

A shapefile contains binary-encoded geographic data in a particular format.

A good discussion is [here](#).

The specification was developed at least partly by ESRI, which makes geographic information software.

The binary encoding was undoubtedly designed to save space, back when space (storage, transmission bandwidth) was much more expensive. Now, the opaque data seems to me to be a liability, but ESRI likes "vendor lock-in."

There is actually not just one file, but always a minimum of three including

- .shp
- .shx
- .dbf

inside a zip container. .shp has the coordinates, in binary format. .shx is an index to the shapefile, including the bounding box and offsets to all the parts. .dbf is a database of record attributes for the entities in the shapefile.

The Shapefiles from the US Census also have .prj and .xml. .prj contains a projection of the coordinate data. We'll look at that later.

## **obtaining the data**

from [Eric Celeste](#), I got a [link](#) for shapefiles at the US Census.

Shapefiles are designed for use with GIS software. GIS stands for Geographic Information System.

They are also available from the census people in KML format. KML [stands for](#) Keyhole Markup Language, which is an XML-type format. Keyhole was the company acquired by Google in its development of Google Earth.

There are a variety of available collections, including

- congressional district
- county

- states

and more.

Also, three resolutions. The largest is called 500k and takes up about 10 MB.

The smallest file with state data is `cb_2018_us_state_20m.zip`. The others are 5m and 20m.

These are zip files. One of the files inside the container is the shapefile.

## geopandas

pip install geopandas  
geopandas wants the zip file

```
import geopandas as gpd

fn = 'cb_2018_us_state_20m.zip'
df = gpd.read_file(fn)
```

The result is a data frame.

```
sel = df['NAME'] == 'Colorado'

g = df.loc[sel].geometry
from shapely.geometry import mapping
D = mapping(g)
```

`D['features']` gives

```
[{'id': '42',
  'type': 'Feature',
  'properties': {},
  'geometry': {'type': 'Polygon',
               'coordinates': (((-109.059962, 38.499987),
                                (-109.05122383101701, 39.36667754958629),
                                ...
                                (-109.059962, 38.499987))),
               }
  'bbox': (-109.060062, 36.992426, -102.041876, 41.003073)
}]
```

