In [6]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('Datasets/auto-mpg.csv')
df.info()
```

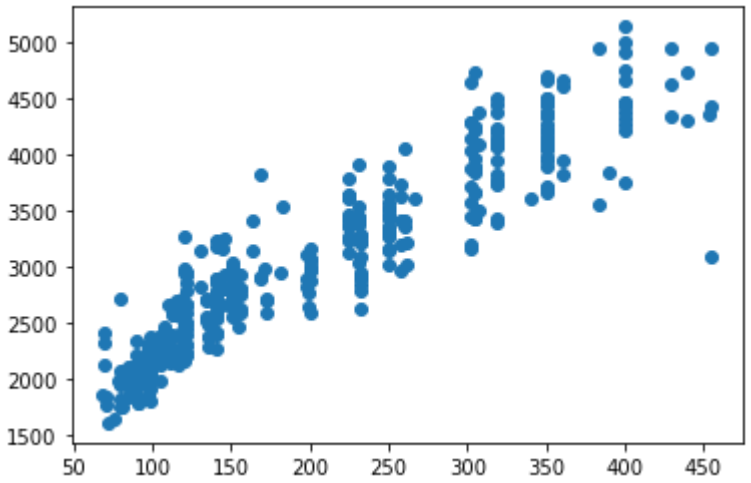
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64  
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    object 
 4   weight        398 non-null    int64  
 5   acceleration  398 non-null    float64
 6   model year    398 non-null    int64  
 7   origin        398 non-null    int64  
 8   car name      398 non-null    object 
dtypes: float64(3), int64(4), object(2)
memory usage: 28.1+ KB
```

In [5]:
```
1 df.corr()
```

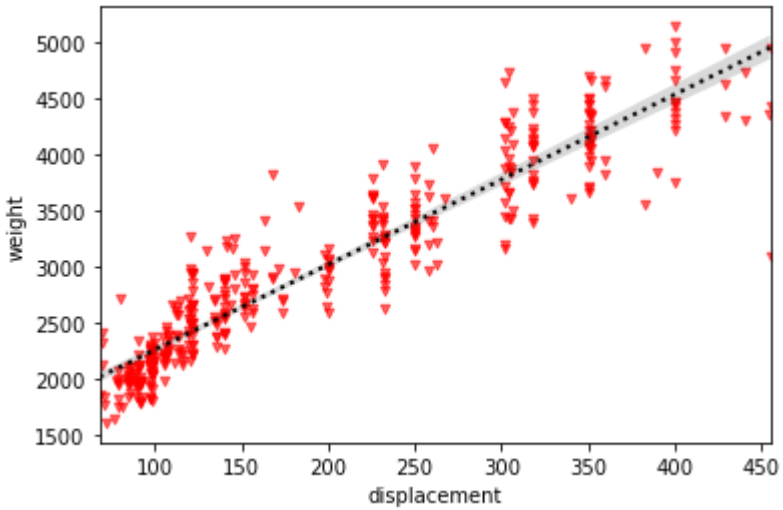
Out[5]:

	mpg	cylinders	displacement	weight	acceleration	model year	origin
mpg	1.000000	-0.775396	-0.804203	-0.831741	0.420289	0.579267	0.563450
cylinders	-0.775396	1.000000	0.950721	0.896017	-0.505419	-0.348746	-0.562543
displacement	-0.804203	0.950721	1.000000	0.932824	-0.543684	-0.370164	-0.609409
weight	-0.831741	0.896017	0.932824	1.000000	-0.417457	-0.306564	-0.581024
acceleration	0.420289	-0.505419	-0.543684	-0.417457	1.000000	0.288137	0.205873
model year	0.579267	-0.348746	-0.370164	-0.306564	0.288137	1.000000	0.180662
origin	0.563450	-0.562543	-0.609409	-0.581024	0.205873	0.180662	1.000000

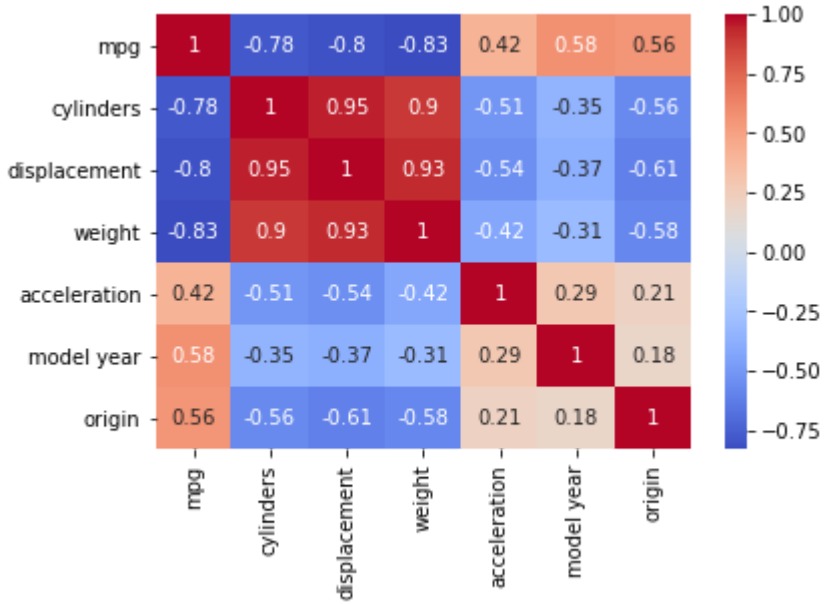
In [4]:
```
1 plt.scatter(df.displacement,df.weight)
2 plt.show()
```



In [13]:
```
sns.regplot(x=df.displacement,y=df.weight,marker='v',
            scatter_kws={'color':'red','s':20,'alpha':0.6,},
           line_kws={'color':'black','ls':':','lw':2})
plt.show()
```



In [16]:
```
sns.heatmap(df.corr(),cmap='coolwarm',annot=True)
plt.show()
```



In [18]:
```
from wordcloud import WordCloud, STOPWORDS
```

In [34]:
```
stopwords=set(STOPWORDS)
stopwords.add('said')
print(stopwords)
```

```
{'did', 'said', 'during', 'having', 'but', 'in', 'above', 'www', "what's", 'up', "they'd", 'once', 'very', 'get', 'ha
d', 'than', 'yourself', 'could', 'am', "hadn't", "wasn't", 'there', "we'd", "shan't", "doesn't", 'k', "let's", 'himsel
f', 'like', 'a', 'have', 'with', 'http', "when's", 'theirs', 'my', 'com', 'is', "mustn't", 'too', 'both', "they're", "i
t's", 'if', 'an', 'how', "we're", 'off', 'his', "i'll", 'at', 'when', "didn't", "i'm", 'over', 'however', 'under', "the
y've", 'its', 'has', "they'll", 'again', 'yourselves', "why's", 'do', 'from', 'hers', "she'd", 'been', 'them', 'you',
'being', 'same', 'otherwise', "shouldn't", 'their', 'those', 'each', 'own', 'herself', "haven't", 'should', 'any', 'wha
t', 'it', "aren't", 'he', 'shall', 'just', "couldn't", "weren't", 'who', "you've", 'few', "isn't", 'nor', 'hence', 'n
o', 'or', "you'll", 'ought', 'such', 'then', 'ourselves', 'r', 'this', 'they', "can't", 'we', 'be', "that's", 'other',
'on', 'doing', 'i', "who's", 'because', 'itself', 'of', 'about', "he'll", 'ours', 'where', 'that', "i've", "she'll", "w
here's", 'as', 'for', 'therefore', "you're", 'cannot', 'me', 'since', 'not', 'she', 'by', 'ever', 'why', 'also', 'som
e', 'through', 'him', 'to', 'her', 'which', 'most', 'all', "we've", 'whom', 'and', "he'd", "here's", "she's", 'these',
"won't", "how's", 'can', 'out', "don't", 'the', 'against', 'myself', 'were', 'before', 'down', "hasn't", 'themselves',
'further', 'was', "you'd", "he's", 'else', 'so', 'until', 'only', 'yours', 'more', 'does', "i'd", "we'll", 'into', 'bel
ow', 'your', "wouldn't", 'are', 'while', 'would', 'here', 'after', 'between', "there's", 'our'}
```

In [21]:
```
import requests
```

In [26]:
```
url=("https://www.gutenberg.org/files/11/11-0.txt")
response=requests.get(url)
text=response.text
print(text)
```

```
*** START OF THE PROJECT GUTENBERG EBOOK 11 ***
[Illustration]



Alice’s Adventures in Wonderland

by Lewis Carroll

THE MILLENNIUM FULCRUM EDITION 3.0

Contents

 CHAPTER I.     Down the Rabbit-Hole
 CHAPTER II.    The Pool of Tears
 CHAPTER III.   A Caucus-Race and a Long Tale
 CHAPTER IV.    The Rabbit Sends in a Little Bill
 CHAPTER V.     Advice from a Caterpillar
```

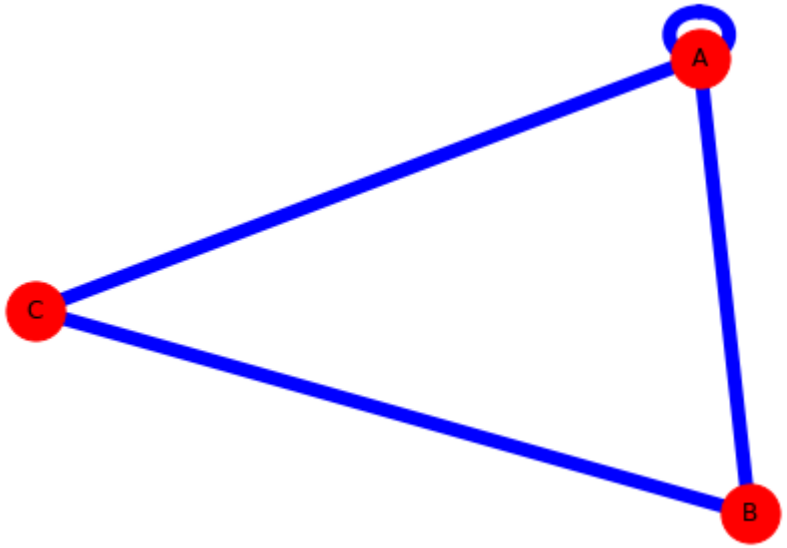
In [36]:
```
alice_wc=WordCloud(background_color='white',
                   max_words=2000,
                   stopwords=stopwords)
```

In [37]:
```
alice_wc.generate(text)
plt.imshow(alice_wc)
plt.axis('off')
plt.show()
```

In [41]:
```
import networkx as nx
```

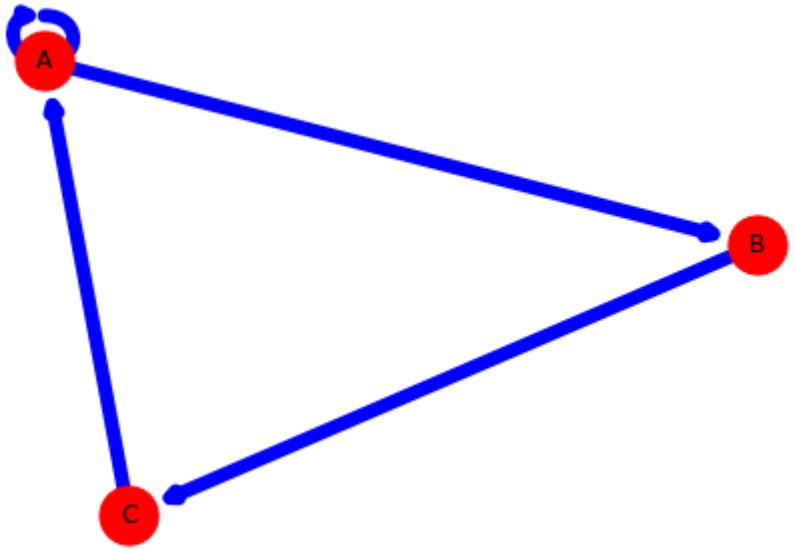
In [45]:

```
G=nx.Graph()
G.add_node('A')
G.add_node('B')
G.add_node('C')
G.add_edge('A','B')
G.add_edge('B','C')
G.add_edge('C','A')
G.add_edge('A','A')
nx.draw(G,node_size=850,node_color='red',width=7,with_labels=True,edge_color='blue')
plt.show()


```

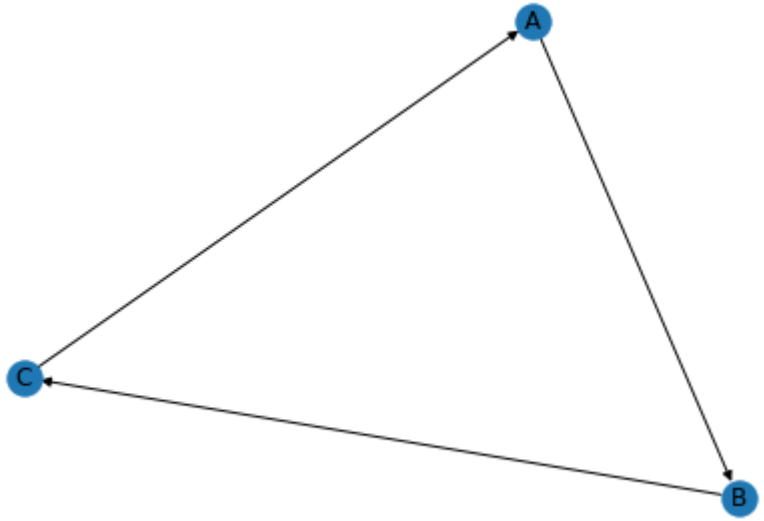


In [50]:

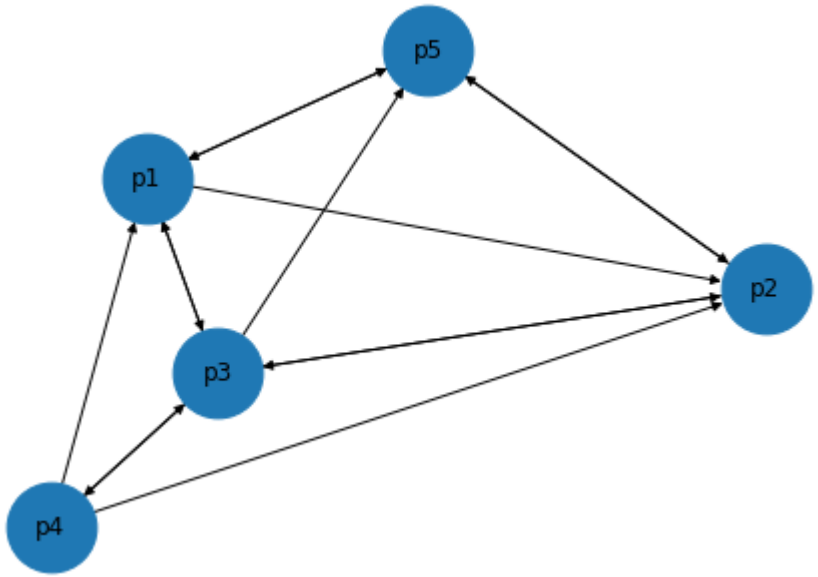
```
G=nx.DiGraph()
G.add_node('A')
G.add_node('B')
G.add_node('C')
G.add_edge('A','B')
G.add_edge('B','C')
G.add_edge('C','A')
G.add_edge('A','A')
nx.draw(G,node_size=850,node_color='red',width=7,with_labels=True,edge_color='blue')
plt.show()


```



In [53]:
```
G=nx.DiGraph()
G.add_nodes_from(['A','B','C'])
G.add_edges_from([('A','B'),('B','C'),('C','A')])
nx.draw(G,with_labels=True)
```



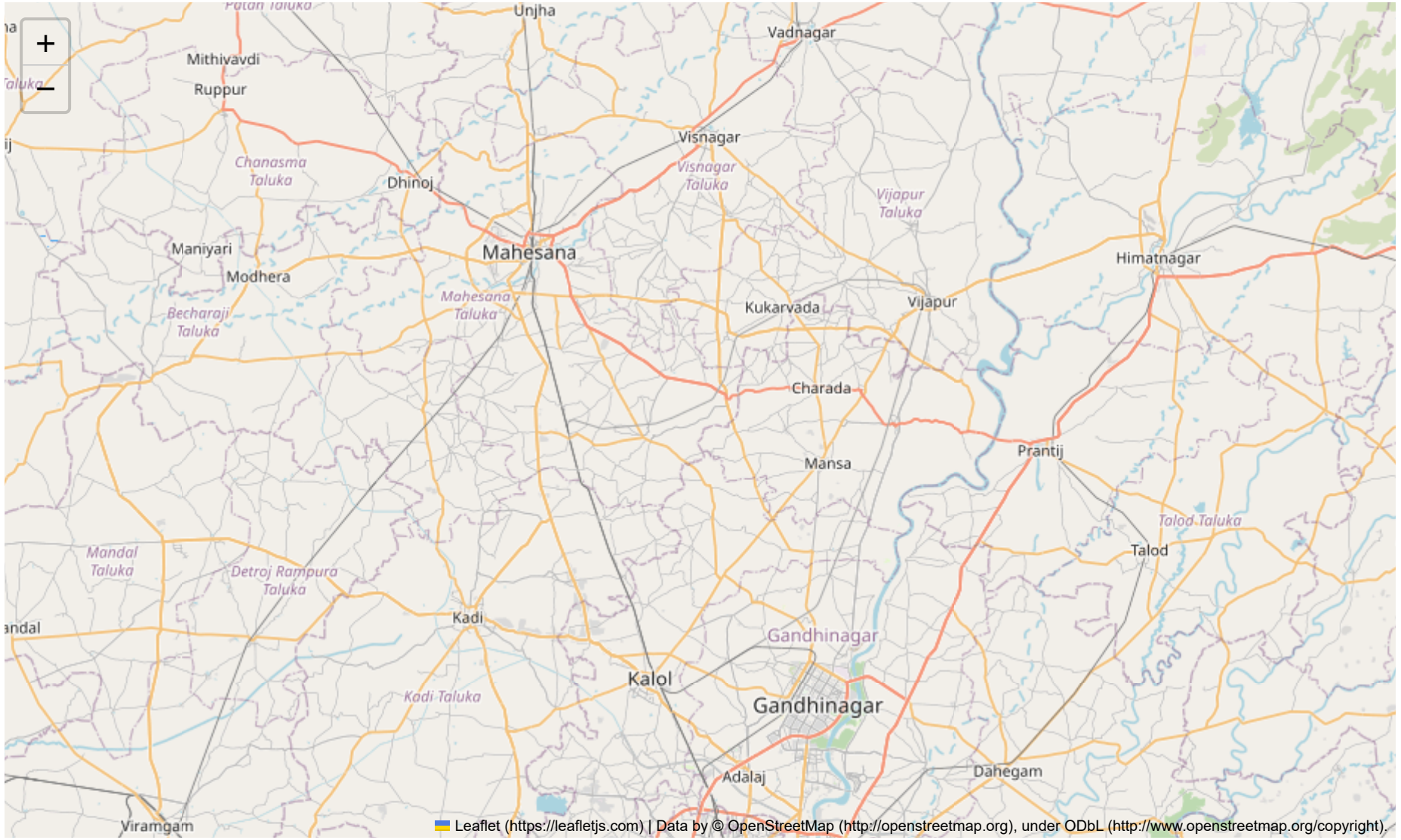
In [59]:
```
insta={'p1':[0,1,1,0,1],
       'p2':[0,0,1,0,1],
       'p3':[1,1,0,1,1],
       'p4':[1,1,1,0,0],
       'p5':[1,1,0,0,0]
       }
G=nx.DiGraph()
for i , follower in enumerate(insta):
    for j, follows in enumerate(insta[follower]):
        if follows:
            G.add_edge(follower,list(insta.keys())[j])
nx.draw(G,with_labels=True,node_size=2000)
```



In [62]:
```
1
```

In [65]:
```
import folium
mapobj=folium.Map(location=[23.4377300754168,72.585859],
                  zoom_start=10,tiles='OpenStreetMap')
mapobj
```

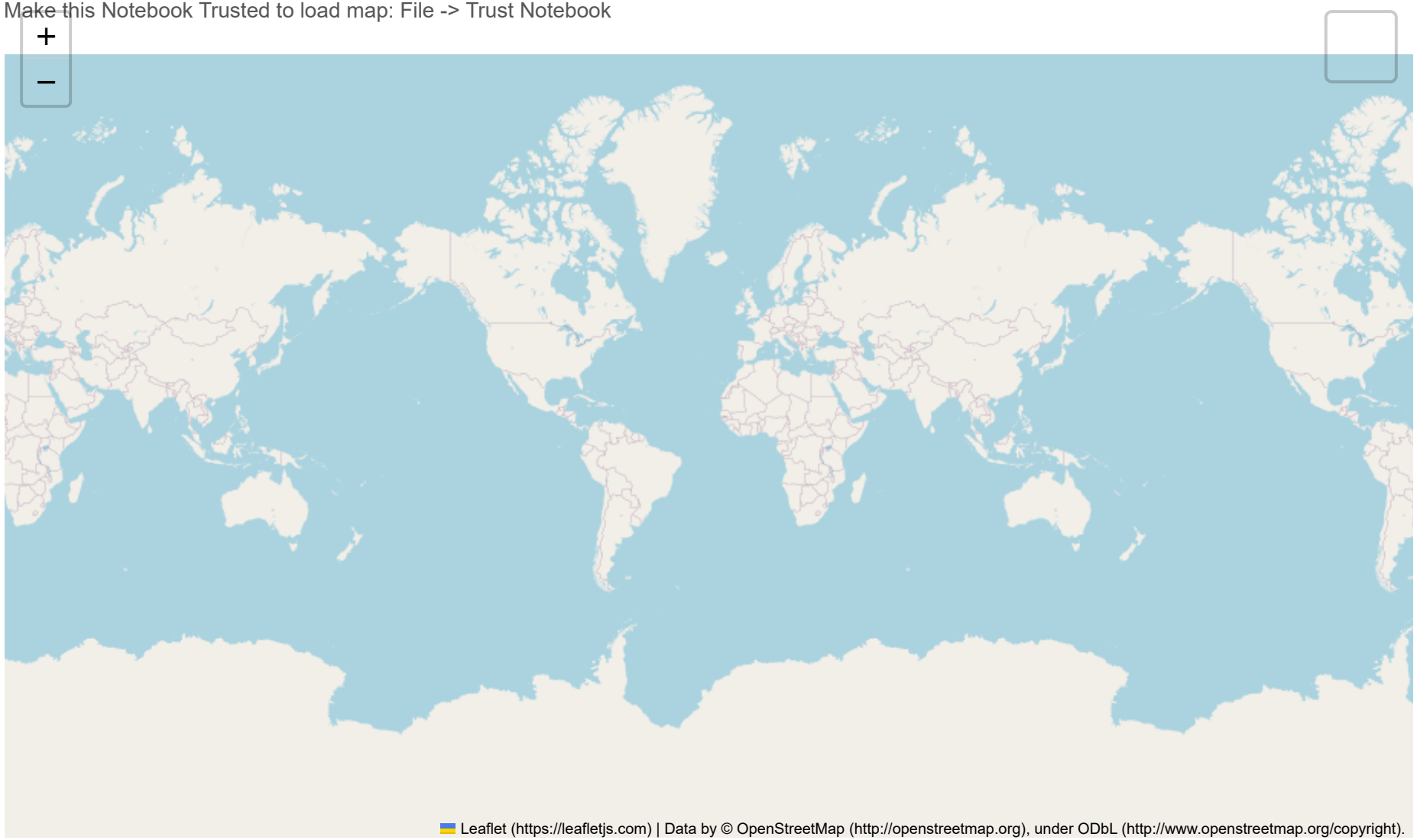
Out[65]:



In [69]:
```
1 import folium
2 myobj=folium.Map()
3 folium.LayerControl().add_to(myobj)
4 myobj
```

Out[69]:

Make this Notebook Trusted to load map: File -> Trust Notebook



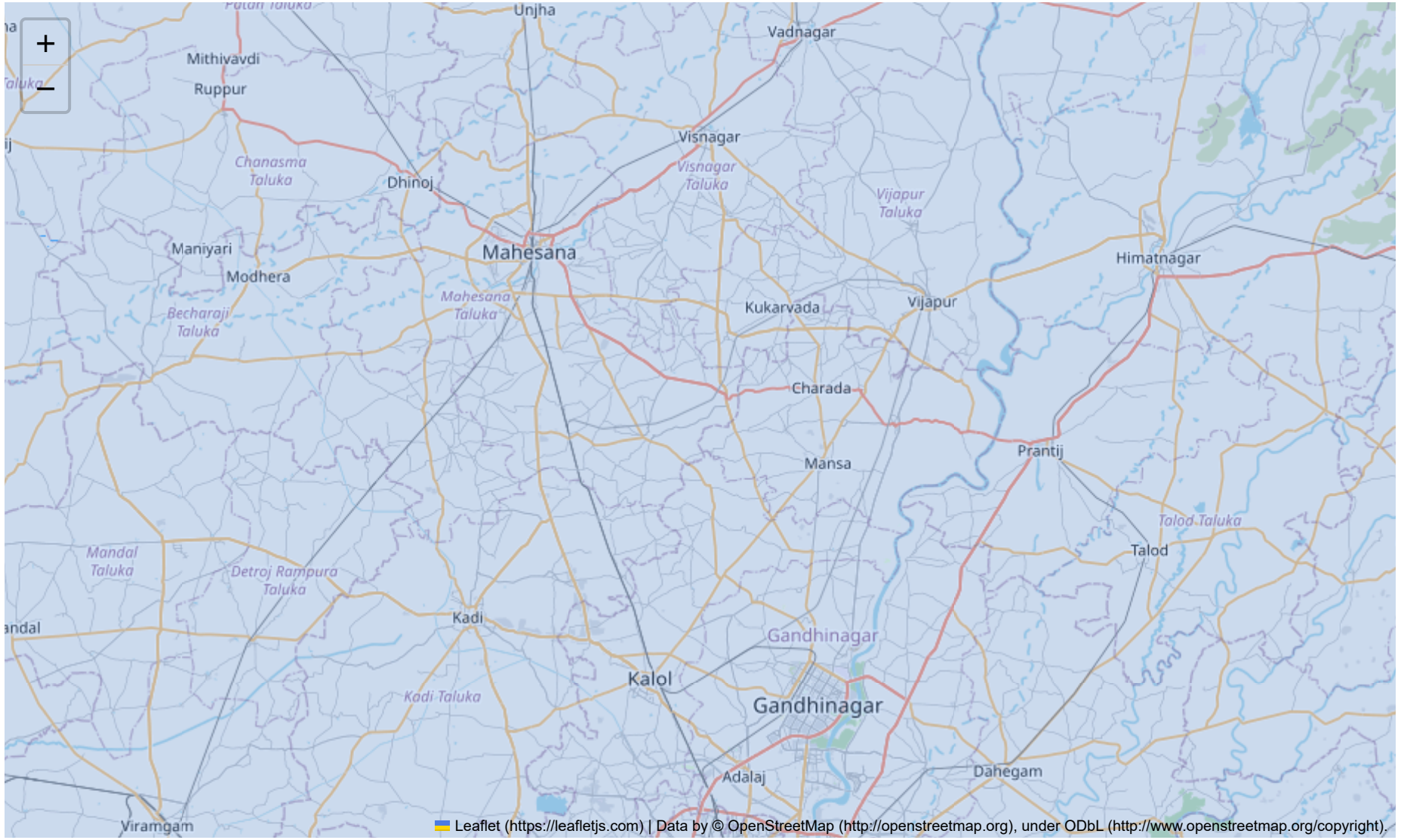
In [71]:

```
mapobj=folium.Map()
folium.TileLayer('OpenStreetMap').add_to(mapobj)
folium.TileLayer('stamenterrain',attr='stamenterrain').add_to(mapobj)
folium.TileLayer('stamenwatercolor',attr='stamenwatercolor').add_to(mapobj)
folium.LayerControl().add_to(mapobj)
mapobj
mapobj.save('output.html')
```

In [75]:

```
import folium
mapobj=folium.Map(location=[23.4377300754168,72.585859],
                  zoom_start=10,tiles='OpenStreetMap')
layer1=folium.GeoJson(data=(open('Datasets/Indian_States.geojson','r').read()),
                      name='India')
layer1.add_to(mapobj)
mapobj.save('india.html')
mapobj
```

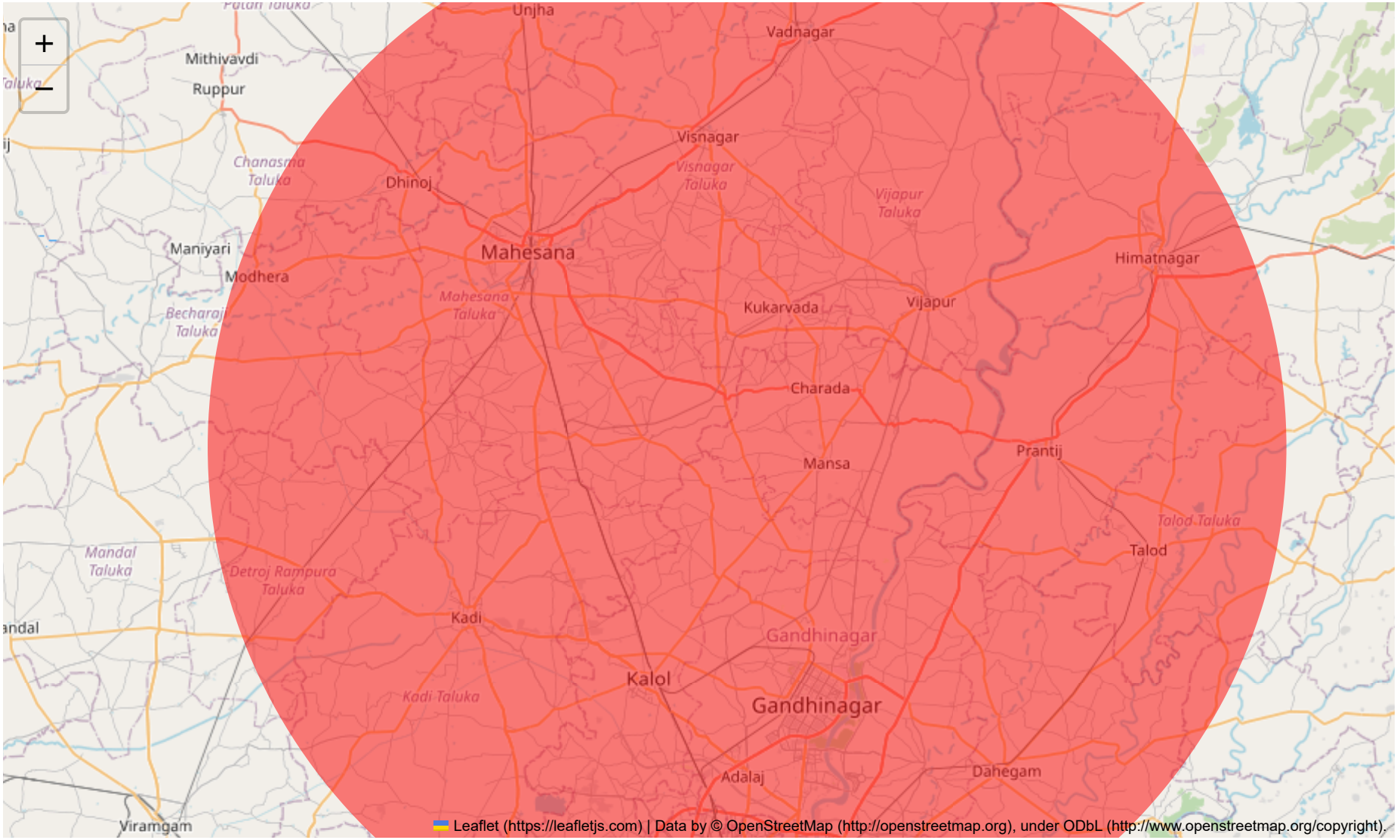
Out[75]:


Unjha
Vadnagar
Mithivavdi
Ruppur
Visnagar
Visnagar Taluka
Chanasma Taluka
Dhinoj
Vijapur Taluka
Maniyari
Modhera
Mahesana
Himatnagar
Mahesana Taluka
Becharaji Taluka
Kukarvada
Vijapur
Charada
Prantij
Mansa
Talod Taluka
Talod
Mandal Taluka
Detroj Rampura Taluka
Kadi
Gandhinagar
Kalol
Kadi Taluka
Gandhinagar
Adalaj
Dahegam
Viramgam
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [81]:

```
import folium
mapobj=folium.Map(location=[23.4377300754168,72.585859],
                  zoom_start=10,tiles='OpenStreetMap')
folium.Circle(location=[23.4377300754168,72.585859],radius=50000,
              color='black',
              weight=0,
              fill_color='red',
              fill_opacity=0.5).add_to(mapobj)

mapobj
```

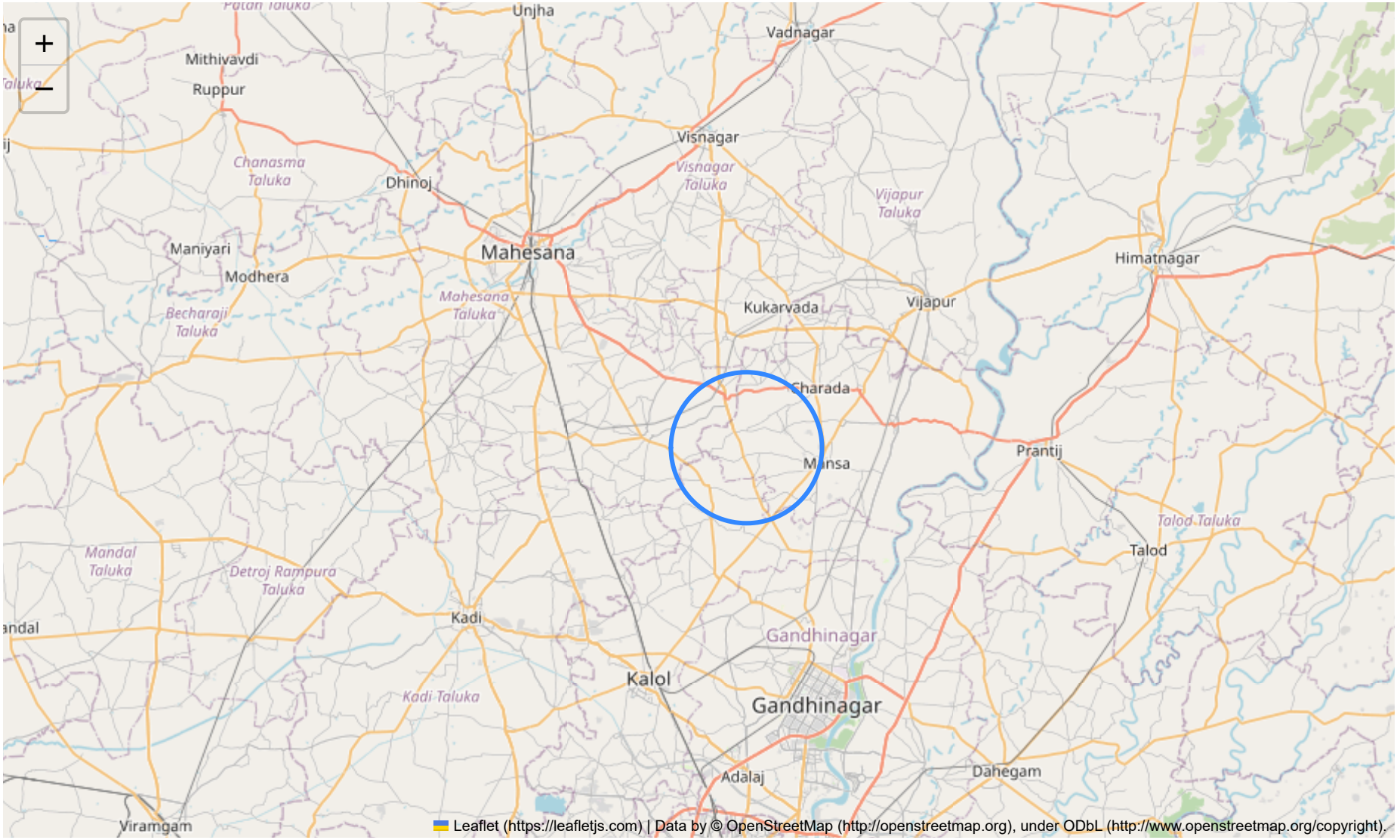
Out[81]:


Unjha
Vadnagar
Mithivavdi
Ruppur
Visnagar
Visnagar Taluka
Chanasma Taluka
Dhinoj
Vijapur Taluka
Maniyari
Mahesana
Himatnagar
Modhera
Mahesana Taluka
Becharaji Taluka
Kukarvada
Vijapur
Charada
Prantij
Mansa
Talod Taluka
Talod
Mandal Taluka
Detroj Rampura Taluka
Kadi
Gandhinagar
Kalol
Kadi Taluka
Gandhinagar
Adalaj
Dahegam
Viramgam
Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).

In [86]:

```
import folium
mapobj=folium.Map(location=[23.4377300754168,72.585859],
                  zoom_start=10,tiles='OpenStreetMap')
folium.CircleMarker(location=[23.4377300754168,72.585859],radius=50,).add_to(mapobj)

mapobj
```

Out[86]:



In [6]:
```
1 import folium
```

In [16]:

```
import folium

m = folium.Map(location=[23.4377, 72.5858], zoom_start=5)

folium.Circle(
    location=[23.4377, 72.5858],
    radius=5000,
    fill=True,
    tooltip="B3 is best",
    popup=folium.Popup('''<h2>This is popup</h2>
                          <a href="http://localhost:8888/tree">Vishal10</a>''', max_width=500)).add_to(m)


m

```

Out[16]:

In [ ]:
```
1
```

In [ ]:
```
1
```