Map visualization - restaurants of Houston

This project visualizes the locations of more than 7,000 restaurants in Houston, TX

In [1]:

```
# Install and import folium

!pip install folium
import folium
```

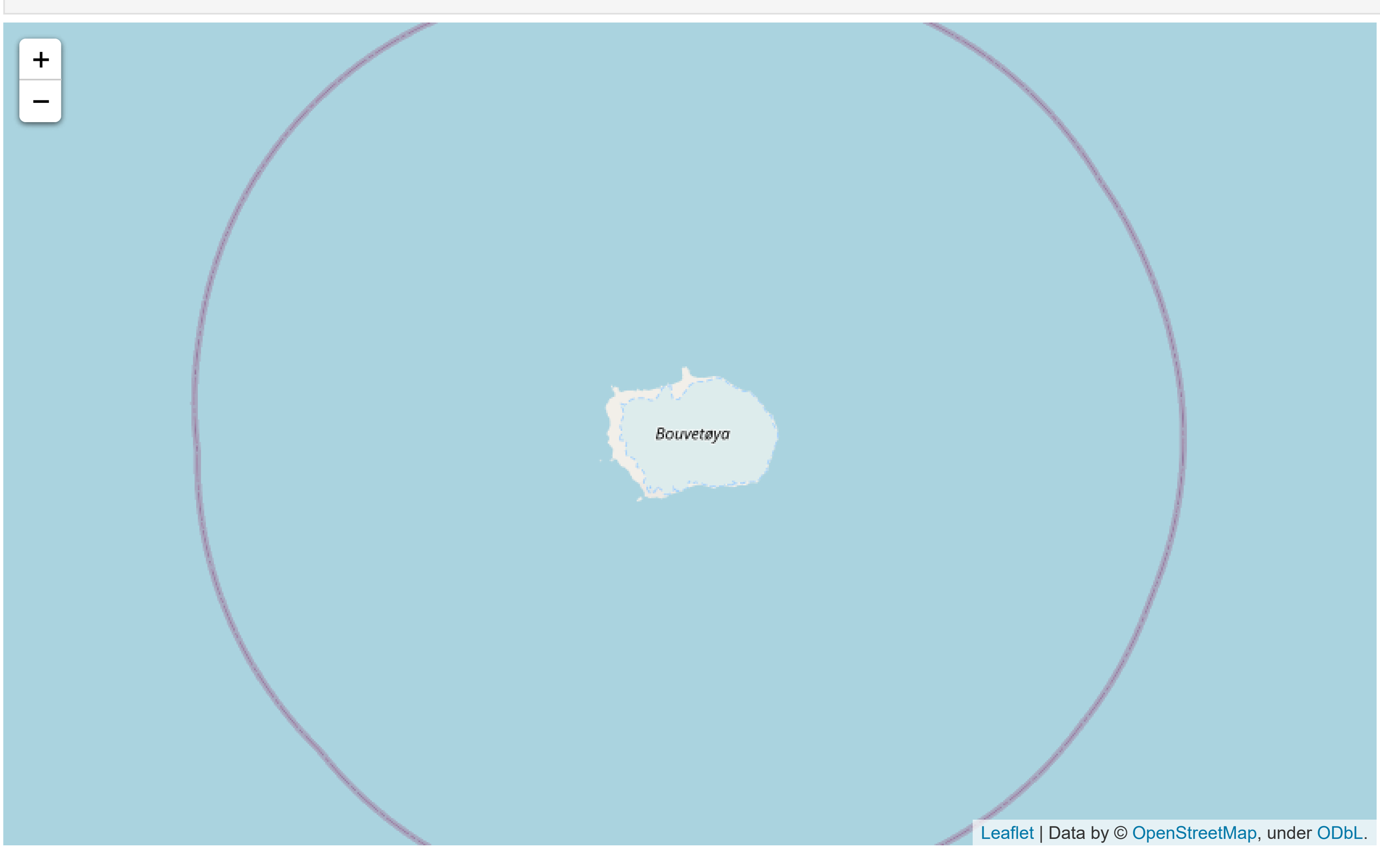
```
Requirement already satisfied: folium in d:\mambaforge\lib\site-packages (0.12.1)
Requirement already satisfied: jinja2>=2.9 in d:\mambaforge\lib\site-packages (from folium) (3.0.1)
Requirement already satisfied: branca>=0.3.0 in d:\mambaforge\lib\site-packages (from folium) (0.4.2)
Requirement already satisfied: requests in d:\mambaforge\lib\site-packages (from folium) (2.26.0)
Requirement already satisfied: numpy in d:\mambaforge\lib\site-packages (from folium) (1.21.2)
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) (2
021.5.30)
Requirement already satisfied: charset-normalizer~=2.0.0 in d:\mambaforge\lib\site-packages (from requests->fol
ium) (2.0.0)
Requirement already satisfied: idna<4,>=2.5 in d:\mambaforge\lib\site-packages (from requests->folium) (3.1)
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 - Norwey) 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]:

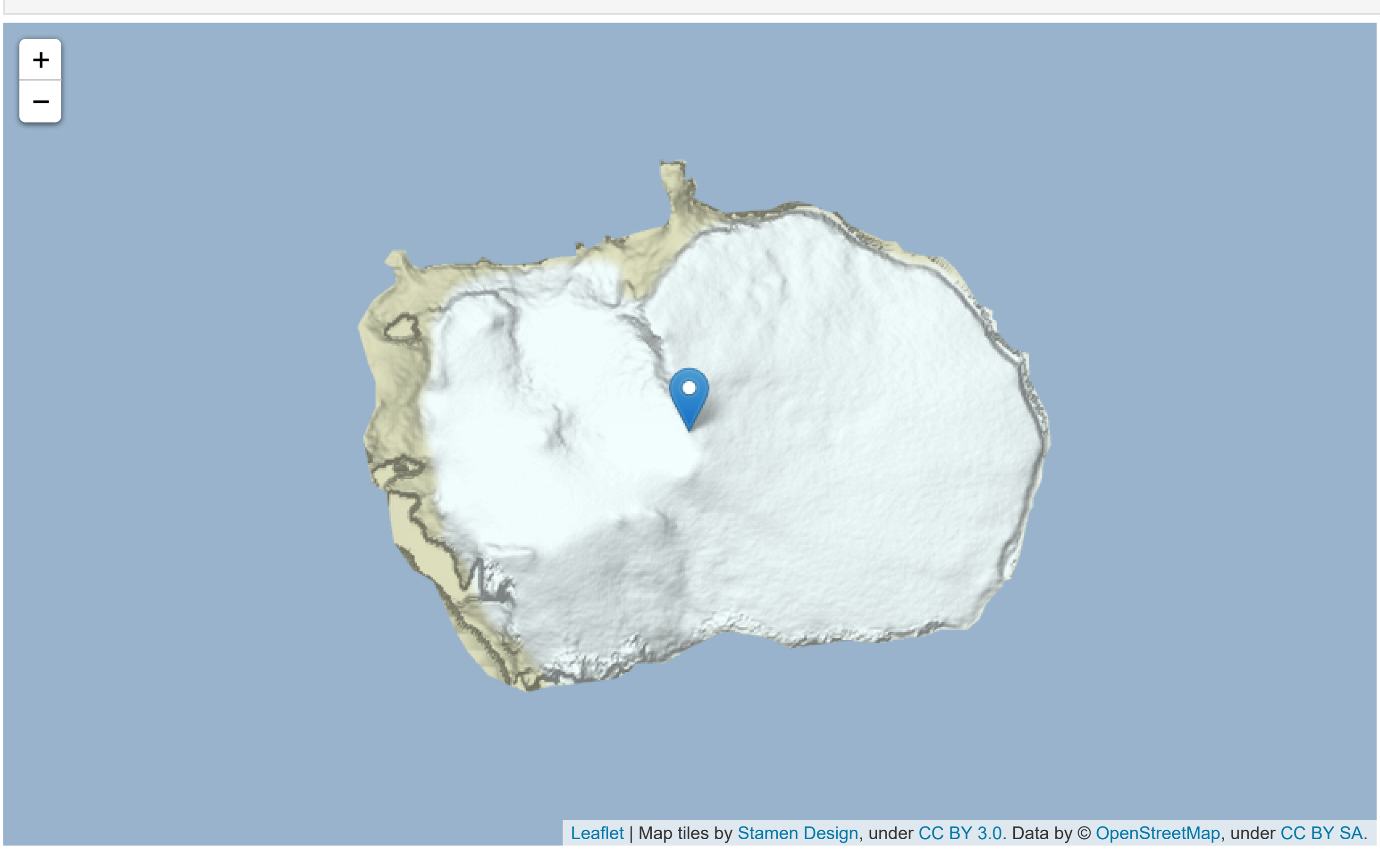


In [3]:

```
# 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, Norwey").add_to(test_map)
test_map.save("test_map.html")
test_map
```

Out[3]:



In [4]:

```
# Import the restaurants file

import pandas as pd
df = pd.read_csv(r"restaurants.csv")
df.head(2)
```

Out[4]:

	alias	display_phone	distance	id	image_url	is_closed	name	phone	pri
0	bird-haus-houston	NaN	1353.224697	y8zMac68WqA37d4IVYCtdw	https://s3-media4.fl.yelpcdn.com/bphoto/XcbSFu...	False	Bird Haus	NaN	
1	guss-world-famous-fried-chicken-houston-2	(832) 740-4022	679.718934	eO55csoh_thLYu8boiP6iQ	https://s3-media1.fl.yelpcdn.com/bphoto/MNgTNT...	False	Gus's World Famous Fried Chicken	1.832740e+10	

2 rows × 24 columns

In [21]:

```
# df.describe may give some insight on rating and review_count. It will not be useful for the other columns

df.describe()
```

Out[21]:

	distance	phone	rating	review_count	latitude	longitude	zip_code
count	7130.000000	6.739000e+03	7130.000000	7130.000000	7126.000000	7126.000000	7120.00000
mean	4730.087560	1.593446e+10	3.363815	98.659748	29.780699	-95.458246	77045.72500
std	3214.376725	2.193370e+09	1.000404	190.339715	0.173649	0.753400	640.76392
min	9.007948	1.201227e+10	0.000000	0.000000	21.434787	-158.184206	23185.00000
25%	2400.577172	1.281889e+10	2.500000	7.000000	29.716332	-95.545828	77023.00000
50%	4090.946125	1.713529e+10	3.500000	28.000000	29.756907	-95.457451	77054.00000
75%	6155.908918	1.713881e+10	4.000000	105.000000	29.838765	-95.370839	77076.00000
max	35276.053768	1.979731e+10	5.000000	4270.000000	38.395365	-95.075330	77598.00000

In [6]:

```
df.columns
```

Out[6]:

```
Index(['alias', 'display_phone', 'distance', 'id', 'image_url', 'is_closed',
       'name', 'phone', 'price', 'rating', 'review_count', 'transactions',
       'url', 'latitude', 'longitude', 'address1', 'address2', 'address3',
       'city', 'country', 'display_address', 'state', 'zip_code',
       'Restaurant Category'],
      dtype='object')
```

In [7]:

```
# Check for missing values
df.isnull().any()
```

Out[7]:

```
alias                  False
display_phone           True
distance               False
id                     False
image_url               True
is_closed              False
name                   False
phone                   True
price                   True
rating                 False
review_count           False
transactions           False
url                    False
latitude                True
longitude               True
address1                True
address2                True
address3                True
city                   False
country                False
display_address        False
state                  False
zip_code                True
Restaurant Category    False
dtype: bool
```

In [8]:

```
# Check for the sums of the missing values

df.isnull().sum()
```

Out[8]:

```
alias                     0
display_phone           391
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
```

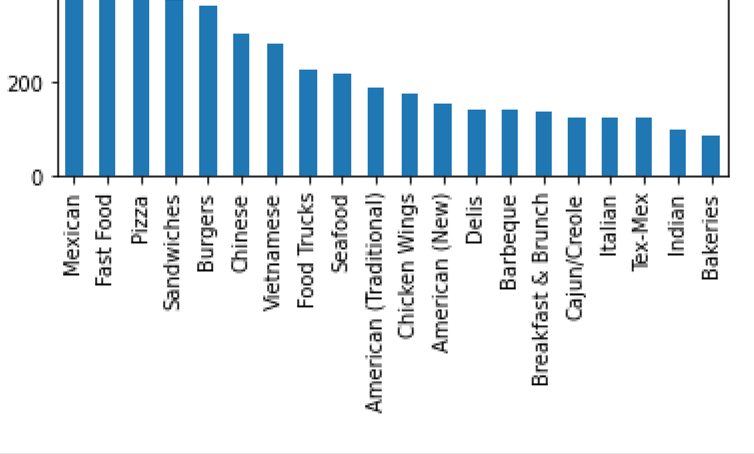
In [10]:

```
# Plot the restaurant category data

import matplotlib.pyplot as plt
df["Restaurant Category"].value_counts()[:20].plot(kind = "bar")
```

Out[10]:

<AxesSubplot:>



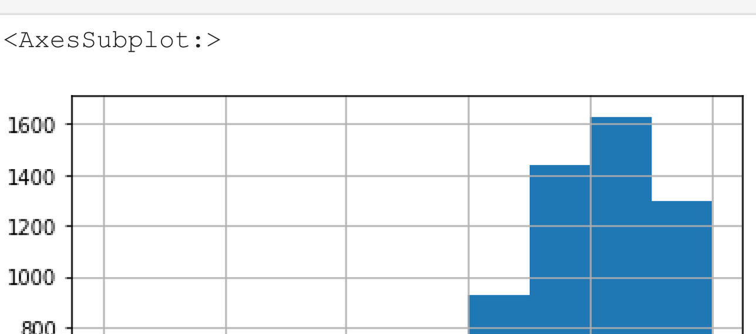
In [11]:

```
# Distribution of the ratings on a histogram

df["rating"].hist()
```

Out[11]:

<AxesSubplot:>



In [12]:

```
df.head(1)
```

Out[12]:

	alias	display_phone	distance	id	image_url	is_closed	name	phone	price	rating
0	bird-haus-houston	NaN	1353.224697	y8zMac68WqA37d4IVYCtdw	https://s3-media4.fl.yelpcdn.com/bphoto/XcbSFu...	False	Bird Haus	NaN	$$	4.5

1 rows × 24 columns

In [13]:

```
# Filter the columns that will be used in the visualization

rest = df[["rating","latitude","longitude","name","address1"]] # double brackets make a new data frame. if neec
rest.head()
```

Out[13]:

	rating	latitude	longitude	name	address1
0	4.5	29.760360	-95.361582	Bird Haus	1010 Prairie St
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
rest.shape
```

```
d:\mambaforge\lib\site-packages\pandas\util\_decorators.py:311: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  return func(*args, **kwargs)
```

Out[17]:

```
(7126, 5)
```

In [18]:

```
rest.isnull().sum()
# rest.isnull().any()
```

Out[18]:

```
rating       0
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

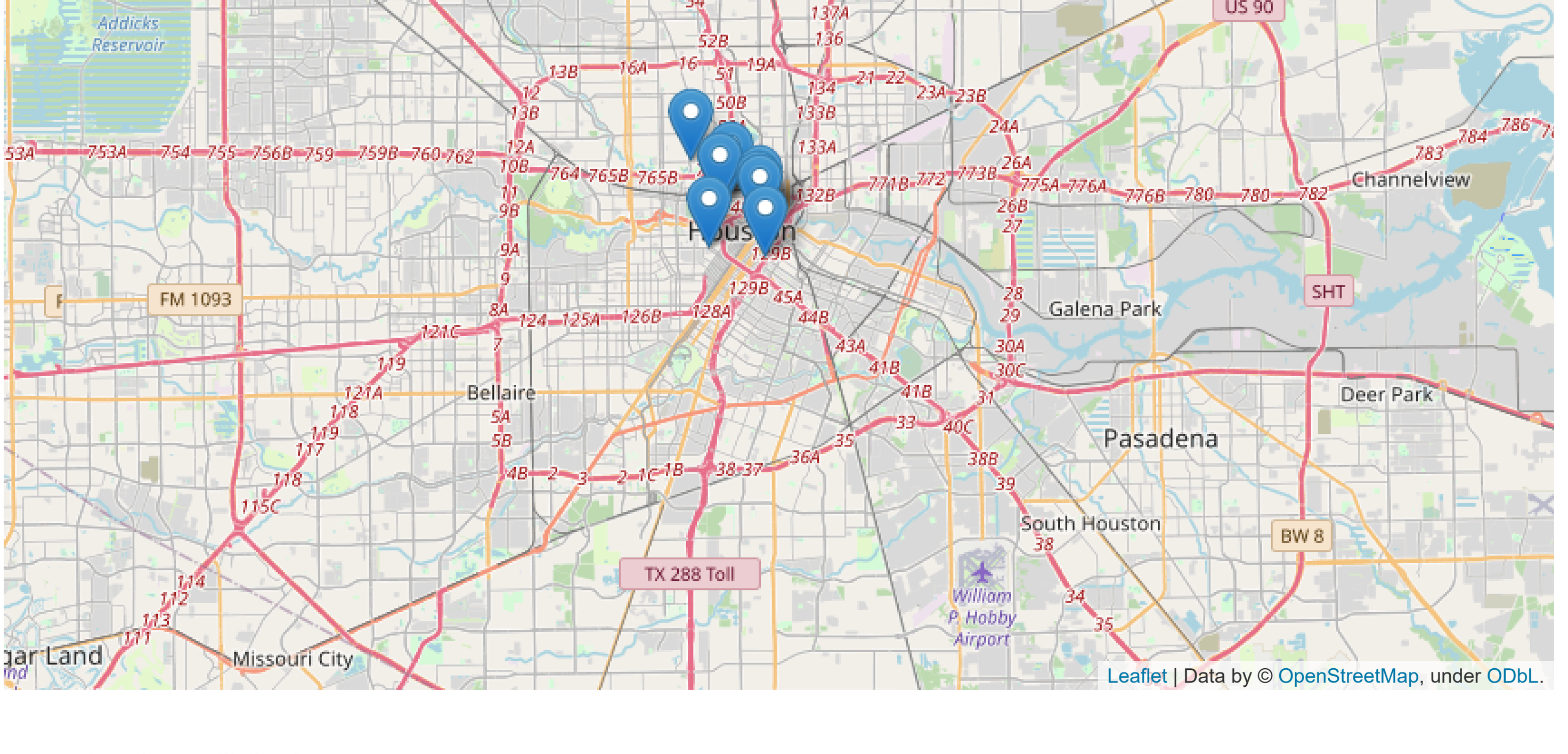
In [20]:

```
# CREATE THE MAP. THIS PROCESS TAKES 1-2 MINUTES TO COMPLETE
# REPLACE range(10) 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(10):
    folium.Marker([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
```

Out[20]:



END OF CODE