Activity: Cartogram Using The `folium` Module

Now, let's a simple map of Mumbai to get started with cartograms. The location coordinates of the Mumbai city is `19.0760` and `72.8777`. To get the coordinates of a city, just google `city_name coordinates`.

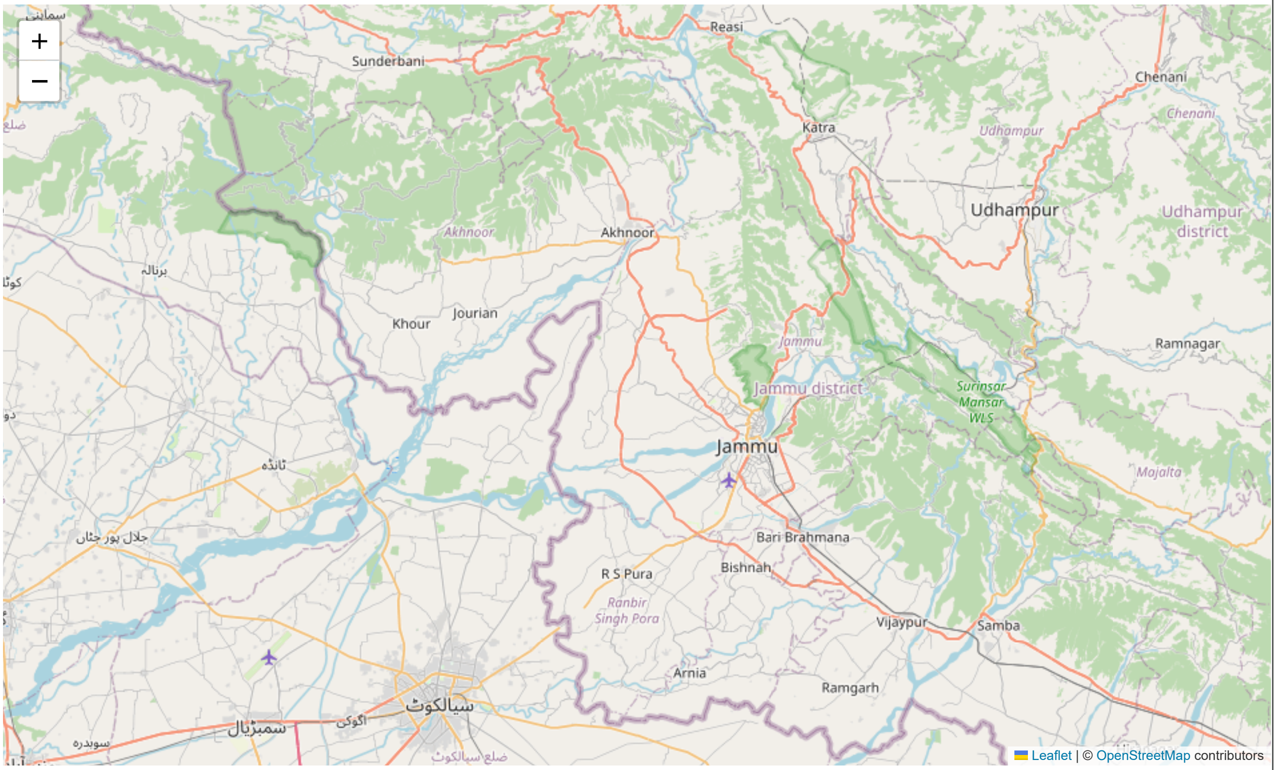
To create a cartogram (or a map) you first need to import the `folium` module.

```
import folium
```

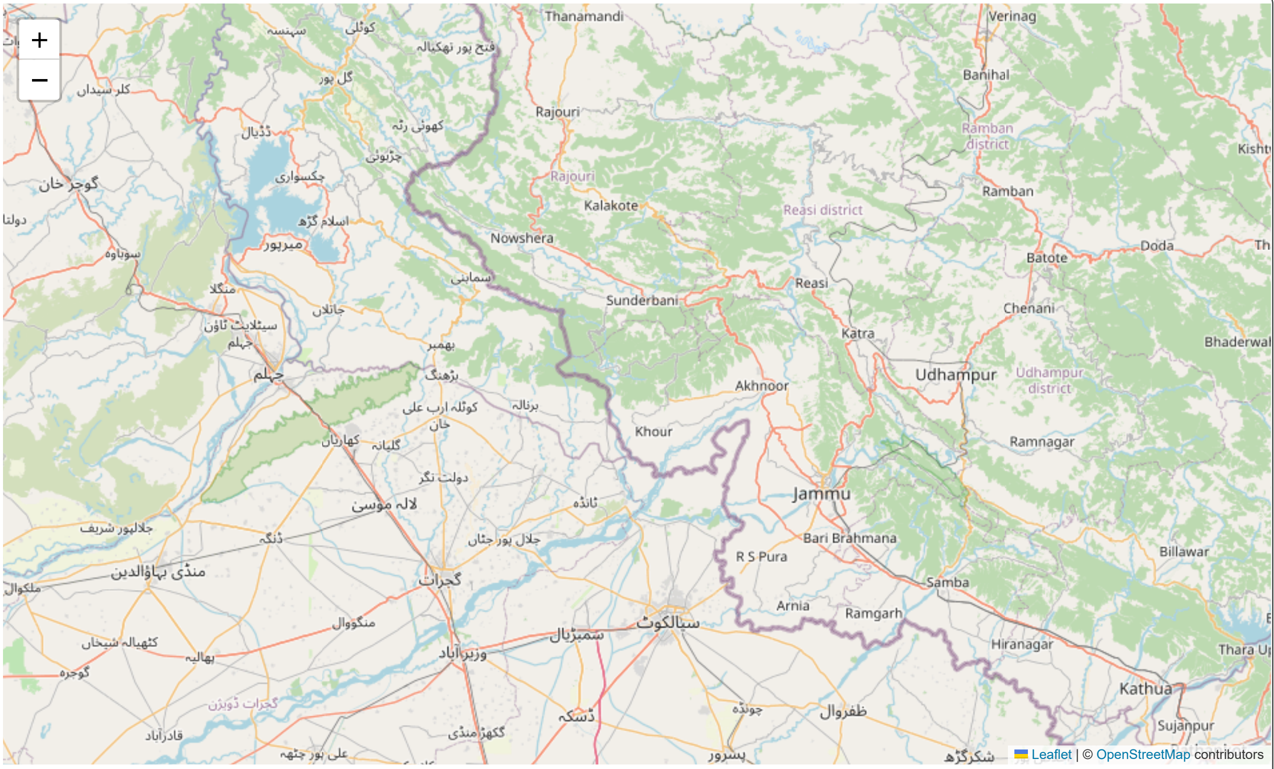
Then you need to call the `Map()` function which exists in the `folium` module. It mandatorily requires the coordinates (latitude and longitude) of a place.

```
folium.Map(location=[latitude, longitude])
```

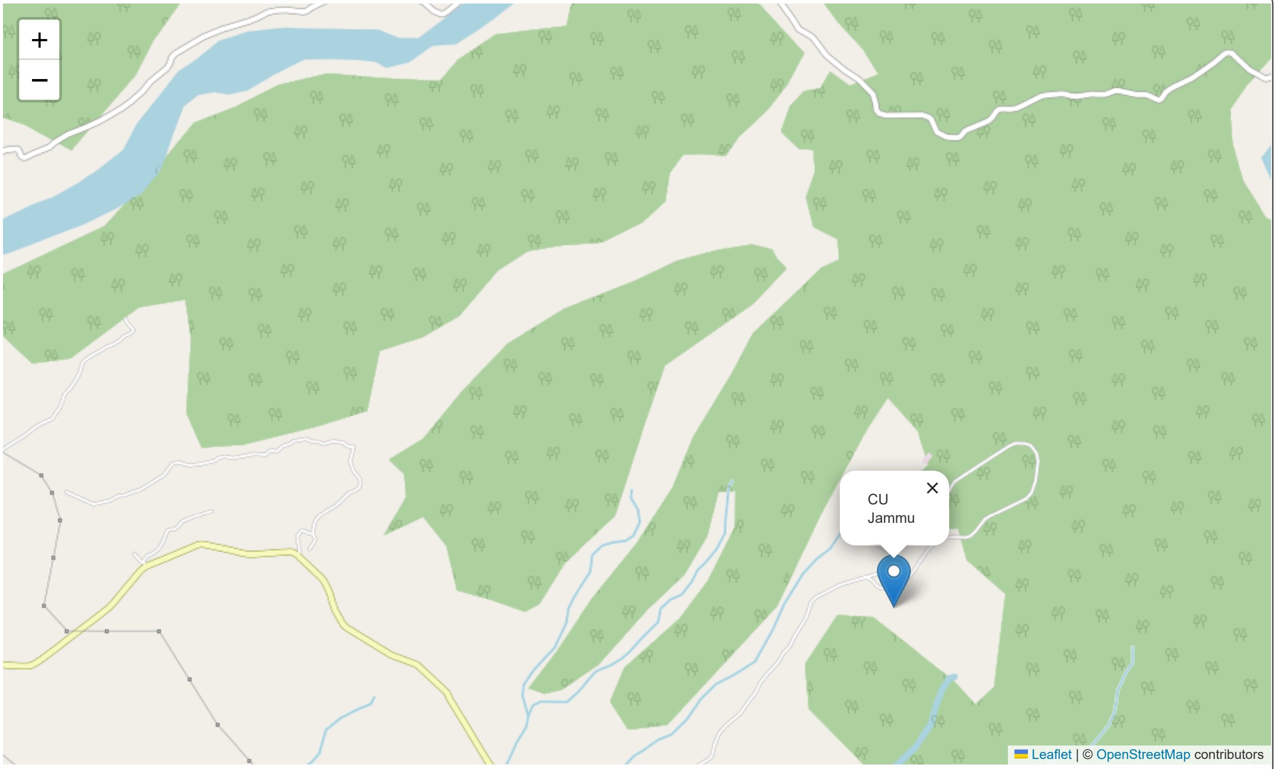
```
# Create a map for the CU Jammu location.
import folium
cu_map=folium.Map(location=[32.6343,75.0129] )
cu_map
```



```
folium.Map(location=[32.6343,75.0129],zoom_start=9)
```



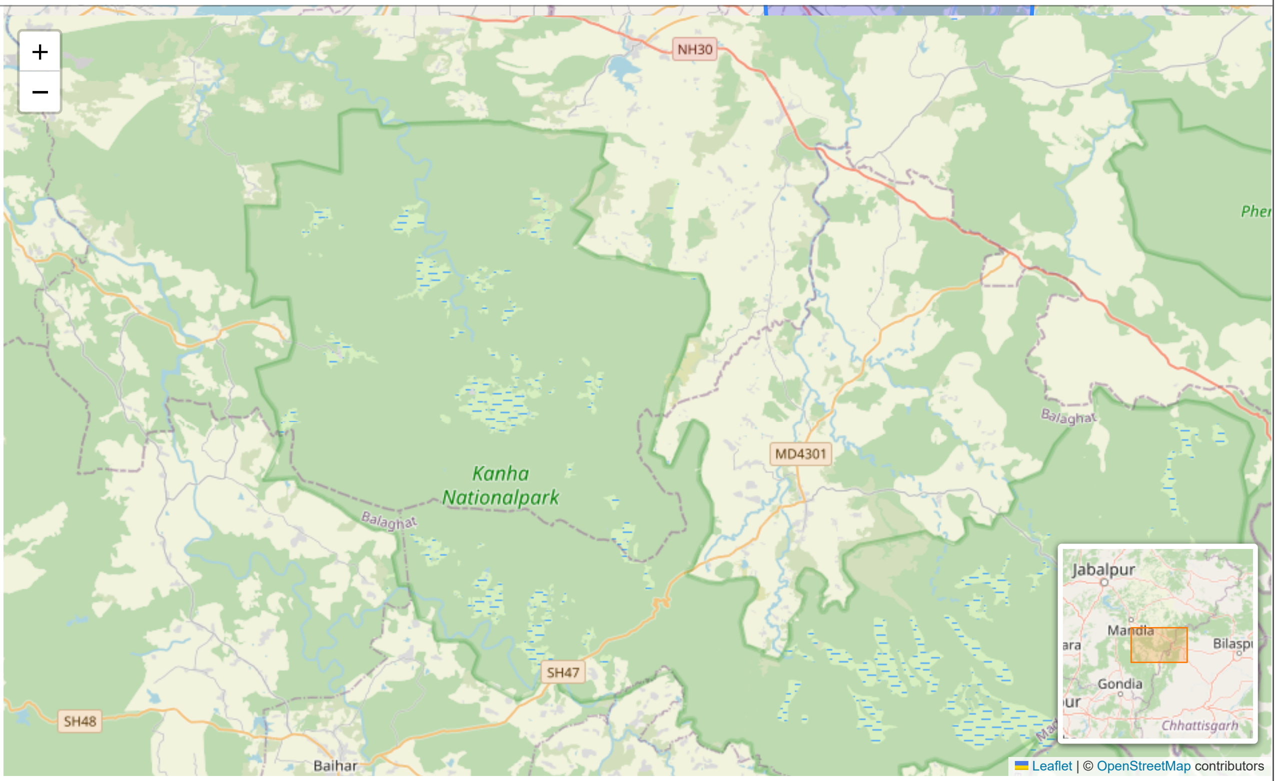
```
# Create a map fwith marker added to CU Jammu location.
folium.Marker(location=[32.6343,75.0129],popup="CU Jammu").add_to(cu_map)
cu_map
```



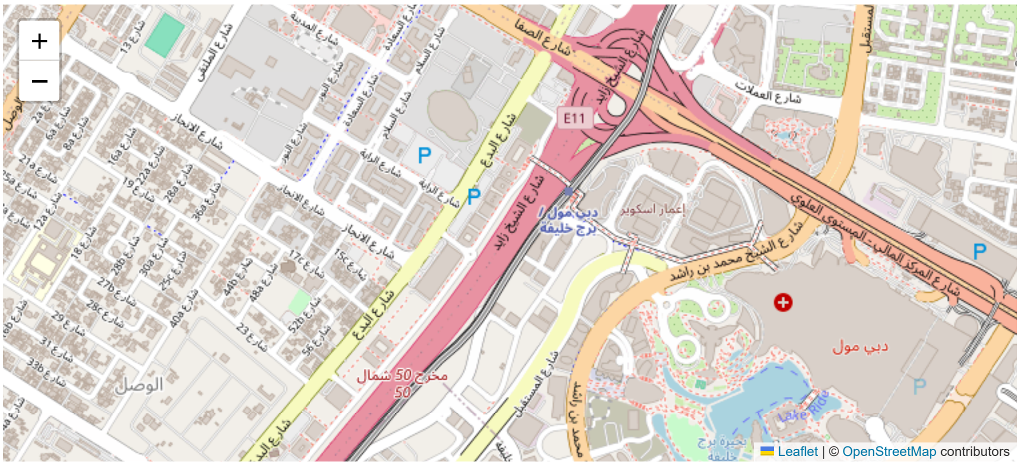
```
# Create a region.
m = folium.Map(location=[32.6343,75.0129])
folium.Marker(location=[32.6343,75.0129], popup =  'CU_JAMMU').add_to(m)
folium.CircleMarker(location=(32.6343,75.0129),radius=100, fill_color='blue').add_to(m)
m
```

+
−
Reasi
Sunderbani
Chenani
Chenani
Katra
Udhampur
Udhampur
Akhnoor
Akhnoor
Udhampur district
Jourian

```
# Create a minimap.
from folium import plugins
m = folium.Map(location=(20.5937, 78.9629), zoom_start=11)
minimap = plugins.MiniMap()
m.add_child(minimap)
m
```



```
# Create a map for the DUBAI.
map_dubai=folium.Map(location=[ 25.197525, 55.2744],width="80%",height="60%",zoom_start
map_dubai
```



Meteorite Landings Map

```
# Connect to the drive
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=
```

```
cd/content/gdrive/My Drive
```

```
/content/gdrive/My Drive
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

met_df = pd.read_csv('meteorite-landings.csv')
```

Meteorite Landings - Histogram

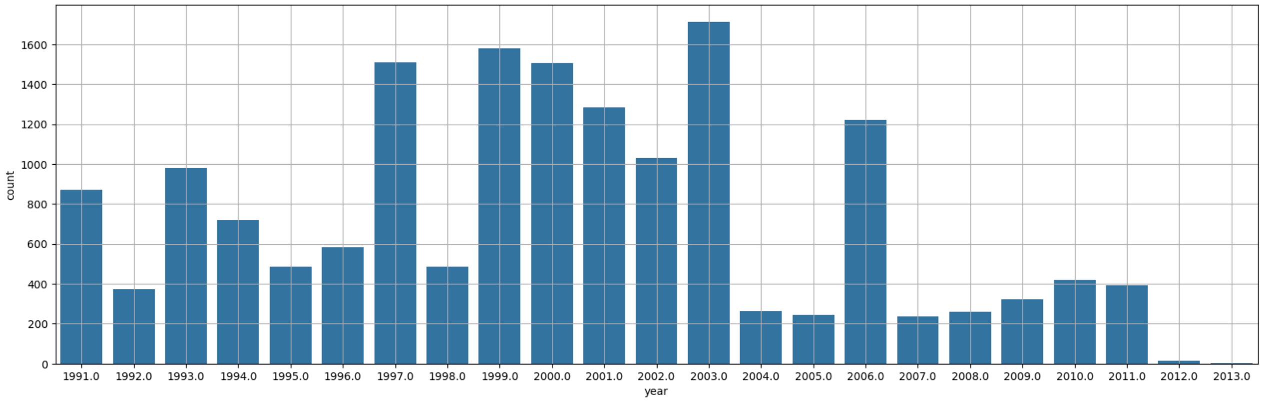
```
# Below are the activities that have been covered in the previous class.
# 1. Import the necessary libraries for this class and create a DataFrame.

# 2. Find the number of rows and columns in the DataFrame.
print(met_df.shape)

# 3. Rows containing the year values less than 860 and greater than 2016.
correct_years_df = met_df[(met_df['year'] >= 860) & (met_df['year'] <= 2016)]

# 4. Rows having the 'reclong' values greater than or equal to -180 degrees and less t
correct_long_df = correct_years_df[(correct_years_df['reclong'] >= -180) & (correct_ye

# 5. Rows containing the '0 reclat' and '0 reclong' values from the 'correct_long_df'.
correct_lat_long_df = correct_long_df[~((correct_long_df['reclat'] == 0 ) & (correct_l

# 6. Indices of the rows having missing mass values.
row_indices = correct_lat_long_df[correct_lat_long_df['mass'].isnull() == True].index
```

```
# 7. Missing values in the 'mass' column in the 'correct_lat_long_df' DataFrame with m
median_mass = correct_lat_long_df['mass'].median()
correct_lat_long_df.loc[row_indices, 'mass'] = median_mass

# 8. Convert the 'year' values into an integer type values.
correct_lat_long_df.loc[:, 'year'] = correct_lat_long_df.loc[:, 'year'].astype('int')

met_after_1990_df=correct_lat_long_df[correct_lat_long_df["year"]>1990]
```

```
(45716, 10)
```

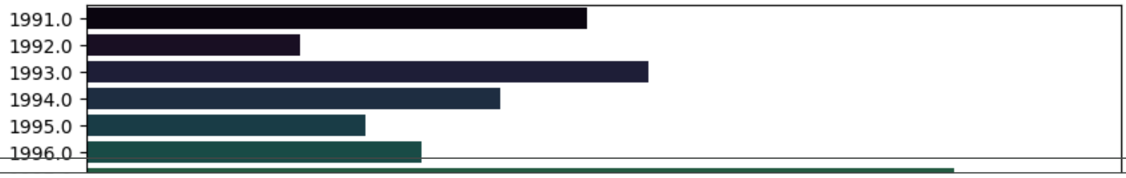
Graph

```
# Create a count plot for the 'year' values in the 'met_after_1990_df' DataFrame on th

import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize= (20,6))
sns.countplot(x= "year",data=met_after_1990_df)
plt.grid()
plt.show()
```



```
#Create a count plot for the 'year' values in the 'met_after_1990_df' DataFrame on the
plt.figure(figsize= (10,6))
sns.countplot(y= "year",data=met_after_1990_df, palette= "cubehelix")

plt.show()
```

```
/tmp/ipython-input-1266544160.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and

  sns.countplot(y= "year",data=met_after_1990_df, palette= "cubehelix")
```



```
# Create a cartogram for the landing sites of the meteorites found in the withered cor
import folium

Valid_df=correct_lat_long_df[(correct_lat_long_df["nametype"]=="Valid") & (correct_lat
Relict_df=correct_lat_long_df[(correct_lat_long_df["nametype"]=="Relict") & (correct_l

map1 = folium.Map(location=[0, 0], width='90%', height='90%', zoom_start=1)
for i in Relict_df.index:
  folium.Marker(location=[Relict_df.loc[i, 'reclat'], Relict_df.loc[i, 'reclong']],pop
map1
```