

# 01\_viz

February 4, 2019

```
In [1]: from IPython.display import Image, IFrame
```

## 1 Manipulating & Visualising spatial data

```
In [2]: %matplotlib inline
```

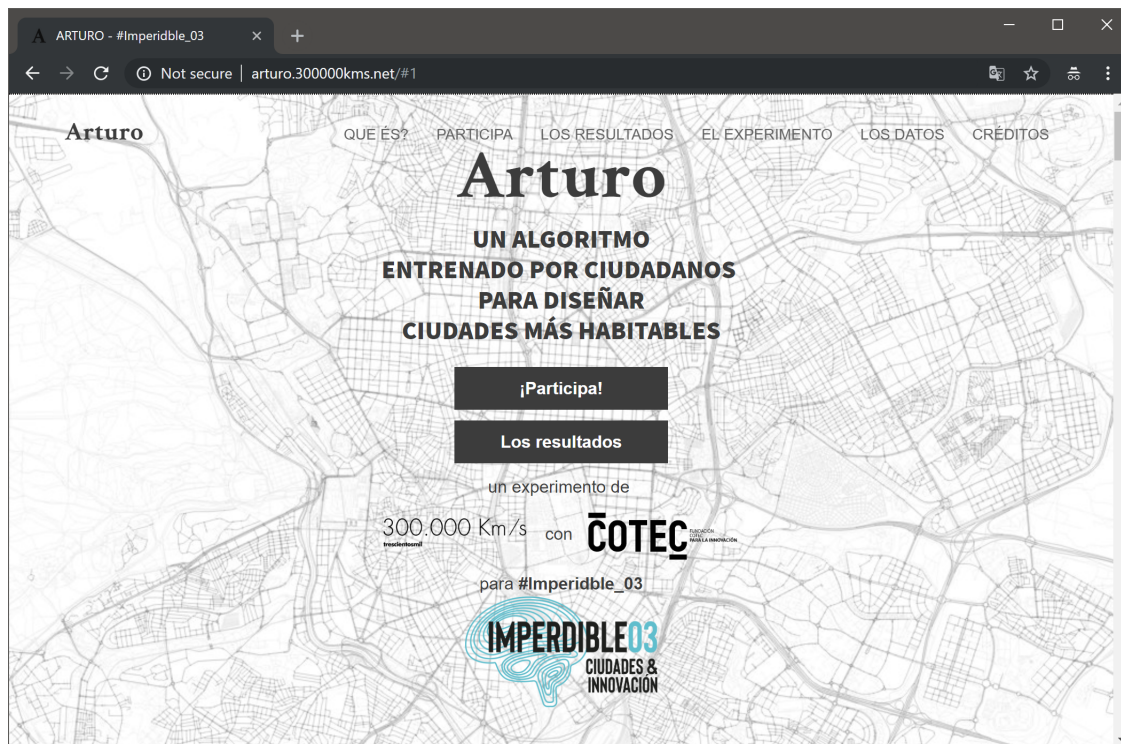
```
import geopandas as gpd
import contextily as ctx
import matplotlib.pyplot as plt
from qgrid import show_grid
```

### 1.1 Sources

#### 1.1.1 “Arturo”

```
In [3]: Image('../figs/arturo.png', width=750)
```

Out[3]:



### 1.1.2 Inside AirBnb

```
In [4]: #Image('../figs/insideabb.png', width=750)
```

## 1.2 Data

```
In [5]: db = gpd.read_file('../data/demo_data.gpkg')
```

```
In [6]: db.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 128 entries, 0 to 127
Data columns (total 15 columns):
neighbourhood          128 non-null object
neighbourhood_group    128 non-null object
land_use_mix           128 non-null float64
closeness_small_parks  128 non-null float64
residence_ratio        128 non-null float64
age_diversity          128 non-null float64
population_density     128 non-null float64
arturo_score           128 non-null float64
abb_review_scores_location 126 non-null float64
abb_price_usd          127 non-null float64
N. Abb                 127 non-null float64
```

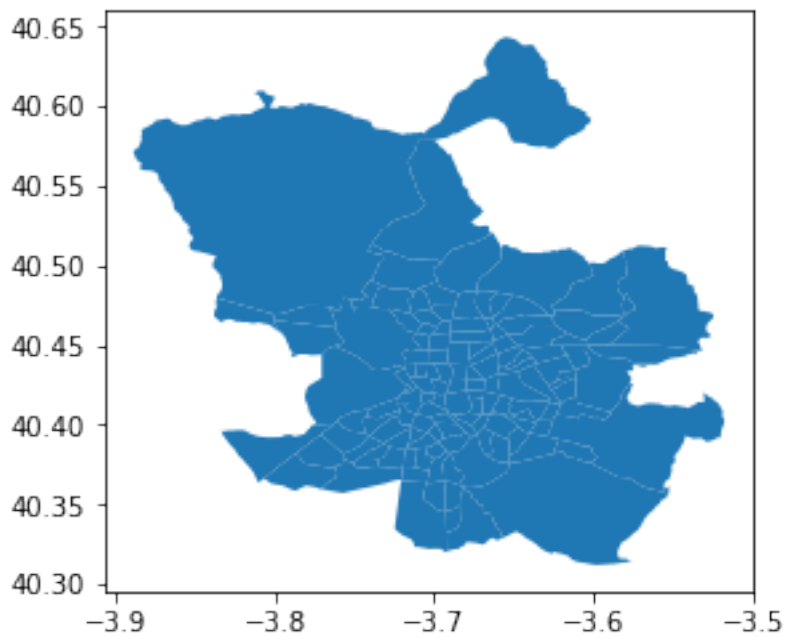
```
N. streets          128 non-null int64
Scored streets      128 non-null int64
pct_scored          128 non-null float64
geometry            128 non-null object
dtypes: float64(10), int64(2), object(3)
memory usage: 15.1+ KB
```

```
In [7]: show_grid(db)
```

```
QgridWidget(grid_options={'fullWidthRows': True, 'syncColumnCellResize': True, 'forceFitColumns':
```

```
In [8]: db.plot()
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8c6db43978>
```



### 1.3 Queries

- Atomic

```
In [9]: db.set_index('neighbourhood').loc['Atocha', 'arturo_score']
```

```
Out[9]: 25.037677469271745
```

- Value-based

```
In [10]: db.query("neighbourhood_group == 'Centro'")
```

```
Out[10]:
```

	neighbourhood	neighbourhood_group	land_use_mix	closeness_small_parks	\
0	Palacio	Centro	0.000365	79.556113	
1	Embajadores	Centro	0.000549	191.088707	
2	Cortes	Centro	0.000346	184.110366	
3	Justicia	Centro	0.000440	193.695258	
4	Universidad	Centro	0.000481	156.970200	
5	Sol	Centro	0.000615	122.989820	

	residence_ratio	age_diversity	population_density	arturo_score	\
0	0.708910	0.002835	5343.068770	28.029880	
1	0.671200	0.002734	11271.551466	28.549250	
2	0.475500	0.002756	4536.868443	29.068393	
3	0.691089	0.002774	6030.387988	27.994642	
4	0.738066	0.002778	10863.626270	27.944427	
5	0.583537	0.002930	5061.064928	29.524770	

	abb_review_scores_location	abb_price_usd	N. Abb	N. streets	\
0	9.878536	94.704802	1416.0	604	
1	9.675348	117.799054	2538.0	421	
2	9.911258	102.711364	880.0	271	
3	9.929070	140.807287	1043.0	301	
4	9.883041	82.927123	1825.0	436	
5	9.923518	101.430099	1216.0	272	

	Scored streets	pct_scored	\
0	152	25.165563	
1	134	31.828979	
2	100	36.900369	
3	97	32.225914	
4	168	38.532110	
5	84	30.882353	

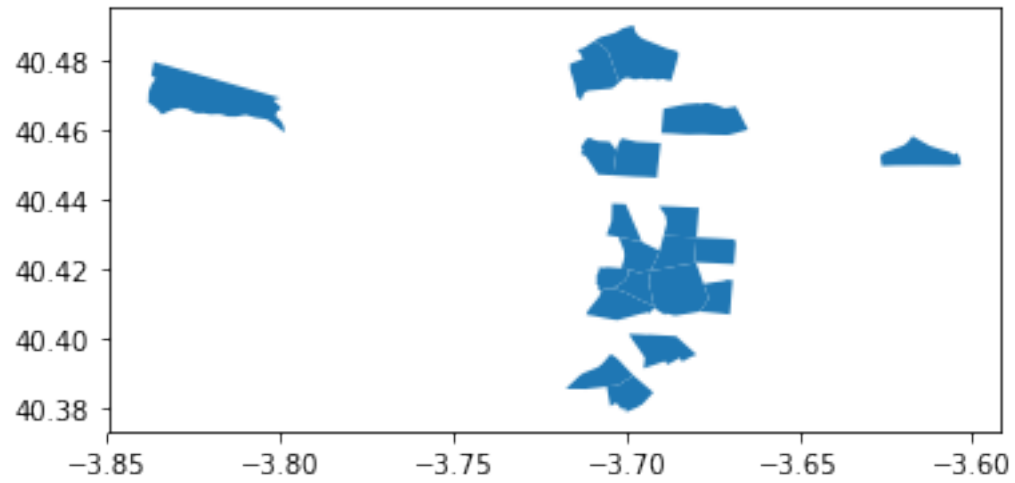
	geometry
0	(POLYGON ((-3.70584 40.420297, -3.706248 40.42...
1	(POLYGON ((-3.703837 40.414318, -3.70277 40.41...
2	(POLYGON ((-3.697959 40.419286, -3.696453 40.4...
3	(POLYGON ((-3.695463 40.418977, -3.696453 40.4...
4	(POLYGON ((-3.701075 40.421339, -3.701547 40.4...
5	(POLYGON ((-3.704747 40.42027, -3.701501 40.42...

- Condition-based

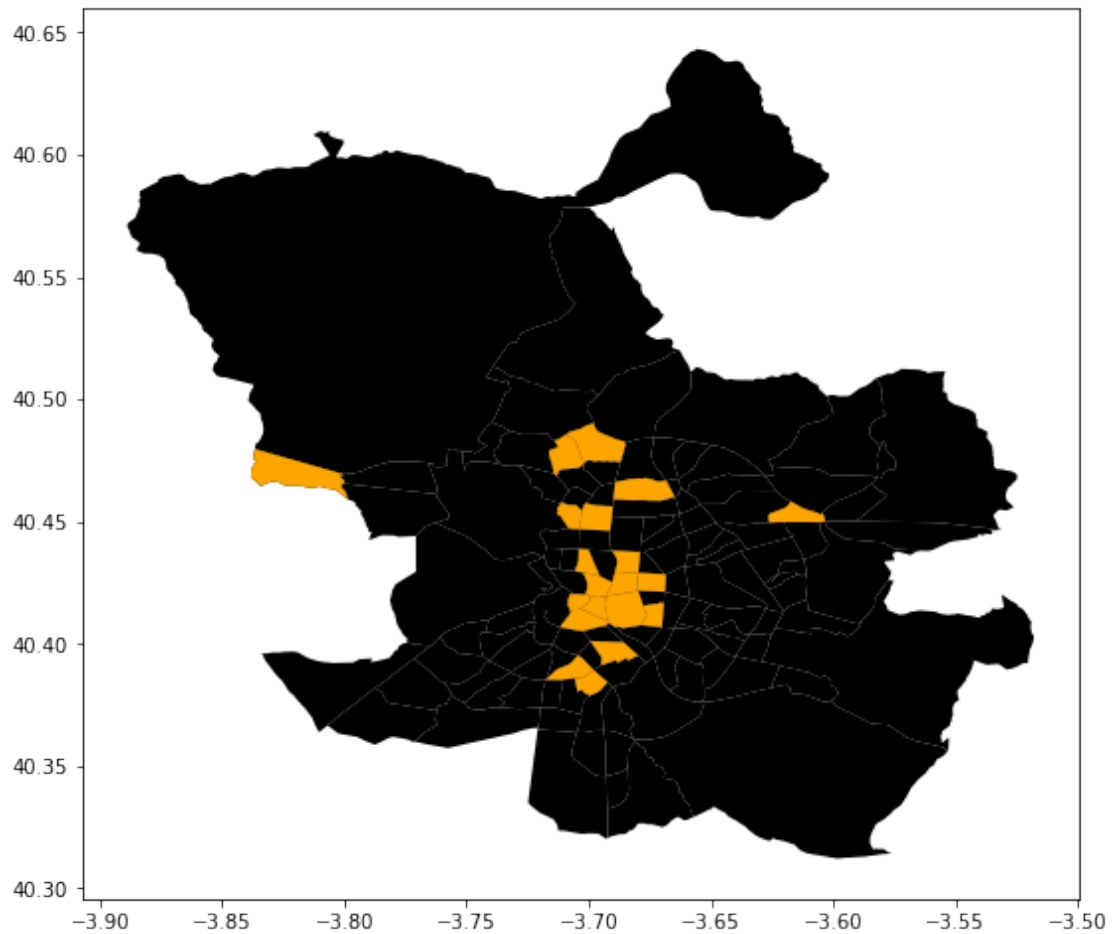
```
In [11]: caros = db.query("abb_price_usd > 100")
```

```
caros.plot()
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8c6d800f60>
```



```
In [12]: f, ax = plt.subplots(1, figsize=(9, 9))
         db.plot(color='k', ax=ax)
         caros.plot(color='orange', ax=ax)
         plt.show()
```



```
In [13]: db.abb_price_usd.describe()
```

```
Out[13]: count      127.000000
         mean        69.256524
         std         42.574609
         min         20.638889
         25%         42.390351
         50%         60.315789
         75%         79.847826
         max         319.400000
         Name: abb_price_usd, dtype: float64
```

## CHALLENGE

- Find the neighbourhoods where the average AirBnb price is greater than 150
- Make a map highlighting the neighbourhoods in the Hortaleza neighbourhood group
- Find the most attractive neighborhood based on the Arturo score

## 1.4 Basemaps (and projections)

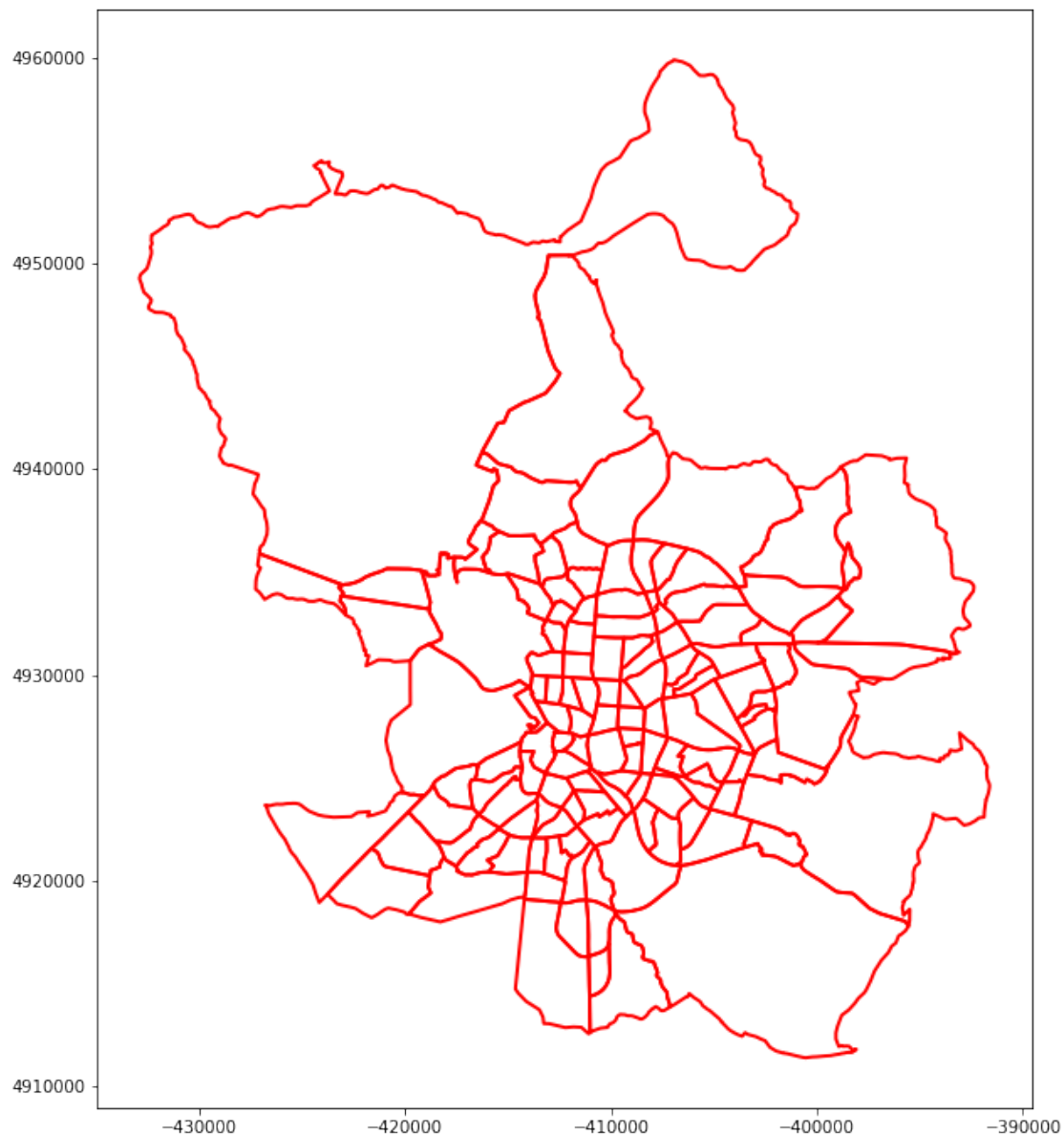
```
In [14]: db.crs
```

```
Out[14]: {'init': 'epsg:4326'}
```

```
In [15]: #IFrame('http://epsg.io/', 750, 400)
```

```
In [16]: db_wm = db.to_crs(epsg=3857)
```

```
In [17]: f, ax = plt.subplots(1, figsize=(12, 12))
db_wm.plot(facecolor='none', edgecolor='red', linewidth=2, ax=ax)
#ctx.add_basemap(ax, url=ctx.sources.ST_TONER)
plt.show()
```



## CHALLENGE

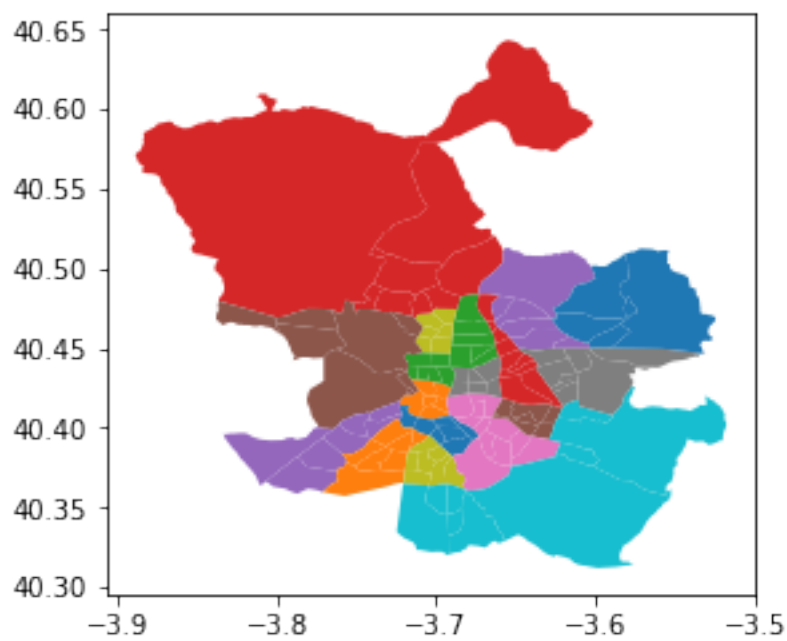
- Make a map of the Madrid neighbourhoods using the Terrain (ST\_Terrain) basemap

## 1.5 Choropleths

### 1.5.1 Categorical

```
In [18]: db.plot(column='neighbourhood_group')
```

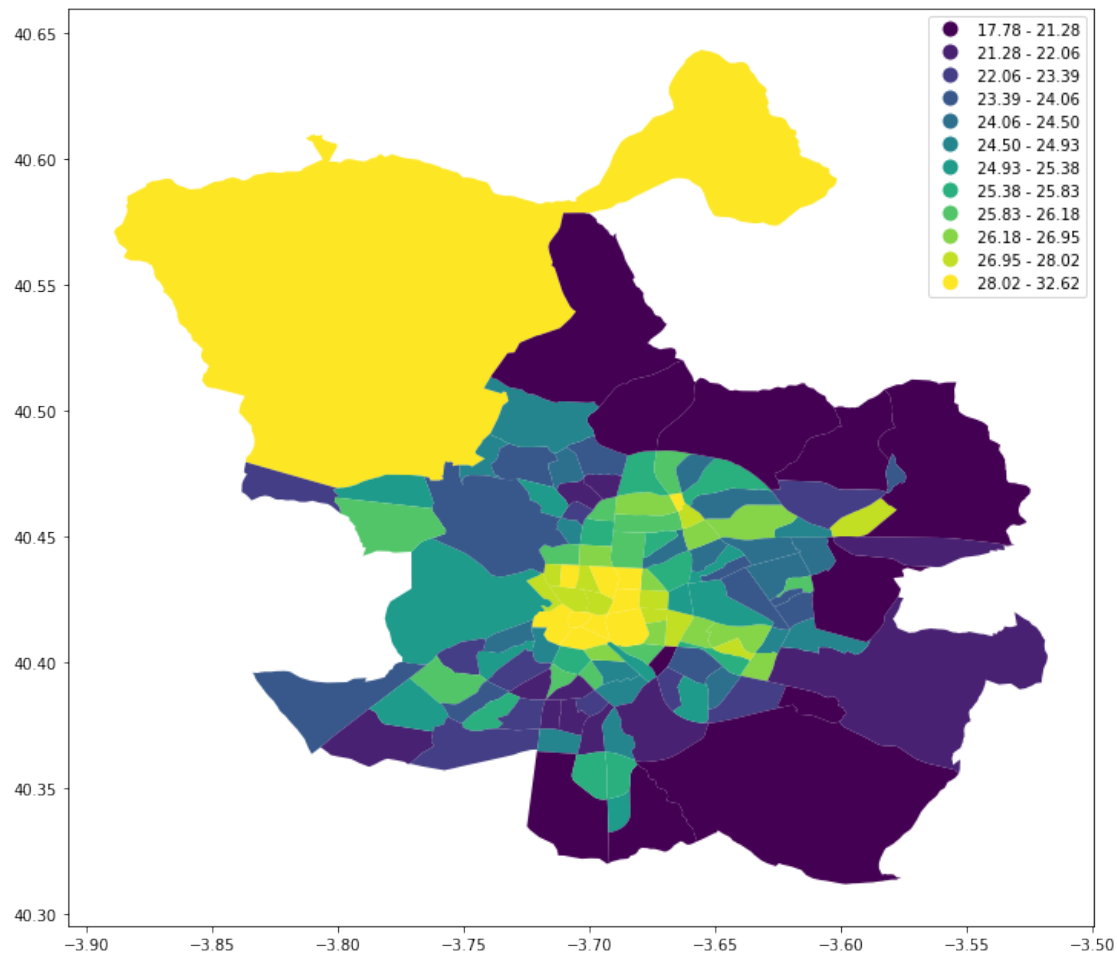
```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8c6d65bcc0>
```



### 1.5.2 Continuous

```
In [19]: f, ax = plt.subplots(1, figsize=(12, 12))
         db.plot(column='arturo_score', scheme='quantiles', k=12,
                 legend=True, ax=ax)
         plt.show()
```





## CHALLENGE

- Make a quantile choropleth of `abb_review_scores_location`
- Make a quantile choropleth of `abb_price_usd`
- Compare the three maps above. Do they display similar patterns?

### 1.5.3 Value by alpha

```
In [20]: from pysal.viz.splot.mapping import vba_choropleth
```

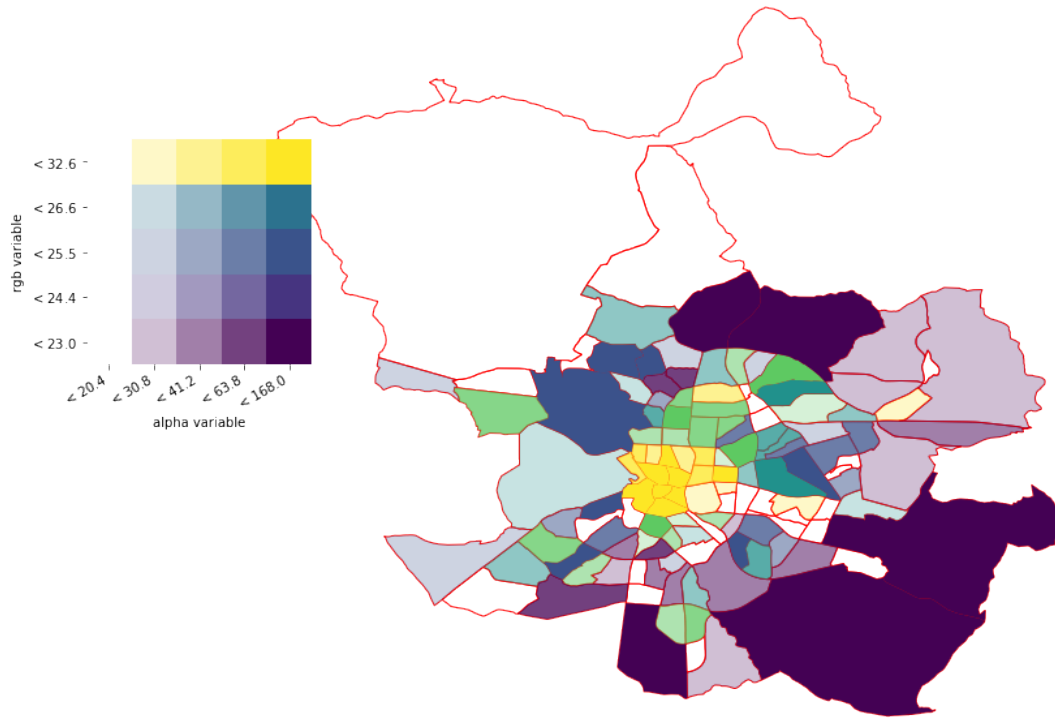
```
/opt/conda/lib/python3.6/site-packages/pysal/model/spvcm/abstracts.py:10: UserWarning: The `dill`
  from .sqlite import head_to_sql, start_sql
```

```
In [21]: f, ax = plt.subplots(1, figsize=(12, 12))
         db.plot(ax=ax, facecolor='none', edgecolor='red')
         vba_choropleth(db['arturo_score'], db['Scored streets'], db,
```

```

alpha_mapclassify=dict(classifier='quantiles', k=5),
rgb_mapclassify=dict(classifier='quantiles', k=5),
legend=True, cmap='viridis', ax=ax)
ax.set_title('')
plt.show()

```



**CHALLENGE** - Make a similar VBA map for average Airbnb price (abb\_price\_usd) using the number of Airbnb properties (N. Abb) to tweak transparency