02 background map and grid

April 5, 2022

1 Maps of italian regions

This notebook creates a simple lightweights map of any Italian region currently set to 6 = FVG

1. create a json file of a single region from the complete geo-json source (italia_geo.json), which contains polygons and attributes (names and codes of regions, provinces, municipaities)
2. create a csv file with the coordinates of each municipality (center of corresponding polygon)
3. create grid file nxn that covers the whole region and saves to .csv

includes simple functions to load and plot map, points and grid

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

2 Create FVG_mappa.json base map from JSON

[ ]:
```
file=r"./maps/ITA/Italia_geo.json"

from pandas_geojson import read_geojson, filter_geojson
geo_json = read_geojson(file)
geo_json['type']
```

[ ]:
```
'FeatureCollection'
```

[ ]:
```
import pandas as pd
nn = len(geo_json['features'])
cols = [  'istat_r', 'istat_p','istat_c','nome', 'poly']
df = pd.DataFrame(columns=cols)
for i in range(nn):
    new_row = {'istat_r' :␣
  ↪geo_json['features'][i]['properties']['reg_istat_code'],
                'istat_p':␣
  ↪geo_json['features'][i]['properties']['prov_istat_code'],
                'istat_c':␣
  ↪geo_json['features'][i]['properties']['com_istat_code'],
                'nome'   : geo_json['features'][i]['properties']['name'],
                'poly'   :␣
 ↪geo_json['features'][i]['geometry']['coordinates'][0][0]}
                   # consider only first polygon

    df = df.append(new_row, ignore_index = True)

# save to a json file - not a standard format!
df.to_json(r'./maps/FVG/FVG_mappa.json', orient='records')
print("fatto! :-D")
```

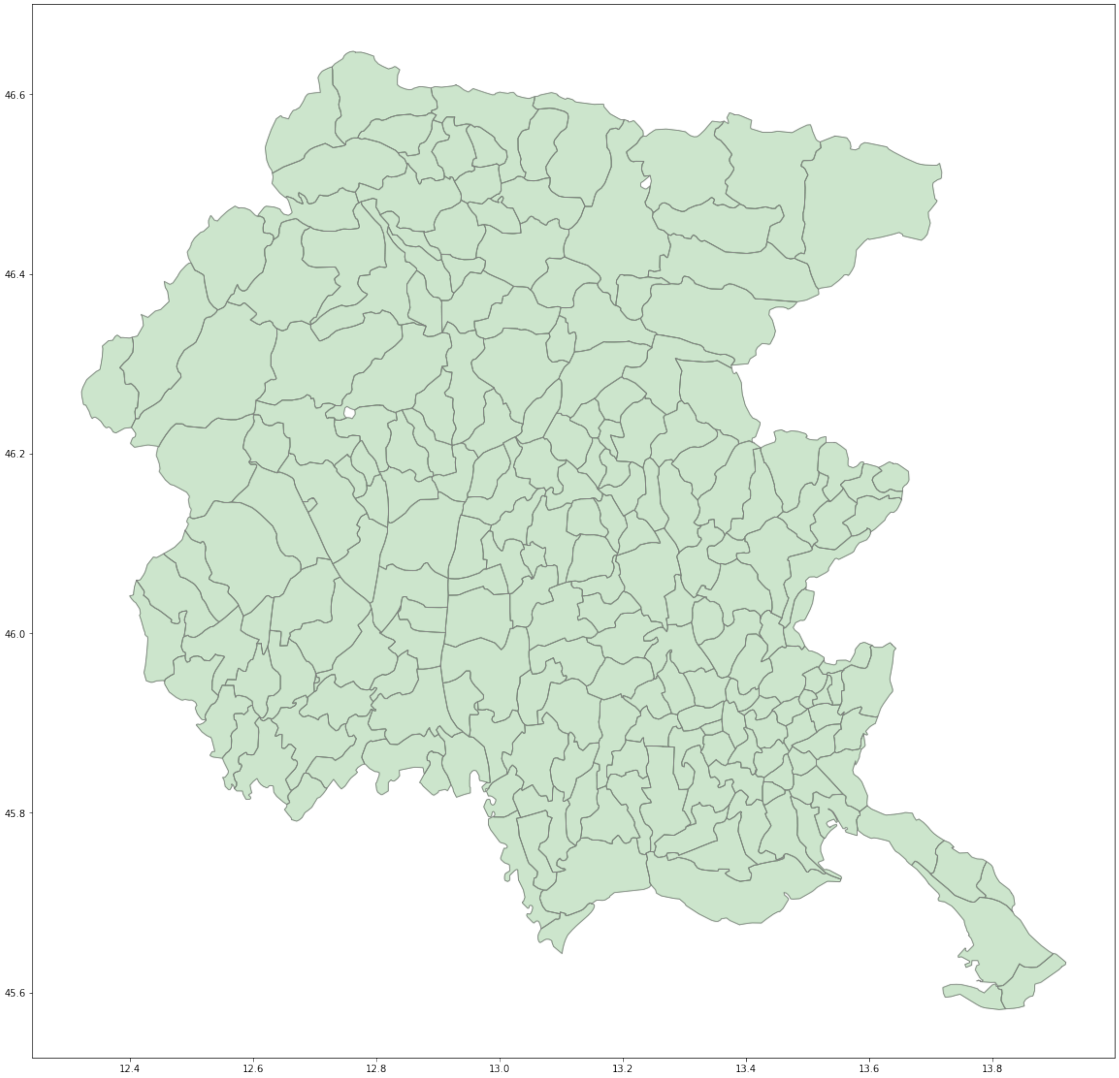
```
fatto! :-D
```

3 Examples of use

[ ]:
```
# read data from json
df.to_json(r'./maps/FVG/FVG_mappa.json', orient='records')
```

[ ]:
```
def plot_map(df, color_fill = "green", alpha = .2, color_border= "gray" ,␣
 ↪linewidth = .8):
    for _, row in df.iterrows():
        x, y = list(zip(*row.poly))
        plt.fill(x,y, color = color_fill, alpha = alpha)
        plt.plot(x,y, color = color_border, linewidth=linewidth)
```

[ ]:
```
#load data from file
df = pd.read_json(r'./maps/FVG/FVG_mappa.json', orient='records')
# filter a single region
df=df[ df.istat_r==6]
# plot the map
fig, ax = plt.subplots(figsize=(20,20))
plot_map(df)
plt.show()
```



4 Create a grid n x n

[ ]:

```
# find min and max coordinates
x_min = df.iloc[0].poly[0][0]
x_max = x_min
y_min = df.iloc[0].poly[0][1]
y_max = y_min
for _, row in df.iterrows():
    x, y = list(zip(*row.poly))
    x_min = np.min(np.append(x,x_min))
    x_max = np.max(np.append(x,x_max))
    y_min = np.min(np.append(y,y_min))
    y_max = np.max(np.append(y,y_max))
```

```
x_min, x_max, y_min, y_max
```

[ ]:

```
(12.3209357495, 13.9188530967, 45.5809267176, 46.6478098111)
```

[ ]:

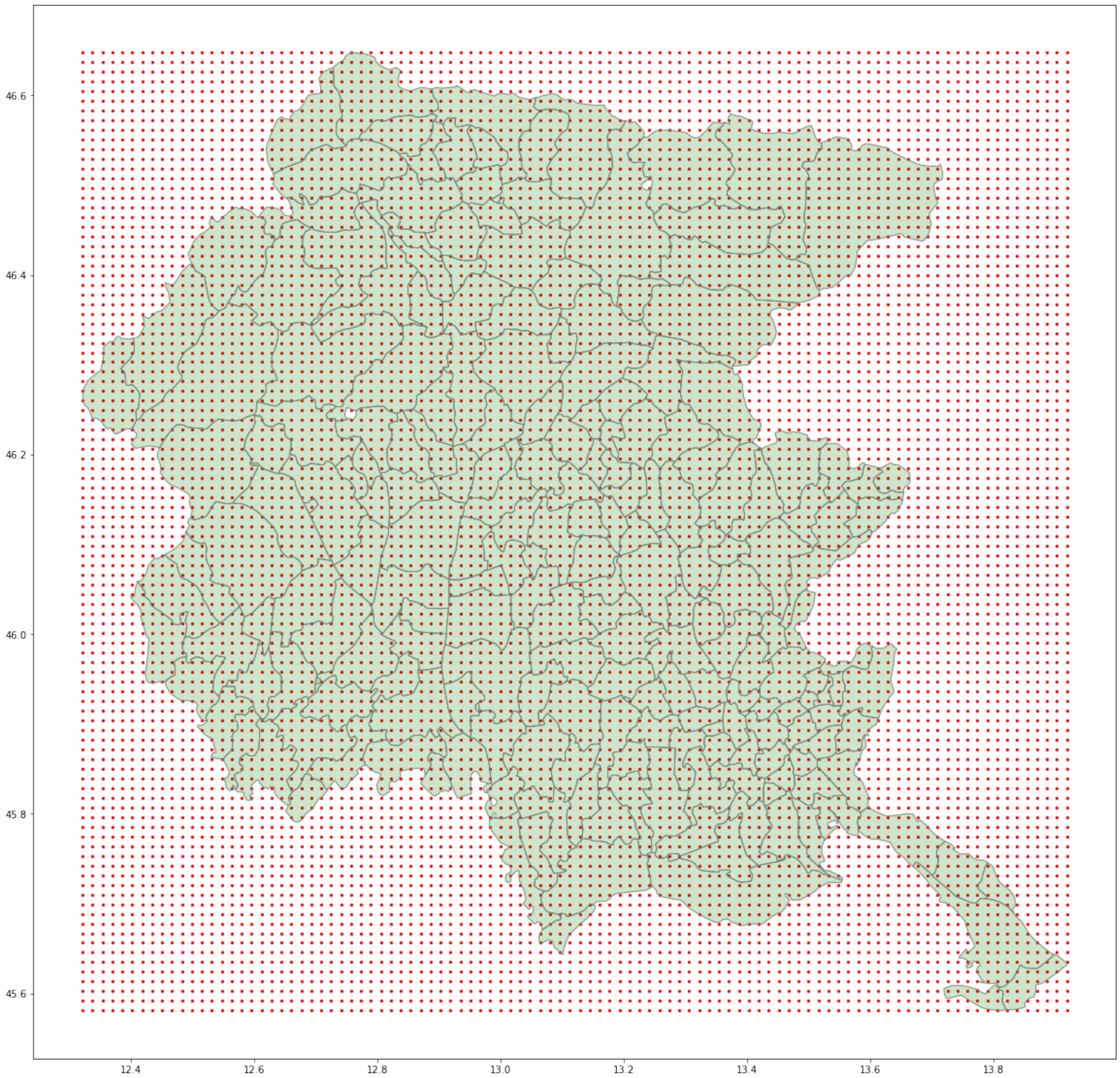
```
nn = 100
xx = np.linspace(x_min, x_max,nn)
yy = np.linspace(y_min, y_max,nn)
x_grid, y_grid = np.meshgrid(xx, yy, indexing='ij')

grid = pd.DataFrame(columns = ['x','y'])
grid['x']=np.reshape(x_grid,-1)
grid['y']=np.reshape(y_grid,-1)

grid.to_csv(r'./maps/FVG/FVG_grid.csv', index=False)



fig, ax = plt.subplots(figsize=(20,20))
plot_map(df)
plt.scatter(x_grid, y_grid, s=20, marker = '.', color = 'red')
plt.show()
```

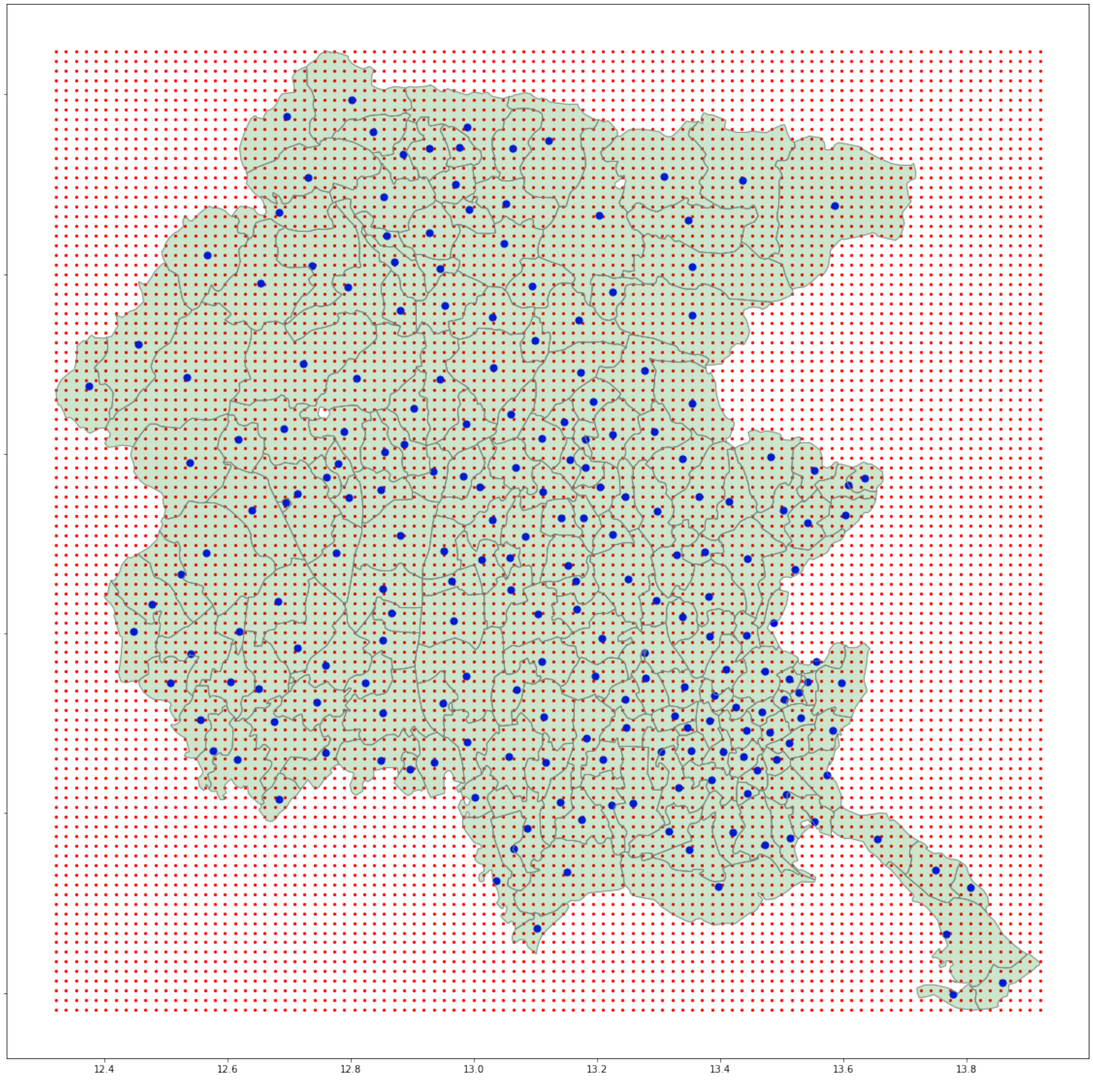


[ ]:

```
# find centroids
centroids = df#[  'istat_r', 'istat_p','istat_c','nome', 'poly']
xc = []
yc = []
for _, row in centroids.iterrows():
    x, y = list(zip(*row.poly))
    xc.append(np.mean(x))
    yc.append(np.mean(y))

centroids['xc']=xc
centroids['yc']=yc
cols_to_use = ['istat_r','istat_p','istat_c','nome','xc','yc']
centroids = centroids [cols_to_use]
centroids.to_csv(r'./maps/FVG/FVG_centroids.csv', index=False)
```

```
fig, ax = plt.subplots(figsize=(20,20))
plot_map(df)
plt.scatter(x_grid, y_grid, s=20, marker = '.', color = 'red')
plt.scatter(xc, yc,         s=50, marker = 'o', color = 'blue')
plt.show()
```



5 companies

plot companiies' location on the map

[ ]:

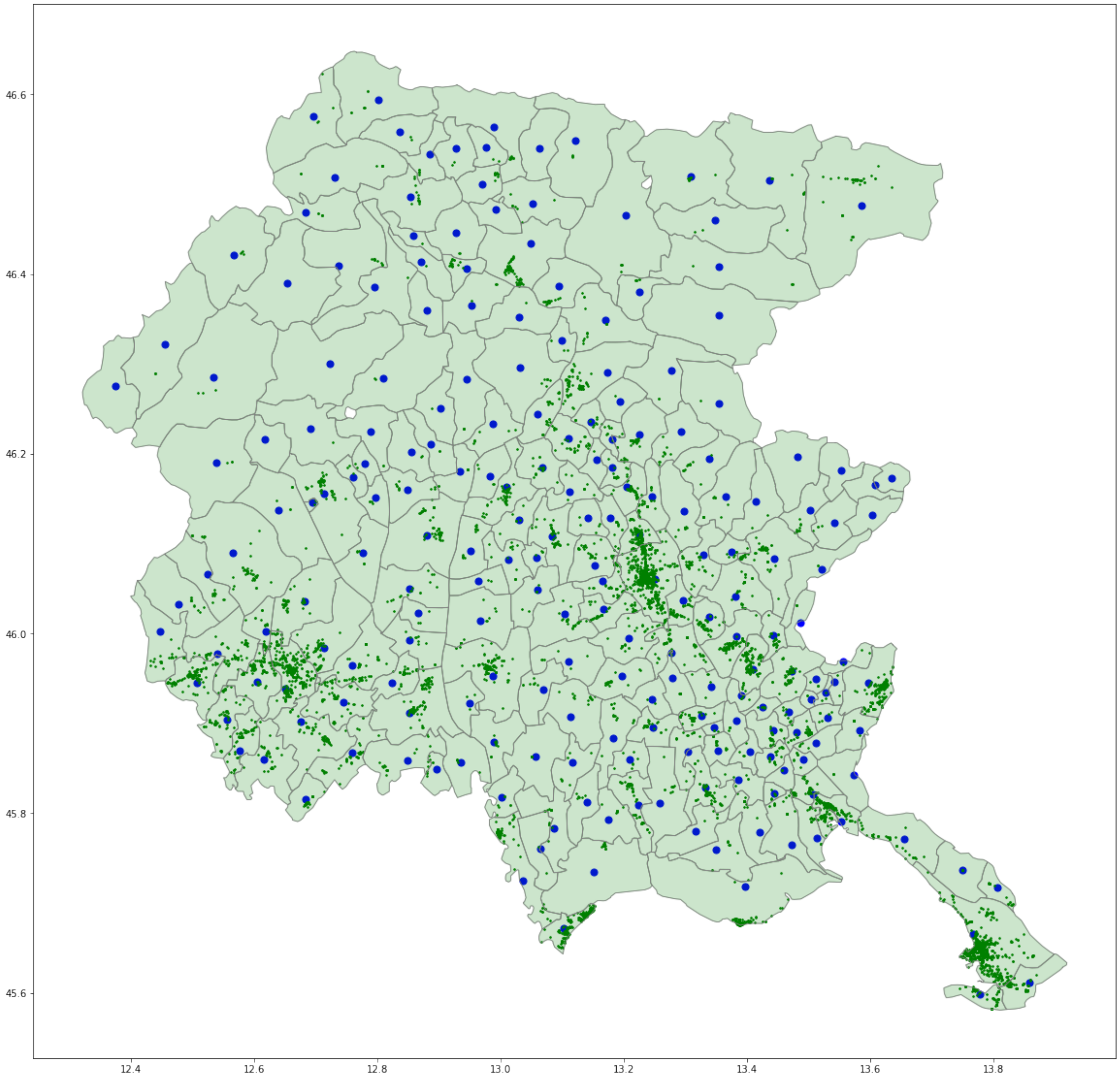
```
comp = pd.read_csv(r'./maps/FVG/FVG_companies.csv', dtype='str')
comp.columns = ['ind', 'cf', 'company','unit', 'lat', 'lon']
```

```
comp['lat'] = comp['lat'].astype(float)
comp['lon'] = comp['lon'].astype(float)
```

[ ]:

```
fig, ax = plt.subplots(figsize=(20,20))
plot_map(df)
#plt.scatter(x_grid, y_grid, s=20, marker = '.', color = 'red')
plt.scatter(xc, yc,         s=50, marker = 'o', color = 'blue')

x = comp['lon']
y = comp['lat']
plt.scatter(x,y, color = "green", marker = "X", s=2)
plt.show()
```


46.6
46.4
46.2
46.0
45.8
45.6
12.4
12.6
12.8
13.0
13.2
13.4
13.6
13.8