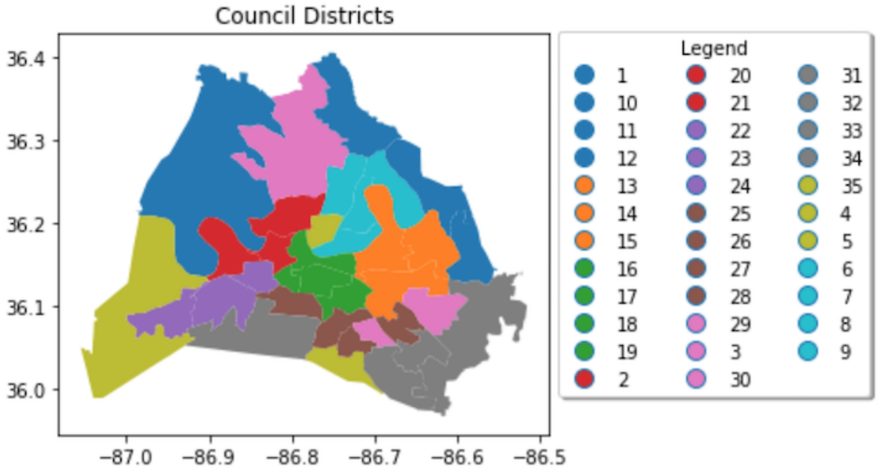
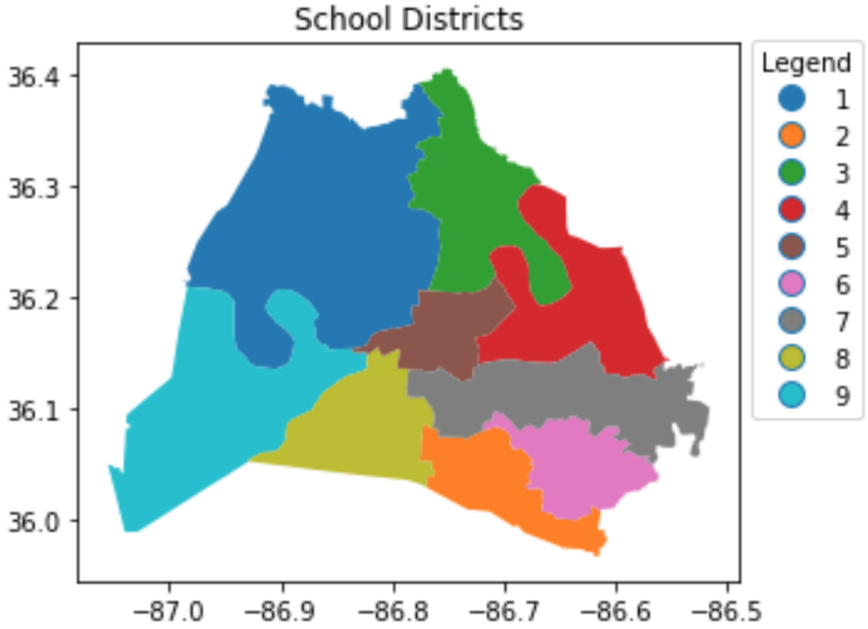
VISUALIZING GEOSPATIAL DATA IN PYTHON

Spatial joins

Mary van Valkenburg
Data Science Program Manager, Nashville Software School

Council districts and school districts





In this section, we will learn how to spatially join two data frames. In the left diagram, we have council districts, and in the right diagram we have school districts. In the next slide, we will see the use of geopandas' .sjoin() argument.

The .sjoin() op argument

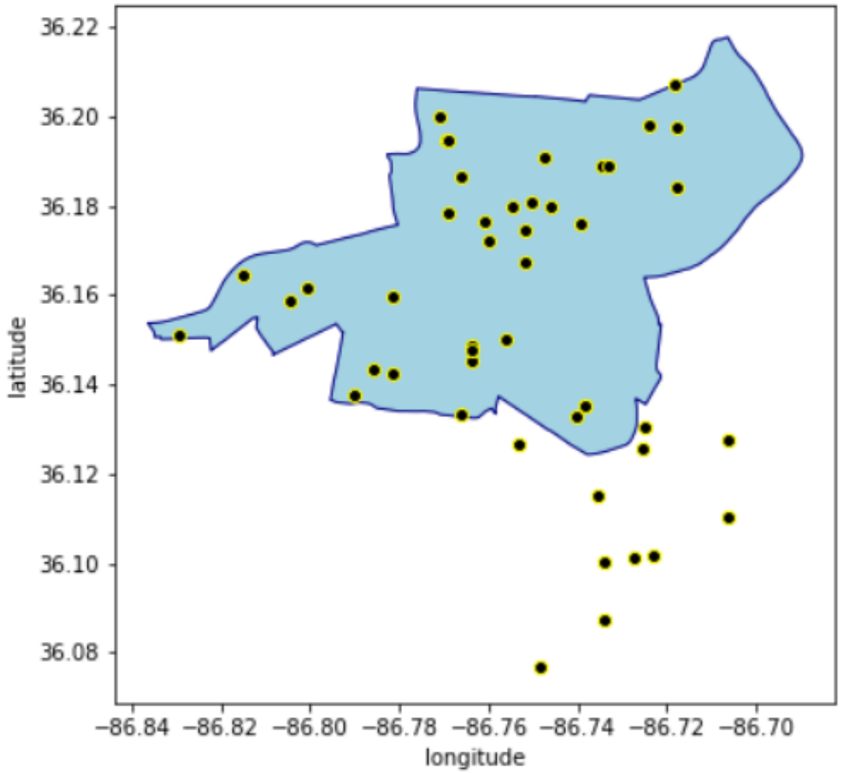
```
import geopandas as gpd

gpd.sjoin(blue_region_gdf, black_point_gdf, op = <operation>)
```

Above, the sjoin() method joins two dataframes called “blue_region_gdf” and “black_point_gdf”. The “op” argument can be any of the following operation can be ***intersects***, ***contains***, or ***within***

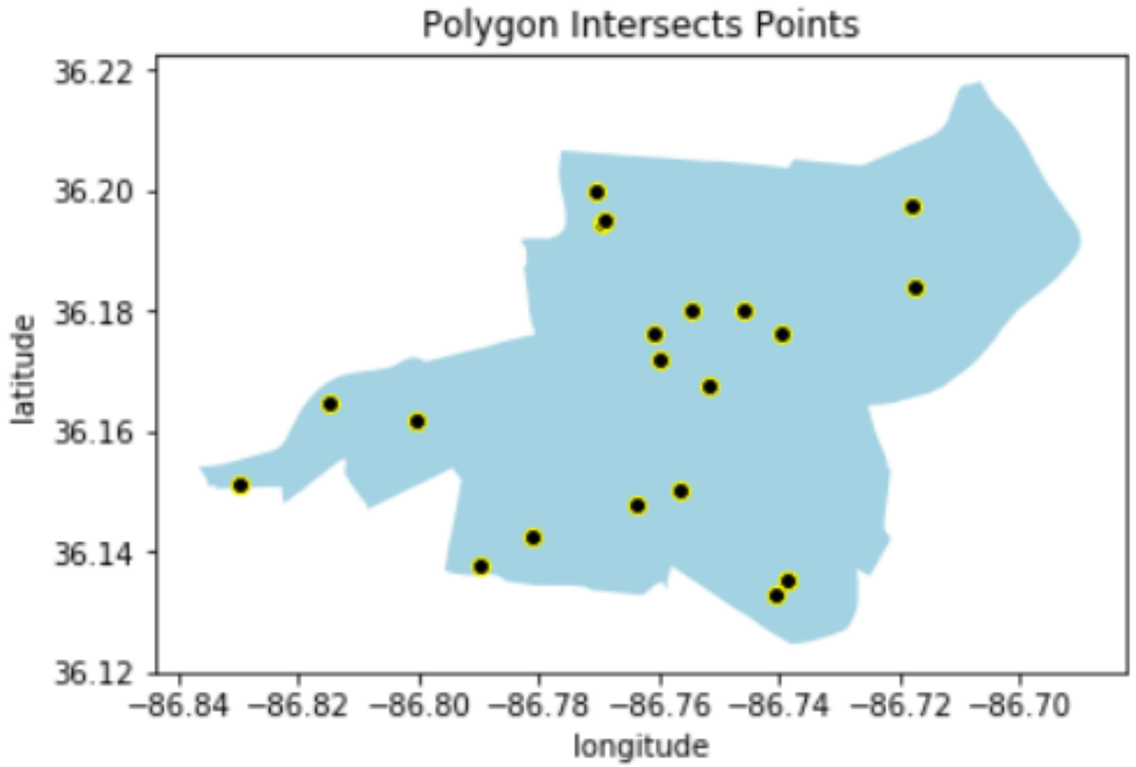
Using .sjoin()

For instance, we have the following map containing a polygon and dots. We can see what happens when we use each of the three arguments separately.



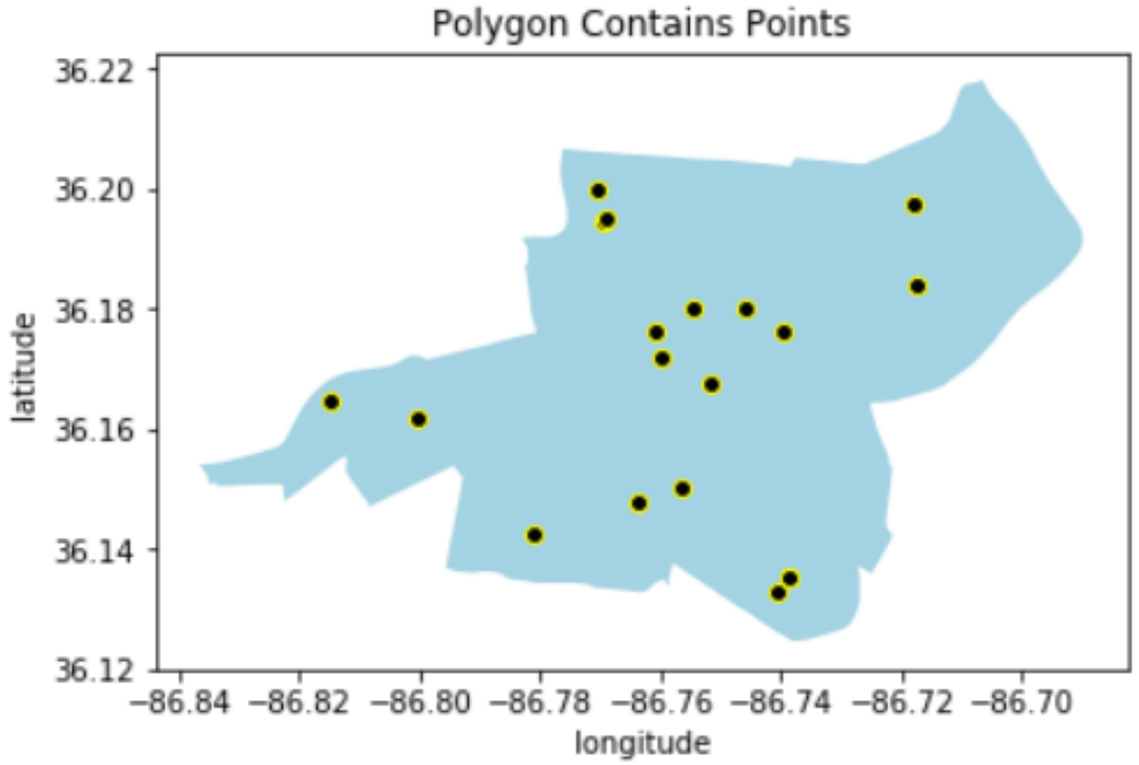
op = 'intersects'

```
gpd.sjoin(blue_region_gdf, black_point_gdf, op = 'intersects')
```



op = 'contains'

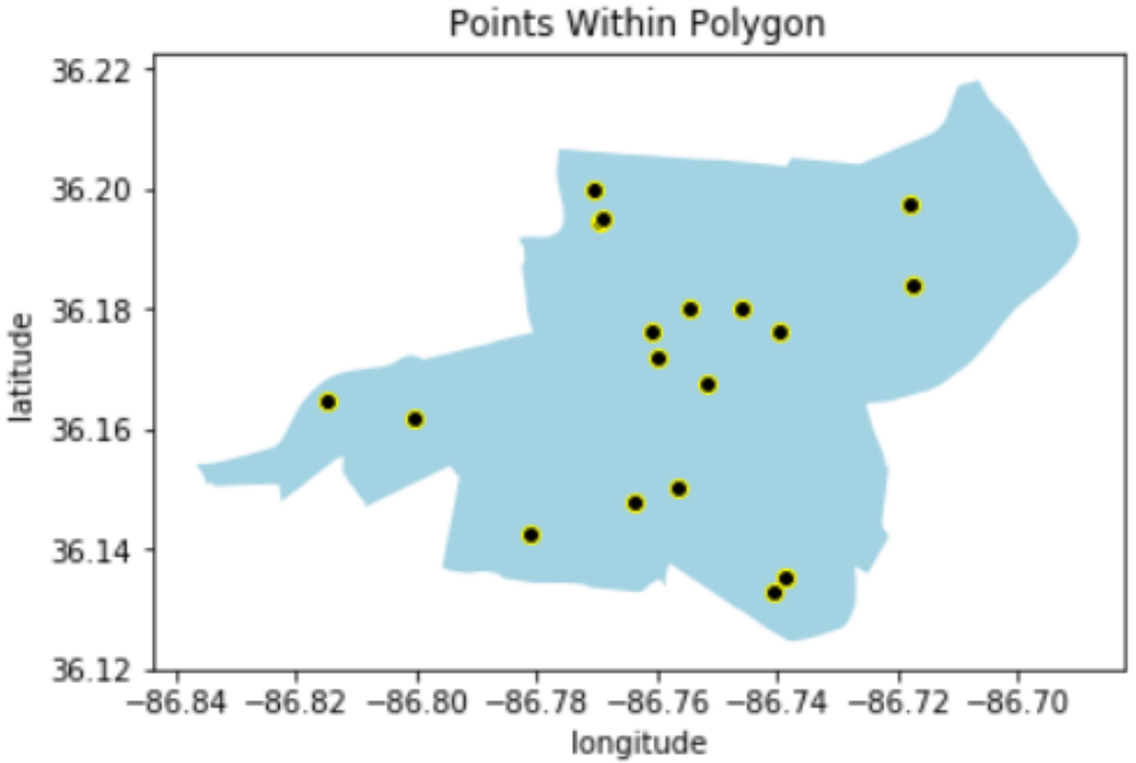
```
gpd.sjoin(blue_region_gdf, black_point_gdf, op = 'contains')
```



op = 'within'

The dataframes are switched when using “within” instead of “contains”.

```
gpd.sjoin(black_point_gdf, blue_region_gdf, op = 'within')
```



The sjoin.() op argument - within

Finding out the number of council districts within school districts.

```
# find council districts within school districts

within_gdf =gpd.sjoin(council_districts, school_districts, op='within')
print('council districts within school districts: ', within_gdf.shape[0])
```

```
council districts within school districts:  11
```

The sjoin.() op argument - contains

Finding out the number of council districts within school districts by using “contains” argument.

```
# find school districts that contain council districts

contains_gdf=pd.sjoin(school_districts, council_districts, op='contains')
print('school districts contain council districts: ', contains_gdf.shape[0])
```

```
school districts contain council districts:  11
```

The sjoin.() op argument - intersects

```
# find council districts that intersect with school districts

intersect_gdf=gpd.sjoin(council_districts, school_districts, op='intersects')
print('council districts intersect school districts: ', intersect.shape[0])
```

```
council districts intersect school districts:  100
```

Columns in a spatially joined GeoDataFrame

```
within_gdf=gpd.sjoin(council_districts, school_districts, op = 'within')
within_gdf.head()
```

```
  first_name_left  last_name_left district_left  index_right
0 Nick             Leonardo              1              0
1 DeCosta          Hastings              2              0
2 Nancy            VanReece              8              1
3 Bill             Pridemore             9              1
9 Doug             Pardue               10              1
```

Aggregating spatially joined data

```
# Aggregate council districts by school district
# to see how many council districts are within each school district.

# first rename district_left and district_right
within_gdf.district_left = council_district
within_gdf.district_right = school_district

within_gdf[['council_district', 'school_district']
          ].groupby('school_district'
                    ).agg('count'
                          ).sort_values('council_district', ascending = False)
```

```
                 council_district
school_district
              3                3
              1                2
              9                2
              2                1
              5                1
              6                1
              8                1
```