

Polygon, Point, LineString e MultiPolygon

Vimos na última aula como a coluna `geometry` de um **GeoDataFrame** é construída. Utilizando o conhecimento aprendido, julgue os itens abaixo. Observe que cada alternativa apresenta um código Python e a figura geométrica gerada por ele.

Para testar os códigos abaixo, não se esqueça de importar o `geopandas` e o `shapely`, como fizemos na última aula:

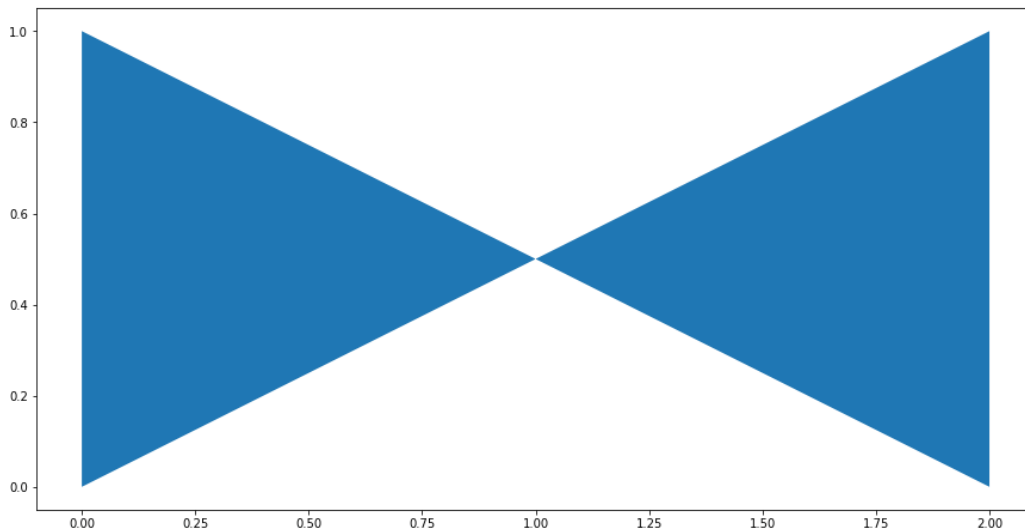
```
import geopandas as gpd
from shapely.geometry import Polygon, Point, LineString, MultiPolygon
```

Selecione 3 alternativas

A**In [1]:**

```
p1 = Polygon([(0, 0), (2, 1), (2, 0), (0, 1)])

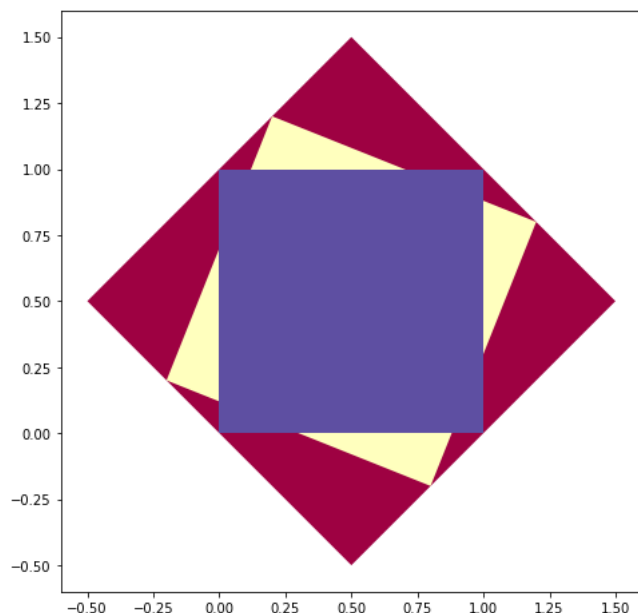
g = gpd.GeoSeries([p1])
g.plot(cmap='tab10', figsize=(15,8))
```

Out [1]:**B****In [1]:**

```
figuras = []
for i in [0.5, 0.2, 0]:
    figuras.append(Polygon([(0 - i, 0 + i),
                           (1 - i, 0 - i),
                           (1 + i, 1 - i),
                           (0 + i, 1 + i)]))
```

```
g = gpd.GeoSeries(figuras)
g.plot(cmap='Spectral', figsize=(15,8))
```

Out [1]:



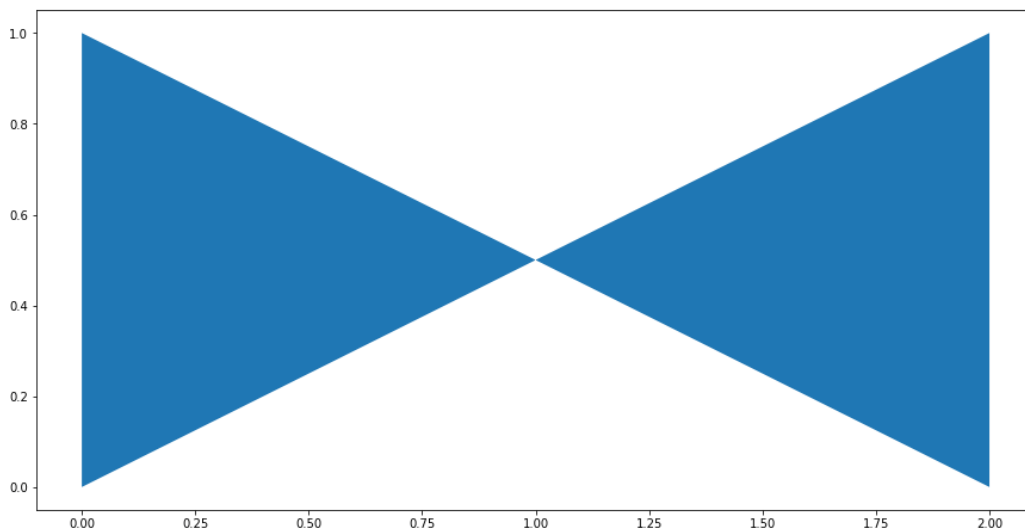
C

In [1]:

```
p1 = Polygon([(0, 0), (1, 0.5), (0, 1)])
p2 = Polygon([(2, 0), (1, 0.5), (2, 1)])

g = gpd.GeoSeries([p1, p2])
g.plot(cmap='tab10', figsize=(15,8))
```

Out [1]:



D

In [1]:

```
p1 = Point(0.5, 0.5).buffer(0.5)
p2 = MultiPolygon([Point(0.25, 0.6).buffer(0.1),
                    Point(0.75, 0.6).buffer(0.1)])
p3 = MultiPolygon([Point(0.25, 0.6).buffer(0.02),
                    Point(0.75, 0.6).buffer(0.02)])
p4 = LineString([(0.2, 0.25), (0.8, 0.25)])

g = gpd.GeoSeries([p1, p2, p3, p4])
g.plot(cmap='Wistia', figsize=(15,8))
```

Out [1]:

