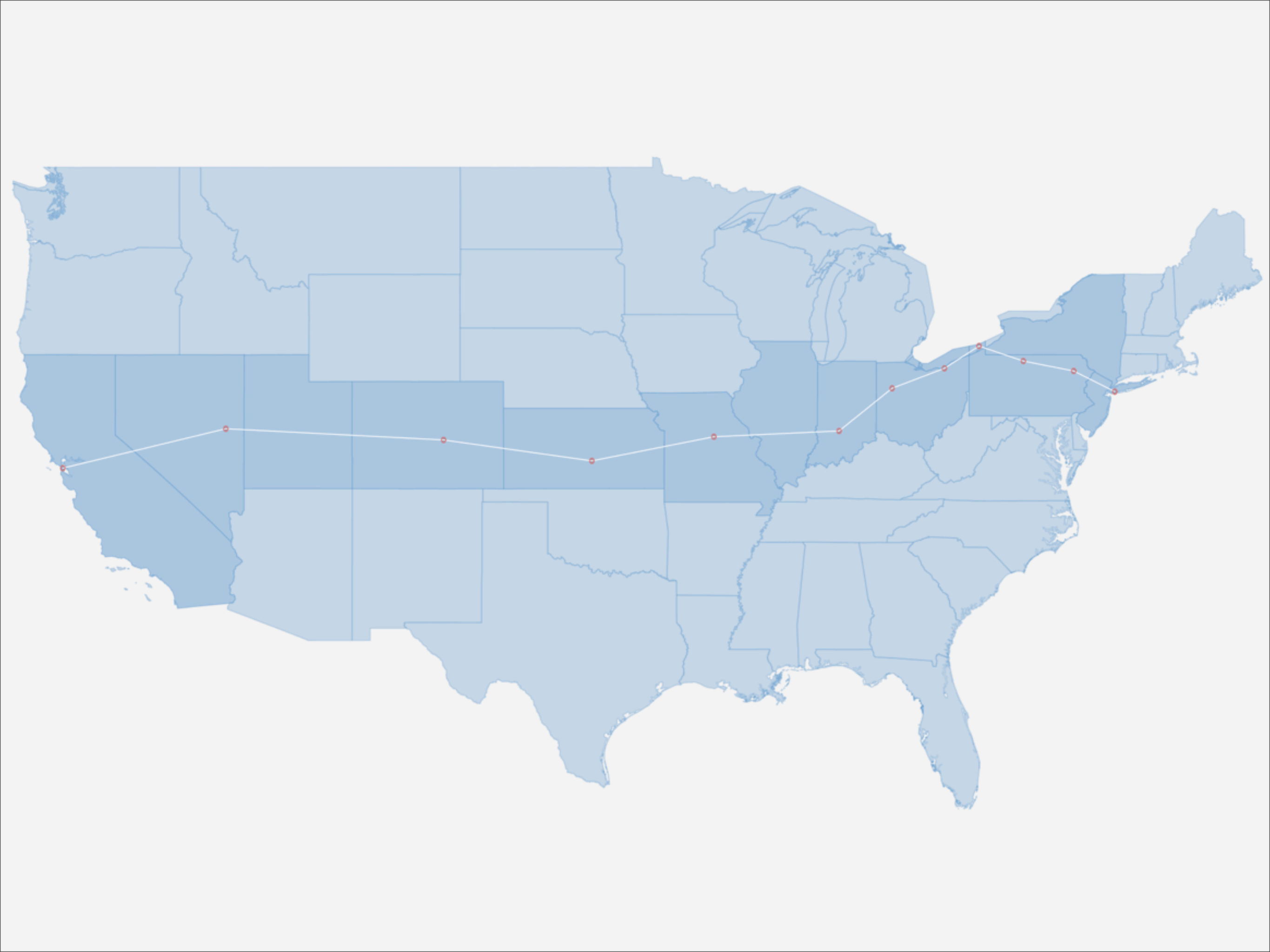


# GIS in Python Using Shapely

A talk by Zain Memon  
@ PyData NYC 2012



# Shapely is:

- "Swiss army knife" spatial analysis tool
- Pythonic way to access GEOS
- PostGIS operations in Python

# Shapely doesn't:

- Read/write GIS data formats  
(Fiona does)
- Display or render geodata  
(Descartes does)

# Point Set Theory

- Three data types: **points**, **curves**, and **surfaces**
- Each has three properties: an **interior** set, a **boundary** set, and an **exterior** set

# Points

```
from shapely.geometry import Point
```

```
# 2d geometry
```

```
pt = Point(x, y)
```

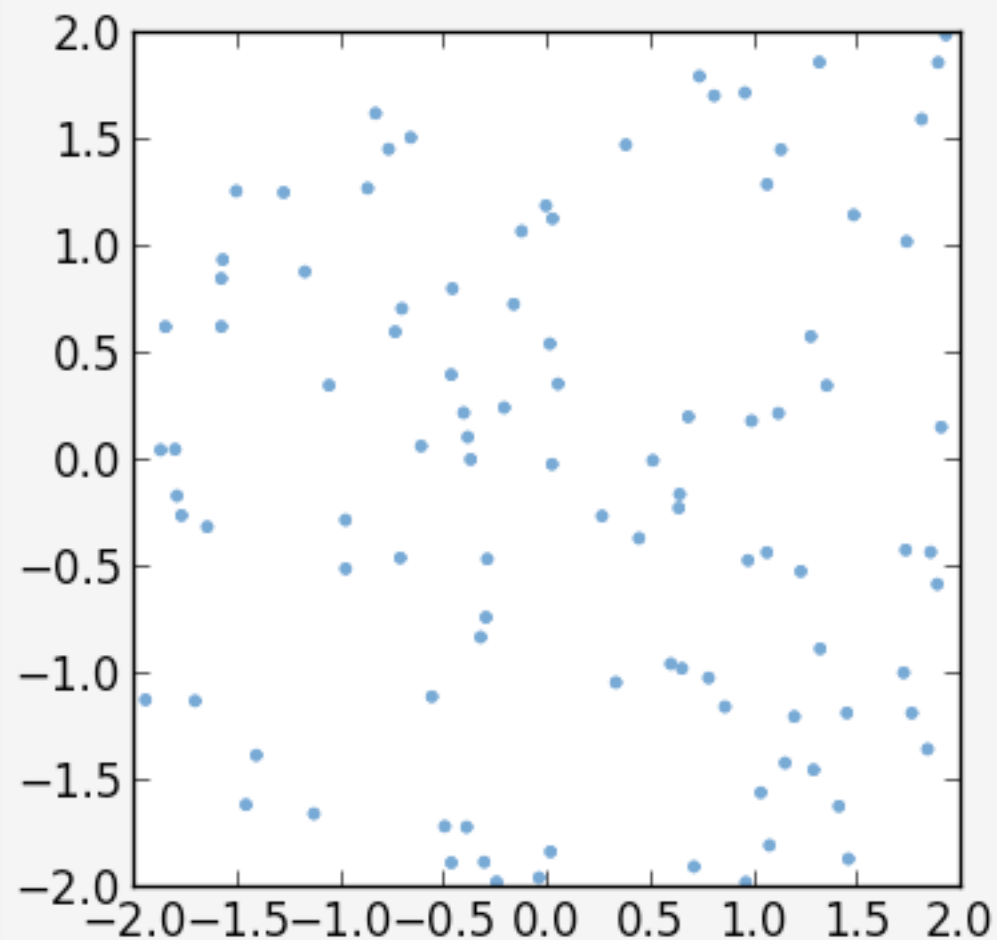
```
# 3d geometry
```

```
pt = Point(x, y, z)
```

```
# geography
```

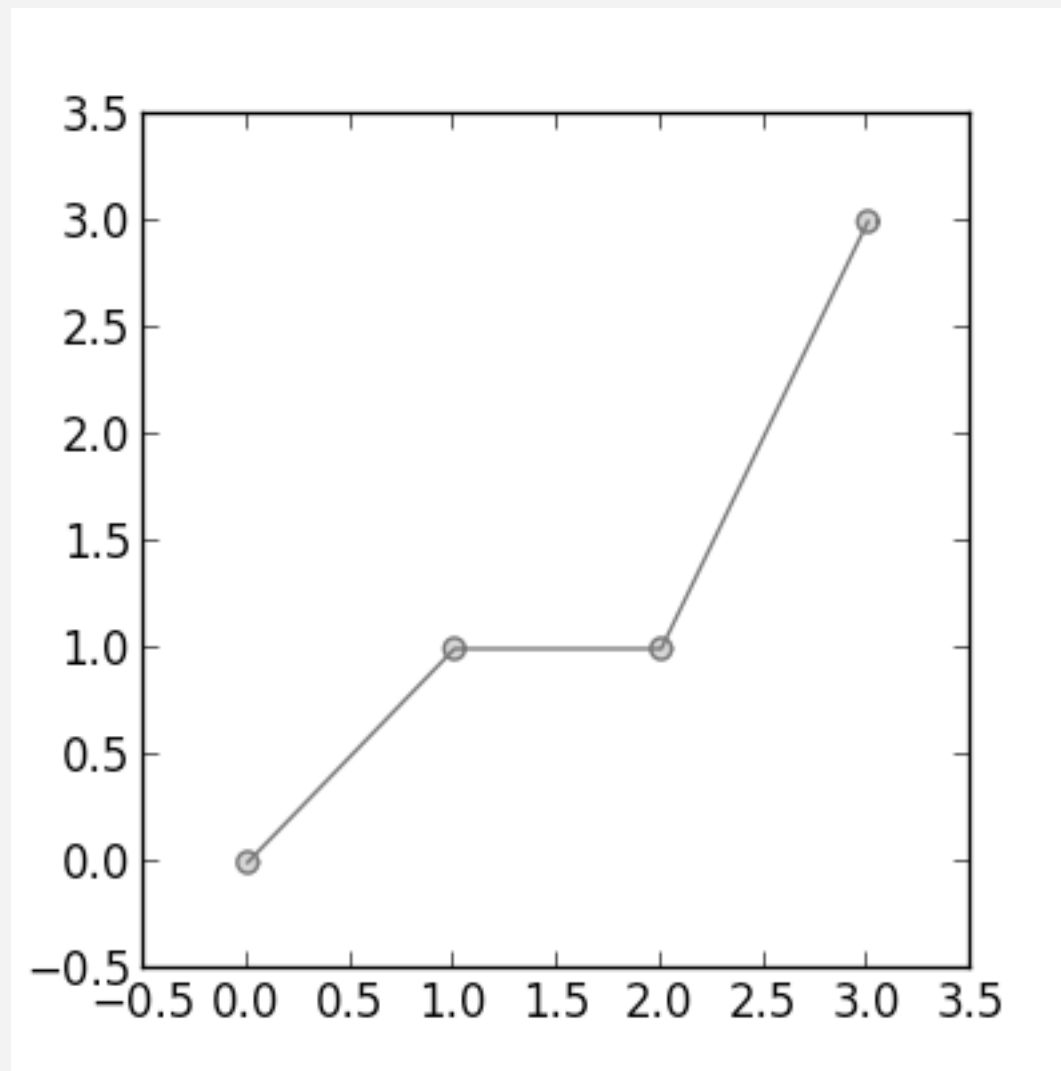
```
pt = Point(longitude, latitude)
```

# Points



```
[Point(random.uniform(-2,2),  
random.uniform(-2,2)) for i in  
range(100)]
```

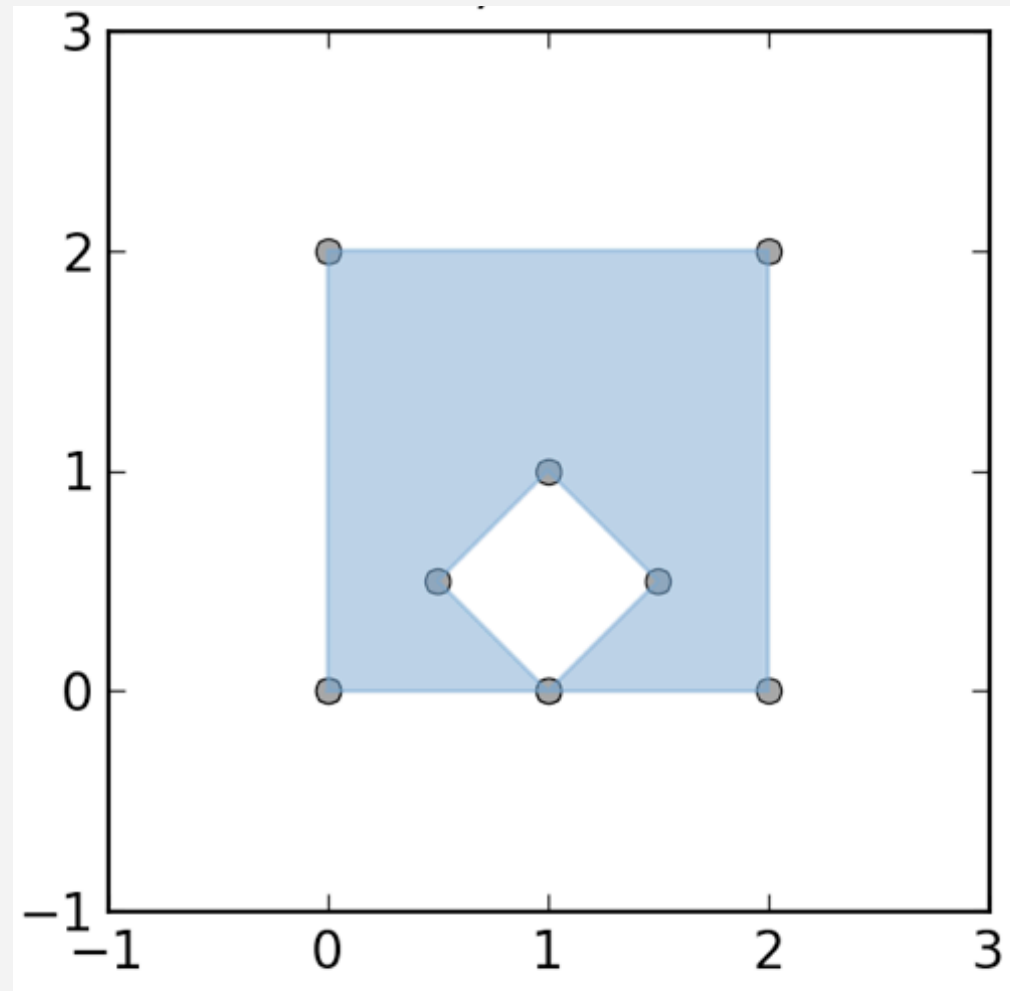
# LineString



```
LineString([(0,0), (1, 1), (2, 1), (3,3)])
```



# Polygons

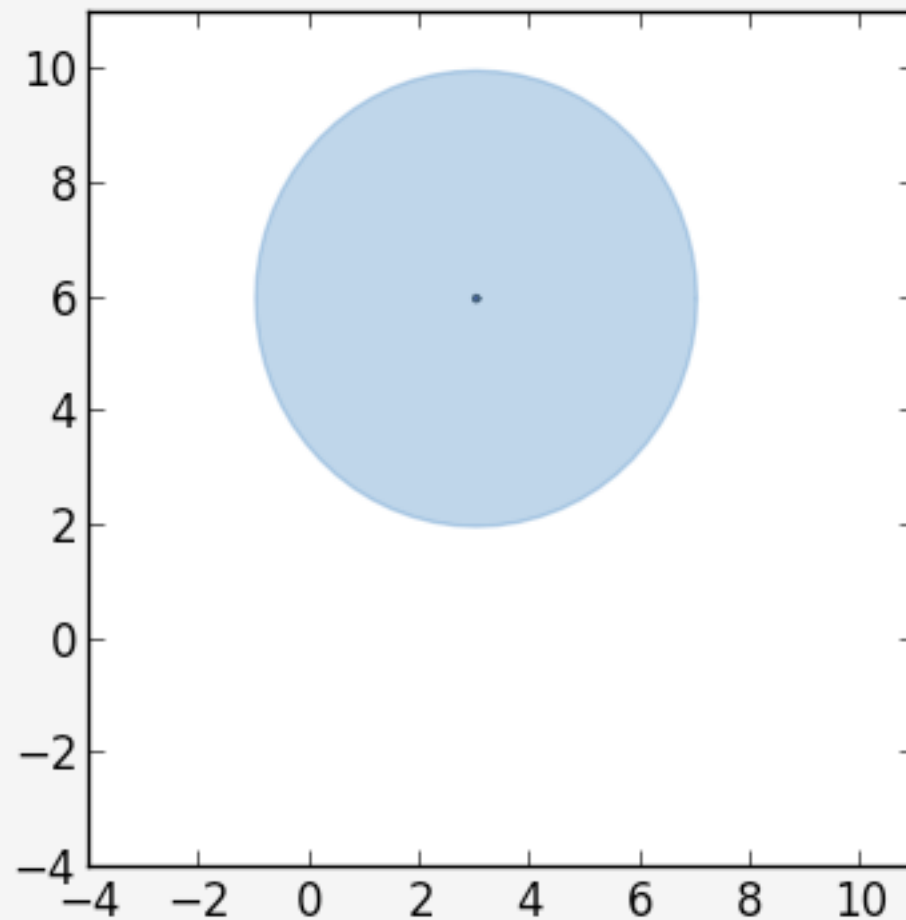


```
ext = [(0, 0), (0, 2), (2, 2), (2, 0), (0, 0)]  
int = [(1, 0), (0.5, 0.5), (1, 1), (1.5, 0.5), (1, 0)]  
polygon = Polygon(ext, [int])
```

# Collections

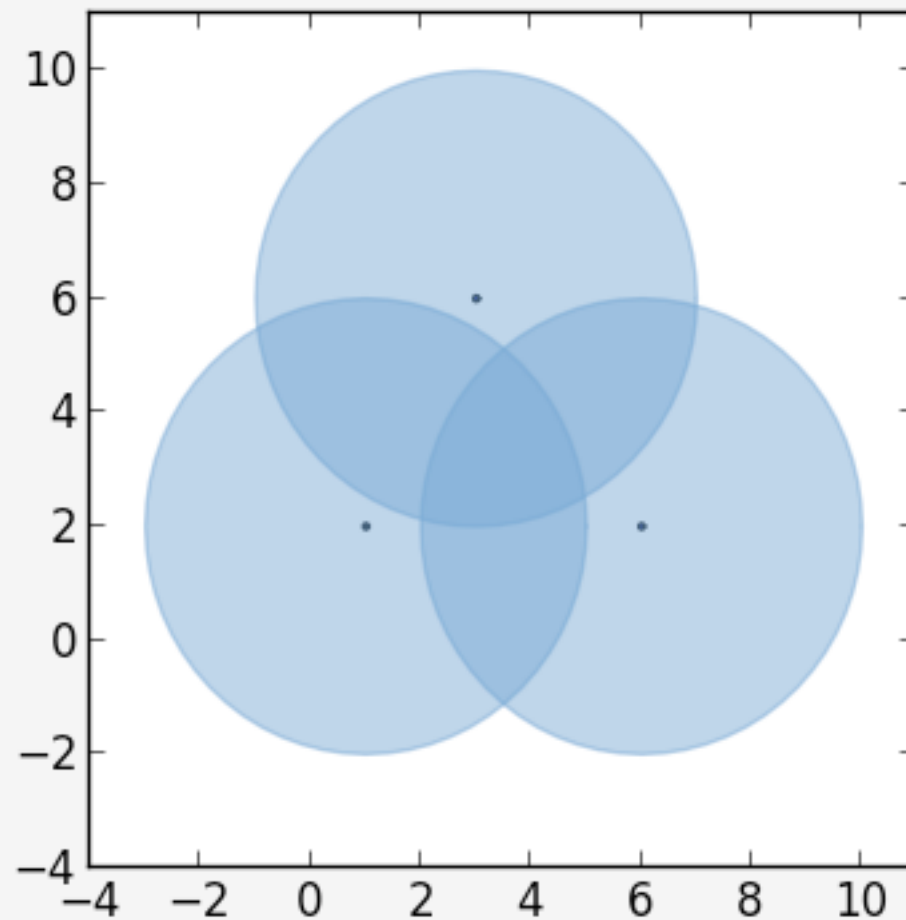
MultiPoints,  
MultiLineStrings,  
& MultiPolygons

# Buffering



`Point(3,6).buffer(4)`

# Buffering

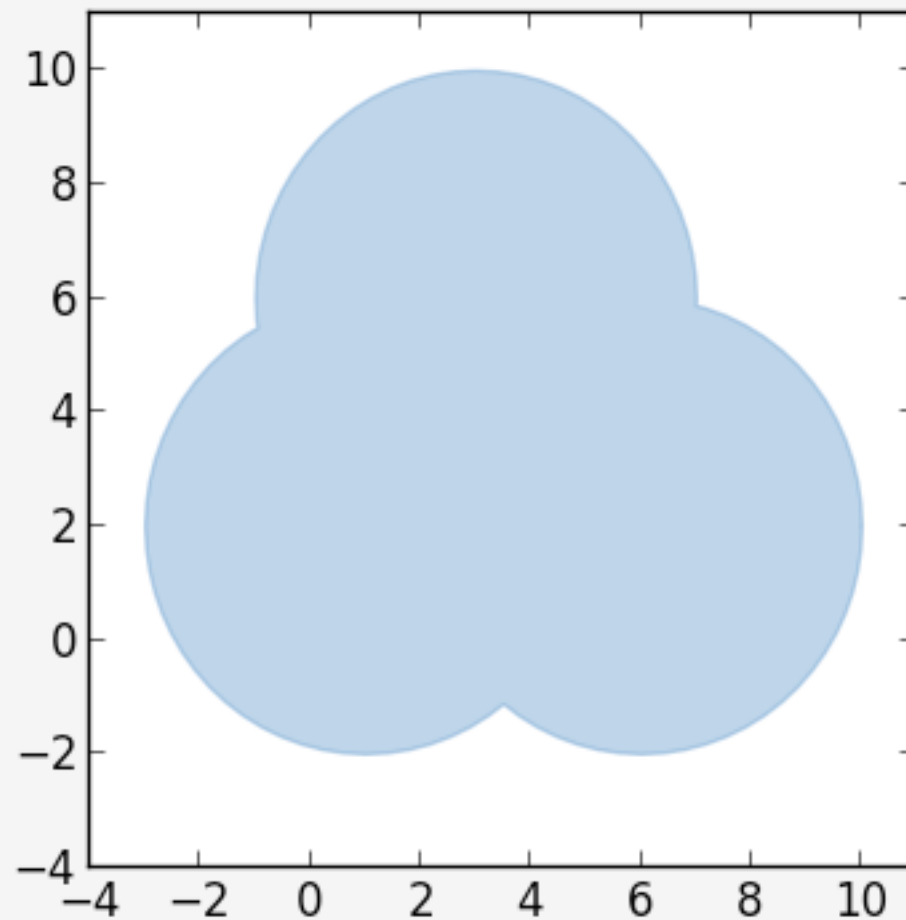


`Point(3,6).buffer(4)`

`Point(6,2).buffer(4)`

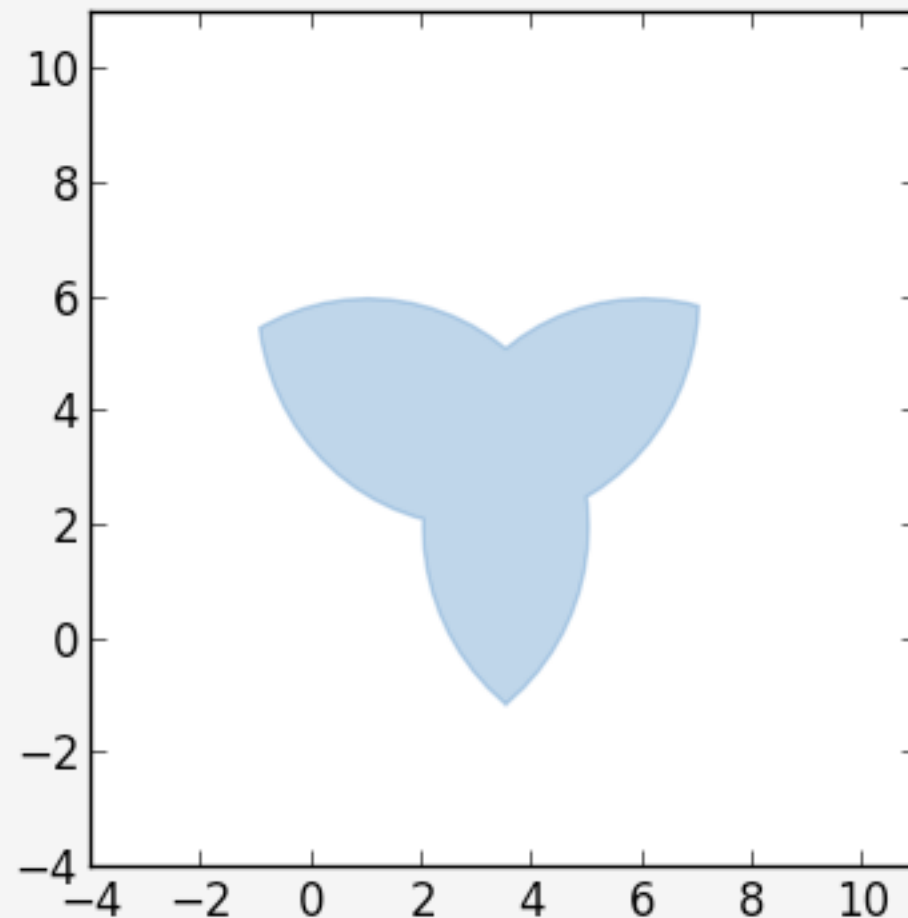
`Point(1,2).buffer(4)`

# Union



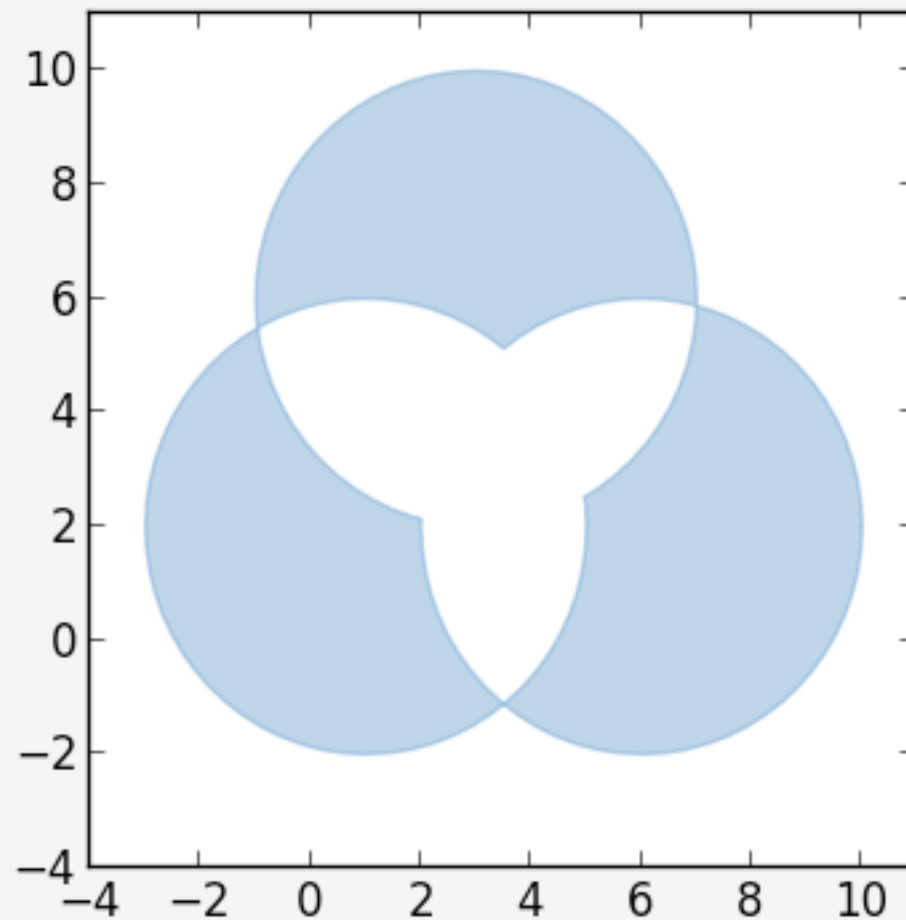
```
shapely.ops.cascaded_union(circles)
```

# Intersection



```
c1.intersection(c2) +  
c2.intersection(c3) +  
c3.intersection(c1)
```

# Difference



`union_shape.difference(inter_shape)`

# Binary Predicates

`geom.contains(other_geom)`

`geom.crosses(other_geom)`

`geom.disjoint(other_geom)`

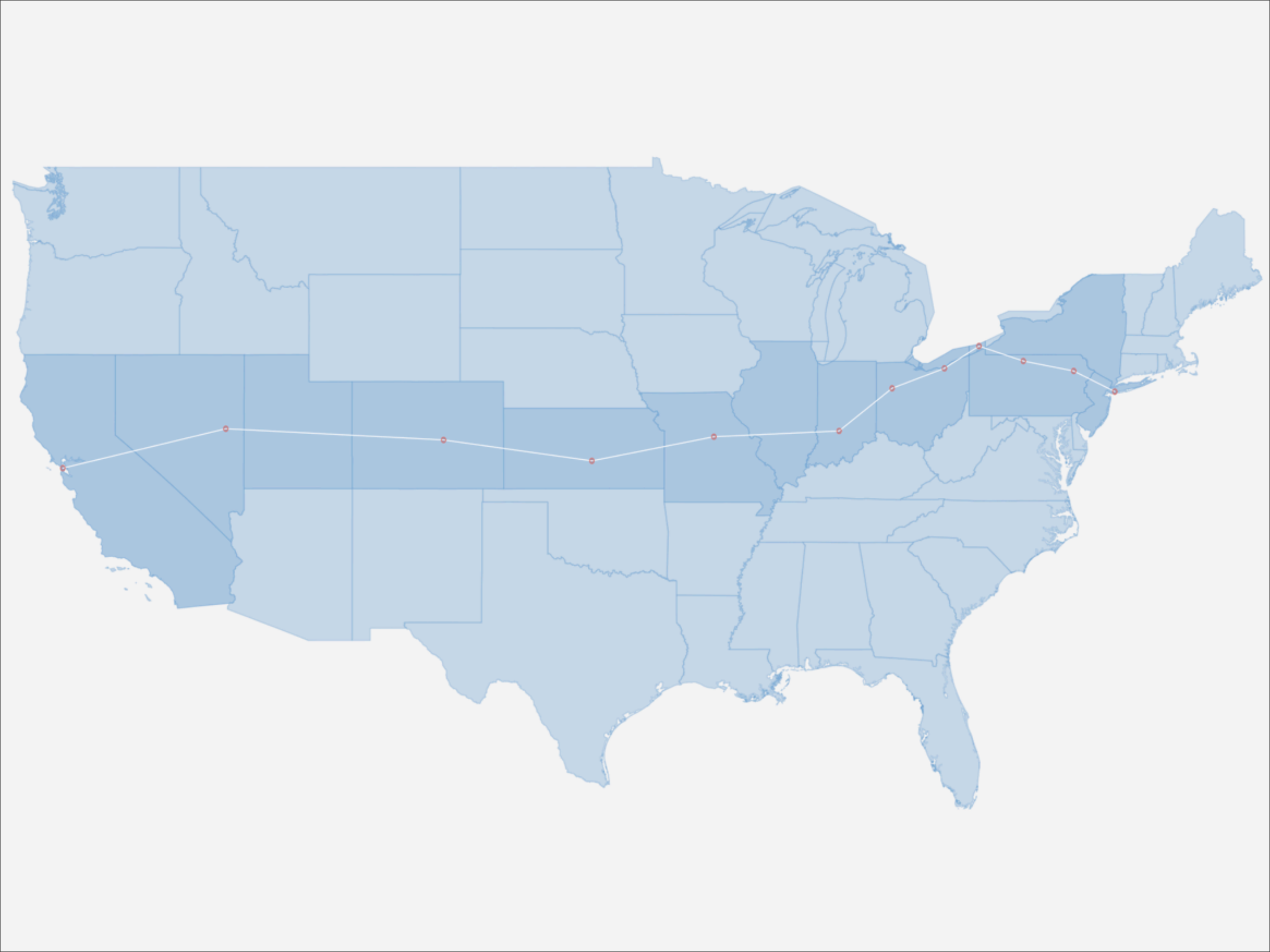
`geom.equals(other_geom)`

`geom.intersects(other_geom)`

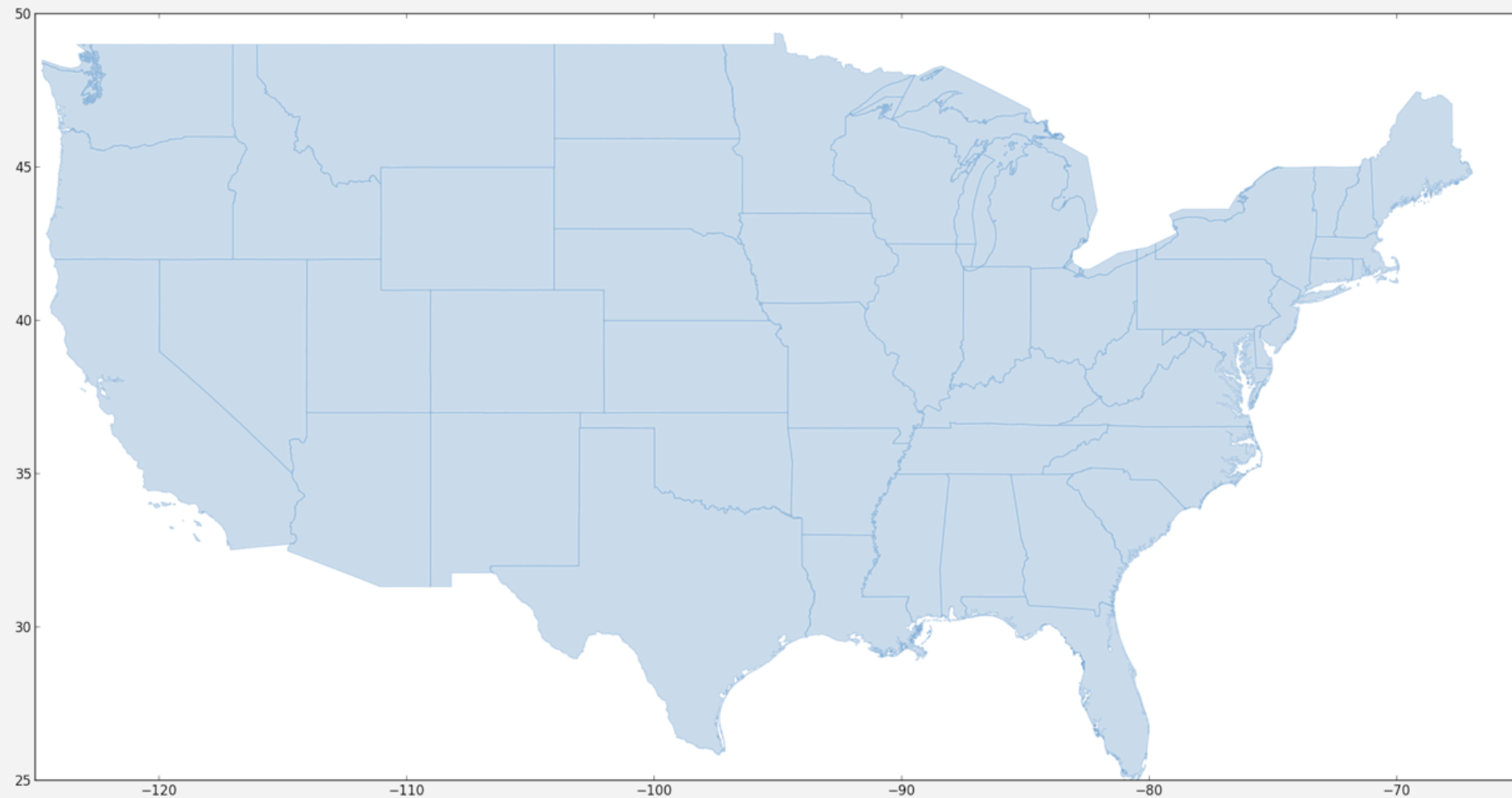
`geom.touches(other_geom)`

`geom.within(other_geom)`





# Data Sources



```
for f in fiona.collection("states.shp"):  
    state_shape = shapely.geometry.shape(f['geometry'])
```

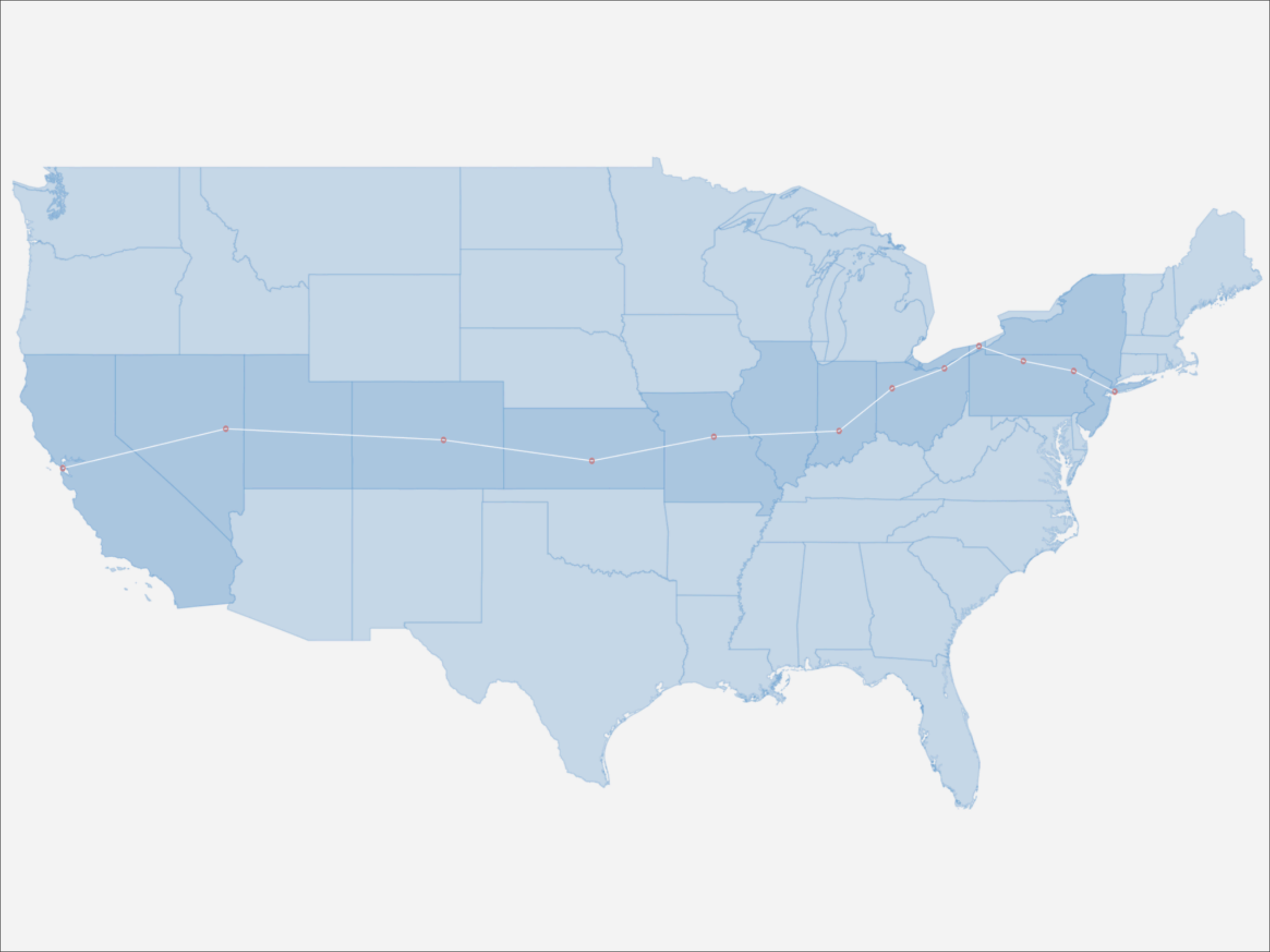
# Line Segment

```
latlons = [(37.766, -122.43), ...]  
path = [(x, y) for y, x in latlons]  
ls = LineString(path)
```

# Coloring

```
for f in fiona.collection("states.shp"):
    state_shape = shapely.geometry.shape(f['geometry'])

    alpha = 0.5
    if ls.intersects(state_shape):
        alpha += 0.2
```



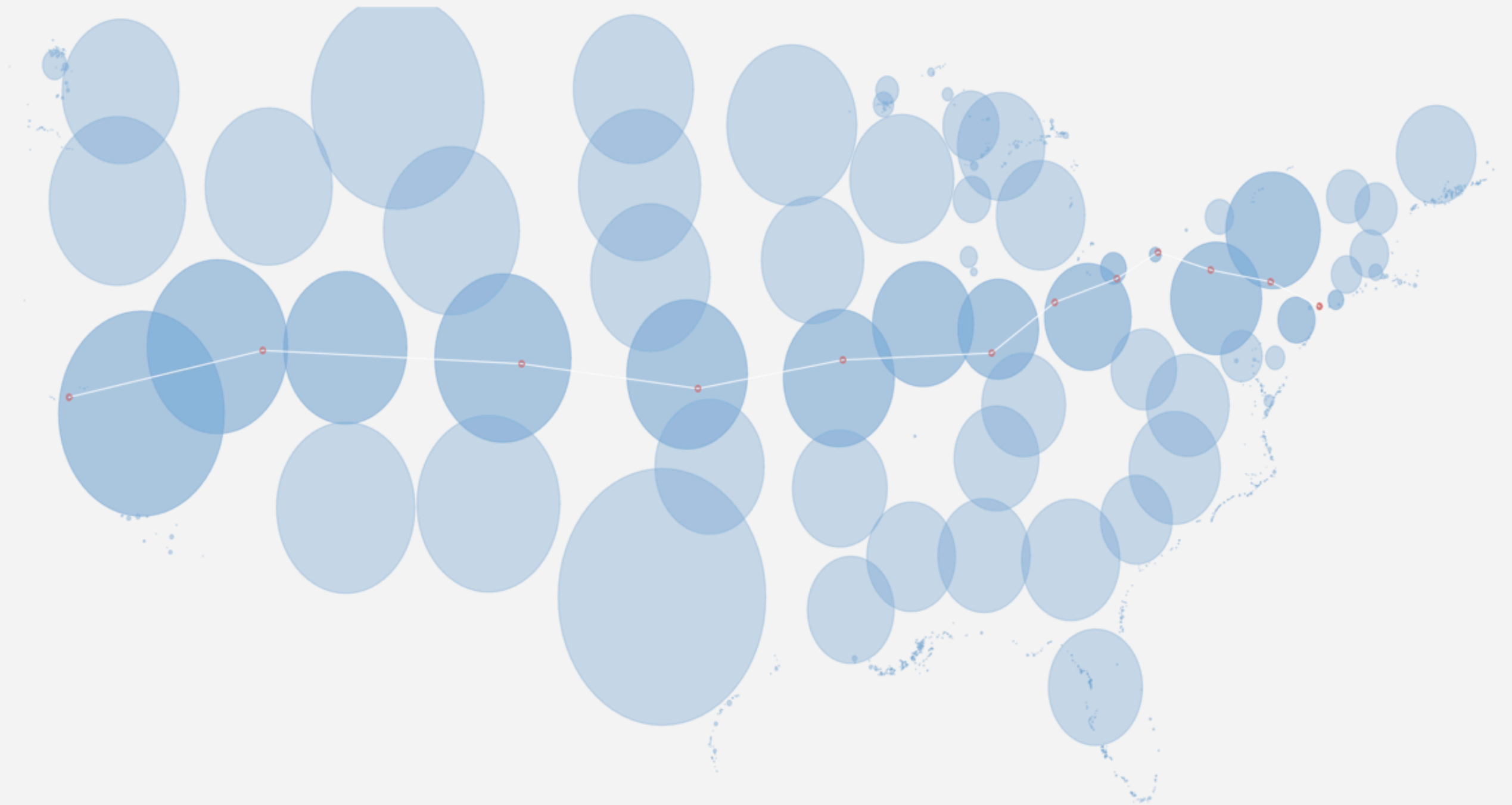
# General Attributes

`geom.area`

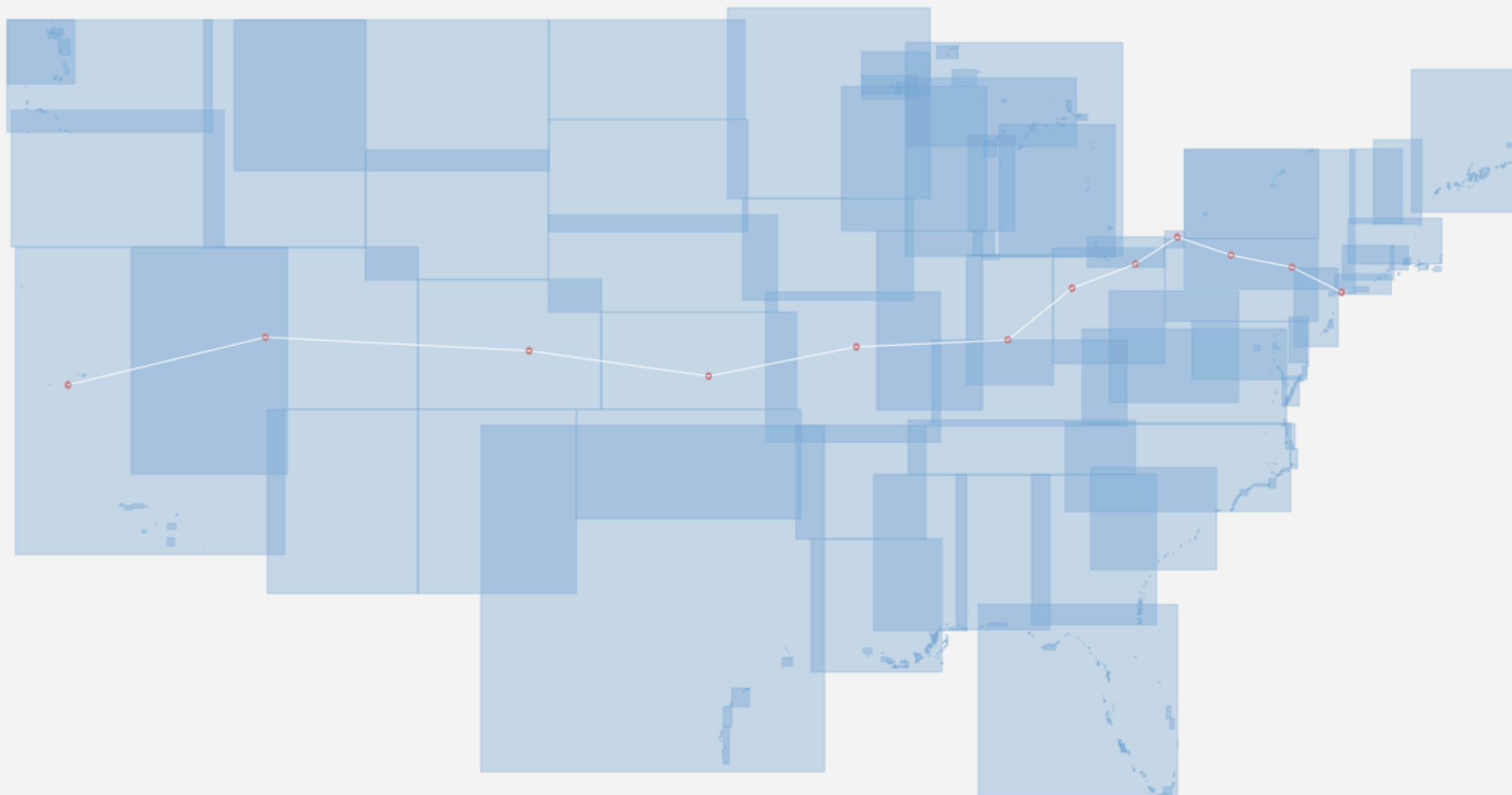
`geom.bounds`

`geom.length`

`geom.distance(to_geom)`

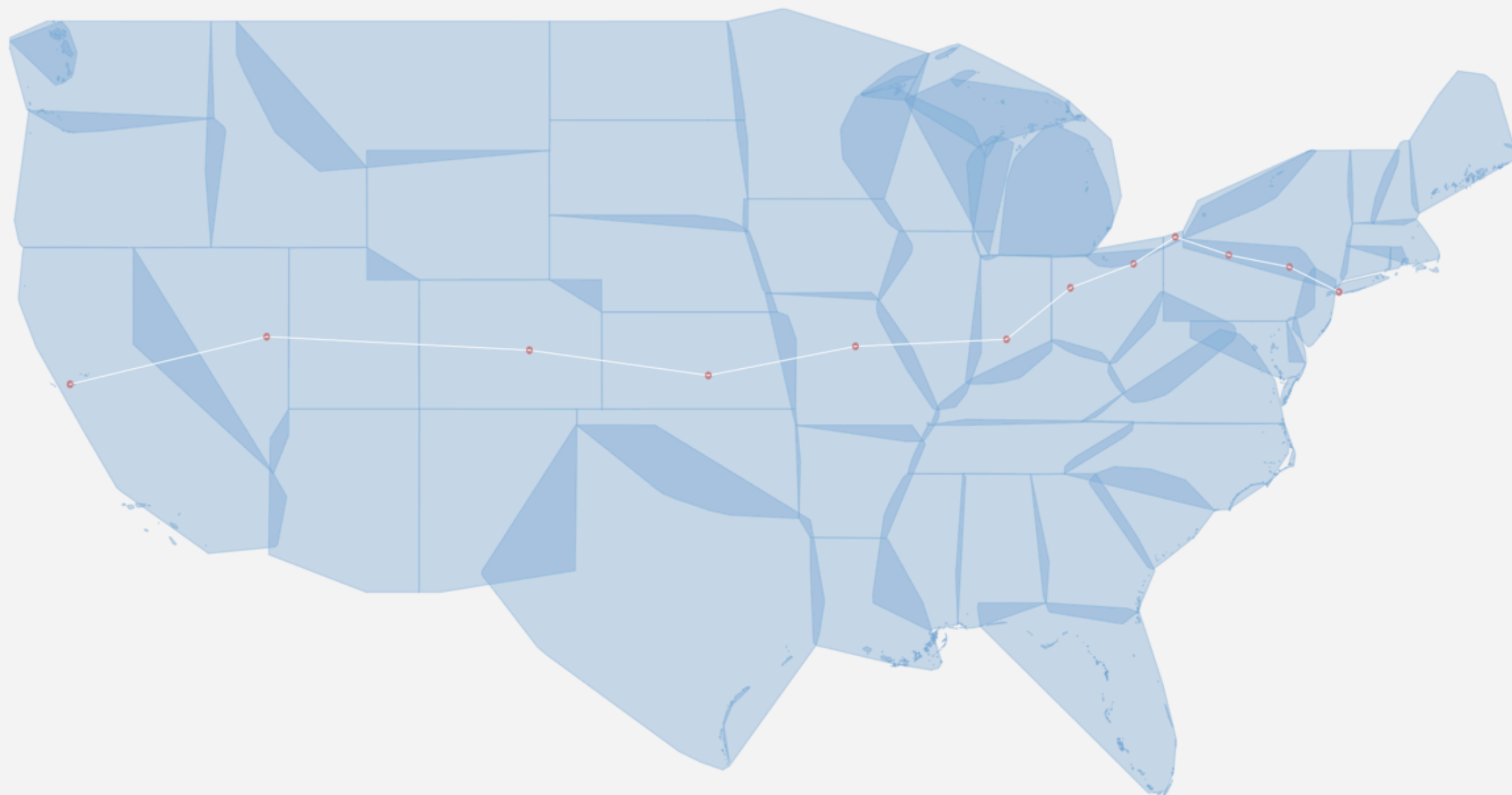


```
poly.centroid.buffer(math.sqrt(poly.area) / 2)
```



`poly.envelope`





`poly.convex_hull`

# Handy Libraries

- Shapely
- Fiona
- Decartes
- PyProj
- RTree
- GeoDjango
- TileStache

This has been

# GIS in Python Using Shapely

Zain Memon <[zain@inzain.net](mailto:zain@inzain.net)>

Slides to be posted @ [github.com/zain](https://github.com/zain)