Exploring ipywidgets

In [1]:

```
import ipywidgets
```

IntSlider widget

In [2]:

```
slider = ipywidgets.IntSlider(value=5,
                             min=0,
                             max=10,
                             step=1,
                             )

def square(number):
    return number*number

ipywidgets.interact(square,number=slider)
```

number 2

4

Out[2]:

```
<function __main__.square(number)>
```

link widget

In [3]:

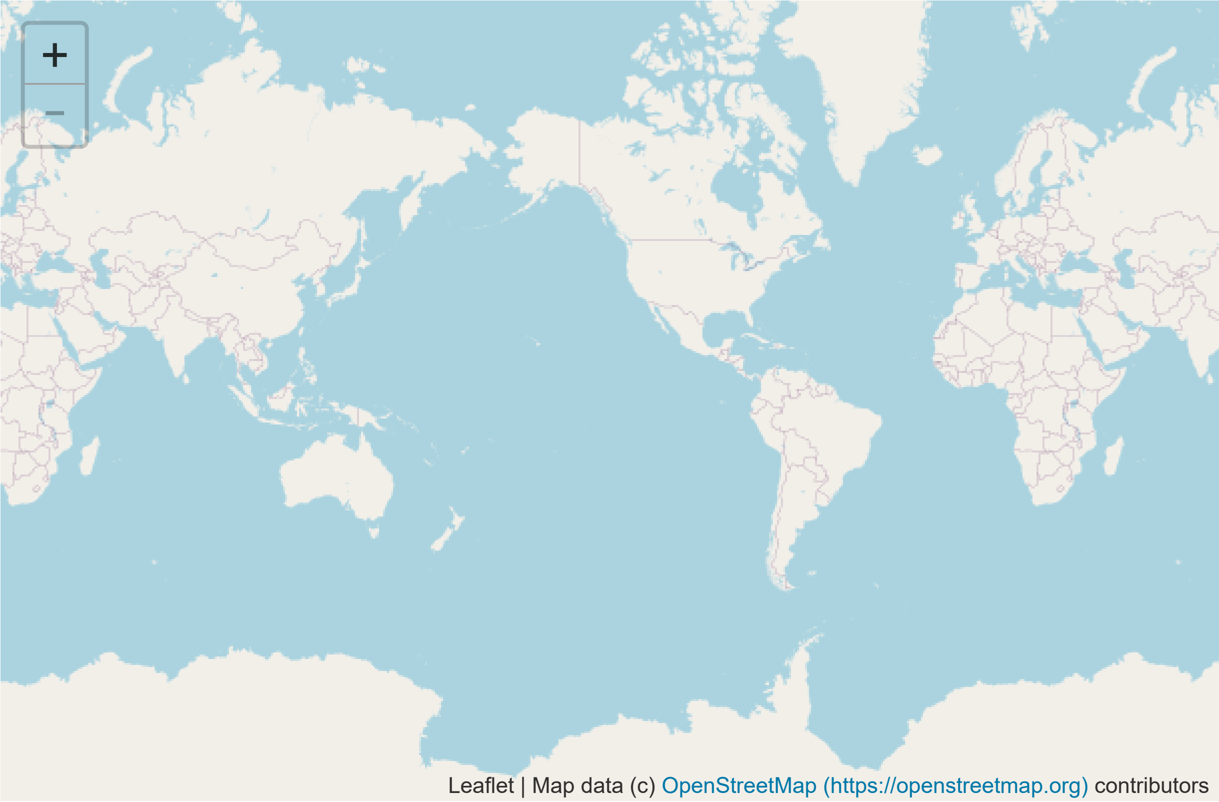
```
a=ipywidgets.widgets.FloatText()
b=ipywidgets.widgets.FloatLogSlider()
link= ipywidgets.link((a,'value'),(b,'value'))
display(a,b)
```

```
79.43282347242814
```

79.4

In [4]:

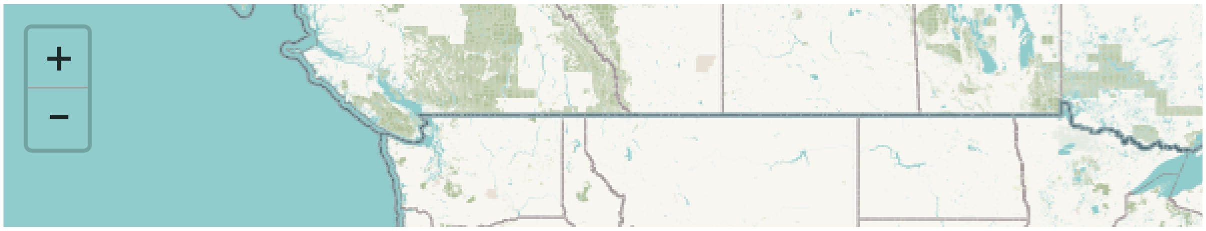
```
import ipyleaflet
from ipyleaflet import Map

bmap=ipyleaflet.Map(zoom=1)
bmap
```

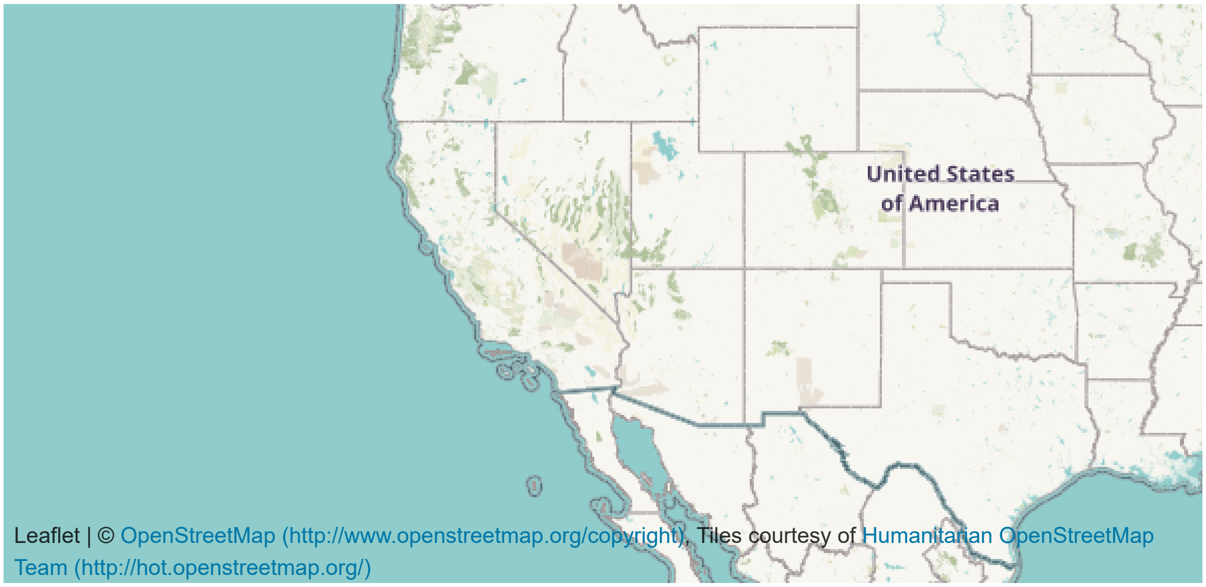

+
−
Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

Exploring ipyleaflet

```
In [5]: from ipyleaflet import basemaps

        radioButton = ipywidgets.RadioButtons(options=['Positron', 'DarkMatter', 'WorldSt
                                                       'WorldTopoMap', 'WorldImagery', '
                                                       'HyddaFull', 'Night', 'ModisTerra
                                                       'Toner', 'Watercolor'],value='Pos
                                              description='Map: ')

        def toggleMap(map):
            if map=='Positron': m=Map(zoom=2,basemap=basemaps.CartoDB.Positron)
            if map == 'DarkMatter': m = Map(zoom=1, basemap=basemaps.CartoDB.DarkMatter)
            if map == 'WorldStreetMap': m = Map(center=(40.67, -73.94), zoom=10, basemap=
            if map == 'DeLorme': m = Map(center=(40, -99), zoom=4, basemap=basemaps.Esri.
            if map == 'WorldTopoMap': m = Map(center=(40, -99), zoom=4, basemap=basemaps.
            if map == 'WorldImagery': m = Map(center=(40, -99), zoom=4, basemap=basemaps.
            if map == 'NatGeoWorldMap': m = Map(center=(20.5,78), zoom=4, basemap=basemap
            if map == 'HikeBike': m = Map(center=(39.73,-104.98), zoom=10, basemap=basema
            if map == 'HyddaFull': m = Map(center=(40, -99), zoom=4, basemap=basemaps.Hyd
            if map == 'Night': m = Map(center=(40, -99), zoom=4, basemap=basemaps.NASAGIB
            if map == 'ModisTerra': m = Map(center=(40, -99), zoom=4, basemap=basemaps.NA
            if map == 'Mapnik': m = Map(center=(40, -99), zoom=4, basemap=basemaps.OpenSt
            if map == 'HOT': m = Map(center=(40, -99), zoom=4, basemap=basemaps.OpenStree
            if map == 'OpenTopoMap': m = Map(center=(40, -99), zoom=4, basemap=basemaps.O
            if map == 'Toner': m = Map(center=(20.5,78), zoom=5, basemap=basemaps.Stamen.
            if map == 'Watercolor': m = Map(center=(40, -99), zoom=4, basemap=basemaps.St
            display(m)

        ipywidgets.interact(toggleMap, map=radioButton)
```

Map: ○ Positron
○ DarkMatter
○ WorldStreetMap
○ DeLorme
○ WorldTopoMap
○ WorldImagery
○ NatGeoWorldMap
○ HikeBike
○ HyddaFull
○ Night
○ ModisTerra
○ Mapnik
● HOT
○ OpenTopoMap
○ Toner
○ Watercolor



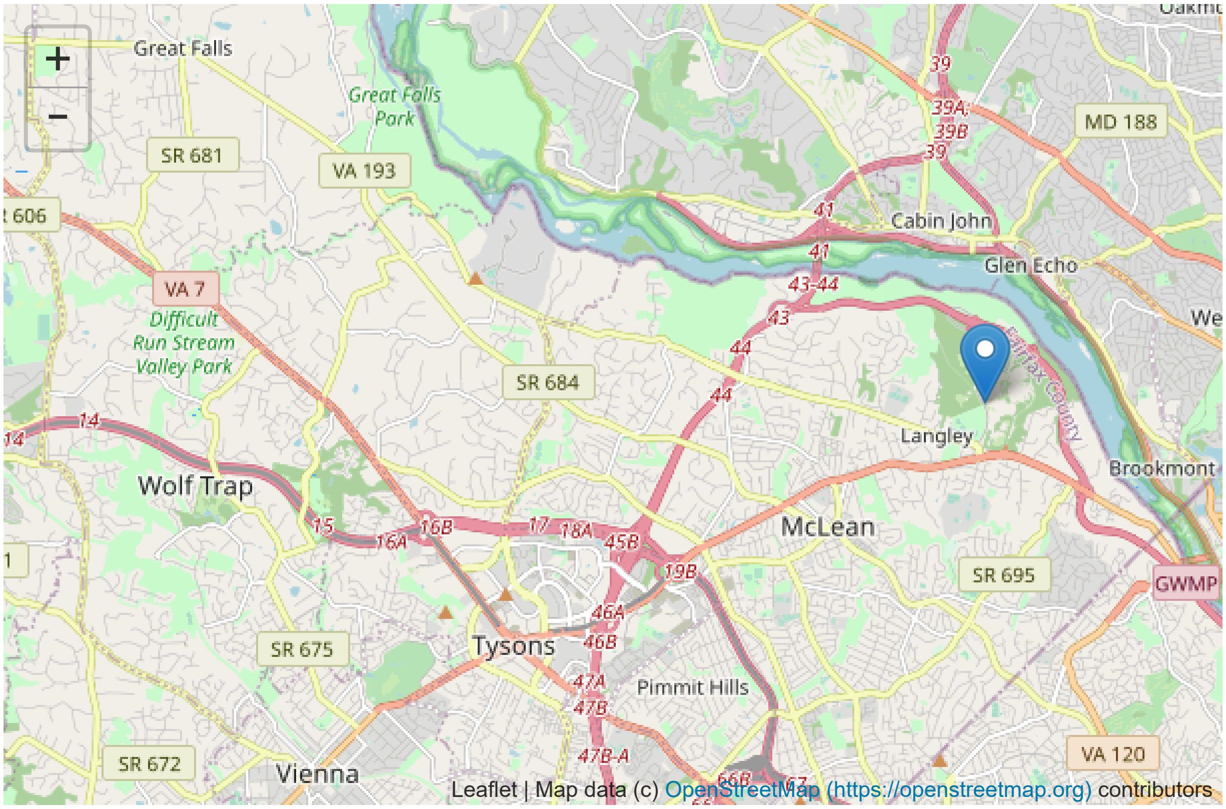


Leaflet | © OpenStreetMap (http://www.openstreetmap.org/copyright), Tiles courtesy of Humanitarian OpenStreetMap Team (http://hot.openstreetmap.org/)

```
Out[5]: <function __main__.toggleMap(map)>
```

Adding Marker

```
In [6]: import geocoder
        from ipyleaflet import Marker
        location = geocoder.osm('1000 Colonial Farm Rd, McLean, VA 22101, United States')
        latlng=[location.lat,location.lng]
        tha=Map(center=latlng)
        marker=Marker(location=latlng,title='CIA HQ')
        tha.add_layer(marker)
        tha
```

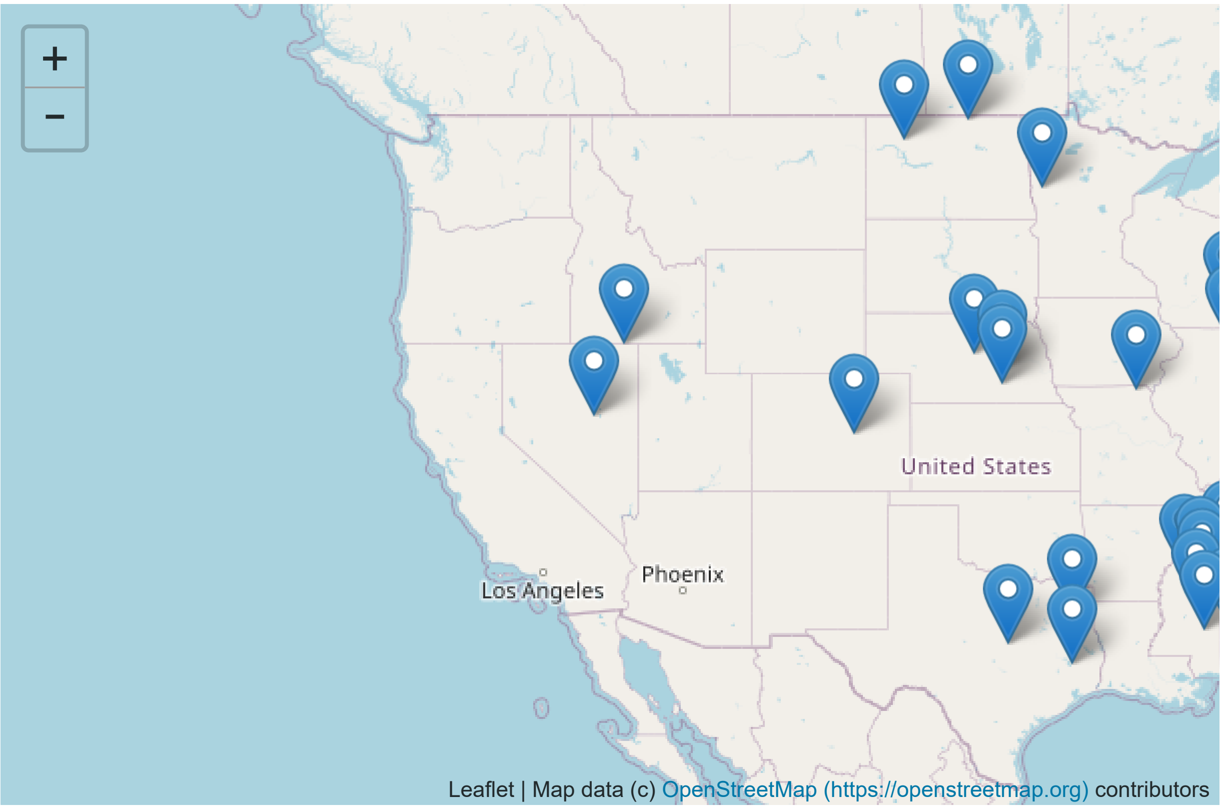


Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

Multiple Custom Markers

In [7]:

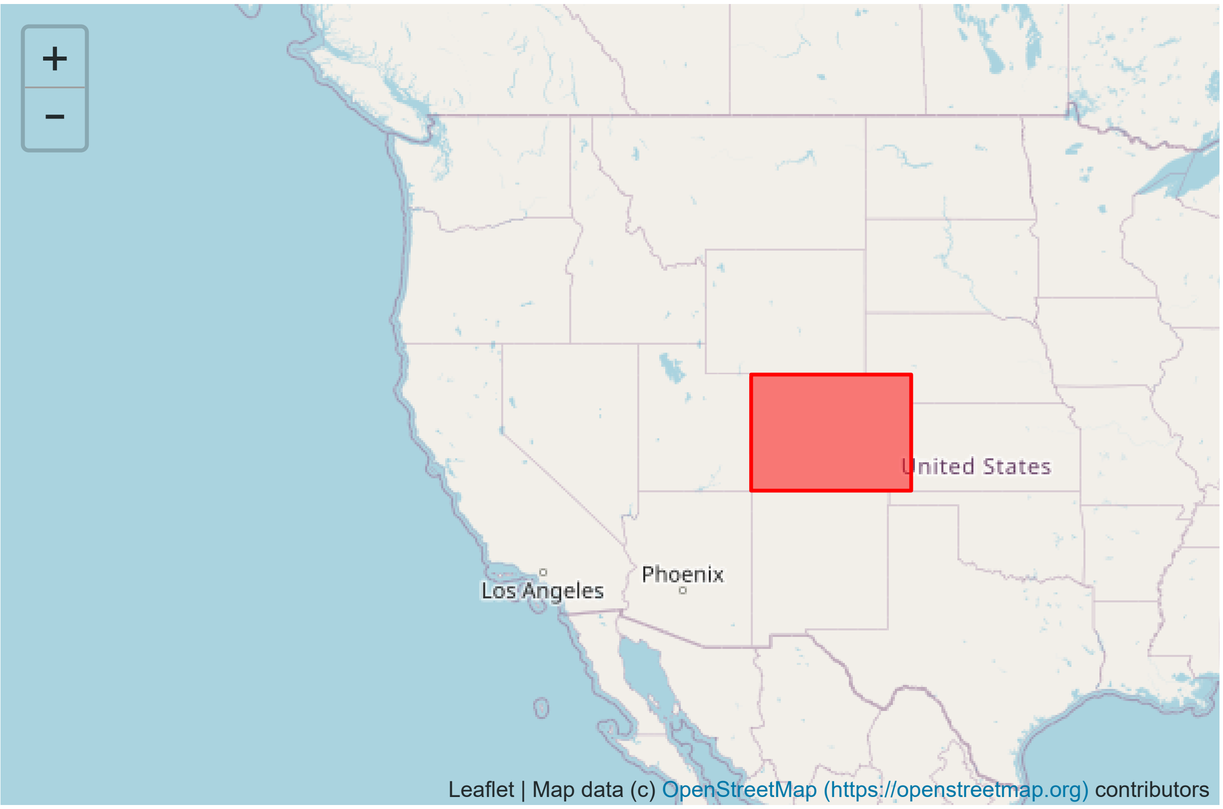
```
from vega_datasets import data
airports = data.airports()

airports = airports[:35]
amap= Map(center=(40,-99),zoom=4)
for (index,rows) in airports.iterrows():
    marker = Marker(location=[rows.loc['latitude'],rows.loc['longitude']],
                    title = rows.loc['name']+ ' '+rows.loc['city']+' '+rows.loc['s
    amap.add_layer(marker)

amap
```


+
−
United States
Los Angeles
Phoenix
Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

GeoJSON Layering

In [8]:

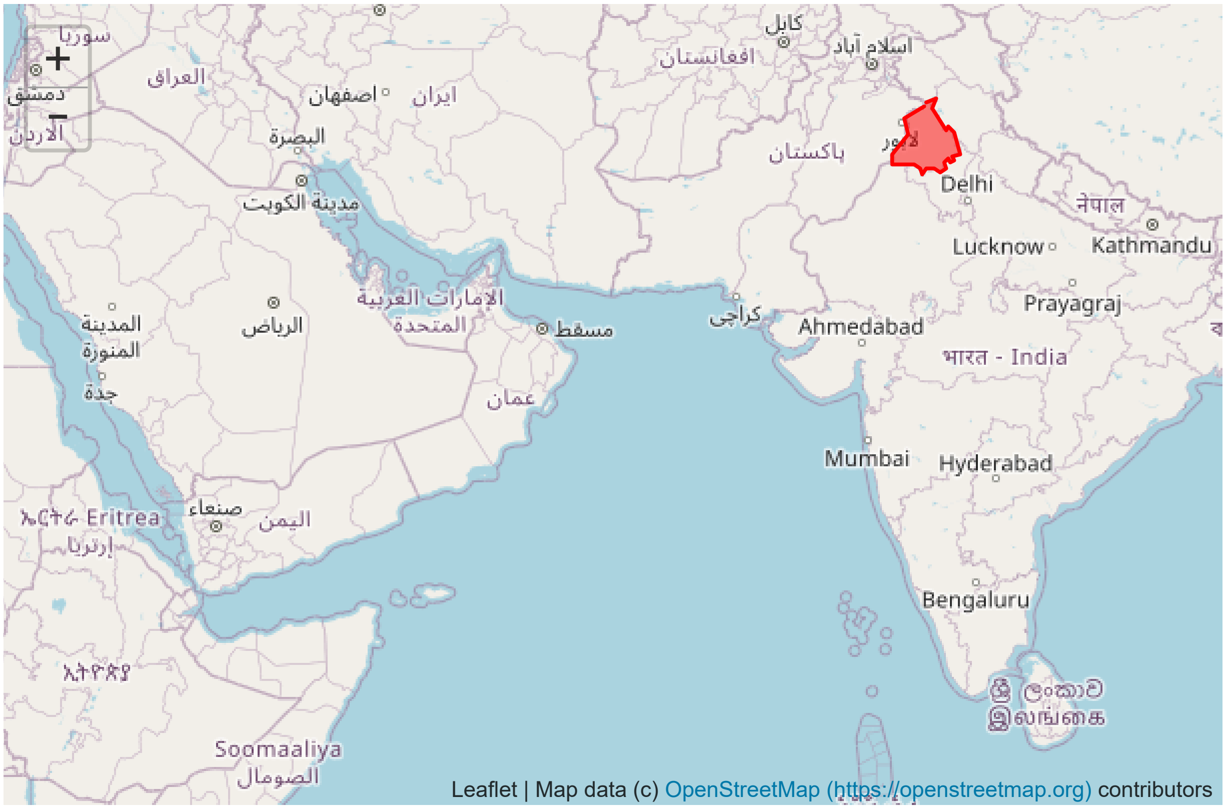
```
from ipyleaflet import GeoJSON
import json
with open('map.geojson') as file:
    geojson_colorado = json.load(file)

usmap=Map(center=(40,-99),zoom=4)
geojson_layer = GeoJSON(data=geojson_colorado,
                        style={'color':'red',
                              'opacity':1,
                              'weight':1.9,
                               'fill': 'green',
                               'fillOpacity': 0.5})

usmap.add_layer(geojson_layer)
usmap
```



Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

In [9]:

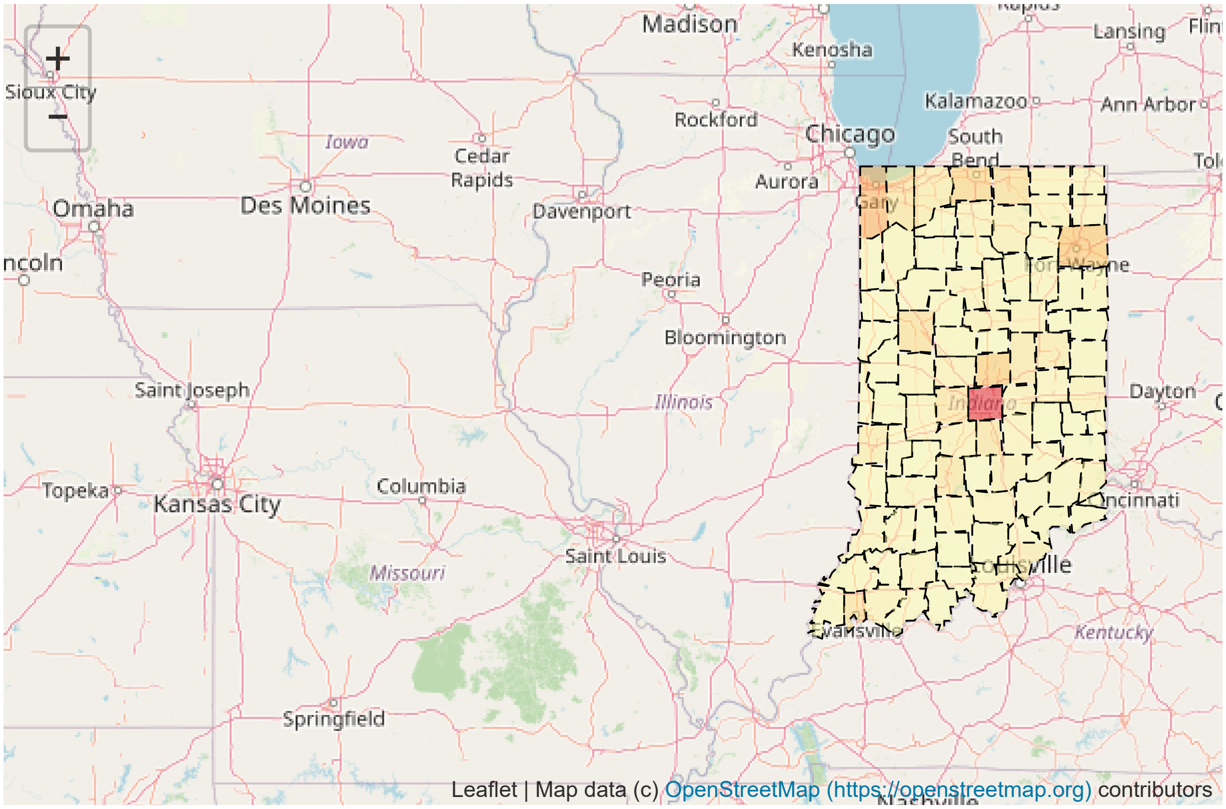
```
indimap=Map(center=(20.5,78),zoom=4)
with open('map1.geojson') as file:
    geojson_punjab = json.load(file)

geojson_layer = GeoJSON(data=geojson_punjab,
                        style={'color':'red',
                              'opacity':1,
                              'weight':1.9,
                               'fill': 'green',
                               'fillOpacity': 0.5})

indimap.add_layer(geojson_layer)
indimap
```



Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

Choropleth Map

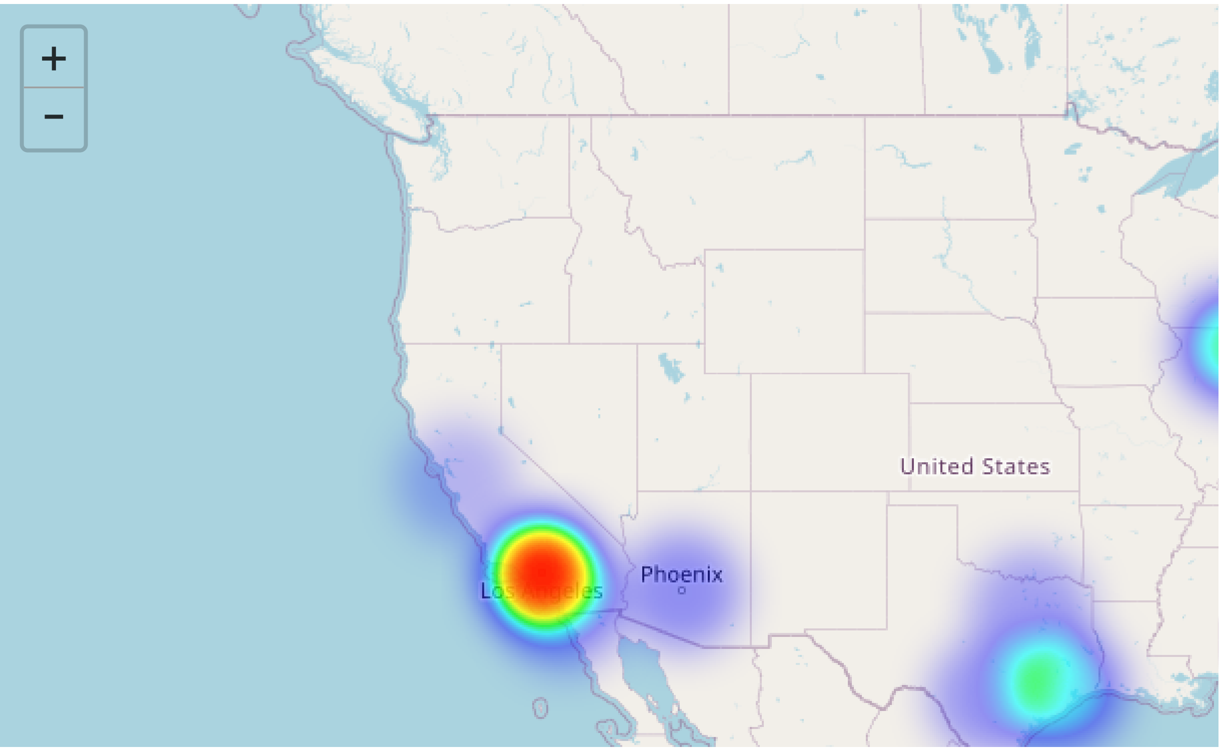
In [10]:
```
import pandas as pd
import numpy as np
from branca.colormap import linear
with open('map2.geojson') as f:
    geo_json_data = json.load(f)

pop_df = pd.read_csv('indiana_population_by_county.csv')

choro_map_data =  dict(zip(pop_df['County'].tolist(), pop_df['Population'].tolist

for i in geo_json_data['features']:
    i['id'] = i['properties']['NAME_L']

layer = ipyleaflet.Choropleth(geo_data=geo_json_data,
                              choro_data=choro_map_data,
                              colormap=linear.YlOrRd_04,
                              border_color='black',
                              style={'fillOpacity': 0.5, 'dashArray': '5, 5'})

imap = Map(center=(39.78, -86.15), zoom=6)
imap.add_layer(layer)
imap
```

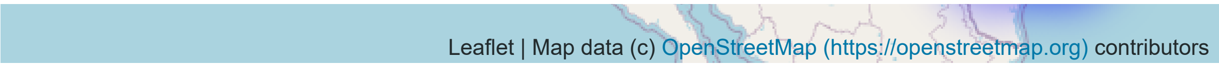


HeatMap

In [11]:

```
#getting cities of US
new_york_city = geocoder.osm('New York City, New York')
los_angeles = geocoder.osm('Los Angeles, California')
chicago = geocoder.osm('Chicago, Illinois')
houston = geocoder.osm('Houston, Texas')
phoenix = geocoder.osm('Phoenix, Arizona')
philadelphia = geocoder.osm('Philadelphia, Pennsylvania')
san_antonio = geocoder.osm('San Antonio, Texas')
san_diego = geocoder.osm('San Diego, California')
dallas = geocoder.osm('Dallas, Texas')
san_jose = geocoder.osm('San Jose, California')

#extracting latitude,longitude and population of the cities(population is the int
new_york_city_latlng = [new_york_city.lat, new_york_city.lng, 8398748/1000]
los_angeles_latlng = [los_angeles.lat, los_angeles.lng, 3990456/1000]
chicago_latlng = [chicago.lat, chicago.lng, 2705994/1000]
houston_latlng = [houston.lat, houston.lng, 2325502/1000]
phoenix_latlng = [phoenix.lat, phoenix.lng, 1660272/1000]
philadelphia_latlng = [philadelphia.lat, philadelphia.lng, 1584138/1000]
san_antonio_latlng = [san_antonio.lat, san_antonio.lng, 1532233/1000]
san_diego_latlng = [san_diego.lat, san_diego.lng, 1425976/1000]
dallas_latlng = [dallas.lat, dallas.lng, 1345047/1000]
san_jose_latlng = [san_jose.lat, san_jose.lng, 1030119/1000]

large_cities = [new_york_city_latlng, los_angeles_latlng, chicago_latlng, houstor
                philadelphia_latlng, san_antonio_latlng, san_diego_latlng, dallas

usmap = Map(center=(40,-99),zoom=4)

heat = ipyleaflet.Heatmap(locations=large_cities,radius=30,blur=20)

usmap.add_layer(heat)

usmap
```



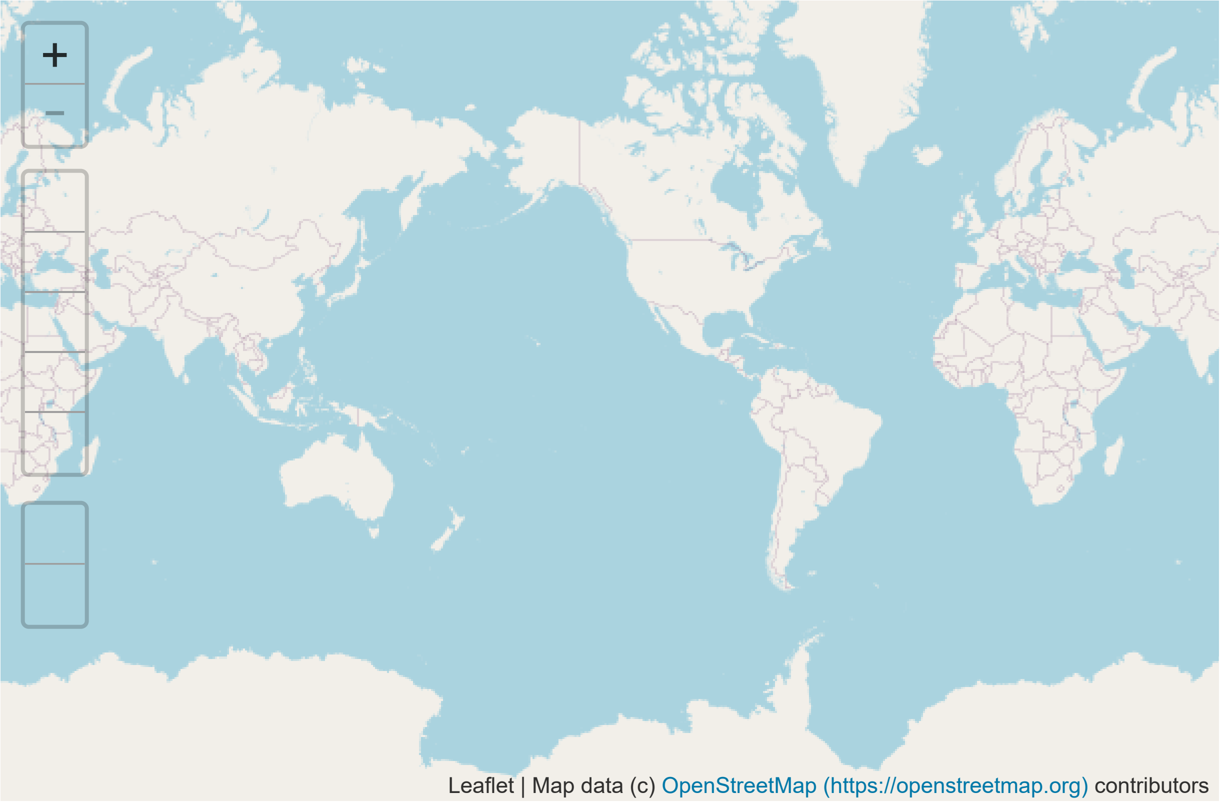


Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

DrawControl

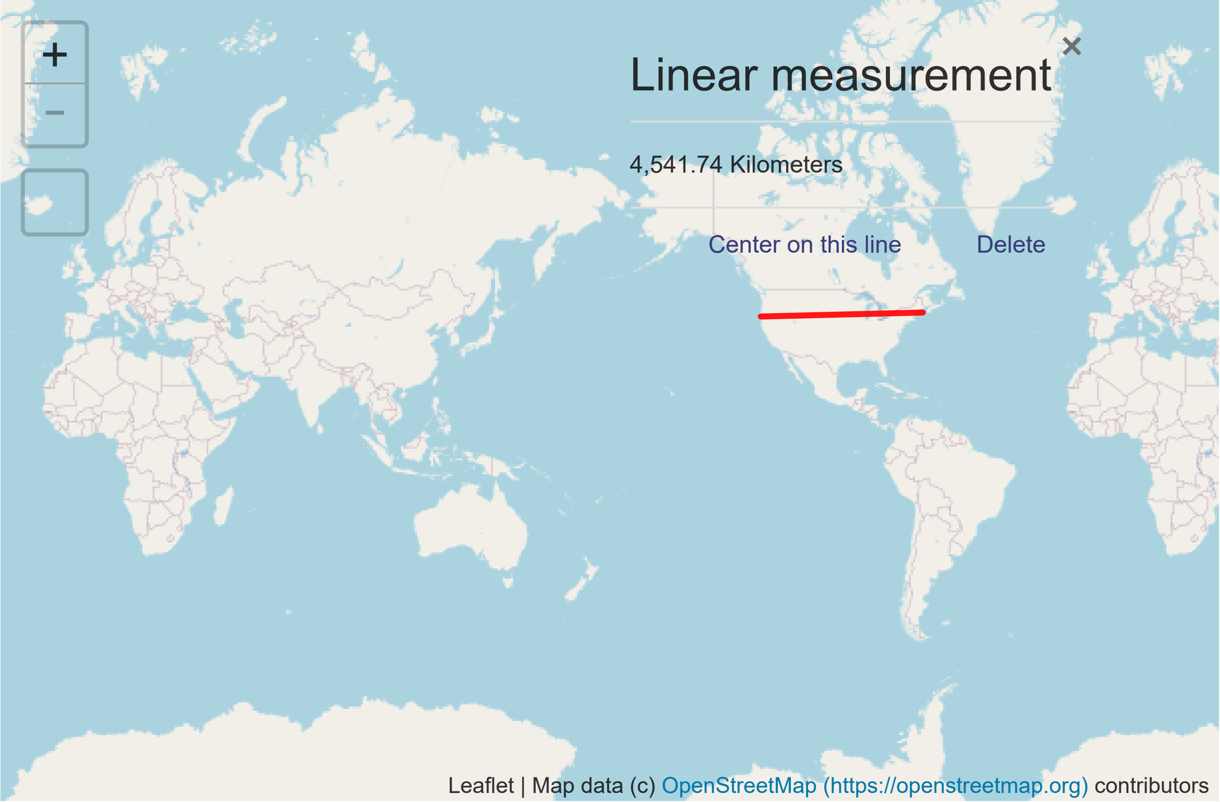
In [12]:

```
from ipyleaflet import DrawControl

# create map
draw_control_map = ipyleaflet.Map(zoom=1)

# create control
draw_control = DrawControl()

# add control to map
draw_control_map.add_control(draw_control)

# add extra options to control
draw_control.circle = {
    "shapeOptions": {
        "fillColor": "blue",
        "color": "blue",
        "fillOpacity": 0.5
    }
}

draw_control.rectangle = {
    "shapeOptions": {
        "fillColor": "blue",
        "color": "blue",
        "fillOpacity": 0.5
    }
}

# display map
draw_control_map
```


+
−
Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

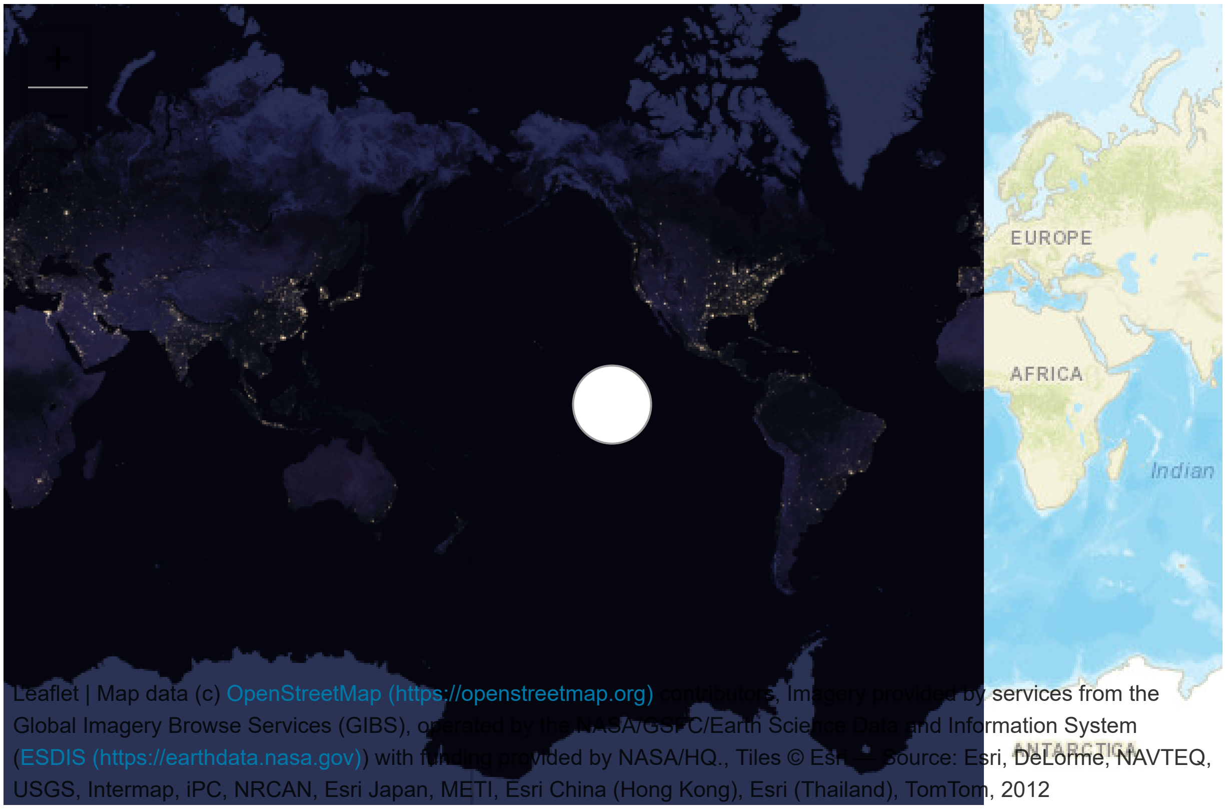
MeasureControl

In [13]:
```
from ipyleaflet import MeasureControl
map2 = Map(zoom=1)
measure = MeasureControl(position='topleft',active_color='blue',primary_length_ur
map2.add_control(measure)
measure.completed_color='red'
map2
```


+
−
×
Linear measurement
4,541.74 Kilometers
Center on this line
Delete
Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

SplitMapControl

```
In [14]: from ipyleaflet import basemap_to_tiles, SplitMapControl

         split_map = ipyleaflet.Map(zoom=1)

         # create right and left layers
         right_layer = basemap_to_tiles(basemaps.Esri.WorldStreetMap)
         left_layer = basemap_to_tiles(basemaps.NASAGIBS.ViirsEarthAtNight2012)

         # create control
         control = SplitMapControl(left_layer=left_layer, right_layer=right_layer)

         # add control to map
         split_map.add_control(control)

         # display map
         split_map
```

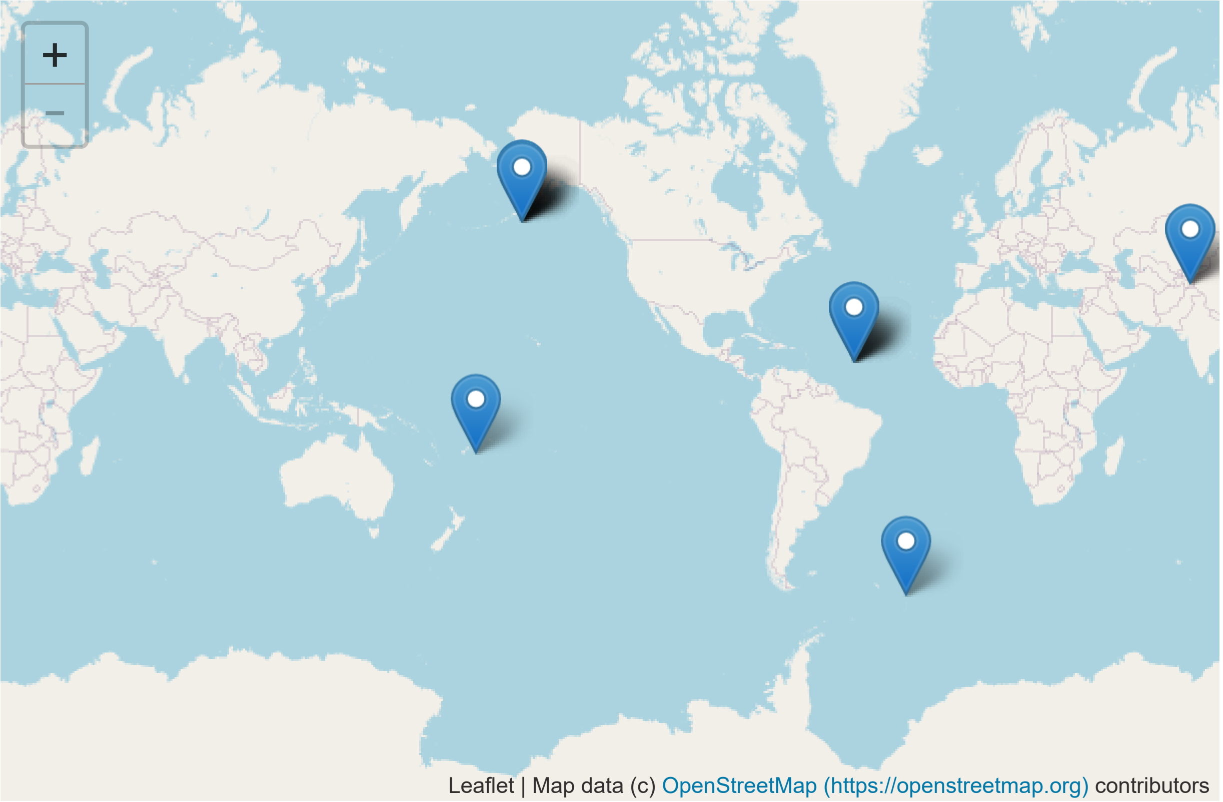


Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors, Imagery provided by services from the Global Imagery Browse Services (GIBS), operated by the NASA/GSFC/Earth Science Data and Information System (ESDIS (https://earthdata.nasa.gov)) with funding provided by NASA/HQ., Tiles © Esri — Source: Esri, DeLorme, NAVTEQ, USGS, Intermap, iPC, NRCAN, Esri Japan, METI, Esri China (Hong Kong), Esri (Thailand), TomTom, 2012

Interactive Slider

In [16]:

```
earthquakes = pd.read_csv('database.csv')
earthquakes = earthquakes[['Date', 'Latitude', 'Longitude', 'Magnitude']]
earthquakes = earthquakes.head(50)

selection_slider = ipywidgets.SelectionSlider(options=list(earthquakes['Date']),
                                              value='01/02/1965',
                                              description='Slider')
wmap=Map(zoom=1)
display(wmap)

def plot_earthquakes(date):
    e = earthquakes.loc[earthquakes['Date'] == date]
    for (index, row) in e.iterrows():
        marker = Marker(location=[row.loc['Latitude'], row.loc['Longitude']])
        wmap.add_layer(marker)
    print(earthquakes.loc[earthquakes['Date'] == date])

ipywidgets.interact(plot_earthquakes, date=selection_slider)
```


+
−
Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

Slider ○ 02/04/1965

```
          Date  Latitude  Longitude  Magnitude
15  02/04/1965   -51.840    139.741        6.1
16  02/04/1965    51.251    178.715        8.7
17  02/04/1965    51.639    175.055        6.0
18  02/04/1965    52.528    172.007        5.7
19  02/04/1965    51.626    175.746        5.8
20  02/04/1965    51.037    177.848        5.9
21  02/04/1965    51.730    173.975        5.9
22  02/04/1965    51.775    173.058        5.7
23  02/04/1965    52.611    172.588        5.7
```

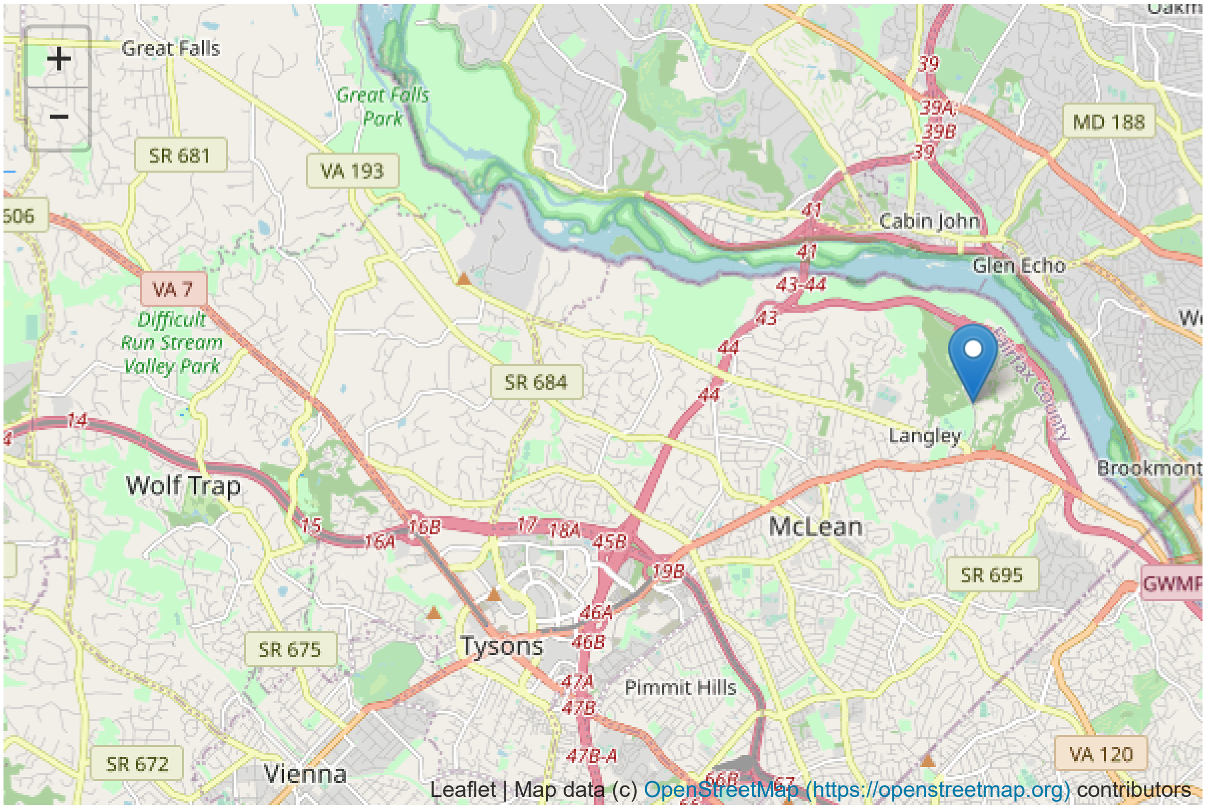
```
24  02/04/1965    51.831    174.368        5.7
25  02/04/1965    51.948    173.969        5.6
26  02/04/1965    51.443    179.605        7.3
27  02/04/1965    52.773    171.974        6.5
28  02/04/1965    51.772    174.696        5.6
29  02/04/1965    52.975    171.091        6.4
30  02/04/1965    52.990    170.874        5.8
31  02/04/1965    51.536    175.045        5.8
32  02/04/1965    13.245    -44.922        5.8
33  02/04/1965    51.812    174.206        5.7
```

```
Out[16]: <function __main__.plot_earthquakes(date)>
```

Ploting Location with Text Widget

In [17]:
```
address_box = ipywidgets.Text(value='',
                             description='Address',
                             placeholder='Type here')

def mark(address):
    location = geocoder.osm(address)
    map1 = Map(center=(location.lat,location.lng))
    latlng=[location.lat,location.lng]
    marker = Marker(location=latlng,title=str(address))
    map1.add_layer(marker)
    display(map1)

ipywidgets.interact(mark,address=address_box)
```

Address 1000 Colonial Farm Rd, McLean,



Leaflet | Map data (c) OpenStreetMap (https://openstreetmap.org) contributors

Out[17]:
```
<function __main__.mark(address)>
```

In [ ]: