

```
# Install and import folium
!pip install folium
import folium
```

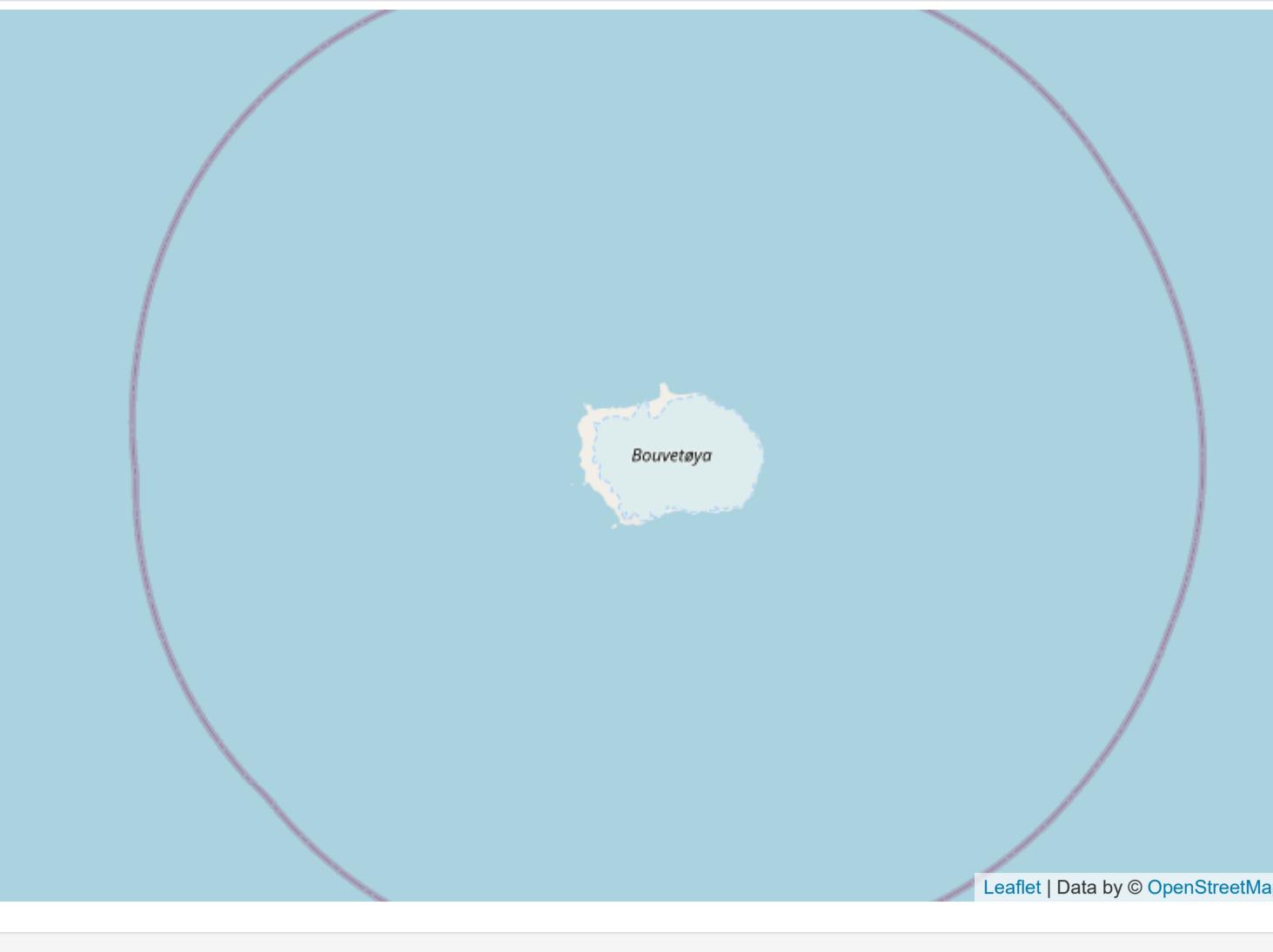
```
import folium

Requirement already satisfied: folium in d:\mambaforge\lib\s...
Requirement already satisfied: requests in d:\mambaforge\lib...
Requirement already satisfied: jinja2>=2.0 in d:\mambaforge\...
```

```
Requirement already satisfied: branca>=0.3.0 in d:\mambaforge\lib\site-packages (from folium) (0.4.2)
Requirement already satisfied: MarkupSafe>=2.0 in d:\mambaforge\lib\site-packages (from jinja2>=2.9->folium) (2.0.1)
Requirement already satisfied: certifi>=2017.4.17 in d:\mambaforge\lib\site-packages (from requests->folium) (2021.5.30)
Requirement already satisfied: idna<4,>=2.5 in d:\mambaforge\lib\site-packages (from requests->folium) (3.1)
Requirement already satisfied: charset-normalizer~2.0.0 in d:\mambaforge\lib\site-packages (from requests->folium) (2.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in d:\mambaforge\lib\site-packages (from requests->folium) (1.26.6)

In [2]: # A simple visualization of the most remote island (Bouvet Island - Norway) on earth.

test_coor = [-54.419992, 3.356570]
test_map = folium.Map(location=test_coor, zoom_start=10) # start at zoom level 10
test_map

Out[2]: A map centered on Bouvetøya, the most remote island in the world, located in the South Atlantic Ocean. The map shows the island's shape and its label "Bouvetøya". A large circular overlay is drawn around the island, representing a 1000 km radius. The map is displayed in a Jupyter Notebook cell with zoom controls (+/-) and a Leaflet attribution.
```

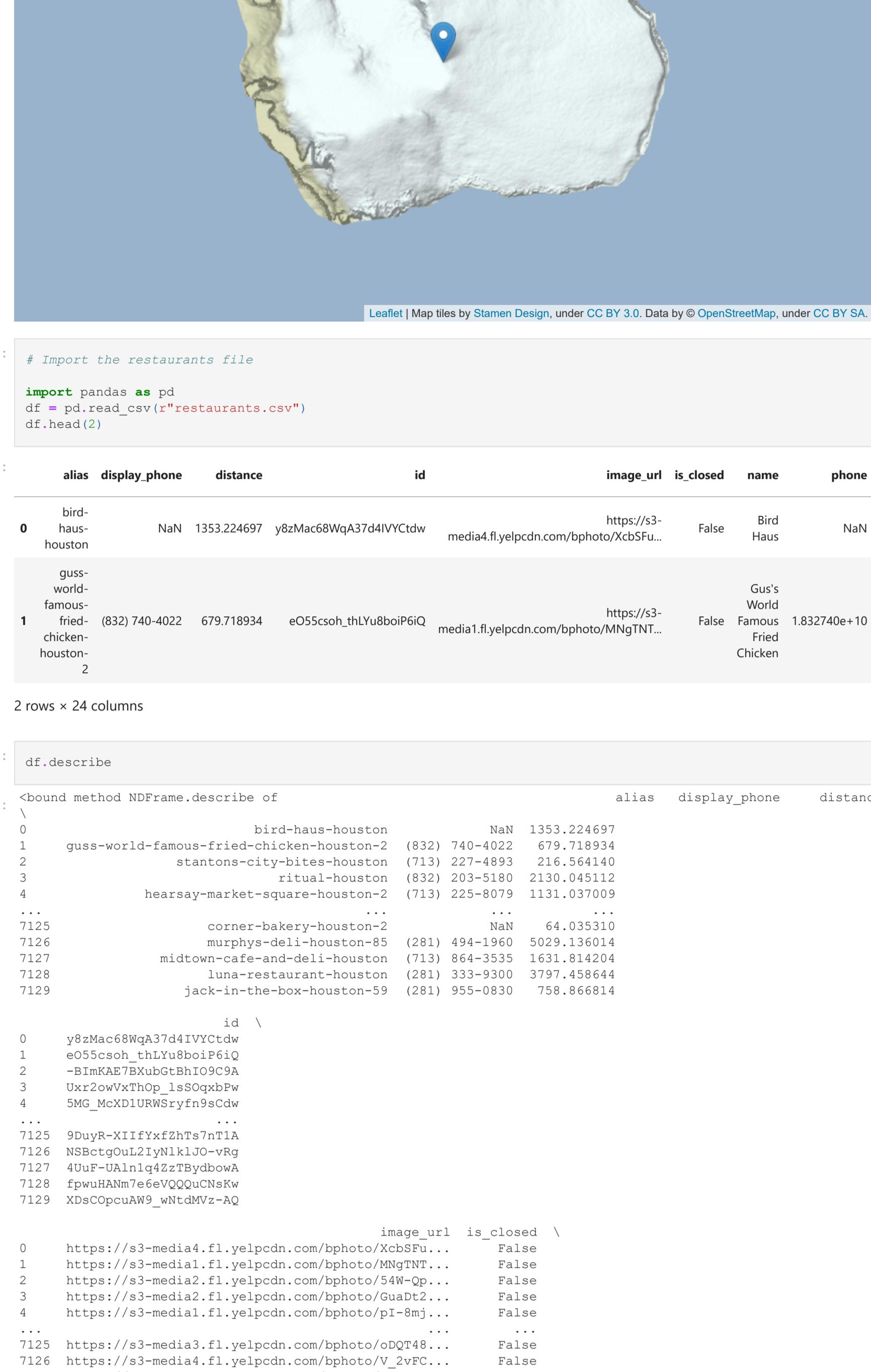
Leaflet | Data by © OpenStreetMap, under ODbL.

```
In [3]: # Adding a marker saving the map as an html file

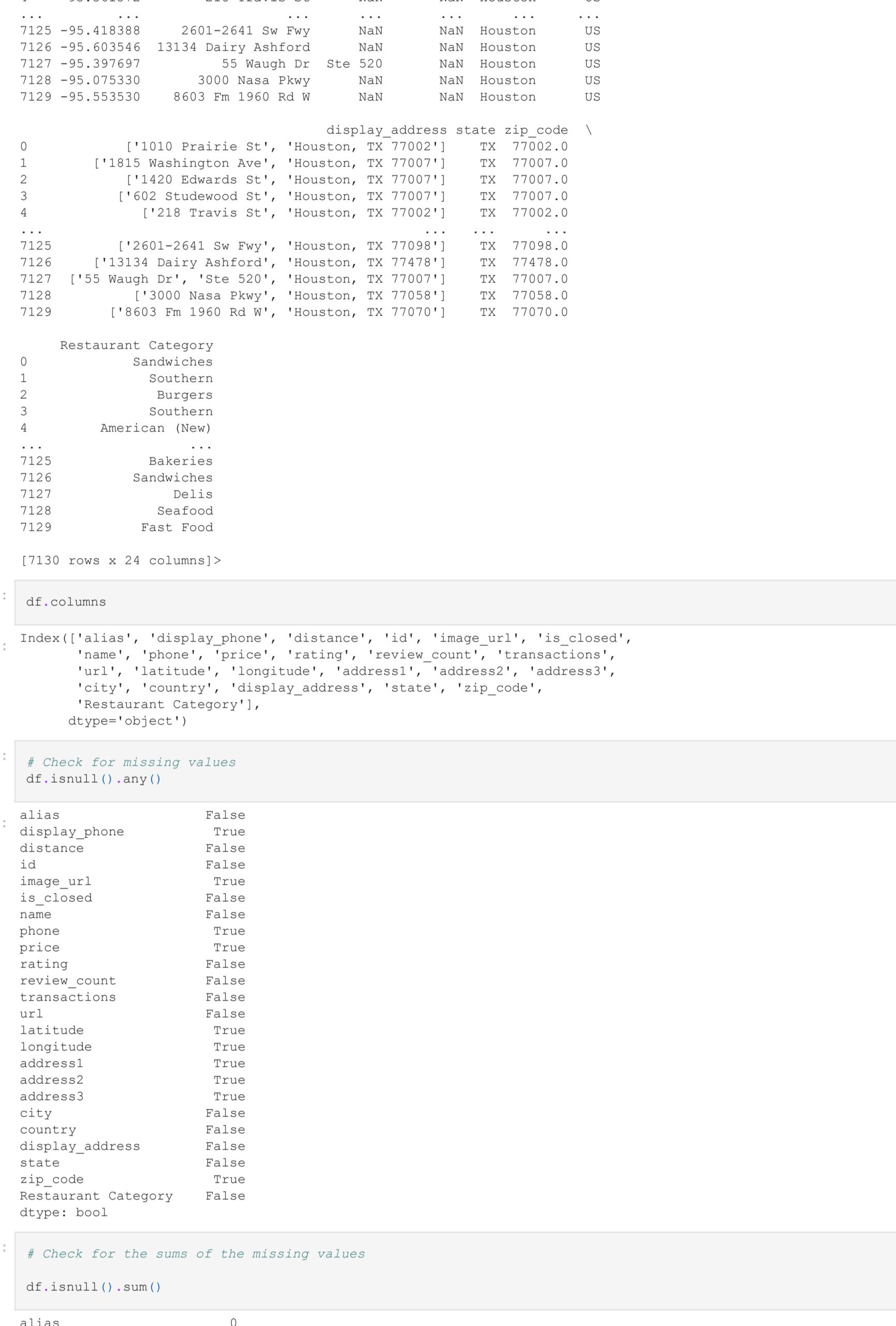
test_coor = [-54.419992, 3.356570]
test_map = folium.Map(location=test_coor, zoom_start=12,tiles="Stamen Terrain")
folium.Marker([-54.419992, 3.356570], popup = "Bouvet Island, Norway").add_to(test_map)
test_map.save("test_map.html")
test_map
```

Out[3]:

A 3D surface plot representing a geological outcrop. The vertical axis is labeled '10 m' at the top. The plot shows two distinct rock layers: a upper layer in light yellow and a lower layer in white. The boundary between them is irregular, showing various fractures and weathering. The top surface of the yellow layer has some small, jagged peaks.



7120	https://s3-media1.fl.yelpcdn.com/bphoto/vfpI6c...	False				
7129	https://s3-media1.fl.yelpcdn.com/bphoto/vfpI6c...	False				
0	Bird Haus	NaN \$\$ 4.5 ... \				
1	Gus's World Famous Fried Chicken	1.832740e+10 \$ 4.0 ...				
2	Stanton's City Bites	1.713227e+10 \$\$ 4.5 ...				
3	Ritual	1.832204e+10 \$\$ 4.0 ...				
4	Hearsay Market Square	1.713226e+10 \$\$ 4.0 ...				
...				
7125	Corner Bakery	NaN \$\$ 2.5 ...				
7126	Murphy's Deli	1.281494e+10 \$ 3.5 ...				
7127	Midtown Cafe & Deli	1.713864e+10 \$\$ 1.0 ...				
7128	Luna Restaurant	1.281334e+10 \$\$ 1.5 ...				
7129	Jack in the Box	1.281955e+10 \$ 2.0 ...				
longitude	address1	address2	address3	city	country	\
0	-95.361582	1010 Prairie St	NaN	NaN	Houston	US
1	-95.376550	1815 Washington Ave	NaN	NaN	Houston	US
2	-95.372068	1420 Edwards St	NaN	NaN	Houston	US
3	-95.387350	602 Studewood St	NaN	NaN	Houston	US



distance	0
id	0
image_url	284
is_closed	0
name	0
phone	391
price	1518
rating	0
review_count	0
transactions	0
url	0
latitude	4

```
longitude          4
address1         102
address2         5372
address3         6961
city              0
country             0
display_address      0
state              0
zip_code           10
Restaurant Category    0
dtype: int64
```

```
In [9]: # Number of restaurants by category

df[\"Restaurant Category\"].value_counts()

Out[9]: Mexican      885
Fast Food      516
Pizza         402
Sandwiches     378
Burgers        362
...
Irish Pub       1
Performing Arts 1
Unknown          1
Fondue          1
Golf             1
Name: Restaurant Category, Length: 177, dtype: int64
```

```
df[ Restaurant Category ].value_counts()[:20].plot(kind = 'bar')
```

Out[10]: <AxesSubplot:>

Category	Value Count
1	850
2	520
3	400
4	380
5	370
6	300
7	280
8	220
9	210
10	200
11	190
12	180
13	170
14	160
15	150
16	140
17	130
18	120
19	110
20	100

A horizontal bar chart showing the count of food categories. The categories are listed on the y-axis: Mexican, Fast Food, Pizza, Sandwiches, Burgers, Chinese, Vietnamese, Food Trucks, Seafood, American (Traditional), Chicken Wings, American (New), Delis, Barbeque, Breakfast & Brunch, Cajun/Creole, Italian, Tex-Mex, Indian, and Bakeries. The x-axis represents the count, with a single bar at 0 for each category.

A histogram showing the distribution of distance values. The x-axis represents distance, ranging from 0 to 5. The y-axis represents frequency, ranging from 0 to 1600. The distribution is right-skewed, with the highest frequency occurring at distance 4.

Distance Bin	Frequency
[1.0, 1.5)	200
[1.5, 2.0)	350
[2.0, 2.5)	600
[2.5, 3.0)	650
[3.0, 3.5)	900
[3.5, 4.0)	1450
[4.0, 4.5)	1650
[4.5, 5.0)	1300

	rating	latitude	longitude	name	address1	image	is_open	category	review_count	distance	price	stars
0	bird-haus-houston	NaN	1353.224697	y8zM	media4.fl.yelpcdn.com/bphoto/XcbSF...	https://s3-	False	Bird Haus	NaN	\$\$	4.5	

1	4.0	29.767530	-95.376550	Gus's World Famous Fried Chicken	1815 Washington Ave
2	4.5	29.770945	-95.372068	Stanton's City Bites	1420 Edwards St
3	4.0	29.781930	-95.387350	Ritual	602 Studewood St
4	4.0	29.763046	-95.361572	Hearsay Market Square	218 Travis St

In [14]:

```
# Check for missing values in the filtered data frame  
  
rest.isnull().sum()
```

Out[14]:

```
rating      0  
latitude    4  
longitude   4  
name        0  
address1    102  
dtype: int64
```

```
In [15]: # Check for any duplicates  
  
rest.duplicated().sum()  
  
Out[15]: 0  
  
In [16]: rest.shape  
  
Out[16]: (7130, 5)  
  
In [17]: # Drop the rows with missing coordinates  
  
rest.dropna(subset = ["latitude", "longitude"], axis=0, inplace=True) # axis=0 deletes the row, axis=1 deletes
```



```
latitude      0
longitude     0
name          0
address1      98
dtype: int64

In [19]: rest.head(1)

Out[19]:

|   | rating | latitude | longitude  | name      | address1        |
|---|--------|----------|------------|-----------|-----------------|
| 0 | 4.5    | 29.76036 | -95.361582 | Bird Haus | 1010 Prairie St |



In [20]: # CREATE THE MAP. THIS PROCESS TAKES 1-2 MINUTES TO COMPLETE
# REPLACE range(100) WITH Range(len(rest["rating"])) TO SEE THE ENTIRE DATA

coor=[29.76087, -95.35462]
my_map = folium.Map(location=coor, zoom_start=11)

for i in range(100):
```

Out[20]:

```
    rest.iloc[i]["latitude"],
        rest.iloc[i]["longitude"]),
    popup = (rest.iloc[i]["rating"],
              rest.iloc[i]["name"],
              rest.iloc[i]["address1"])).add_to(my_map)

my_map.save("rest_map.html")
my_map
```

