrotters*** interested in finding a new travel destination.

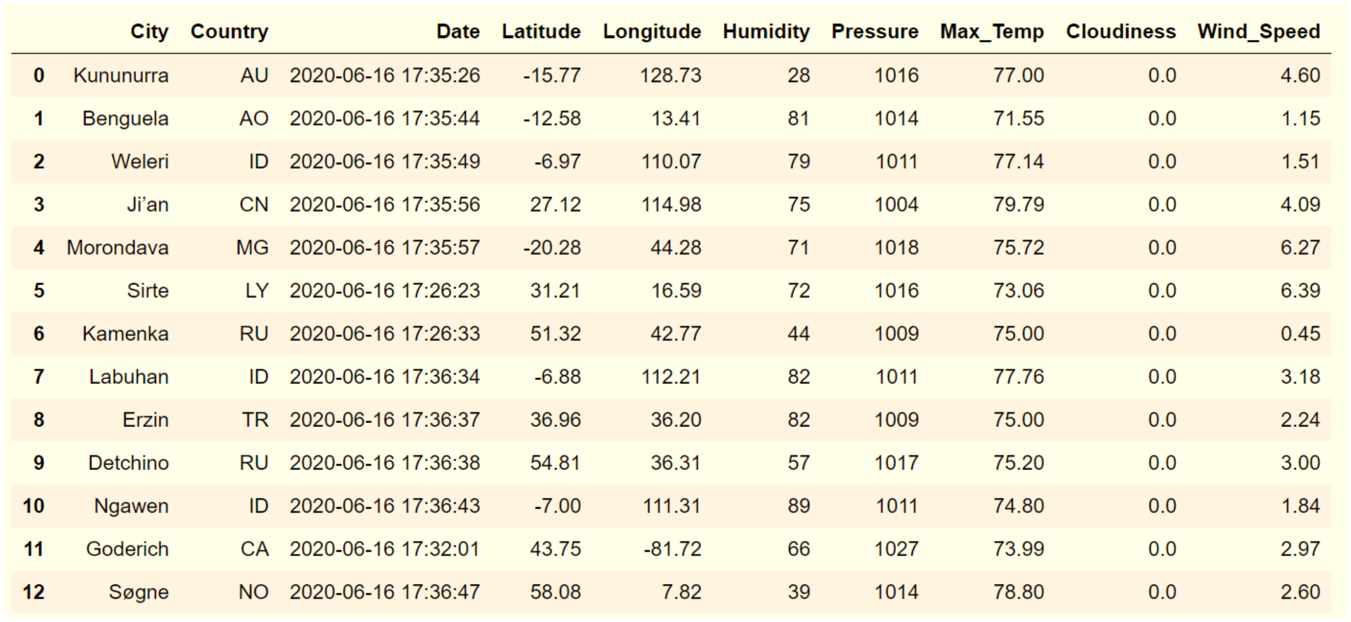
API calls & Libraries used:

1. We used the list of cities generated in ***WeatherPy*** as a starting list for vacation locations.
2. ***maps*** API and ***places*** API from Google were used to obtain more information on the selected cities.
3. A "try-except" loop was employed to retrieve data through successive API calls. Errors detected during execution (exceptions) were handled by the "except" portion and printed out explicitly.

Data selection: Perfect weather conditions for selection are ***(a)*** Maximum temperature less than ***80F*** but higher than ***70F*** ***(b)*** Wind speed less than ***10mph*** and ***(c) Zero*** cloudiness

Data Analysis:

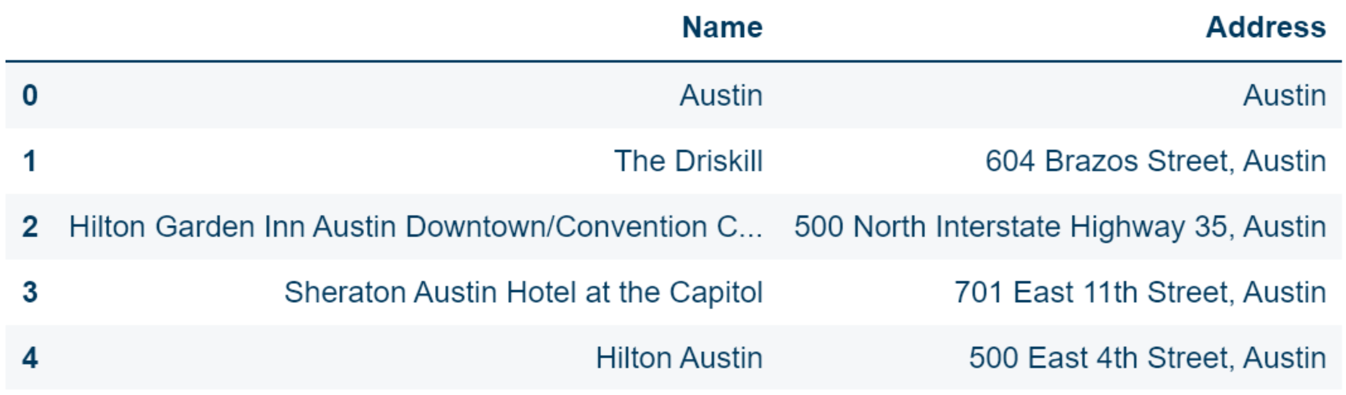
By using the selection criteria, we segmented our dataframe and found a list of 13 cities that satisfy all the conditions. The complete dataframe is shown in Table 1.



	City	Country	Date	Latitude	Longitude	Humidity	Pressure	Max_Temp	Cloudiness	Wind_Speed
0	Kununurra	AU	2020-06-16 17:35:26	-15.77	128.73	28	1016	77.00	0.0	4.60
1	Benguela	AO	2020-06-16 17:35:44	-12.58	13.41	81	1014	71.55	0.0	1.15
2	Weleri	ID	2020-06-16 17:35:49	-6.97	110.07	79	1011	77.14	0.0	1.51
3	Ji'an	CN	2020-06-16 17:35:56	27.12	114.98	75	1004	79.79	0.0	4.09
4	Morondava	MG	2020-06-16 17:35:57	-20.28	44.28	71	1018	75.72	0.0	6.27
5	Sirte	LY	2020-06-16 17:26:23	31.21	16.59	72	1016	73.06	0.0	6.39
6	Kamenka	RU	2020-06-16 17:26:33	51.32	42.77	44	1009	75.00	0.0	0.45
7	Labuhan	ID	2020-06-16 17:36:34	-6.88	112.21	82	1011	77.76	0.0	3.18
8	Erzin	TR	2020-06-16 17:36:37	36.96	36.20	82	1009	75.00	0.0	2.24
9	Detchino	RU	2020-06-16 17:36:38	54.81	36.31	57	1017	75.20	0.0	3.00
10	Ngawen	ID	2020-06-16 17:36:43	-7.00	111.31	89	1011	74.80	0.0	1.84
11	Goderich	CA	2020-06-16 17:32:01	43.75	-81.72	66	1027	73.99	0.0	2.97
12	Søgne	NO	2020-06-16 17:36:47	58.08	7.82	39	1014	78.80	0.0	2.60

Table 1. Dataframe of selected cities

To understand the working of this base url***:"https://maps.googleapis.com /maps /api /place/nearbysearch/json"*** we searched for hotels within 5km radius in Austin. The API returned 20 hotels which were stored in a dataframe. The first 5 rows of the dataframe is shown in Table 2.



	Name	Address
0	Austin	Austin
1	The Driskill	604 Brazos Street, Austin
2	Hilton Garden Inn Austin Downtown/Convention C...	500 North Interstate Highway 35, Austin
3	Sheraton Austin Hotel at the Capitol	701 East 11th Street, Austin
4	Hilton Austin	500 East 4th Street, Austin

Table 2. First 5 hotels within 5km of downtown Austin

Now we are ready to run through the list of 13 selected cities to find the nearest hotel.

Heatmap displaying Humidity:

Figure 2 shows a heatmap of humidity of all the cities (~ 583) in our dataframe.

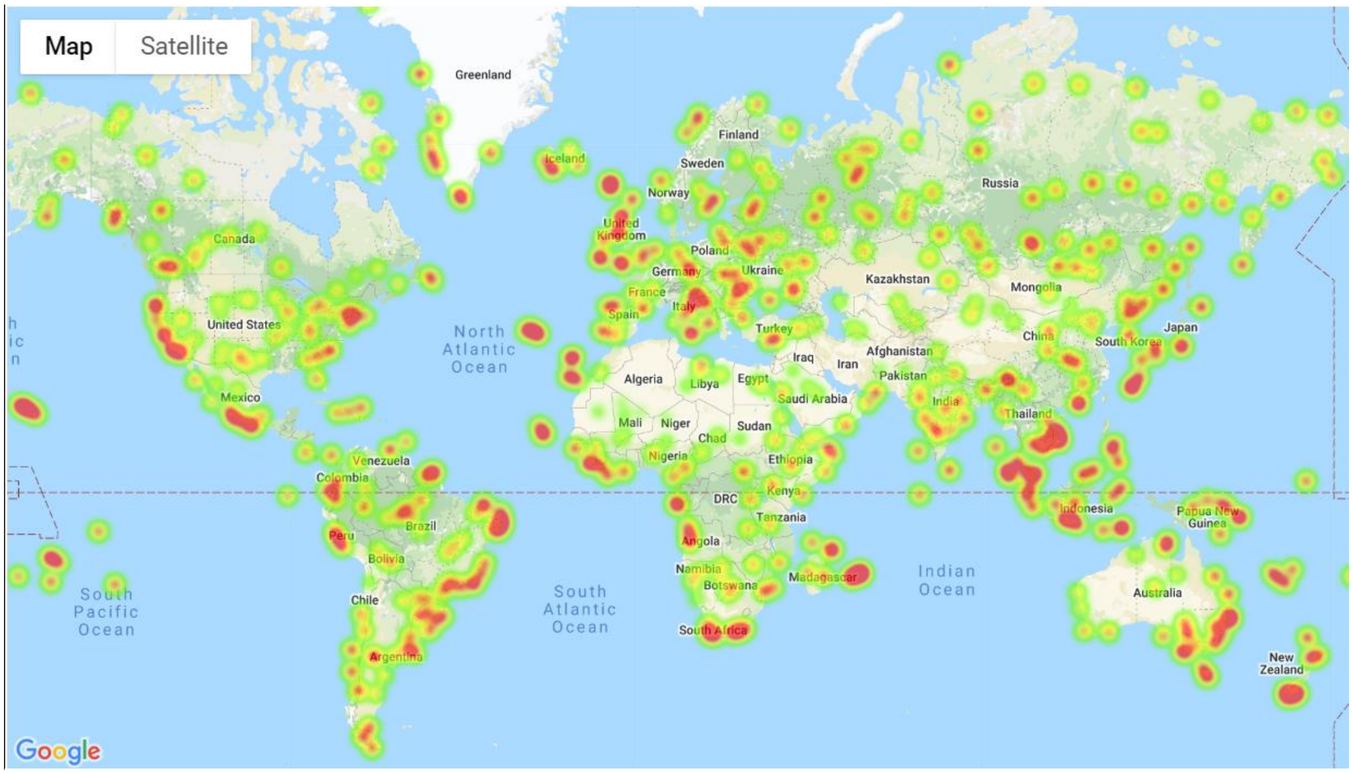


Figure 2. Heatmap displaying humidity of all the cities in our dataframe.

Observations:

As expected, humidity is high around the coastal areas, particularly in the south-eastern corner of the map. Humidity is high near the equator and decreases for higher latitudes.

Searching for Hotels:

base url: ***“https://maps.goog-leapis.com /maps /api /place/near-bysearch/json”***

keyword: ***“hotel”***

radius: ***“5000”***

Using the above parameters, we searched for hotels within 5 km of every city in our data-frame.

The response was appended in a column “Name” in the dataframe (see Table 3).

	City	Country	Latitude	Longitude	Name
0	Kununurra	AU	-15.77	128.73	Kununurra
1	Benguela	AO	-12.58	13.41	Benguela
2	Weleri	ID	-6.97	110.07	Lebo
3	Ji’an	CN	27.12	114.98	Ji’an
4	Morondava	MG	-20.28	44.28	Morondava
5	Sirte	LY	31.21	16.59	Sirte
6	Kamenka	RU	51.32	42.77	Kamenka
7	Labuhan	ID	-6.88	112.21	Musholla Babut Taubat
8	Erzin	TR	36.96	36.20	Erzin
9	Detchino	RU	54.81	36.31	Detchino
10	Ngawen	ID	-7.00	111.31	Bandjarredja
11	Goderich	CA	43.75	-81.72	Goderich
12	Søgne	NO	58.08	7.82	Søgne

Table 3. Hotels within 5km of selected cities

Destination Hotels (with markers):

Figure 3 shows the nearest hotel for every selected city appended to the heatmap in Figure 2.

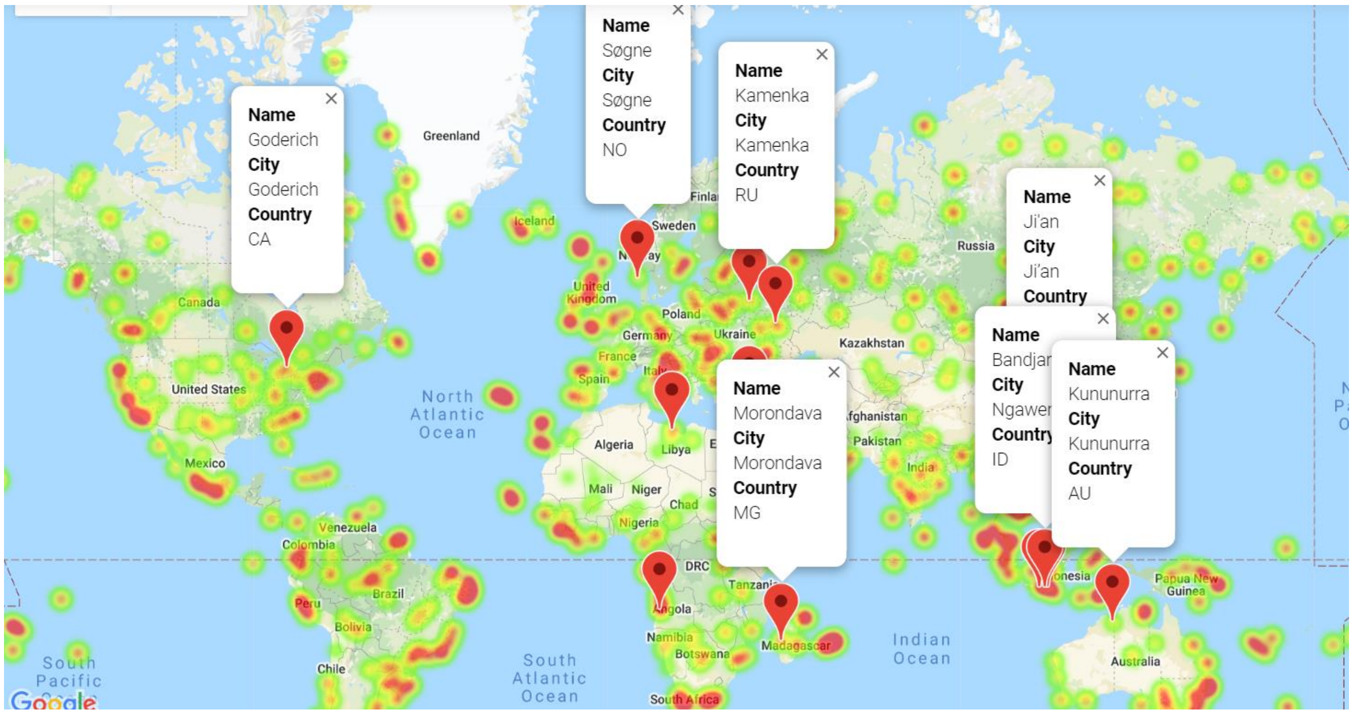


Figure 3. Hotel locations with markers for selected cities

Observations:

We could find one hotel within 5 kms in every city in the dataframe.

Searching for Natural Features:

To select the best vacation spot, we would also like to know the natural features (like *mountains*, *lakes*, *beach* etc.) around the selected cities. This would make the location more attractive to enjoy nature as well as for relaxation. For this purpose, we employed the places API with following parameters:

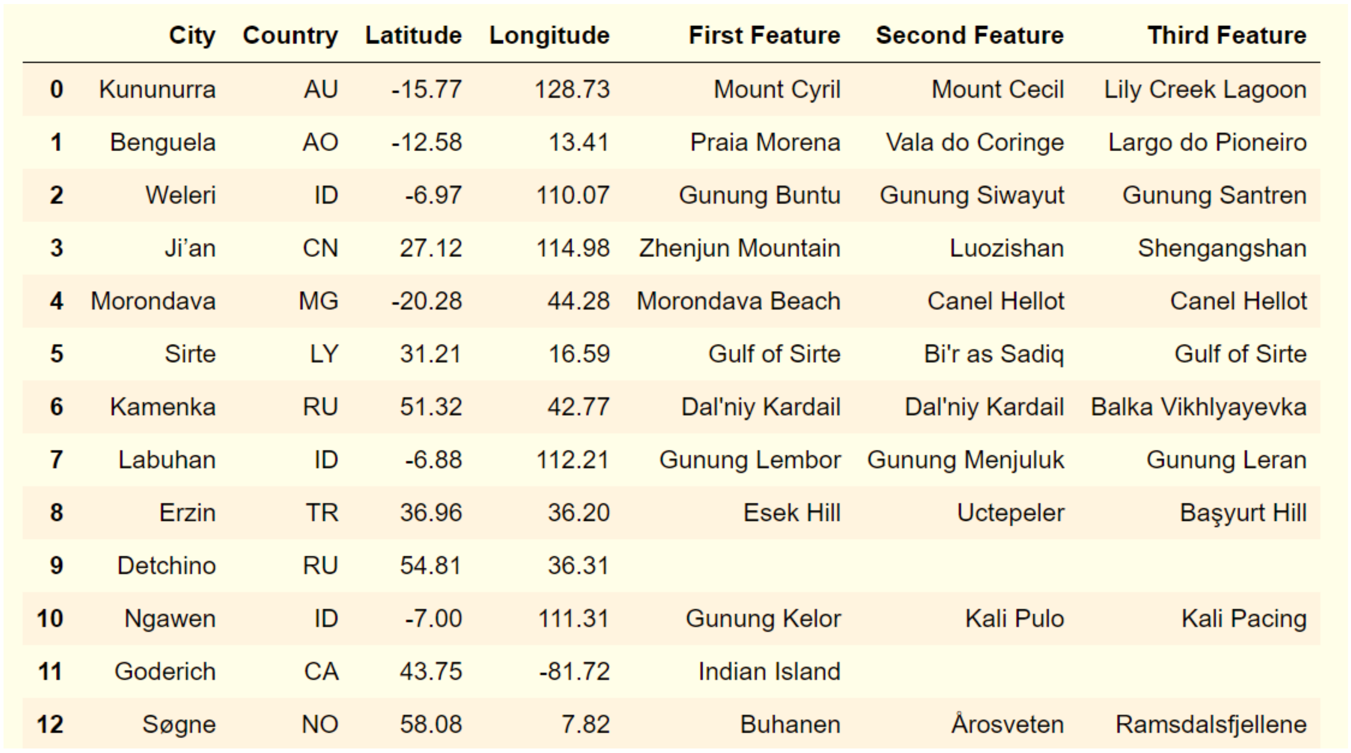
base url***: "https://maps.googleapis.com /maps /api /place/nearbysearch/json"***

keyword***: "natural_feature"***

radius***: "10000"***

The ***first, second and third natural feature*** for every city was saved in a dataframe (see Table 4). Not every city provided all the features - "try-except" loop within the script printed out the missing data.

The API couldn't find any natural feature around ***Detchino (Russia)*** and found only one natural feature around ***Goderich (Canada).***

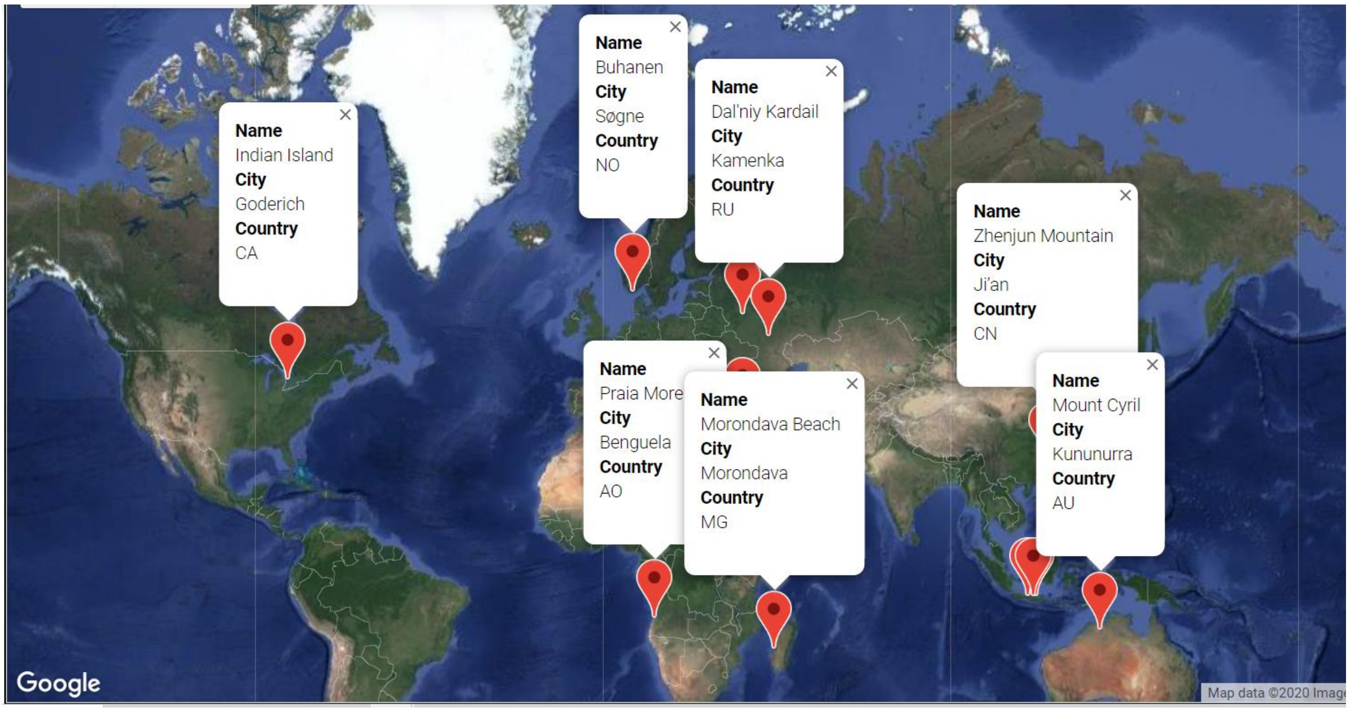


	City	Country	Latitude	Longitude	First Feature	Second Feature	Third Feature
0	Kununurra	AU	-15.77	128.73	Mount Cyril	Mount Cecil	Lily Creek Lagoon
1	Benguela	AO	-12.58	13.41	Praia Morena	Vala do Coringe	Largo do Pioneiro
2	Weleri	ID	-6.97	110.07	Gunung Buntu	Gunung Siwayut	Gunung Santren
3	Ji’an	CN	27.12	114.98	Zhenjun Mountain	Luozishan	Shengangshan
4	Morondava	MG	-20.28	44.28	Morondava Beach	Canel Hellot	Canel Hellot
5	Sirte	LY	31.21	16.59	Gulf of Sirte	Bi'r as Sadiq	Gulf of Sirte
6	Kamenka	RU	51.32	42.77	Dal'niy Kardail	Dal'niy Kardail	Balka Vikhlyayevka
7	Labuhan	ID	-6.88	112.21	Gunung Lembor	Gunung Menjuluk	Gunung Leran
8	Erzin	TR	36.96	36.20	Esek Hill	Uctepeler	Başyurt Hill
9	Detchino	RU	54.81	36.31			
10	Ngawen	ID	-7.00	111.31	Gunung Kelor	Kali Pulo	Kali Pacing
11	Goderich	CA	43.75	-81.72	Indian Island		
12	Søgne	NO	58.08	7.82	Buhanen	Årosveten	Ramsdalsfjellene

Table 4. First three natural features (mountain, beach, lake etc.) around every city in the dataframe

Destination Natural Features (with markers):

Figure 4 shows the nearest natural feature for every selected city in a Google Map *(type: “Hybrid”).*



Name Buhanen **City** Søgne **Country** NO

Name Dal'niy Kardail **City** Kamenka **Country** RU

Name Indian Island **City** Goderich **Country** CA

Name Zhenjun Mountain **City** Ji’an **Country** CN

Name Praia More **City** Benguela **Country** AO

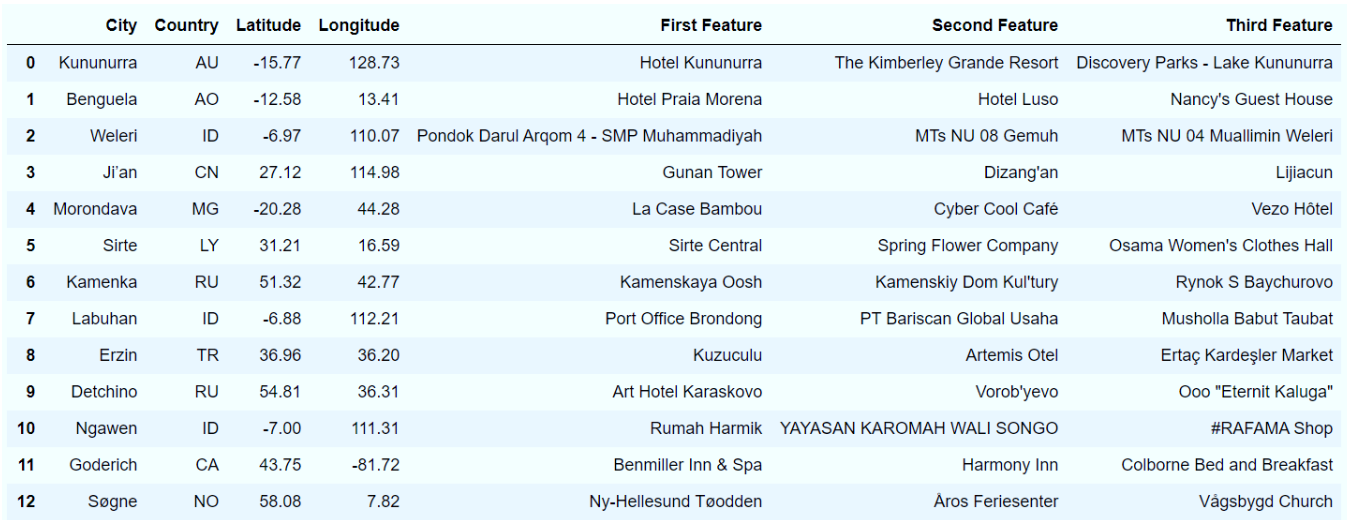
Name Morondava Beach **City** Morondava **Country** MG

Name Mount Cyril **City** Kununurra **Country** AU

Figure 4. First natural feature in the vicinity of every city in the dataframe

Searching for Point of Interest:

Apart from natural features, it would be nice to know popular venues (like architectural wonders, museums, church etc.) around the selected cities. This information would make the location more attractive to experience art and culture of a foreign land.



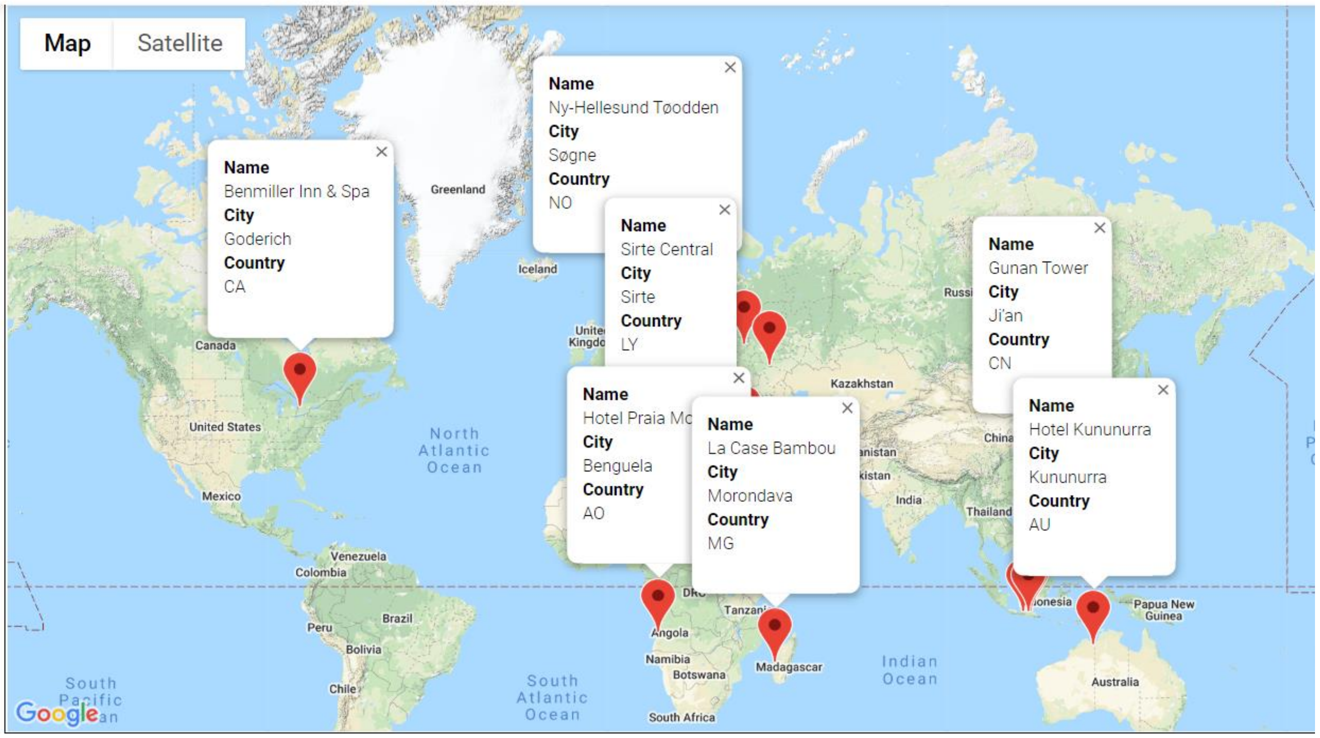
	City	Country	Latitude	Longitude	First Feature	Second Feature	Third Feature
0	Kununurra	AU	-15.77	128.73	Hotel Kununurra	The Kimberley Grande Resort	Discovery Parks - Lake Kununurra
1	Benguela	AO	-12.58	13.41	Hotel Praia Morena	Hotel Luso	Nancy's Guest House
2	Weleri	ID	-6.97	110.07	Pondok Darul Arqom 4 - SMP Muhammadiyah	MTs NU 08 Gemuh	MTs NU 04 Muallimin Weleri
3	Ji'an	CN	27.12	114.98	Gunan Tower	Dizang'an	Lijiacun
4	Morondava	MG	-20.28	44.28	La Case Bambou	Cyber Cool Café	Vezo Hôtel
5	Sirte	LY	31.21	16.59	Sirte Central	Spring Flower Company	Osama Women's Clothes Hall
6	Kamenka	RU	51.32	42.77	Kamenskaya Oosh	Kamenskiy Dom Kul'tury	Rynok S Baychurovo
7	Labuhan	ID	-6.88	112.21	Port Office Brondong	PT Bariscan Global Usaha	Musholla Babut Taubat
8	Erzin	TR	36.96	36.20	Kuzuculu	Artemis Otel	Ertaç Kardeşler Market
9	Detchino	RU	54.81	36.31	Art Hotel Karaskovo	Vorob'yevo	Ooo "Eternit Kaluga"
10	Ngawen	ID	-7.00	111.31	Rumah Harmik	YAYASAN KAROMAH WALI SONGO	#RAFAMA Shop
11	Goderich	CA	43.75	-81.72	Benmiller Inn & Spa	Harmony Inn	Colborne Bed and Breakfast
12	Søgne	NO	58.08	7.82	Ny-Hellesund Tødden	Åros Feriesenter	Vågsbygd Church

Table 5. Three topmost popular venues around every city in the dataframe

The ***first, second and third popular venues*** for every city was saved in a dataframe (see Table 5).

Destination Point of Interest (with markers):

Figure 5 shows the nearest popular venues for every selected city in a Google Map *(type: "Terrain")*



Name Benmiller Inn & Spa **City** Goderich **Country** CA

Name Ny-Hellesund Tødden **City** Søgne **Country** NO

Name Sirte Central **City** Sirte **Country** LY

Name Gunan Tower **City** Ji'an **Country** CN

Name Hotel Praia Mo **City** Benguela **Country** AO

Name La Case Bambou **City** Morondava **Country** MG

Name Hotel Kununurra **City** Kununurra **Country** AU

Figure 5. First point-of-interest near every city in the dataframe

Conclusion:

- We started this exercise with a paired set of 1500 random points
- Using ***Citypy*** library we found 683 cities corresponding to these random co-ordinates.
- Using ***OpenWeatherMap*** API we found weather data for ***583 cities*** - skipped 53 cities.
- Using ***maps*** API from Google we created a heatmap displaying humidity of all the cities.
- We selected 13 cities as probable vacation destination based on weather conditions
- Using ***places*** API from Google we discovered nearest hotel for the selected destinations.
- Using ***places*** API, we found the natural features in the vicinity of the selected cities.
- Using ***places*** API, we found the popular venues in the vicinity of the selected cities.

Now, we have all the data we need to find a perfect vacation spot. If we visit the city of Ji'an in China, we can enjoy the beauty of Zhenjun mountain and admire the Gunan tower near the bank of Ganjiang river. If we visit Kununurra in Western Australia, a small town built on big dreams, we can explore the foothills of Mount Cyril and recharge our batteries in serene cabins near Lake Kunnunurra. We finally decided to travel to Morondava (5th city in our dataframe) – a laid back beach town in Madagascar, where "mora, mora" (means slowly, slowly) is a way of life.

This is where our quest ends... which started with pairs of 1500 random numbers.

Figure 6. Morondava beach