Spatially-Enabled DataFrames vs GeoPandas

Comparing Python Geospatial Frameworks

Bryan Chastain, PhD
GDIT, supporting USEPA

https://github.com/bchastain/devsummit2023

GENERAL DYNAMICS Information Technology

1

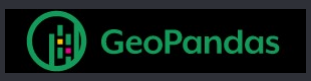
Data Frames

- Inspiration from popular R statistical language led to development of data frame approach for Python
- Many goals
  - Data structures to make working with statistical or “labeled” data sets easy and intuitive for non-experts
  - Create a friendly backbone for implementing statistical models
  - Provide an integrated set of tools for common analyses
  - Implement statistical models!
- Takes some inspiration from R but aims also to improve
- Core idea: ndarrays with labeled axes and lots of methods
- Etymology: **pa**nel **da**ta **s**tructures

https://github.com/bchastain/devsummit2023

2

EPA GeoPandas



- Created during 2013 SciPy conference
- Make working with geographic data like working with other kinds of data in python
- Work with existing tools
  - Desktop GIS (ArcGIS, QGIS)
  - Geospatial databases (e.g., PostGIS)
  - Web maps (Leaflet, D3, etc.)
  - Python data tools (pandas, numpy, etc.)
- https://geopandas.org

https://github.com/bchastain/devsummit2023

3

EPA GeoPandas can do

- Geometry operations (Shapely)
- Data alignment (pandas)
- Coordinate transformations (pyproj)
- Read/write GIS file formats (Fiona)
- Create a GeoDataFrame from PostGIS table
- Output any object as geoJSON
- Plotting (matplotlib)

https://github.com/bchastain/devsummit2023

4

Spatially-Enabled DataFrames

- Similarly, Esri released their Spatially-Enabled DataFrame (SEDF) in 2018
  - Replaced short-lived SpatialDataFrame
- Part of ArcGIS API for Python
- Can use either ArcGIS or Shapely geometry engine
- Crossplatform
- Two different namespaces
  - geom on Series, spatial on DataFrame

https://github.com/bchastain/devsummit2023

5

Spatial data IO

- Shapefile

```
gpd.read_file('my.shp.zip')
pd.DataFrame.spatial.from_featureclass('my.shp') #sedf
```

- Geodatabase

```
gpd.read_file('my.gdb', driver='FileGDB', layer='layer_a')
pd.DataFrame.spatial.from_featureclass('my.gdb\layer_a')
```

- Geojson

```
gpd.read_file('my.geojson')
FeatureSet.from_geojson('my.geojson').sdf
```

- Others
  - GeoFeather, GeoParquet
- Writing shapefile, geojson work similar to reading, GDB special*
  - `gdf.to_file()` for GPD
  - `sdf.to_featureclass(), .to_geojson` for SEDF

https://github.com/bchastain/devsummit2023

6

Accessing data from Web (esri)

```
url =
'https://services.arcgis.com/cJ9YHowT8TU7DUyn/arcgis/
rest/services/Air_Now_Current_Monitors_Ozone_and_PM/F
eatureServer/0'
# GeoPandas
# Note: need to handle pagination for large datasets
url_q = url + '/query?where=1%3D1&outFields=*&f=json'
data = requests.get(url).text
gpd.read_file(data)

# SEDF
fl = FeatureLayer(url)
pd.DataFrame.spatial.from_layer(fl)
```

https://github.com/bchastain/devsummit2023

7

Accessing data from Web (OGC)

```
# OGC Features API
url = 'https://demo.pygeoapi.io/covid-
19/collections/cases_netherlands_per_municipality
/items?f=json'
data = requests.get(url).json()
# GeoPandas
gpd.GeoDataFrame.from_features(data['features'])
# SEDF
pd.DataFrame.spatial.from_geojson(data)
```

https://github.com/bchastain/devsummit2023

8

Publishing Spatial Data (esri)

- GPD
  - Have to use requests & REST API
  - Just, no…
- SEDF

```
lyr = sdf.spatial.to_featurelayer(
      'census_cities', folder='census')
```

https://github.com/bchastain/devsummit2023

9

Publishing Spatial Data (OGC)

- For OGC Features API, basically the same in both GPD & SEDF
- Use requests to POST geojson

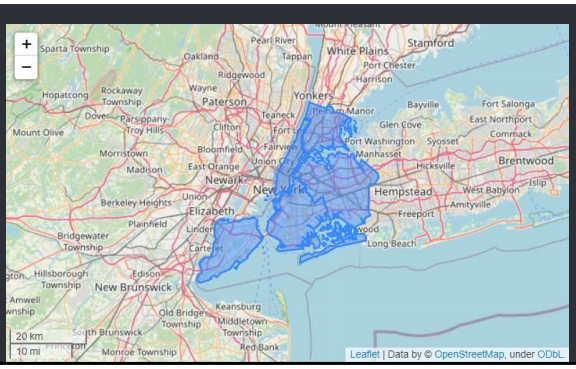
```
Client                                                          Server
  |                                                               |
  | POST /collections/oakland_buildings/items   HTTP/1.1          |
  | Content-Type: application/geo+json                            |
  |                                                               |
  | {                                                             |
  |   "type": "Feature",                                          |
  |   "geometry": {                                               |
  |     "type": "MultiPolygon",                                   |
  |     "coordinates": [                                          |
  |     [[[-122.2694982,37.79045922],[ -122.2693624, 37.79041125],|
  |      [-122.2693518,37.79042521],[ -122.26899, 37.7902858],    |
  |      [-122.2690027,37.79027181],[ -122.2688602, 37.79021705], |
  |      [-122.2687222,37.790445],[-122.2688582, 37.79049813],    |
  |      [-122.2689084,37.79041634],[ -122.2689473, 37.79041058], |
  |      [-122.2691974,37.79051029],[ -122.2692367, 37.7906097],  |
  |      [-122.2692201,37.79064271],[ -122.2693538, 37.79069243], |
  |      [-122.2694982,37.79045922]]]]                            |
  |   },                                                          |
  |   "properties": {                                             |
  |     "shape_len": 666.635209546,                               |
  |     "shape_area": 14016.0452102,                              |
  |     "bldgid3": "11 EMBARCADERO WEST_bldgG102",                |
  |     "bldgid2": "11 EMBARCADERO WEST",                         |
  |     "bldgtype": "Commercial Building",                        |
```

10

Interactive Maps - GeoPandas

- Uses folium for webmap

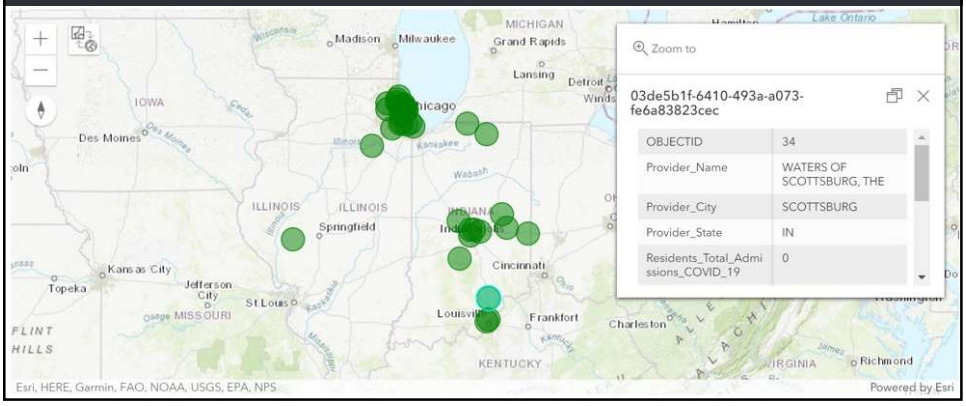
```
import folium
nybb = geopandas.read_file(geopandas.datasets.get_path('nybb'))
nybb.explore()
```



11

Interactive Maps - SEDF

```
m = gis.map('IL, USA', 6)
my_sdf.spatial.plot(m)
```



12

Spatial Join

- Like pandas merge, but based on spatial relationship instead of columnar
- Both GPD & SEDF use inner join by default, but can specify

```
# SEDF
item = gis.content.get("56eac669cfbf4b529c067a65a3c559b5")
# Get all registered pollutant dischargers in TX
df = item.layers[0].query("STATE_CODE='TX'").sdf
cnty_item = gis.content.get("14c5450526a8430298b2fa74da12c2f4")
# Get all TX counties
df_cnty = cnty_item.layers[0].query("STATE_ABBR='TX'").sdf
# Coordinate system mismatch, so need to project Counties
df_cnty.spatial.project(4269)
df_join = df.spatial.join(df_cnty)
# GeoPandas
# Using GPD versions of df & df_cnty above
gdf_cnty.to_crs(4269)
gdf_join = gpd.sjoin(gdf, gdf_cnty)
```

https://github.com/bchastain/devsummit2023

13

Other Geoprocessing in Both!

- Projections
- Spatial Indexing
- Distance between features
- Buffer
- Union/Intersect/Diff
- Simplify
  - `simplify()` in gpd; `generalize()` in SEDF
- Dissolve
- Geocoding
  - GPD – Photon
  - SEDF – Esri World by default

https://github.com/bchastain/devsummit2023

14

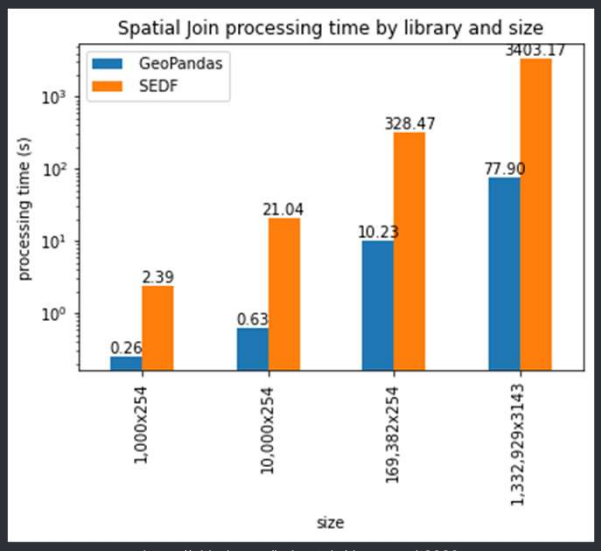
Performance

- Testing performance on spatial joins of various sizes
  - Points, n=1,000; 10,000; 169,382; 1,332,929
  - Polygons, n=254; 3,143
  - Average of 3 runs across each point/poly combination for both GPD & SEDF
- Environment
  - 11th gen Core i9, 32GB RAM, SSD
  - ArcGIS API for Python v2.1.0 (using arcpy)
  - GeoPandas v0.12.2

https://github.com/bchastain/devsummit2023

15

Results



https://github.com/bchastain/devsummit2023

16

EPA Limitations

- Geopackage
  - `gpd.read_file('package.gpkg', layer='cnty')`
  - SEDF??
- PostGIS
  - `gpd.read_postgis(sql, con)`
  - SEDF
    - Can read point data using `from_xy()`
    - No ability to read WKT into SEDF geom

https://github.com/bchastain/devsummit2023

17

EPA Compatibility

- From GPD to SEDF

```
pd.DataFrame.spatial.from_geodataframe(gdf)
```

- From SEDF to GPD
  - No direct function, but can go through FeatureSet:

```
fl = FeatureLayer("https://services.arcgis.com/P3ePLMYs2RVChkJx/arcgis/rest/services/World_Cities/FeatureServer/0")
sdf1 = pd.DataFrame.spatial.from_layer(fl)
fs = sdf1.spatial.to_featureset()
gpd.GeoDataFrame.from_features(json.loads(fs.to_geojson)['features'])
```

https://github.com/bchastain/devsummit2023

18

Conclusions

- What are you currently using?
- Do you need to publish to GIS servers?
- Do you need to work with Geopackages/PostGIS?
- Is performance an issue?
- **You don't have to choose!**

https://github.com/bchastain/devsummit2023

19

Questions?

https://github.com/bchastain/devsummit2023

GENERAL DYNAMICS Information Technology

20