geopandas

July 2, 2023

1 Geopandas

geopandas est devenu le point d'entrée à tout traitement lié aux coordonnées géographiques.

[1]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

[1]:
```
<IPython.core.display.HTML object>
```

1.1 Coordonnées géographiques, et geocoding

[2]:
```
import geopandas
import pandas
```

[3]:
```
df = pandas.DataFrame(dict(name=["A", "B", "C", "D", "E"],
                           address=["50 rue du moulin 08000 Charleville-Mézières",
                                    "50 rue dumoulin 08000 Charleville-Mézières",
                                    "50 rue du moulin 08000 Charleville",
                                    "29 rue Ordener 75018 Paris",
                                    "Gare Lille"]))
df
```

[3]:
```
  name                                     address
0    A  50 rue du moulin 08000 Charleville-Mézières
1    B   50 rue dumoulin 08000 Charleville-Mézières
2    C           50 rue du moulin 08000 Charleville
3    D                   29 rue Ordener 75018 Paris
4    E                                  Gare Lille
```

[4]:
```
from geopy.geocoders import Nominatim
```

[5]:
```
geolocator = Nominatim(timeout=10, user_agent = "myGeolocator")
res = geolocator.geocode("50 rue du moulin 08000 Charleville-Mézières")
res
```

[5]:
```
Location(50, Rue du Moulin, Charleville-Mézières, Ardennes, Grand Est, France
métropolitaine, 08000, France, (49.775247, 4.721863, 0.0))
```

[6]:
```
res.latitude, res.longitude, res.point
```

[6]:
```
(49.775247, 4.721863, Point(49.775247, 4.721863, 0.0))
```

[7]:
```
coordinates = df['address'].apply(geolocator.geocode)
```

[8]:
```
import numpy
from shapely.geometry import Point

df['position'] = [numpy.nan if c is None else
                  Point(c.point.longitude, c.point.latitude) for c in coordinates]
df
```

[8]:
```
  name                                     address  \
0    A  50 rue du moulin 08000 Charleville-Mézières
1    B   50 rue dumoulin 08000 Charleville-Mézières
2    C           50 rue du moulin 08000 Charleville
3    D                   29 rue Ordener 75018 Paris
4    E                                  Gare Lille

                                       position
0                   POINT (4.721863 49.775247)
1                                          NaN
2                   POINT (4.721863 49.775247)
3                 POINT (2.3539607 48.8906617)
4  POINT (3.162718387293441 50.69456630000001)
```

[9]:
```
geodf = geopandas.GeoDataFrame(df)
```

[10]:
```
geodf
```

[10]:
```
  name                                     address  \
0    A  50 rue du moulin 08000 Charleville-Mézières
1    B   50 rue dumoulin 08000 Charleville-Mézières
2    C           50 rue du moulin 08000 Charleville
3    D                   29 rue Ordener 75018 Paris
4    E                                  Gare Lille

                                       position
0                   POINT (4.721863 49.775247)
1                                          NaN
2                   POINT (4.721863 49.775247)
3                 POINT (2.3539607 48.8906617)
4  POINT (3.162718387293441 50.69456630000001)
```

[11]:
```
geodf.dropna()
```

[11]:
```
  name                                     address  \
0    A  50 rue du moulin 08000 Charleville-Mézières
2    C           50 rue du moulin 08000 Charleville
3    D                   29 rue Ordener 75018 Paris
4    E                                  Gare Lille

                                       position
0                   POINT (4.721863 49.775247)
2                   POINT (4.721863 49.775247)
3                 POINT (2.3539607 48.8906617)
4  POINT (3.162718387293441 50.69456630000001)
```

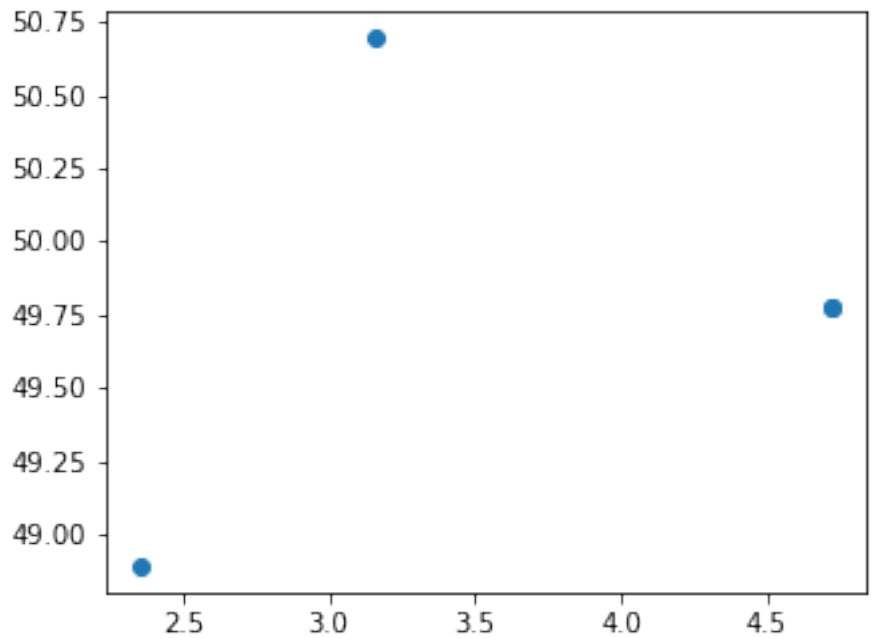
[12]:
```
gdf = geodf.set_geometry('position')
gdf
```

[12]:

	name	address	position
0	A	50 rue du moulin 08000 Charleville-Mézières	POINT (4.72186 49.77525)
1	B	50 rue dumoulin 08000 Charleville-Mézières	None
2	C	50 rue du moulin 08000 Charleville	POINT (4.72186 49.77525)
3	D	29 rue Ordener 75018 Paris	POINT (2.35396 48.89066)
4	E	Gare Lille	POINT (3.16272 50.69457)

1.2 Carte

[13]:
```
gdf.plot();
```



[14]:
```
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import matplotlib.pyplot as plt
```

[15]:
```
fig = plt.figure(figsize=(7,7))
ax = fig.add_subplot(1, 1, 1, projection=ccrs.PlateCarree())
ax.set_extent([-5, 10, 42, 52])

ax.add_feature(cfeature.OCEAN)
ax.add_feature(cfeature.COASTLINE)
ax.add_feature(cfeature.RIVERS)
ax.add_feature(cfeature.LAKES)
ax.add_feature(cfeature.LAND)
ax.add_feature(cfeature.BORDERS, linestyle=':')
ax.plot([2.35, 2.20], [48.85, 48.71], '.')
```

```
ax.text(2.35, 48.85, "Paris")
ax.text(2.20, 48.71, "Saclay", ha="right")
ax.set_title('France');
```

[16]:

```

```

[17]:

```
fig = plt.figure(figsize=(7,7))
ax = fig.add_subplot(1, 1, 1, projection=ccrs.PlateCarree())
ax = gdf.plot(ax=ax)
ax.add_feature(cfeature.OCEAN)
ax.add_feature(cfeature.COASTLINE)
ax.add_feature(cfeature.RIVERS)
ax.add_feature(cfeature.LAKES)
ax.add_feature(cfeature.LAND)
ax.add_feature(cfeature.BORDERS, linestyle=':')
for i in range(gdf.shape[0]):
    if gdf.geometry[i] is None:
        continue
    ax.text(gdf.geometry[i].x, gdf.geometry[i].y, gdf.name[i])
ax;
```

1.3 Régions administratives

[18]:

```
```