Python Tutorial – Part II: Data

Mark A. Austin

University of Maryland

austin@umd.edu
ENCE 201, Fall Semester 2025

September 2, 2025

Overview

Spatial Data and Dataset Transformation

(Working with GeoPandas)

GeoPandas

GeoPandas

GeoPandas is an open source project to make working with geospatial data in Python easier.

Approach:

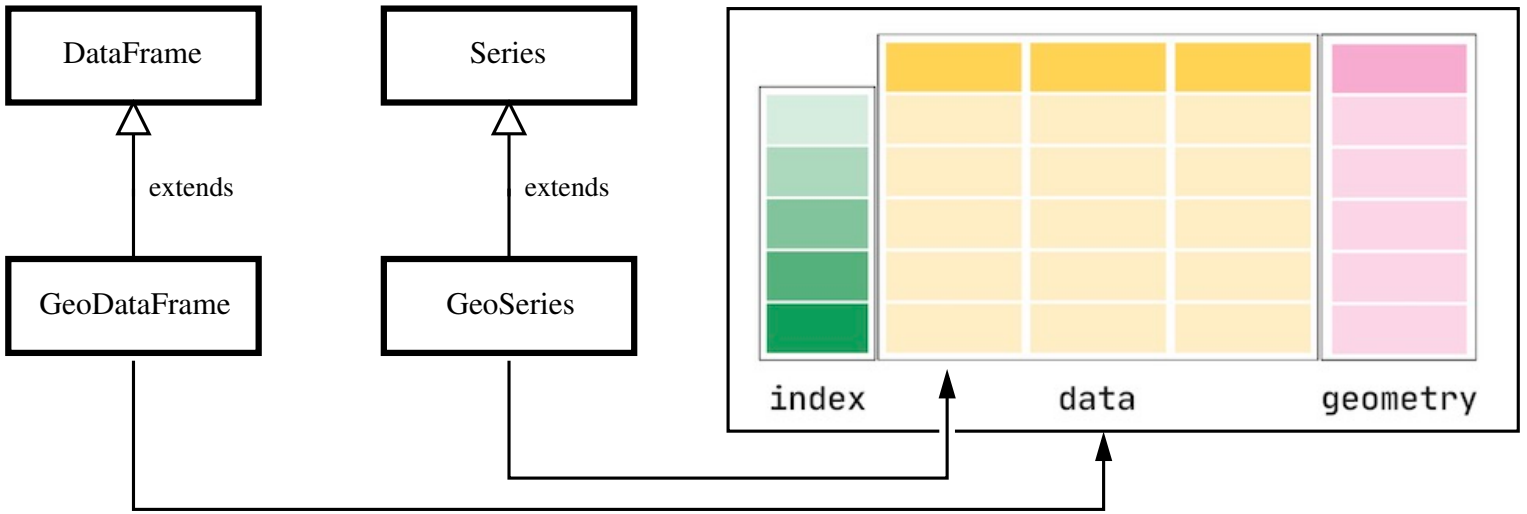
- Extend the datatypes used by Pandas to allow spatial operations on geometric types.
- Geometric operations are performed by shapely.
- Geopandas further depends on fiona for file access and matplotlib for plotting.

Installation

```
prompt >> pip3 install geopandas
```

Working with GeoPandas Dataframes

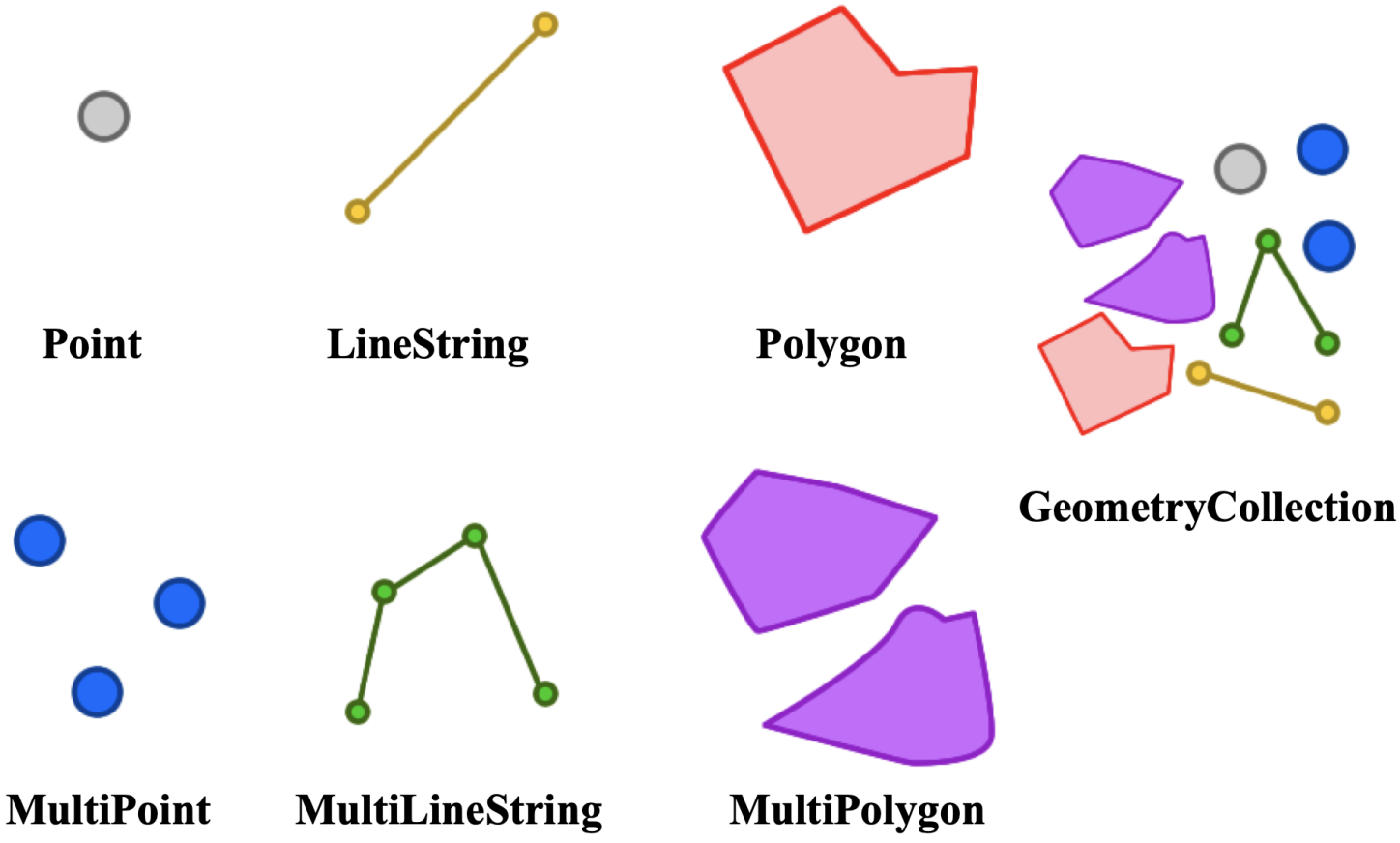
Core Modeling Concepts and Data Structure:



- GeoSeries handle geometries (points, polygons, etc).
- GeoDataFrames store geometry columns and perform spatial operations. They can be assembled from geopandas.GeoSeries.

Working with GeoPandas Dataframes

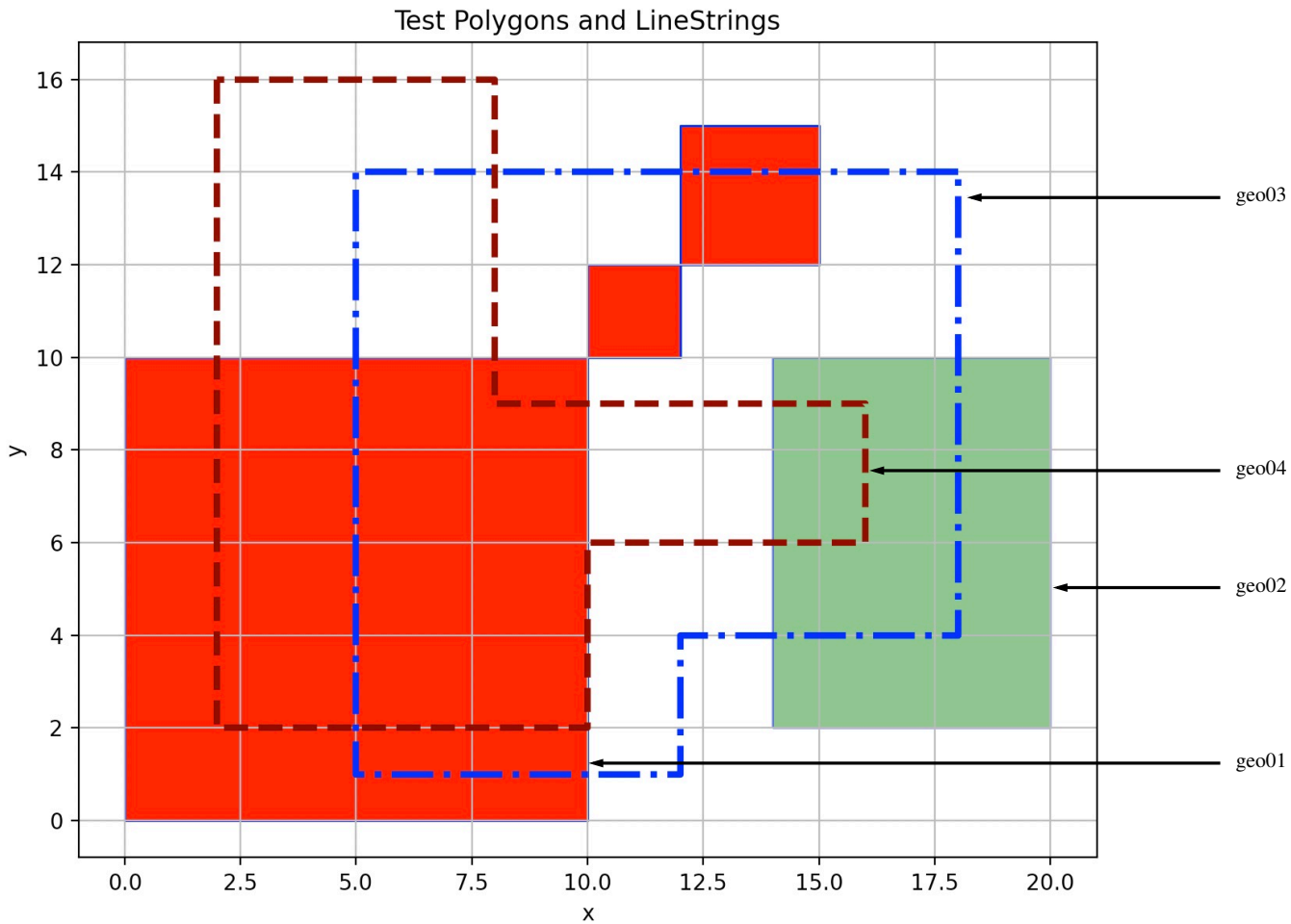
Geometric Objects: points, multi-points, lines, multi-lines, polygons, multi-polygons.



Example 1: Manual Specification of Geometric Shapes

Example 1: Manual specification of polygon and linestring shapes

...



Example 1: Manual Specification of Geometric Shapes

Part I: Problem Setup

```
1   # ========================================================================
2   # TestGeoSeries01.py. Manual assembly of simple geometries.
3   #
4   # Written by: Mark Austin                                        February 2023
5   # ========================================================================
6
7   import geopandas
8   from geopandas import GeoSeries
9   from shapely.geometry import Polygon
10  from shapely.geometry import LineString
11
12  import matplotlib.pyplot as plt
13
14  # ==============================
15  # main method ...
16  # ==============================
17
18  def main():
19      print("--- Enter TestGeoSeries01.main()                ... ");
20      print("--- ======================================== ... ");
21
22      print("--- Part 01: Create individual polygons ... ");
23
24      polygon01 = Polygon([   (0,0),  (10,0), (10,10),  (0,10) ] )
25      polygon02 = Polygon([ (10,10), (12,10), (12,12), (10,12) ] )
26      polygon03 = Polygon([ (12,12), (15,12), (15,15), (12,15) ] )
```

Example 1: Manual Specification of Geometric Shapes

Part I: Problem Setup (Continued) ....

```
27      polygon04 = Polygon([ (14,2), (20,2), (20,10), (14,10) ] )
28
29      print("--- Part 02: Add polygons to GeoSeries ... ");
30
31      geo01 = GeoSeries( [ polygon01, polygon02, polygon03 ]);
32      geo02 = GeoSeries( [ polygon04 ]);
33
34      print("--- Part 03: Create simple linestring GeoSeries ... ");
35
36      line01 = LineString([ (18,14), (5,14), (5,1), (12,1), (12,4), (18,4), (18,14) ] )
37      geo03  = GeoSeries( [ line01 ]);
38      line02 = LineString([ (2,16), (2,2), (10,2), (10,6), (16,6), (16,9), (8,9), (8,16),
39      geo04  = GeoSeries( [ line02 ]);
40
41      print("--- Part 04: Print GeoSeries info and contents ... ");
42
43      print(geo01)
44      print(geo02)
45
46      print("--- Part 05: Area and boundary of geo01 ... ");
47
48      print(geo01.area)
49      print(geo01.boundary)
50
51      print("--- Part 06: Area and boundary of geo02 ... ");
52
53      print(geo02.area)
54      print(geo02.boundary)
```

Example 1: Manual Specification of Geometric Shapes

Part I: Problem Setup (Continued) ....

```
55
56      print("--- Part 07: Spatial relationship of geo01 through geo04 ... ");
57
58      print("--- Compute intersection of (lines) geo03 and geo04 ...")
59      geo02a = geo03.intersects(geo04)
60      print("---   geo03.intersects(geo04) --> {:s} ...".format( str( geo02a[0] ) ))
61      geo02b = geo03.intersection(geo04)
62      print("---   geo03.intersection(geo04) --> {:s} ...".format( str( geo02b[0] ) ))
63
64      print("--- Compute intersection of (region) geo01 and (lines) geo03 and geo04 ...")
65      geo02c = geo01.intersection(geo03)
66      print("---   geo01.intersection(geo03) --> {:s} ...".format( str( geo02c[0] ) ))
67      geo02d = geo01.intersection(geo04)
68      print("---   geo01.intersection(geo04) --> {:s} ...".format( str( geo02d[0] ) ))
69
70      print("--- Compute intersection of (region) geo02 and (lines) geo03 and geo04 ...")
71      geo02e = geo02.intersection(geo03)
72      print("---   geo02.intersection(geo03) --> {:s} ...".format( str( geo02e[0] ) ))
73      geo02f = geo02.intersection(geo04)
74      print("---   geo02.intersection(geo04) --> {:s} ...".format( str( geo02f[0] ) ))
75
76      print("--- Part 08: Plot polygons ... ");
77
78      ax = geo01.plot( color='blue', edgecolor='black')
79      ax.set_aspect('equal')
80      ax.set_title("Test Polygons and LineStrings")
```

Example 1: Manual Specification of Geometric Shapes

Part I: Problem Setup (Continued) ....

```
81
82      # Plot polygons ...
83
84      geo01.plot(ax=ax, edgecolor='blue', color='red',   alpha= 1.0 )
85      geo02.plot(ax=ax, edgecolor='blue', color='green', alpha= 0.5 )
86
87      # Plot linestring ...
88
89      geo03.plot(ax=ax, color='blue',   alpha= 1.0, linewidth=3.0, linestyle='dashdot' )
90      geo04.plot(ax=ax, color='maroon', alpha= 1.0, linewidth=3.0, linestyle='dashed' )
91
92      plt.xlabel('x')
93      plt.ylabel('y')
94      plt.grid(True)
95      plt.show()
96
97      print("--- ======================================== ... ");
98      print("--- Leave TestGeoSeries01.main()             ... ");
99
100 # ================================================
101 # call the main method ...
102 # ================================================
103
104 main()
```

Source Code: See: python-code.d/geopandas/

Example 1: Manual Specification of Geometric Shapes

Part II: Abbreviated Output:

```
--- Enter TestGeoSeries01.main()              ...
--- Part 01: Create individual polygons ...
--- Part 02: Add polygons to GeoSeries ...
--- Part 03: Create simple linestring GeoSeries ...

--- Part 04: Print GeoSeries info and contents ...

0    POLYGON ((0.00000 0.00000, 10.00000 0.00000, 1...
1    POLYGON ((10.00000 10.00000, 12.00000 10.00000...
2    POLYGON ((12.00000 12.00000, 15.00000 12.00000...
dtype: geometry
0    POLYGON ((14.00000 2.00000, 20.00000 2.00000, ...
dtype: geometry

--- Part 05: Area and boundary of geo01 ...

0    100.0
1      4.0
2      9.0
dtype: float64
0    LINESTRING (0.00000 0.00000, 10.00000 0.00000,...
1    LINESTRING (10.00000 10.00000, 12.00000 10.000...
2    LINESTRING (12.00000 12.00000, 15.00000 12.000...
dtype: geometry
```

Example 1: Manual Specification of Geometric Shapes

Part II: Abbreviated Output:

```
--- Part 06: Area and boundary of geo02 ...

0    48.0
dtype: float64
0    LINESTRING (14.00000 2.00000, 20.00000 2.00000...
dtype: geometry

--- Part 07: Spatial relationship of geo01 through geo04 ...

--- Compute intersection of (lines) geo03 and geo04 ...

---   geo03.intersects(geo04) --> True ...
---   geo03.intersection(geo04) --> MULTIPOINT (5 2, 8 14) ...

--- Compute intersection of (region) geo01 and (lines) geo03 and geo04 ...

---   geo01.intersection(geo03) --> LINESTRING (5 10, 5 1, 10 1) ...
---   geo01.intersection(geo04) --> MULTILINESTRING ((10 2, 10 6), (2 10, 2 2, 10 2), (10 9, 8 9, 8 10))

--- Compute intersection of (region) geo02 and (lines) geo03 and geo04 ...

---   geo02.intersection(geo03) --> LINESTRING (14 4, 18 4, 18 10) ...
---   geo02.intersection(geo04) --> LINESTRING (14 6, 16 6, 16 9, 14 9) ...

--- Part 08: Plot polygons ...
--- Leave TestGeoSeries01.main()              ...
```