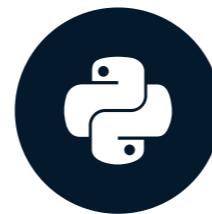


GeoSeries attributes and methods I

VISUALIZING GEOSPATIAL DATA IN PYTHON



Mary van Valkenburg

Data Science Program Manager,
Nashville Software School

Shapely attributes and methods

```
# the geometry column is a GeoSeries  
type(school_districts.geometry)
```

geopandas.geoseries.GeoSeries

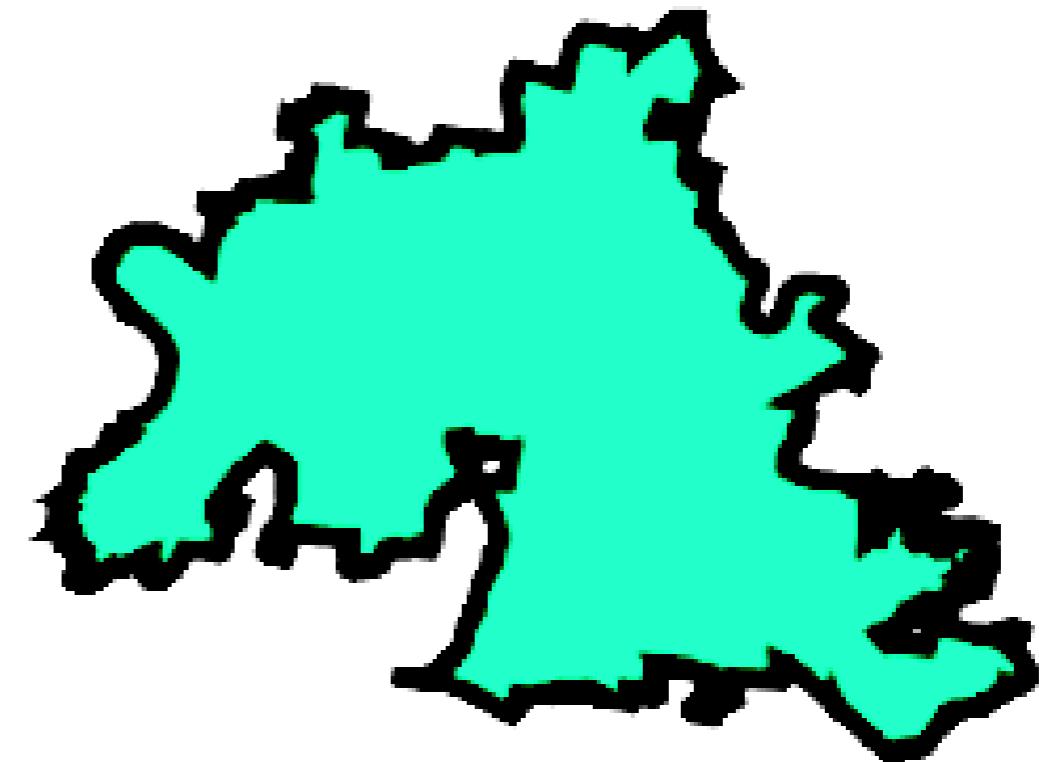
- `GeoSeries.area` - returns the area of each geometry in a GeoSeries
- `GeoSeries.centroid` - returns the center point of each geometry in a GeoSeries
- `GeoSeries.distance(other)` - returns the minimum distance to `other`

GeoSeries.area

- returns the area of each geometry in a GeoSeries

```
# area of first polygon in districts  
print(districts.geometry[0].area)
```

```
325.78
```



School district areas

GEOSERIES.AREA

```
# print first 4 rows of school districts and the total number of rows
print(school_districts.head(4))
print('There are ', school_districts.shape[0], ' school districts.' )
```

first_name	last_name	position	district	geometry
Sharon	Gentry	Member	1	(POLYGON ((-86.771 36.383...)))
Jill	Speering	Vice-Chair	3	(POLYGON ((-86.753 36.404...)))
Jo Ann	Brannon	Member	2	(POLYGON ((-86.766 36.083...)))
Anna	Shepherd	Chair	4	(POLYGON ((-86.580 36.209...)))

There are 9 school districts.

```
# calculate area of each school district
district_area = school_districts.geometry.area
# print the areas and crs used
print(district_area.sort_values(ascending = False))
print(school_districts.crs)
```

```
0    0.036641
4    0.023030
8    0.015004
1    0.014205
3    0.014123
5    0.010704
2    0.008328
7    0.007813
6    0.006415
dtype: float64
{'init': 'epsg:4326'}
```

```
# create a copy of school_districts that uses EPSG:3857
school_districts_3857 = school_districts.to_crs(epsg = 3857)
# define a variable for m^2 to km^2 and get area in kilometers squared
sqm_to_sqkm = 10**6
district_area_km = school_districts_3857.geometry.area / sqkm_to_sqm
print(district_area_km.sort_values(ascending = False))
print(school_districts_3857.crs)
```

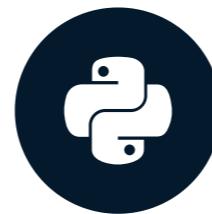
```
0    563.134380
4    353.232132
8    230.135653
1    218.369949
3    216.871511
5    164.137548
2    127.615396
7    119.742279
6     98.469632
dtype: float64
{'init': 'epsg:3857'}
```

Let's Practice!

VISUALIZING GEOSPATIAL DATA IN PYTHON

GeoSeries attributes and methods II

VISUALIZING GEOSPATIAL DATA IN PYTHON



Mary van Valkenburg

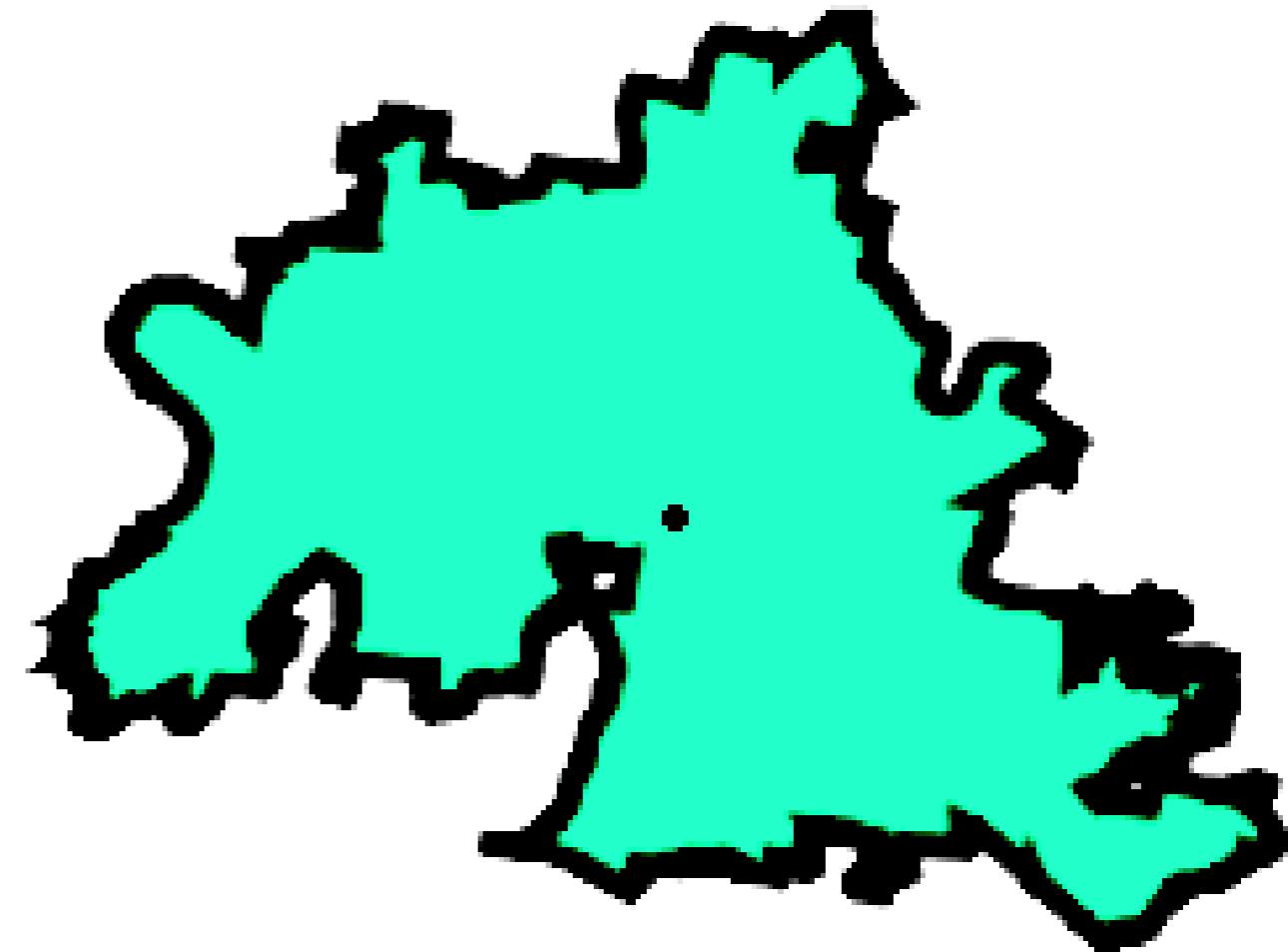
Data Science Program Manager,
Nashville Software School

GeoSeries.centroid

- returns the point at the center of each geometry in a GeoSeries

```
# centroid of first polygon  
print(districts.geometry.centroid[0])
```

```
Point(-87.256 36.193)
```



School district centroids

GEOSERIES.CENTROID

```
# print the first 5 rows of school districts
print(school_districts.head())
```

```
first_name  last_name  district  geometry
Sharon      Gentry      1          (POLYGON ((-86.771 36.383...
Jill        Speering    3          (POLYGON ((-86.753 36.404...
Jo Ann     Brannon     2          (POLYGON ((-86.766 36.083...
Anna        Shepherd    4          (POLYGON ((-86.580 36.209...
Amy         Frogge      9          (POLYGON ((-86.972 36.208...
```

School district centroids

```
# create 'center` column from the centroid
school_districts['center'] = school_districts.geometry.centroid
# create GeoDataFrame with districts and centers
part = ['district', 'center']
school_district_centers = school_districts[part]
school_district_centers.head(3)
```

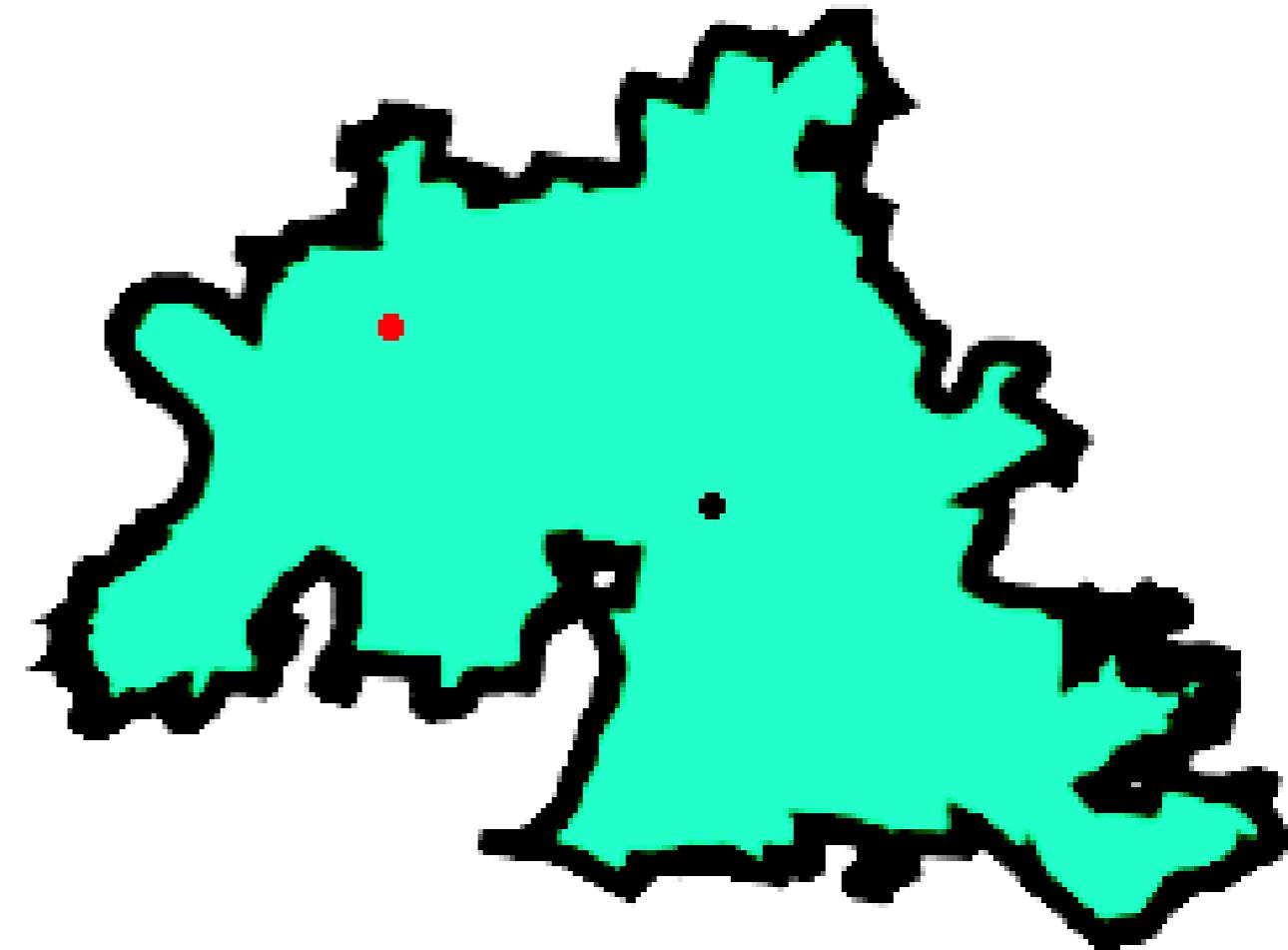
```
district      center
1            POINT (-86.8608659599405 36.2628221811899)
3            POINT (-86.72361421487962 36.28515517790142)
2            POINT (-86.70156420691957 36.03021153030475)
```

GeoSeries.distance()

- `GeoSeries.distance(other)` - returns minimum distance to `other`

```
# distance from red_pt to centroid  
cen = districts.geometry.centroid[0]  
print(red_pt.distance(other = cen))
```

```
24.273
```



Distance between two points

GEOSERIES.DISTANCE(OTHER)

```
district_one = school_districts.loc[school_districts.district == '1']
district_one.head()
```

```
first_name    last_name    district    center           geometry
Sharon        Gentry        1          POINT (-86.860 36.262)  (POLYGON ((-86.771...
```

Distance between two points

```
schools.head(3)

name          lat      lng
AZ Kelley Elem  36.021 -86.658
Alex Green Elem 36.252 -86.832
Amqui Elem     36.27   -86.703

# create geometry in schools
schools['geometry']=schools.apply(lambda x: Point((x.lng, x.lat)), axis=1)

#construct schools GeoDataFrame
school_geo=gpd.GeoDataFrame(schools,crs = district_one.crs,
                             geometry = schools.geometry)
```

Distance between two points

```
# spatial join schools within dist 1
schools_in_dist1 = gpd.sjoin(schools_geo, district_one, op = 'within')
schools_in_dist1.shape
```

```
(30, 8)
```

```
# import pprint to format dictionary output
import pprint

distances = {}
for row in schools_in_dist1.iterrows():
    vals = row[1]
    key = vals['name']
    ctr = vals['center']
    distances[key] = vals['geometry'].distance(ctr)
pprint.pprint(distances)
```

```
{'Alex Green Elementary': 0.030287172719682773,
'Bellshire Elementary': 0.0988045140909651,
'Brick Church College Prep': 0.08961013862715599,
'Buena Vista Elementary': 0.10570511270825833,
'Cockrill Elementary': 0.1077685612196105....
```

Let's Practice!

VISUALIZING GEOSPATIAL DATA IN PYTHON

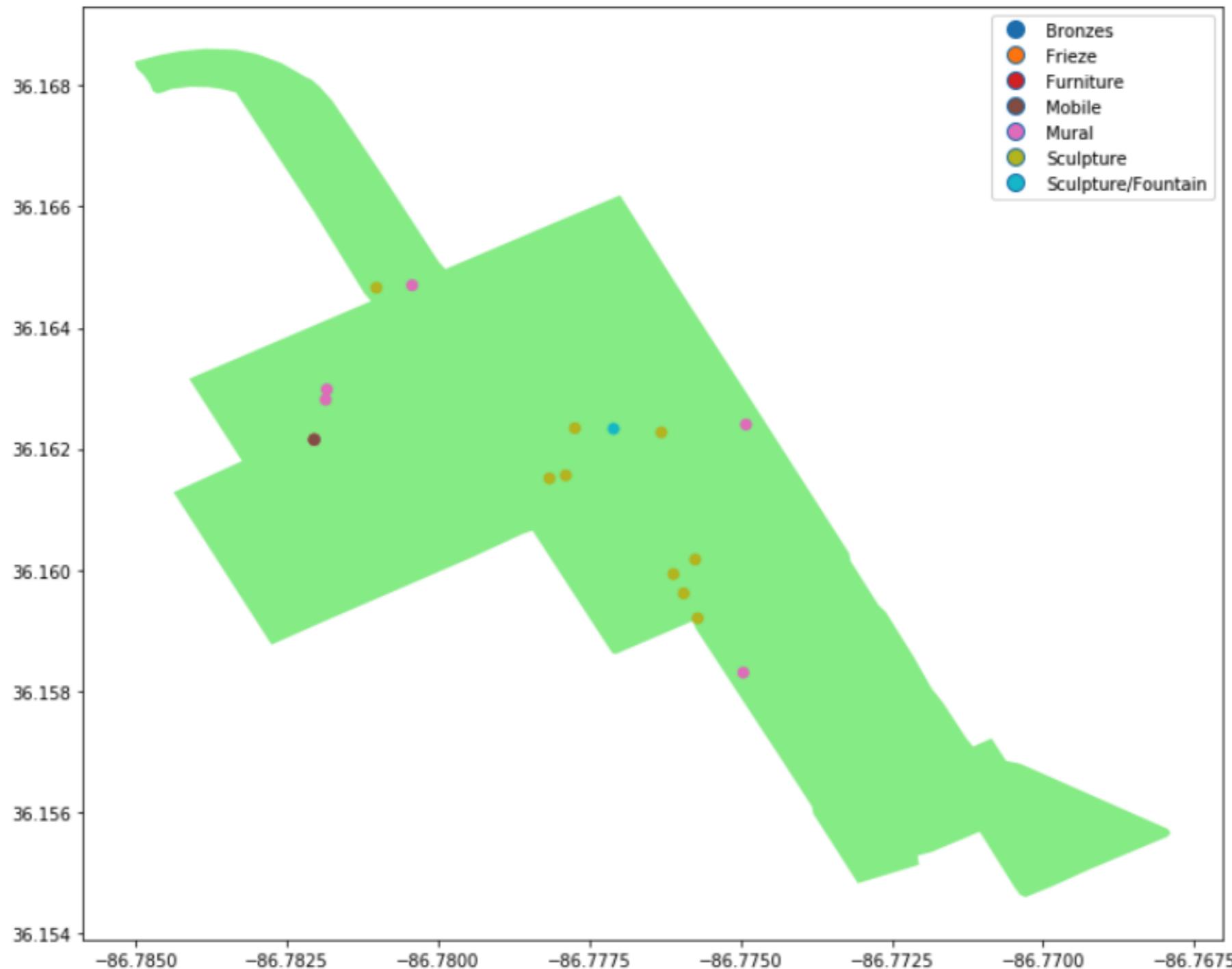
Working with Folium

VISUALIZING GEOSPATIAL DATA IN PYTHON



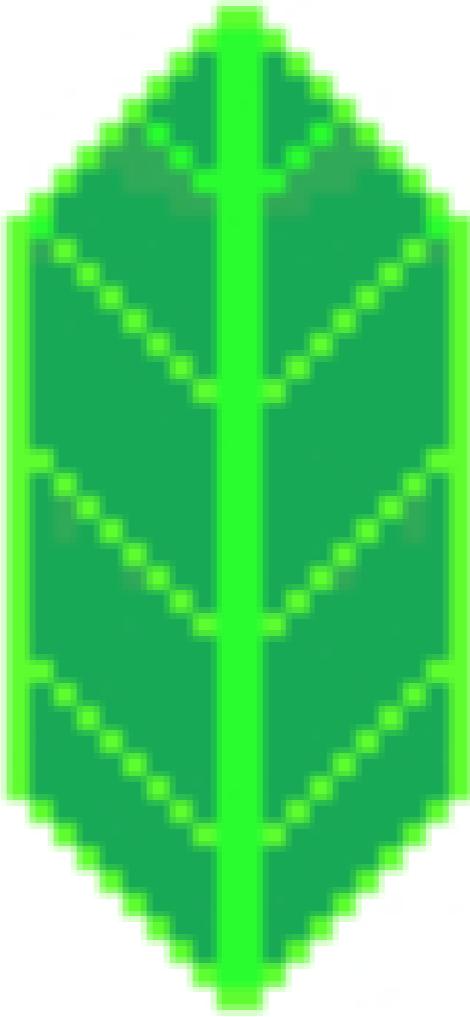
Mary van Valkenburg

Data Science Program Manager,
Nashville Software School



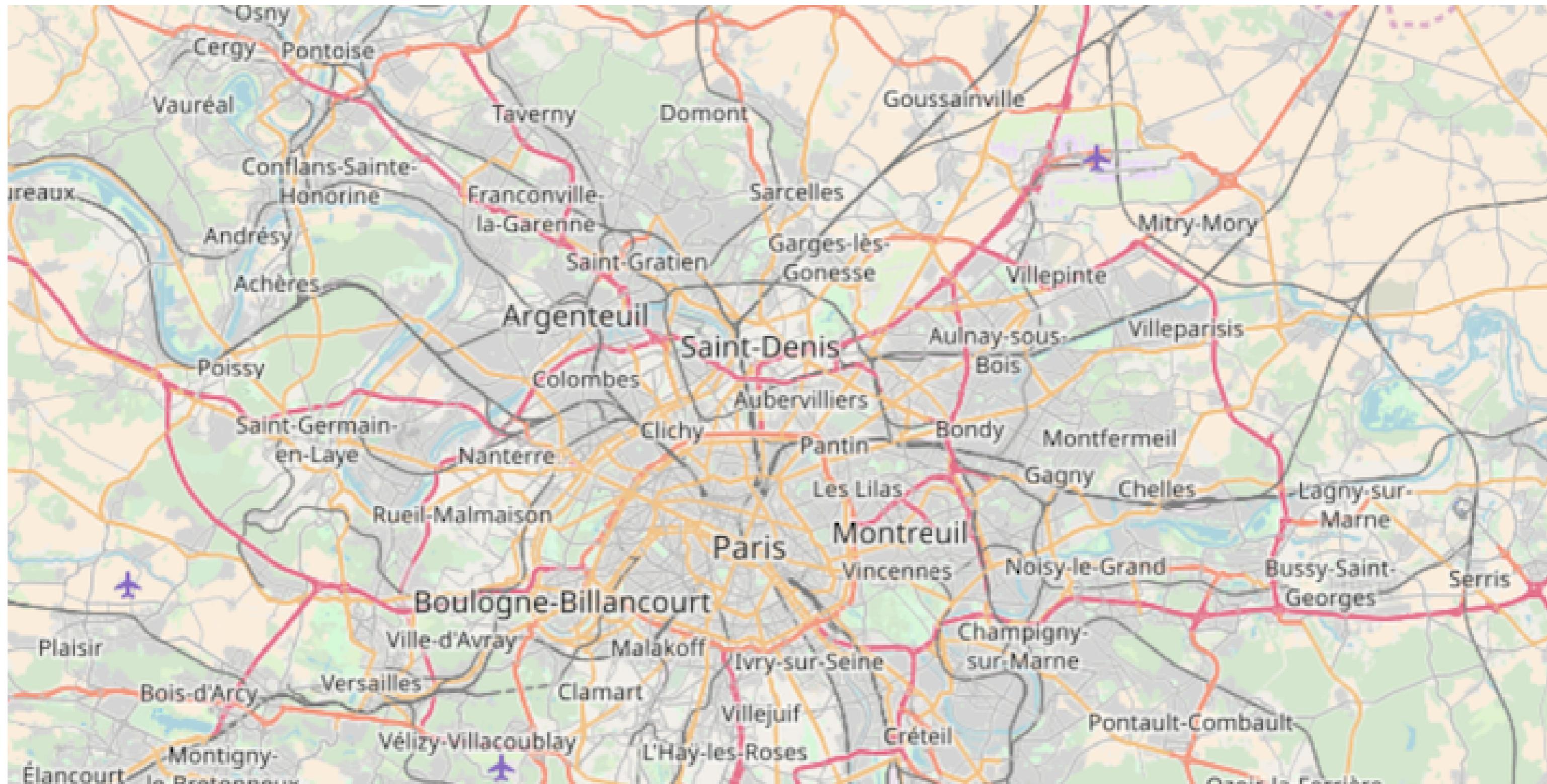
Folium

- python package
- interactive maps
- built upon Leaflet.js



folium.Map()

```
import folium  
  
# construct a map centered at the Eiffel Tower  
eiffel_tower = folium.Map(location = [48.8583736,2.2922926])  
  
# display the map  
display(eiffel_tower)
```



Setting the zoom level

```
import folium  
  
# construct a map centered at the Eiffel Tower  
eiffel_tower = folium.Map([location = 48.8583736,2.2922926], zoom_start = 12)  
  
# display the map  
display(eiffel_tower)
```



Folium location from centroid

```
district_one.head()
```

```
district    center                      geometry
1          POINT (-86.860 36.262)      (POLYGON ((-86.771 36.383...
```

```
center_point = district_one.center[0]
type(center_point)
```

```
<class 'shapely.geometry.point.Point'>
```

Folium location from centroid

```
# reverse the order for folium location array  
district_center = [center_point.y, center_point.x]  
  
# print center point and district_center  
print(center_point)  
print(district_center)
```

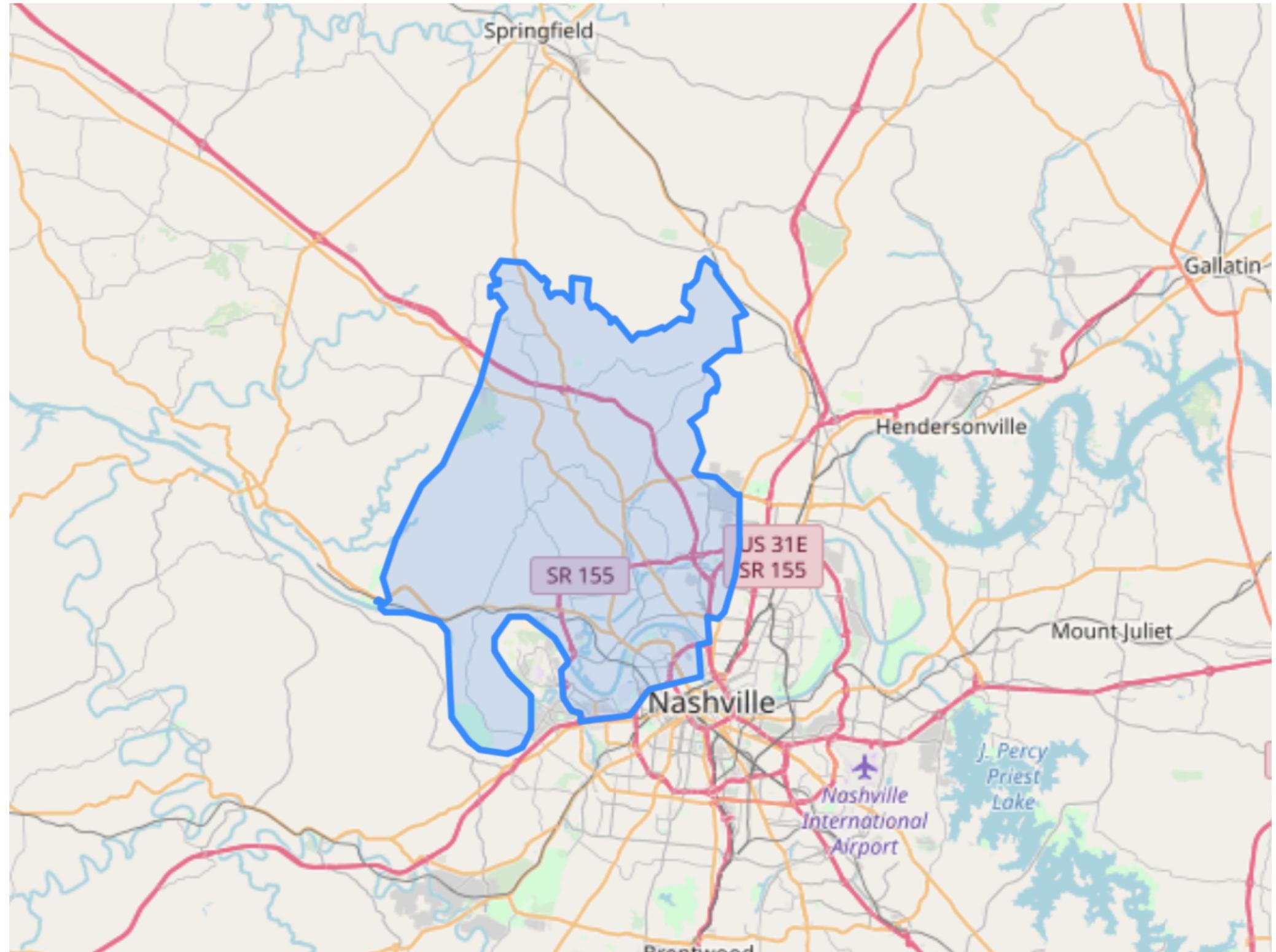
```
POINT (-86.86086595994405 36.2628221811899)  
[36.262822181189904, -86.86086595994405]
```

Adding a polygon to a folium map

```
# create a folium map centered on district 1  
district1_map = folium.Map(location = district_center)
```

```
# add the outline of district one  
folium.GeoJson(district_one.geometry).add_to(district1_map)
```

```
# display the resulting map  
display(district1_map)
```



Let's practice!

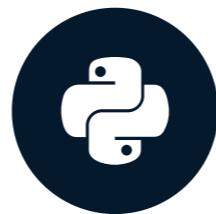
VISUALIZING GEOSPATIAL DATA IN PYTHON

Creating markers and popups in folium

VISUALIZING GEOSPATIAL DATA IN PYTHON

Mary Van Valkenburg

Data Science Program Manager,
Nashville Software School



```
for row in schools_in_dist1.iterrows():
    row_values = row[1]
    print(row_values)
```

```
name          Alex Green Elementary
lat            36.253
lng           -86.8322
geometry      POINT (-86.8322292 36.2529607)
district        1
center      POINT (-86.8608659599405 36.2628221811899)
Name: 1, dtype: object
name          Bellshire Elementary
lat            36.2697
lng           -86.7623
geometry      POINT (-86.76230026 36.26968766)
district        1
center      POINT (-86.8608659599405 36.2628221811899)
Name: 8, dtype: object
```

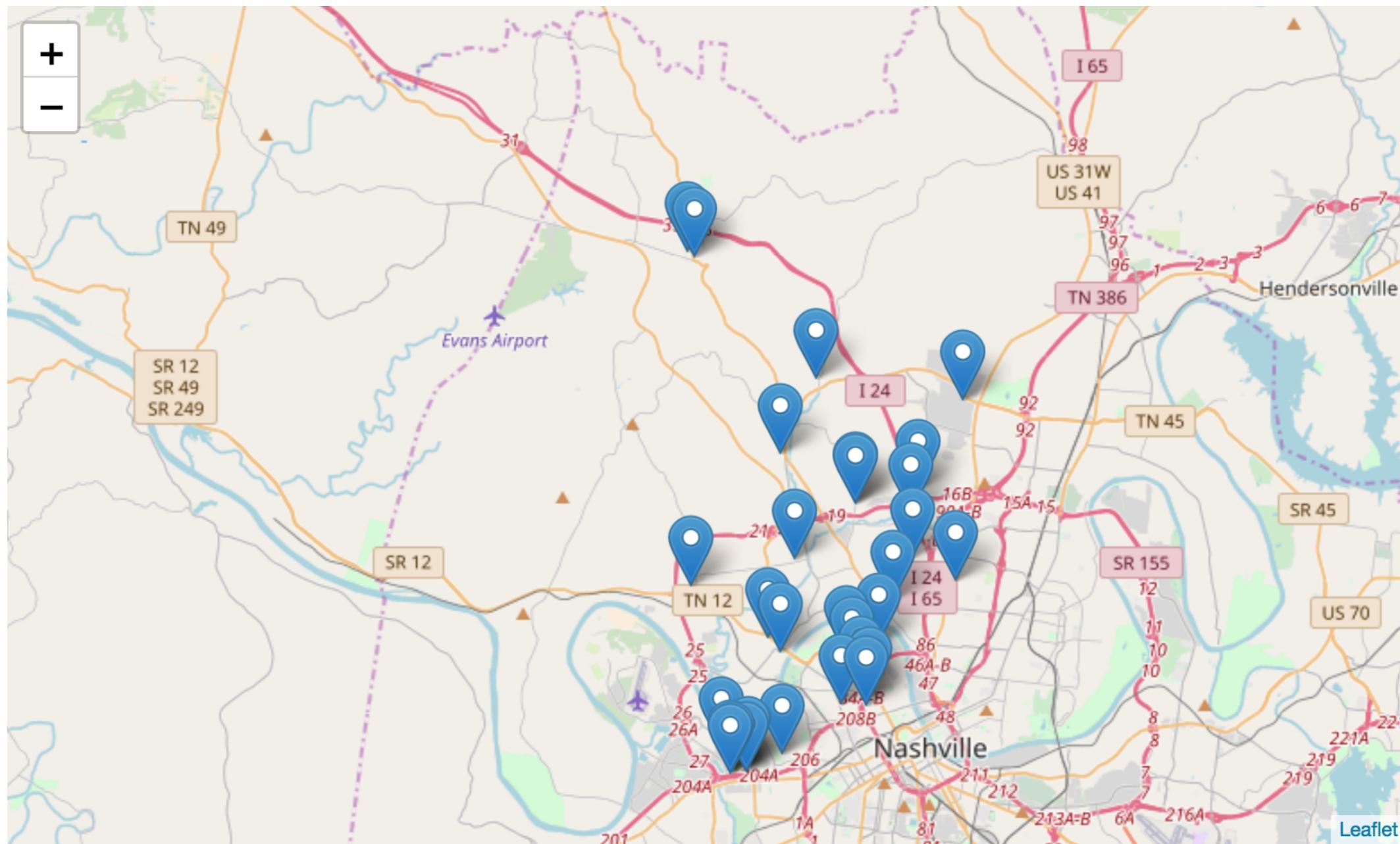
Building marker locations

```
# Construct a folium map for school district 1
district1_map = folium.Map(location = district_center, zoom_start = 11)

#Create a marker for each school
for row in schools_in_dist1.iterrows():
    row_values = row[1]
    location = [row_values['lat'], row_values['lon']]
    marker = folium.Marker(location = location)
    marker.add_to(district1_map)

display(district1_map)
```

Building marker locations



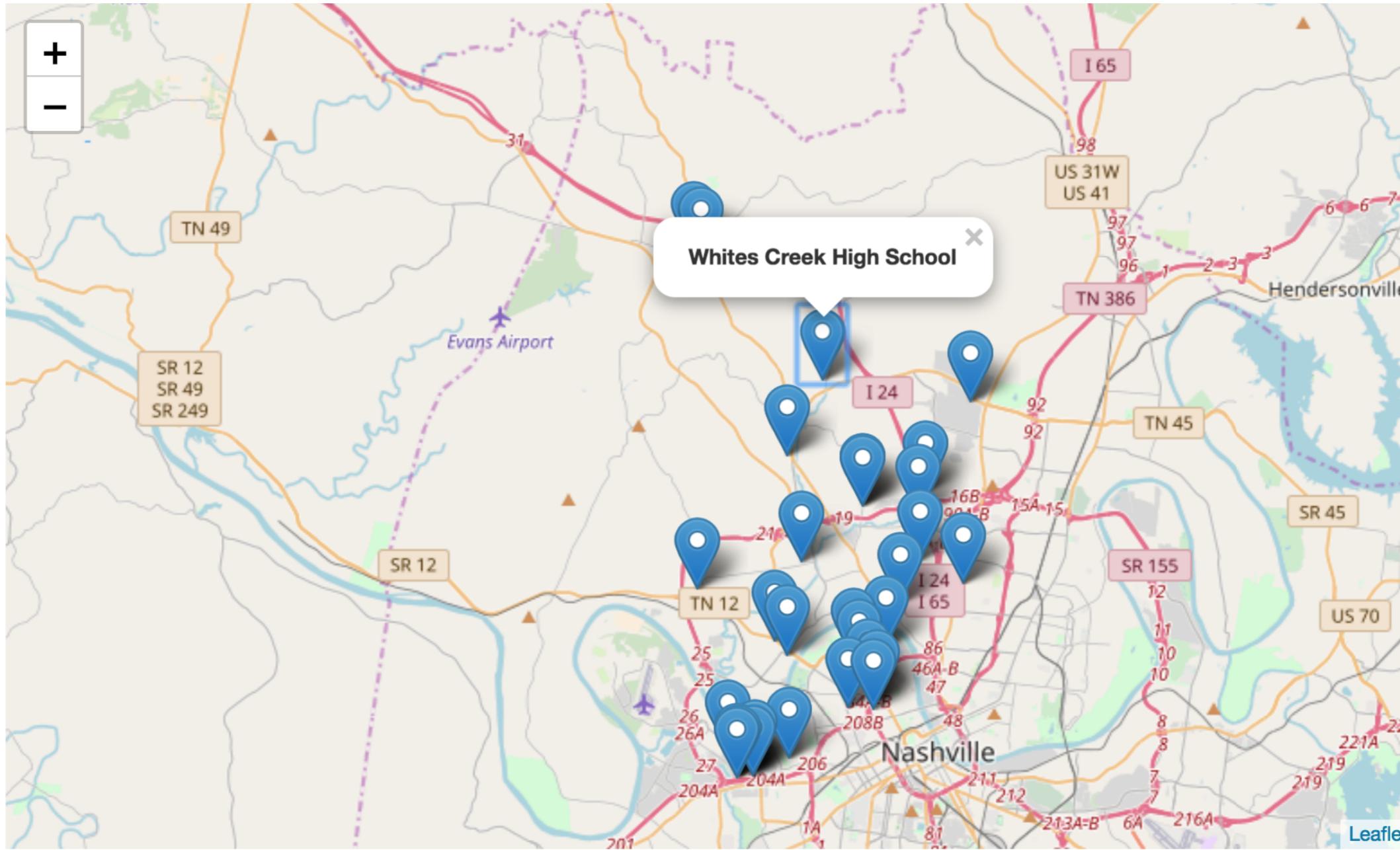
Creating popups from data in a DataFrame

```
# Construct a folium map for school district 1
district1_map = folium.Map(location = district_center, zoom_start = 11)

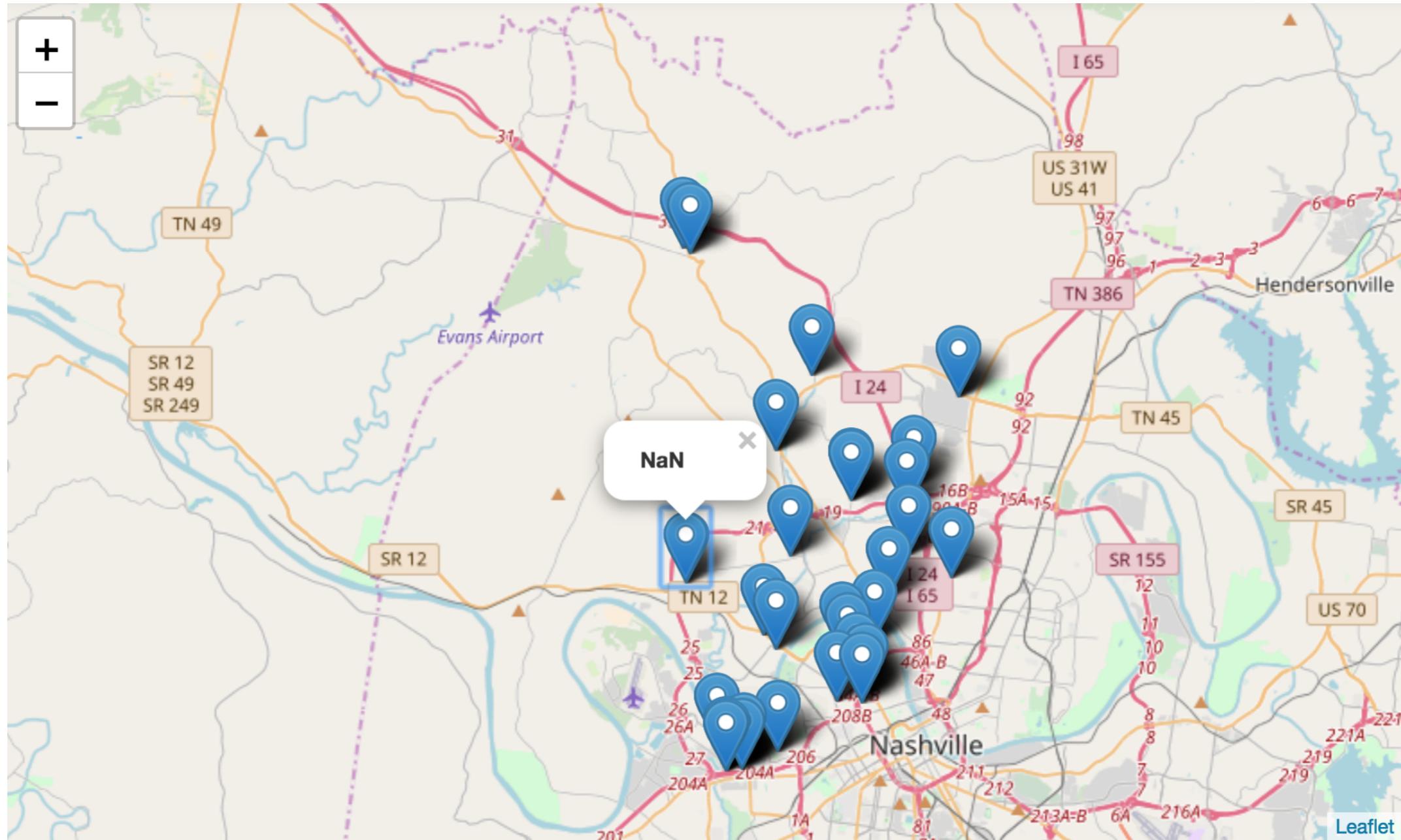
# Create a marker for each school
for row in schools_in_dist1.iterrows():
    row_values = row[1]
    location = [row_values['lat'], row_values['lng']]
    popup = popup = '<strong>' + row_values['name'] + '</strong>'
    marker = folium.Marker(location = location, popup = popup)
    marker.add_to(district1_map)

display(district1_map)
```

Creating popups from data in a DataFrame



Troubleshooting popups



Let's practice!

VISUALIZING GEOSPATIAL DATA IN PYTHON