

Overview

- 1 What is Python?
 - Origins, Features, Framework for Scientific Computing
- 2 Program Development with Python
 - Working with the Terminal
 - Integrated Development Environments
- 3 Data Types, Variables, Arithmetic Expressions, Program Control, and Functions
- 4 First Program (Evaluate and Plot Sigmoid Function)
- 5 Builtin Collections (Lists, Dictionaries, and Sets)
- 6 Numerical Python (NumPy)
- 7 Tabular Data and Dataset Transformation (Pandas)
- 8 Spatial Data and Dataset Transformation (GeoPandas)

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:

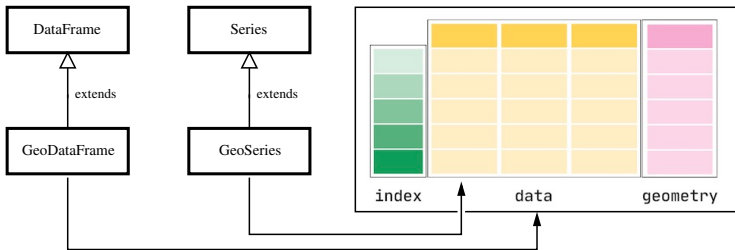
- Extend the datatypes used by Pandas to allow **spatial operations** on **geometric types**.
- Geometric operations are performed by **shapely**.
- Geopandas further depends on **fiona** for **file access** and **matplotlib** for **plotting**.

Installation

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

Working with GeoPandas Dataframes

Core Modeling Concepts and Data Structure:

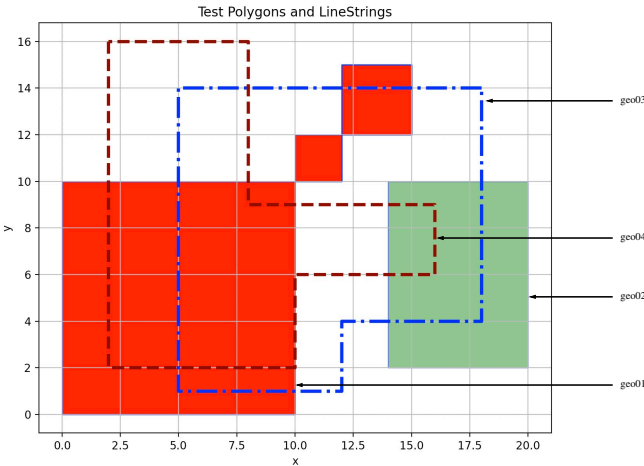


- GeoSeries handle geometries (points, polygons, etc).
- GeoDataFrames store geometry columns and perform spatial operations. They can be assembled from geopandas.GeoSeries.

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
    
```

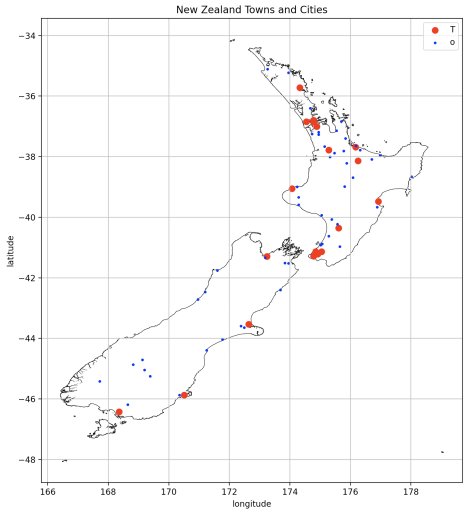
Example 1: Manual Specification of Geometric Shapes

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()      ...
```

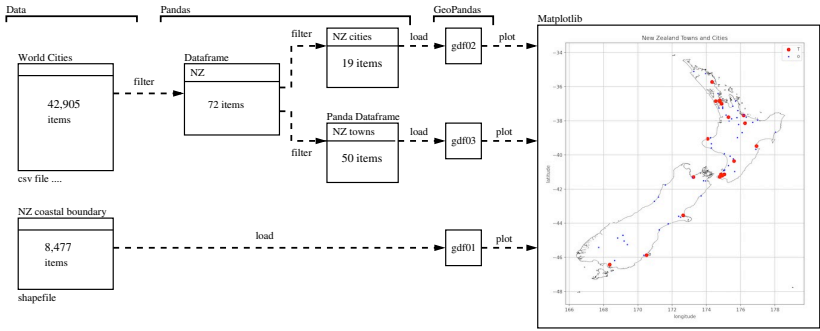
Example 2: Towns and Cities in New Zealand

Example 2: Towns and Cities in New Zealand.



Example 2: Towns and Cities in New Zealand

Part I: Data Processing Pipeline: Use sequence of filters to specialize views of data ...



Example 2: Towns and Cities in New Zealand

Part II: Program Source Code:

```
1 # =====
2 # TestNewZealandDataModel.py. Assemble data model for towns and cities in
3 # New Zealand.
4 #
5 # Written by: Mark Austin February 2023
6 # =====
7
8 from pandas import DataFrame
9 from pandas import Series
10 from pandas import read_csv
11
12 import numpy as np
13 import pandas as pd
14 import geopandas
15
16 import matplotlib.pyplot as plt
17
18 # =====
19 # main method ...
20 # =====
21
22 def main():
23     print("--- Enter TestNewZealandDataModel.main() ... ");
24     print("--- ===== ... ");
25
26     print("--- Part 01: Load world city dataset ... ");
```


Example 2: Towns and Cities in New Zealand

Part II: Program Source Code: (Continued) ...

```
27
28     df = pd.read_csv("../data/cities/world-cities.csv")
29
30     print("--- Part 02: Print dataframe info and contents ... ")
31
32     print(df)
33     print(df.info() )
34
35     print("--- Part 03: Filter dataframe to keep only cities from New Zealand ... ")
36
37     options = ['New Zealand']
38     dfNZ      = df [ df['country'].isin(options) ].copy()
39
40     print("--- Part 04: Filter data to find NZ cities and towns ... ")
41
42     dfNZcities = dfNZ [ (dfNZ['population'] > 40000) ].sort_values( by=['population'] )
43
44     dfNZtowns  = dfNZ [ (dfNZ['population'] > 1000) & (dfNZ['population'] < 40000) ]
45     dfNZtowns  = dfNZtowns.sort_values( by=['population'] )
46
47     print('--- New Zealand Cities:\n', dfNZcities )
48     print('--- New Zealand Towns:\n', dfNZtowns )
49
50     print("--- Part 05: Read NZ coastline shp file into geopandas ... ")
51
52     nzboundarydata = geopandas.read_file("../data/geography/nz/Coastline02.shp")
53     print(nzboundarydata)
```

Example 2: Towns and Cities in New Zealand

Part II: Program Source Code: (Continued) ...

```
55     print("--- Part 06: Define geopandas dataframes ... ")
56
57     gdf01 = geopandas.GeoDataFrame(nzboundarydata)
58     gdf02 = geopandas.GeoDataFrame( dfNZcities ,
59         geometry=geopandas.points_from_xy(dfNZcities.lng, dfNZcities.lat))
60     gdf03 = geopandas.GeoDataFrame( dfNZtowns ,
61         geometry=geopandas.points_from_xy( dfNZtowns.lng, dfNZtowns.lat))
62
63     print(gdf01.head())
64
65     print("--- Part 07: Create boundary map for New Zealand ... ")
66
67     # We can now plot our 'GeoDataFrame'.
68
69     ax = gdf01.plot( color='white', edgecolor='black')
70     ax.set_aspect('equal')
71     ax.set_title("New Zealand Towns and Cities")
72
73     gdf01.plot(ax=ax, color='white')
74
75     gdf02.plot(ax=ax, color = 'red', markersize = 50, label= 'Cities')
76     gdf03.plot(ax=ax, color = 'blue', markersize = 5, label= 'Towns' )
77
78     plt.legend('Towns/Cities:')
79     plt.xlabel('longitude')
80     plt.ylabel('latitude')
```

Example 2: Towns and Cities in New Zealand

Part II: Program Source Code: (Continued) ...

```
81     plt.grid(True)
82     plt.show()
83
84     print("--- ===== ... ");
85     print("--- Leave TestNewZealandDataModel.main() ... ");
86
87     # =====
88     # call the main method ...
89     # =====
90
91     main()
```

Source Code: See: [python-code.d/geopandas/](#)

Example 2: Towns and Cities in New Zealand

Part III: Abbreviated Output:

```
--- Enter TestNewZealandDataModel.main()      ...
--- ===== ...
--- Part 01: Load world city dataset ...
--- Part 02: Print dataframe info and contents ...
      city  city_ascii  lat  ...  capital  population  id
0      Tokyo      Tokyo  35.6839  ...  primary  39105000.0  1392685764
1      Jakarta      Jakarta  -6.2146  ...  primary  35362000.0  1360771077
...
42903 Timmiarmiut Timmiarmiut  62.5333  ...    NaN      10.0  1304206491
42904 Nordvik      Nordvik  74.0165  ...    NaN      0.0  1643587468
[42905 rows x 11 columns]
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42905 entries, 0 to 42904
Data columns (total 11 columns):
```

#	Column	Dtype	#	Column	Dtype
0	city	object	6	iso3	object
1	city_ascii	object	7	admin_name	object
2	lat	float64	8	capital	object
3	lng	float64	9	population	float64
4	country	object	10	id	int64
5	iso2	object			

```
dtypes: float64(3), int64(1), object(7)
memory usage: 3.6+ MB
```

Example 2: Towns and Cities in New Zealand

Part III: Abbreviated Output (Continued) ...

--- Part 03: Filter dataframe to keep only cities from New Zealand ...
--- Part 04: Filter data to find NZ cities and towns ...

```
--- New Zealand Cities:
```

	city	city_ascii	...	population	id
14169	Upper Hutt	Upper Hutt	...	41000.0	1554000042
6159	Invercargill	Invercargill	...	47625.0	1554148942
.....					
741	Wellington	Wellington	...	418500.0	1554772152
516	Auckland	Auckland	...	1346091.0	1554435911

[19 rows x 11 columns]

```
--- New Zealand Towns:
```

	city	city_ascii	...	population	id
42142	Kaikoura	Kaikoura	...	2210.0	1554578431
.....					
14309	Whanganui	Whanganui	...	39400.0	1554827998

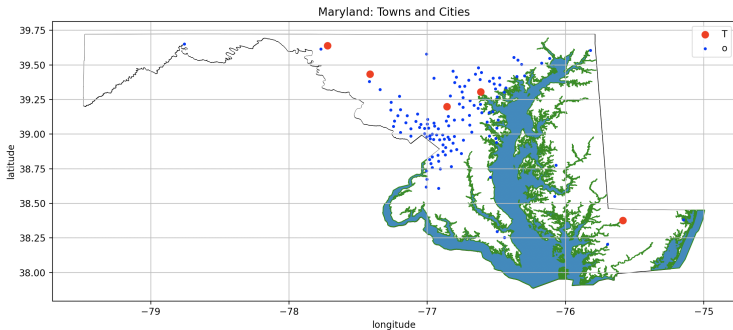
[50 rows x 11 columns]

```
--- Part 05: Read NZ coastline shp file into geopandas ...
0 POLYGON ((174.00369 -40.66489, 174.00372 -40.6...
.....
8476 POLYGON ((173.01384 -34.39348, 173.01395 -34.3...
[8477 rows x 1 columns]
```

--- Part 07: Create boundary map for New Zealand ...
--- ===== ...
--- Leave TestNewZealandDataModel.main() ...

Example 3: Towns and Cities in Maryland

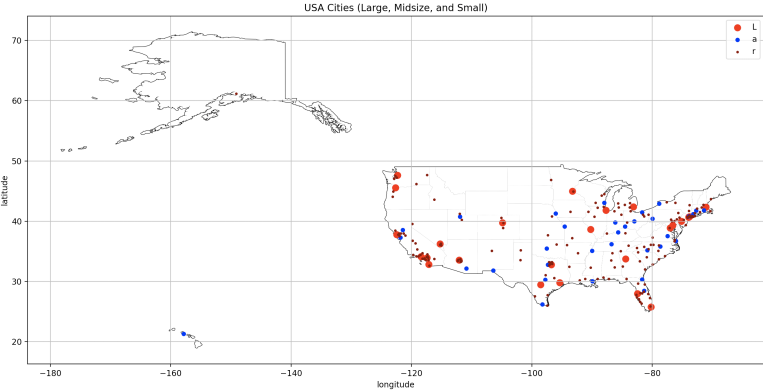
Example 3: Towns and Cities in Maryland.



Cities: Columbia (pop. 103991), Salisbury (pop. 106447), Frederick (pop. 156787), Hagerstown (pop. 184755), Baltimore (pop. 2106068).

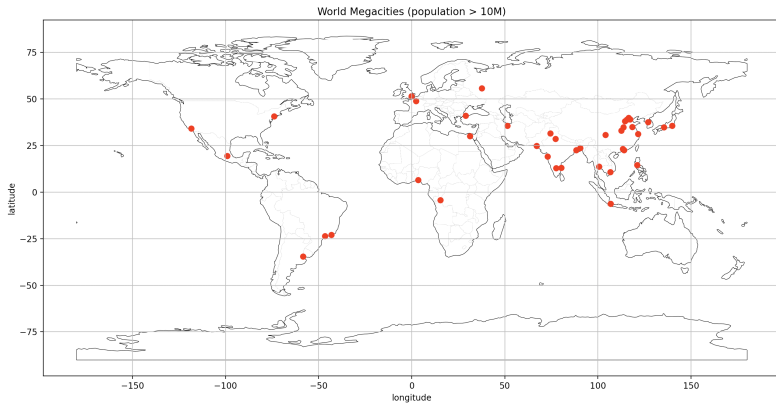
Example 4: Large, Midsize, and Small US Cities

Example 4: Large, Midsize, and Small US Cities



Cities: 26 large (pop. > 2M), 34 midsize (800k < pop. < 2M), 172 small (200k < pop. < 800k).

Example 5: The World's Megacities



Example 5: The World's Megacities

--- Part 02: Filter to keep only large cities (pop. > 10M) ...

	city	city_ascii	...	population	id
0	Tokyo	Tokyo	...	39105000.0	1392685764
1	Jakarta	Jakarta	...	35362000.0	1360771077
2	Delhi	Delhi	...	31870000.0	1356872604
3	Manila	Manila	...	23971000.0	1608618140
4	São Paulo	Sao Paulo	...	22495000.0	1076532519
5	Seoul	Seoul	...	22394000.0	1410836482
6	Mumbai	Mumbai	...	22186000.0	1356226629
7	Shanghai	Shanghai	...	22118000.0	1156073548
8	Mexico City	Mexico City	...	21505000.0	1484247881
9	Guangzhou	Guangzhou	...	21489000.0	1156237133
10	Cairo	Cairo	...	19787000.0	1818253931
11	Beijing	Beijing	...	19437000.0	1156228865
12	New York	New York	...	18713220.0	1840034016
13	Kolkāta	Kolkata	...	18698000.0	1356060520
14	Moscow	Moscow	...	17693000.0	1643318494
15	Bangkok	Bangkok	...	17573000.0	1764068610

... details removed ...

33	London	London	...	11120000.0	1826645935
34	Paris	Paris	...	11027000.0	1250015082
35	Tianjin	Tianjin	...	10932000.0	1156174046
36	Linyi	Linyi	...	10820000.0	1156086320
37	Shijiazhuang	Shijiazhuang	...	10784600.0	1156217541
38	Zhengzhou	Zhengzhou	...	10136000.0	1156183137
39	Nanyang	Nanyang	...	10013600.0	1156192287