# Geographic Information Systems

Jamie Saxon

Introduction to Programming for Public Policy

November 14, 2016

# Geographic Information Systems: Maps

People love maps – emotional response to 'seeing yourself.'

- GIS is a huge field. There are other classes at Harris for this.
  - A lot of work on 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.
    $\Longrightarrow$ Great potential for joins!

# What We'll Cover

1. Making simple maps with GeoPandas (`pandas+`).
   - Finding and importing shapefiles and geojson (like `read_csv()`).
   - Projections (briefly).
2. Attribute and spatial joins.
   - Using the census geolocation API (APIs).
   - Making a map with real data!
3. Making a simple web (!) map with GeoPandas

- Largely revisiting old material.

# 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: **data.cityofchicago.org**, the **US Census**, **USGS**, etc.
- Much of this is provided in 'ESRI Shapefiles' (Environmental Systems Research Institute, major GIS company) or in geojson.
  - Shapefiles come zipped with a lot of other files, it's the shp you want.
  - Let's browse: **census shapefiles**.
- Addresses may be geocoded and coordinates are also points!

# Loading a Shapefile with GeoPandas

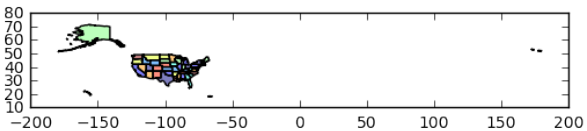
- **GeoPandas** simply adds a geometry series to a pandas DataFrame.
- It is tightly integrated with many other geographic programs, among them **fiona** for reading geojson/shapefiles and **shapely** for geometric operations (intersections, etc.).
- Really easy to import! Both shapefiles and geojson:

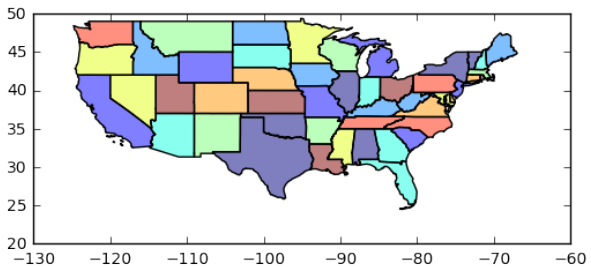
```
import geopandas as gpd
gdf = gpd.read_file("myfile.shp")
gdf.plot() # WOW!!!!
```

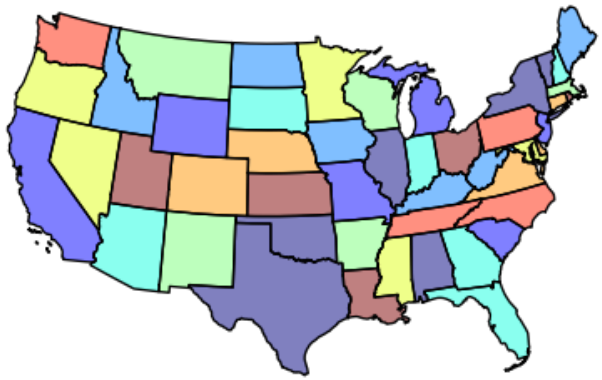
- All of the 'standard' dataframe operations (slicing, indexing, merging) are still available.

# Making a Slightly 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 $\geq 57$).
- We can also use a better projection: `gpd.to_crs(epsg=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:

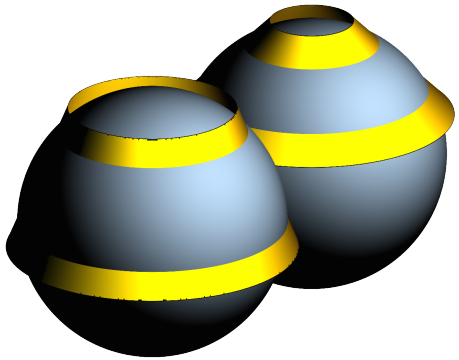
  4269 By default, GeoPandas uses a Plate-Carée projection: a mapping of longitude and latitude lines to horizontal and vertical lines (gross).

  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!!

# Joins

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 choropleth maps!

```
gdf.plot(column = "Percent Mothers Unmarried",
         scheme = "quantiles", k = 5,
         cmap = "rainbow", legend = True,
         alpha = 0.4, linewidth = 0.5,
         figsize = (12, 8))
```

- The built-in method also allows for `quantiles` (default), `equal_intervals` (linear), 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!!
- There are many, many **colormaps**.
- The defaults only allow 9 breaks; see `Adanced.ipynb` for a gradient.

# Point to Polygon: Spatial Joins

- GeoPandas provides many powerful geometric operations.
- Spatial joins (`sjoin`) behave as you might expect:
  `gpd.sjoin(pt_df, poly_df, how = 'left', op = 'within')`
- This joins rows with locations (points) in one dataframe, to regions (polygons) in the other.

# Building a GeoDataFrame from Scratch

- We also need to be able to create a GeoDataFrame from scratch.
- A GeoDataFrame, as we've said, is just a DataFrame with a geometry.
- So we need to build the GeoSeries...
- In this case, the GeoSeries consists of a list of points, which we can construct as

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

- Then create a GeoDatFrame, by setting the geoemtry and crs:

```
gpd.GeoDataFrame(crime_df, crs = tract_df.crs,
geometry=geometry)
```

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

# A Simple Web Map

scraps/web_folium.py

- Folium creates a powerful javascript map on OpenStreetMap.
- Really nice interface, easily embedded in other sites:
  - `<iframe src="map.html" width=800px height=500px></iframe>`

```
import folium

m = folium.Map([39.828175, -98.5795], tiles='cartodbpositron',
               zoom_start=4, max_zoom=14, min_zoom=4,)

ft = "Percent Mothers Unmarried"
cmap = folium.colormap.linear.YlOrRd.scale(merged[ft].min(), merged[ft].max())

folium.GeoJson(merged, style_function=lambda feature: {
                       'fillColor': cmap(feature['properties'][ft]),
                       'fillOpacity' : 0.6,
                       'weight' : 2, 'color' : 'black'
                      }).add_to(m)

cmap.caption = 'Percent Children Born to Single Mothers'
cmap.add_to(m)

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

# Geocoding

- Often, we have latitudes and longitudes (ready to be wrapped as points), but addresses.
- Geocoding is the process of turning addresses into coordinates.
- Many geocoding services can additionally provide census tracts, counties, etc. $\Longrightarrow$ Huge time saver!

# Nominatim/geopy

- geopy plugs into the OpenStreetMap 'Nominatim' API.
- Super easy to use!!

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

# Census API

- 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 layers=9 \
     -F vintage=ACS2015_Current \
     -F benchmark=Public_AR1_Current \
     https://geocoding.geo.census.gov/geocoder/geographies/addressbatch
```