Vaction Picker

Introduction

For this project I will use Foursquare location data to learn about venues in the vicinity of three potential vacation spots. I will sort and summarize the data in order to generate a profile of each location and a recommendation for what type of vacationer might most enjoy each spot.

Specifically, I have chosen three different beach towns in New England for my vacationer to consider.

People interested in this project might include:

1. Travel agencies
2. Developers of travel sites or apps
3. DIY vacation planners

Data

I will use the Foursquare API to gather location data for the three sample vacation spots. Specifically, I will get the name, category, and location (latitude and longitude) of all the venues within 1500 meters of each address.

I will use this data to:

1. Paint a summary picture of each location, including:
   - the number of venues in the area
   - the types of venues in the area
   - the ratio of eateries to other types of venue
2. Group the venues by category to give a vacationer an idea of the diversity of options in the area
3. Display the venues on a map, color coded by category groups, so a vacationer can visualize the area and the venues nearby

---

Methodology

We'll follow the same steps for each location:

1. Get the json file and convert it to a dataframe showing the venues, categories, and location
2. Summarize the nearby venues by the total number in each category
3. Create two additional dataframes, one showing eateries and another showing everything else
4. Sum up all the counts we've done - total venues, number of categories, number of eateries, number of other venues

Following that, I'll create a map for each location to visualize the location and the distribution of nearby venues. Venues will be color coded based on whether they can be described as primarily a place to get food, or another type of venue.

First, we'll get the json file from Foursquare, clean and filter it, and put the data we want into a pandas dataframe. Then we'll view the dataframe.

Block Island Venues:

In [12]:
```
venues = results['response']['groups'][0]['items']
nearby_venues = pd.json_normalize(venues)
filtered_columns = ['venue.name','venue.categories','venue.location.lat','venue.locat
bi_venues=nearby_venues.loc[:,filtered_columns]
bi_venues['venue.categories'] = nearby_venues.apply(get_category_type, axis=1)
bi_venues.columns = [col.split('.')[-1] for col in bi_venues.columns]
bi_venues
```

Out[12]:

	name	categories	lat	lng
0	Eli's	Seafood Restaurant	41.172964	-71.558901
1	The Ice Cream Place	Ice Cream Shop	41.172746	-71.558732
2	Ballard's Beach & Tiki Hut	Beach	41.172766	-71.554753
3	Persephone's Kitchen	Coffee Shop	41.173952	-71.560733
4	The Atlantic Inn	Hotel Bar	41.167991	-71.556781
5	Spring House Hotel	Hotel	41.168123	-71.554724
6	Koru Eco Spa	Spa	41.170147	-71.558525
7	1661 Inn	Hotel	41.169451	-71.555759
8	Mohegan Cafe & Brewery	American Restaurant	41.173618	-71.558588
9	The Beachead	Seafood Restaurant	41.175848	-71.562340
10	The National Hotel	Hotel	41.174101	-71.559400
11	Poor People's Pub	Bar	41.173835	-71.562950
12	Animal Farm	Farm	41.169115	-71.557228
13	Ernie's Old Harbor Restaurant	Breakfast Spot	41.172253	-71.556964
14	Finns Seafood Restaurant	Seafood Restaurant	41.172410	-71.556960
15	Three Sisters	Sandwich Place	41.173181	-71.561980
16	The National Hotel Bar	Hotel Bar	41.173891	-71.559126
17	Club Soda	Bar	41.171344	-71.566891
18	The Harborside Inn	Hotel Bar	41.173086	-71.558471
19	The Manisses Inn	Seafood Restaurant	41.170763	-71.557354
20	The Old Post Office Bagel Shop	Bagel Shop	41.174005	-71.561910
21	Aldo's Reataurant & Bar	Italian Restaurant	41.172509	-71.558796
22	Crescent Beach	Beach	41.179655	-71.564887
23	Aldo's Bakery	Bakery	41.172697	-71.558823
24	Captain Nick's	Bar	41.173893	-71.562769
25	Blue Dory Inn	Bed & Breakfast	41.174239	-71.559961
26	Ben & Jerry's	Ice Cream Shop	41.172606	-71.558198
27	Empire Theatre	Movie Theater	41.171995	-71.557828
28	Rebecca's Seafood Take-Out	Food Truck	41.172424	-71.558037
29	Harbor grill	American Restaurant	41.173017	-71.558435
30	Town Beach	Beach	41.182587	-71.566058
31	Block Island Depot	Deli / Bodega	41.175618	-71.569344
32	Surf Hotel	Hotel	41.174445	-71.559381
33	Island Moped	Rental Service	41.172956	-71.558837
34	Ballard's Inn	Resort	41.172638	-71.554769
35	Old Harbor View Take Out	Seafood Restaurant	41.172676	-71.557905
36	Nana's Ice Cream & Gelato	Ice Cream Shop	41.172750	-71.558701
37	Block Island Trading Co	Gift Shop	41.172709	-71.557576
38	Block Island Grocery	Grocery Store	41.174164	-71.562785
39	Block Island Yoga	Yoga Studio	41.165325	-71.549198
40	Fred Benson Town Beach	Beach	41.182936	-71.566278

Now we have a nice list of all the venues within 1500 meters of our spot on Block Island. I chose 1500 meters, just under a mile, to give a good sense of what's within walking distance.

Let's get a broader view by summarizing the venues according to category.

In [13]:
```
bi_cat_count=bi_venues.categories.value_counts().to_frame()
bi_cat_count.rename(columns={'categories':'number'}, inplace=True)
bi_cat_count
```

Out[13]:

	number
Seafood Restaurant	5
Beach	4
Hotel	4
Bar	3
Ice Cream Shop	3
Hotel Bar	3
American Restaurant	2
Deli / Bodega	1
Sandwich Place	1
Rental Service	1
Yoga Studio	1
Bed & Breakfast	1
Bakery	1
Bagel Shop	1
Coffee Shop	1
Grocery Store	1
Breakfast Spot	1
Gift Shop	1
Food Truck	1
Farm	1
Resort	1
Movie Theater	1
Spa	1
Italian Restaurant	1

I want to know how many places to eat are in the area, but they fall into several different categories in the Foursquare data, so I'll make a new dataframe called 'eateries' that will include every place where I could get food, excluding dessert-only venues and coffee shops.

Eateries:

In [14]:
```
bi_eateries = bi_venues[bi_venues['categories'].str.contains('Restaurant|Bar|Deli|Bake
bi_eateries.drop(['index'],axis=1,inplace=True)
bi_eateries
```

Out[14]:

	name	categories	lat	lng
0	Eli's	Seafood Restaurant	41.172964	-71.558901
1	The Atlantic Inn	Hotel Bar	41.167991	-71.556781
2	Mohegan Cafe & Brewery	American Restaurant	41.173618	-71.558588
3	The Beachead	Seafood Restaurant	41.175848	-71.562340
4	Poor People's Pub	Bar	41.173835	-71.562950
5	Ernie's Old Harbor Restaurant	Breakfast Spot	41.172253	-71.556964
6	Finns Seafood Restaurant	Seafood Restaurant	41.172410	-71.556960
7	Three Sisters	Sandwich Place	41.173181	-71.561980
8	The National Hotel Bar	Hotel Bar	41.173891	-71.559126
9	Club Soda	Bar	41.171344	-71.566891
10	The Harborside Inn	Hotel Bar	41.173086	-71.558471
11	The Manisses Inn	Seafood Restaurant	41.170763	-71.557354
12	The Old Post Office Bagel Shop	Bagel Shop	41.174005	-71.561910
13	Aldo's Reataurant & Bar	Italian Restaurant	41.172509	-71.558796
14	Aldo's Bakery	Bakery	41.172697	-71.558823
15	Captain Nick's	Bar	41.173893	-71.562769
16	Rebecca's Seafood Take-Out	Food Truck	41.172424	-71.558037
17	Harbor grill	American Restaurant	41.173017	-71.558435
18	Block Island Depot	Deli / Bodega	41.175618	-71.569344
19	Old Harbor View Take Out	Seafood Restaurant	41.172676	-71.557905

What about places other than eateries?

In [15]:
```
bi_not_food=eateries = bi_venues[~bi_venues['categories'].str.contains('Restaurant|Ba
bi_not_food.drop('index',axis=1,inplace=True)
bi_not_food
```

Out[15]:

	name	categories	lat	lng
0	The Ice Cream Place	Ice Cream Shop	41.172746	-71.558732
1	Ballard's Beach & Tiki Hut	Beach	41.172766	-71.554753
2	Persephone's Kitchen	Coffee Shop	41.173952	-71.560733
3	Spring House Hotel	Hotel	41.168123	-71.554724
4	Koru Eco Spa	Spa	41.170147	-71.558525
5	1661 Inn	Hotel	41.169451	-71.555759
6	The National Hotel	Hotel	41.174101	-71.559400
7	Animal Farm	Farm	41.169115	-71.557228
8	Crescent Beach	Beach	41.179655	-71.564887
9	Blue Dory Inn	Bed & Breakfast	41.174239	-71.559961
10	Ben & Jerry's	Ice Cream Shop	41.172606	-71.558198
11	Empire Theatre	Movie Theater	41.171995	-71.557828
12	Town Beach	Beach	41.182587	-71.566058
13	Surf Hotel	Hotel	41.174445	-71.559381
14	Island Moped	Rental Service	41.172956	-71.558837
15	Ballard's Inn	Resort	41.172638	-71.554769
16	Nana's Ice Cream & Gelato	Ice Cream Shop	41.172750	-71.558701
17	Block Island Trading Co	Gift Shop	41.172709	-71.557576
18	Block Island Grocery	Grocery Store	41.174164	-71.562785
19	Block Island Yoga	Yoga Studio	41.165325	-71.549198
20	Fred Benson Town Beach	Beach	41.182936	-71.566278

Let's sum it up

In [16]:
```
bi_categories_count=bi_venues.categories.nunique()
bi_venue_count=bi_venues.name.count()
print('Number of venues in Block Island:',bi_venue_count)
print('Different types of venue in Block Island:',bi_categories_count)
print('Number of eateries in Block Island:',bi_eateries.shape[0])
print('Venues other than food places:',bi_not_food.shape[0])
```

```
Number of venues in Block Island: 41
Different types of venue in Block Island: 24
Number of eateries in Block Island: 20
Venues other than food places: 21
```

Let's visualize these venues with a map

Orange circles are eateries, blue circles are other venues

In [32]:
```
bi_map = folium.Map(location=[bi_latitude,bi_longitude],zoom_start=15)
for lat, lng, name, categories in zip(bi_eateries['lat'], bi_eateries['lng'], bi_eate
    label = '{}, {}'.format(name, categories)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker([lat,lng],
                        radius=5,
                        popup=label,
                        color='orange',
                        fill=True,
                        fill_color='orange',
                        fill_opacity=.5,
                        parse_html=False).add_to(bi_map)

for lat, lng, name, categories in zip(bi_not_food['lat'], bi_not_food['lng'], bi_not_
    label = '{}, {}'.format(name, categories)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker([lat,lng],
                        radius=5,
                        popup=label,
                        color='blue',
                        fill=True,
                        fill_color='blue',
                        fill_opacity=.5,
                        parse_html=False).add_to(bi_map)

print('BLOCK ISLAND')
bi_map
```

```
BLOCK ISLAND
```

Out[32]: Make this Notebook Trusted to load map: File -> Trust Notebook

---

I followed the same procedure for the other two vacation spots, Westerly and Cape Cod. To see the details of those spots, check out my notebook, which includes all the steps I took in my analysis. For now, let's just see the summary tables and synthesis.

Results

Let's build a summary table for all three spots

In [36]:
```
data = {'Vacation Spot':['Block Island','Westerly, RI','Cape Cod'],
        'Total Venues':[bi_venues.shape[0],wri_venues.shape[0],cc_venues.shape[0]],
       'Eateries':[bi_eateries.shape[0],wri_eateries.shape[0],cc_eateries.shape[0]],
       'Other Venues':[bi_not_food.shape[0],wri_not_food.shape[0],cc_not_food.shape[0
       'Percentage Eateries':[(bi_eateries.shape[0]/bi_venues.shape[0]),wri_eateries.s
comp_table = pd.DataFrame(data=data).set_index('Vacation Spot')
comp_table
```

Out[36]:

Vacation Spot	Total Venues	Eateries	Other Venues	Percentage Eateries
Block Island	41	20	21	0.487805
Westerly, RI	30	9	21	0.300000
Cape Cod	95	39	56	0.410526

Discussion

We see that our three spots represent a spectrum of possibilities that may appeal to different people. If we order these three beach towns from least venues to most, we could say Westerly has the fewest offerings at 31, Block Island the next at 44, and Cape Cod by far the most at 96. The same order, unsurprisingly, holds true if we order by number of eateries instead. I was a bit surprised, however, to see that Westerly has even more non-food venues than Block Island!

So, for a person who seeks a bustling atmosphere with a wide variety of things to do and places to eat, **Cape Cod** is clearly the place to go. There's sure to be something here to please almost anyone - except perhaps someone who prefers a quieter, less busy location.

For that person, **Westerly** may be the best bet, especially if they're more interested in venues that are not food joints, because Westerly has the lowest ratio of eateries to non-eateries.

And **Block Island** represents a middle ground, with a nice mix of places to eat and things to do. Besides that, it provides an opportunity to to enjoy a ferry ride and experience an island vaction.

Conclusion

With the information we've discovered about the number and types of venues nearby each of our potential vacation spots, coupled with the maps for each location, I think we've enabled prospective vacationers to get a good idea of which spot would appeal most to them. Next step: packing the car!

This method could be tailored to multiple use cases, from travel websites and apps to magazine or blog writers who want an engine for generating summaries or recommendations for different destinations. We could easily use it to prioritize different types of venues for different travelers. One may want to sort destinations by the number of nearby coffee shops, or ice cream parlors, or shopping venues, or parks or beaches.