

Geographic Information Systems: GeoPandas

Jamie Saxon

Introduction to Programming for Public Policy

November 27, 2017

People love maps – emotional response to ‘seeing yourself.’

- ▶ This is a huge field: there are entire classes at Harris and around the University for GIS, spatial statistics, etc.
- ▶ But huge bang for the buck at the entry level.
 - ▶ Easy to make compelling graphics.
 - ▶ Many datasets represent a spatial area or point at a specific time.
⇒ Great potential for joins!

1. Making simple maps with GeoPandas (pandas+).
 - ▶ Finding and importing shapefiles and geojson (like `read_csv()`).
 - ▶ Projections (briefly).
2. Attribute and spatial joins.
 - ▶ Geocoding APIs: geopy/nominatim, Census, etc.
 - ▶ Making a map with real data!
3. Making a simple web (!) map with GeoPandas
 - ▶ Largely revisiting old material, with new functions.

Shapefiles

- ▶ Three forms of geographic objects: points (schools, crimes), lines (roads, rivers), and polygons (lots, census tracts, regions, lakes, etc.).
- ▶ Many, many sources for geographic data: [Chicago](#), [US Census](#), [USGS](#), [Global Administrative Areas](#) (GADM), etc.
- ▶ Much of this is provided in 'ESRI Shapefiles' (Environmental Systems Research Institute, major GIS company) or in geojson. Modern databases (postgres) are helpful for assembling large datasets.
 - ▶ Shapefiles come zipped with a lot of other files. The shp file is the 'master' file, and references the others. That's what you import.
 - ▶ Let's browse: [census shapefiles](#).
- ▶ Addresses may be geocoded and coordinates are also points!

Loading a Shapefile with GeoPandas

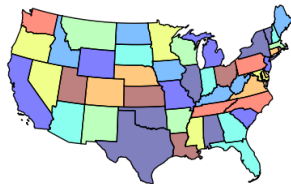
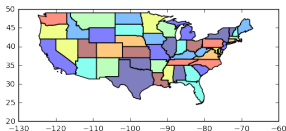
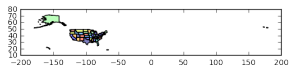
- ▶ **GeoPandas** simply adds a geometry series to a pandas DataFrame.
- ▶ It is tightly integrated with many other geographic programs, among them **fiona** for reading geojson/shapefiles and **shapely** for geometric operations (intersections, etc.).
- ▶ Really easy to import! Both shapefiles and geojson:

```
import geopandas as gpd
gdf = gpd.read_file("myfile.shp")
gdf.plot() # WOW!!!!
```

- ▶ All of the 'standard' dataframe operations (slicing, indexing, merging) are still available.

Making a Better Map

- ▶ Let's restrict ourselves to the contiguous 48 states.
- ▶ Make a mask to get rid of Alaska and Hawaii (STATEFP 2 and 15), and the territories (STATEFP ≥ 57).
 - ▶ Alternative: translate, rotate, and scale them with shapely.
- ▶ We also need a better better projection: `gpd.to_crs(eps=2163)`.



Coordinate Reference Systems (CRS)

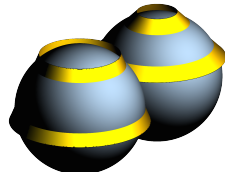
- ▶ To make maps, we need a description of the shape of Earth (an ellipsoid) and an origin/center. This is called a **datum**.
- ▶ We also need a **projection** from 3D to 2D.
- ▶ These are standardized in **EPSG codes**:
 - 4269** By default, GeoPandas uses a plate carée projection: a mapping of longitude and latitude lines to horizontal and vertical lines (gross).
 - 3857** Most online maps use web Mercator, which is conformal (preserves shapes/angles) but much-maligned.
 - 2163** Albers Equal Area is a good conic projection for the US. ✓
- ▶ Inappropriate projections make maps look stupid.



Center of the World



~ Goode Homolosine



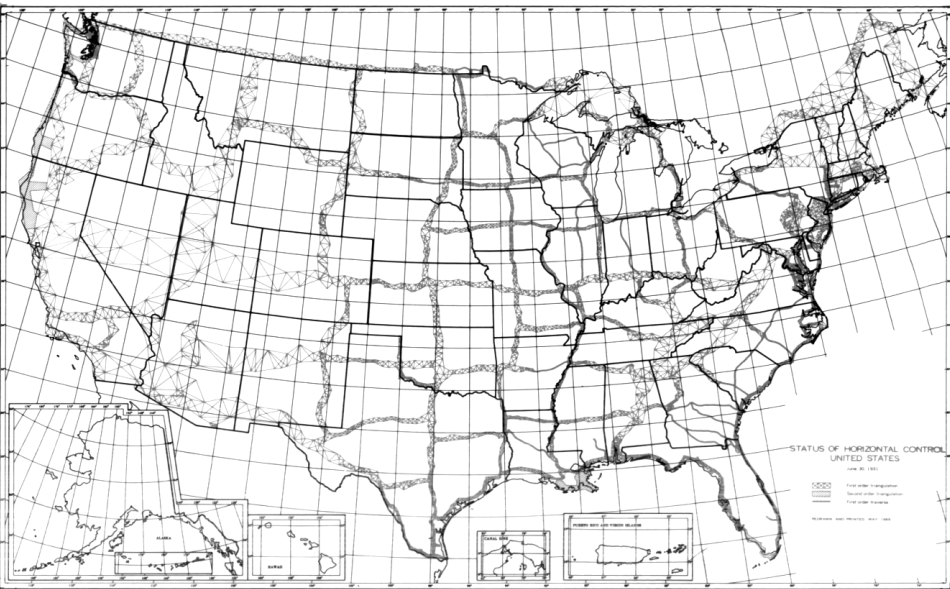
Albers Equal Area!!

STATUS OF HORIZONTAL CONTROL UNITED STATES

(June 30, 1955)

-  First order triangulation
-  Second order triangulation
-  First order traverse

NEARLY 500,000 SQUARE MILES





NAVY OF THE NORTHERN AND NORTHWESTERN LAKES
 Made in accordance with an act of Congress
 and under the direction of the
 HEADQUARTERS OF THE CORPS OF ENGINEERS, WAR DEPARTMENT

CHART OF THE NORTHERN AND NORTHWESTERN LAKES

compiled from the data
 of the
 U. S. LAKE SURVEY.

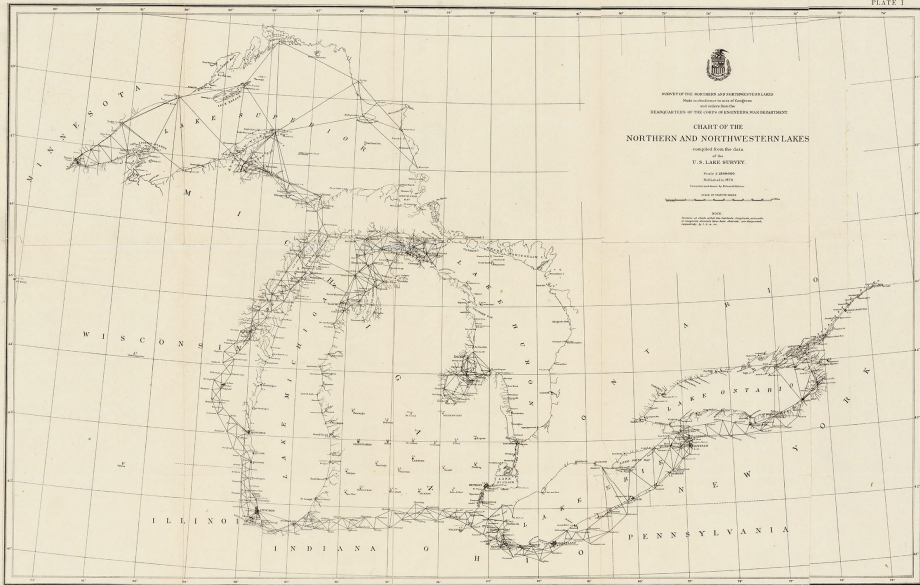
First Edition 1880
 Revised Edition 1875

Revised and corrected by the Hydrographic

DEPARTMENT OF THE ARMY

1875

Scale of miles
 Scale of fathoms



Several distinct goals:

1. Visualize a dataset as a map (join it to a shapefile).
2. Attribute join on two datasets with matching geometries.
 - ▶ Don't care about the geometry, just use it!
3. Datasets with different geometries (e.g., points and polygons).
 - ▶ Use a spatial join; may not care about map!

Attribute Join

- ▶ Attribute joins are the joins we've already been doing with pandas.
- ▶ Prepare them for the join by matching the indices (state codes).

Two examples:

1. Single mothers in the United States.
 - ▶ Join the state shapes to data from the (census API).
2. Voting returns in Pennsylvania from the election return site

Choropleth Maps: Shaded Areas

Easy to make basic, beautiful maps!

```
gdf.plot(column = "Percent Mothers Unmarried",  
         scheme = "quantiles", k = 10,  
         cmap = "rainbow", alpha = 0.4,  
         edgecolor = "black", linewidth = 0.5,  
         legend = True, figsize = (12, 8))
```

- ▶ The built-in method also allows for `equal_intervals` (linear, default), `quantiles`, and `fisher_jenks`.
 - ▶ Fisher Jenks defines categories by minimizing the in-group variance , and maximizing the between-group variance.
 - ▶ Most lay-people will only understand equal intervals!!
 - ▶ Without a scheme, geopandas will make a smooth, equal-interval coloring but a bad legend.
 - ▶ Can also use `vmin` and `vmax` for a smooth colormap.
- ▶ There are many, many [colormaps](#).

Point to Polygon: Spatial Joins

- ▶ Spatial joins (`sjoin`) use properties of two geometries – instead of equality of attributes – to align rows.
 - ▶ One geometry 'contains', is 'within', or 'intersects' another:

```
gpd.sjoin(pt_df, poly_df, how='left', op='within')
```

- ▶ For example: aggregate crimes (points) by community area (polygon).
- ▶ The geometry from the left DataFrame is preserved.

Building a GeoDataFrame from Scratch

- ▶ We also need to be able to create a GeoDataFrame from scratch.
- ▶ A GeoDataFrame, is just a DataFrame with a 'GeoSeries.'
- ▶ The GeoSeries is just a list of points, which we can construct as:

```
from shapely.geometry import Point  
pt = Point(x, y)
```

- ▶ Create the GeoDataFrame by setting the geometry and CRS (4269):

```
gpd.GeoDataFrame(crime_df, crs = {'init':  
'epsg:4269'}, geometry=geometry)
```

**Example: associate murders to
census tracts and community areas.**

- ▶ Folium creates a powerful javascript map on OpenStreetMap.
- ▶ Nice interface, easily embedded in other sites:

- ▶ `<iframe src="map.html" width=800px height=500px></iframe>`

```
import folium

m = folium.Map([40, -98], tiles='cartodbpositron',
               zoom_start=4, max_zoom=14, min_zoom=4)

ft = "Percent Mothers Unmarried"
colormap = folium.LinearColormap(("orange", "white", "purple"),
                                vmin = geo_merge[ft].min(),
                                vmax = geo_merge[ft].max(),
                                caption = ft)

colormap.add_to(m)

folium.GeoJson(geo_merge,
               style_function = lambda feature: {
                   'fillColor': colormap(feature['properties'][ft]),
                   "color" : "black", "weight" : 1, "fillOpacity" : 0.4
               }).add_to(m)

m.save("mothers.html")
```


Other Folium Features

- ▶ You can plot a collection of points with GeoJson, but you can get somewhat more control with

```
folium.Marker([41.7855052, -87.5971531],  
              popup='Harris School').add_to(map)
```

- ▶ See also e.g., CircleMarker, RegularPolygonMarker, etc.
- ▶ Full documentation [here](#).
- ▶ We'll come back to this after our last example.

- ▶ Often, we have latitudes and longitudes (ready to be wrapped as points), but addresses.
- ▶ Geocoding is the process of turning addresses into coordinates.
 - ▶ We have already done this with the google API.
- ▶ Many geocoding services (Census, Texas A&M) also provide census tracts, counties, etc. \implies Huge time saver!

Built-In Geocoding

- ▶ geopy plugs into the OpenStreetMap 'Nominatim' API, as well as google bing, yahoo, and openmapquest.
 - ▶ Some of the others require API keys for large numbers of requests.

```
from geopy.geocoders import Nominatim  
Nominatim().geocode("1155 E. 60th St, Chicago 60637")
```

- ▶ GeoPandas provides tools to do many geocodes in one pass:

```
gpd.tools.geocode(["London", "Paris",  
                  "New York", "Hong Kong"])
```

Census API (For Interest)

- ▶ The Census **geocoding API** matches tracts in **geography** endpoint
 - ▶ Also standard **location** mode.
- ▶ Capable of up to 1000 addresses at a time in batch mode:

```
curl -F addressFile=@short.csv\  
-F benchmark=Public_AR_Current\  
-F vintage=ACS2015_Current \  
https://geocoding.geo.census.gov/geocoder/geographies/addressbatch
```

Merges and Spatial Operations as Geocoding

Spatial operations (`intersects`, `within`, and `contains`) are effectively geocodes:

```
geo_df[geo_df.contains(pt)]["NAME"]
```

Second Folium Example

- ▶ Make a map of places represented in this class (points and countries).
- ▶ Curl these shapefiles for the world:

http://thematicmapping.org/downloads/TM_WORLD_BORDERS_SIMPL-0.3.zip