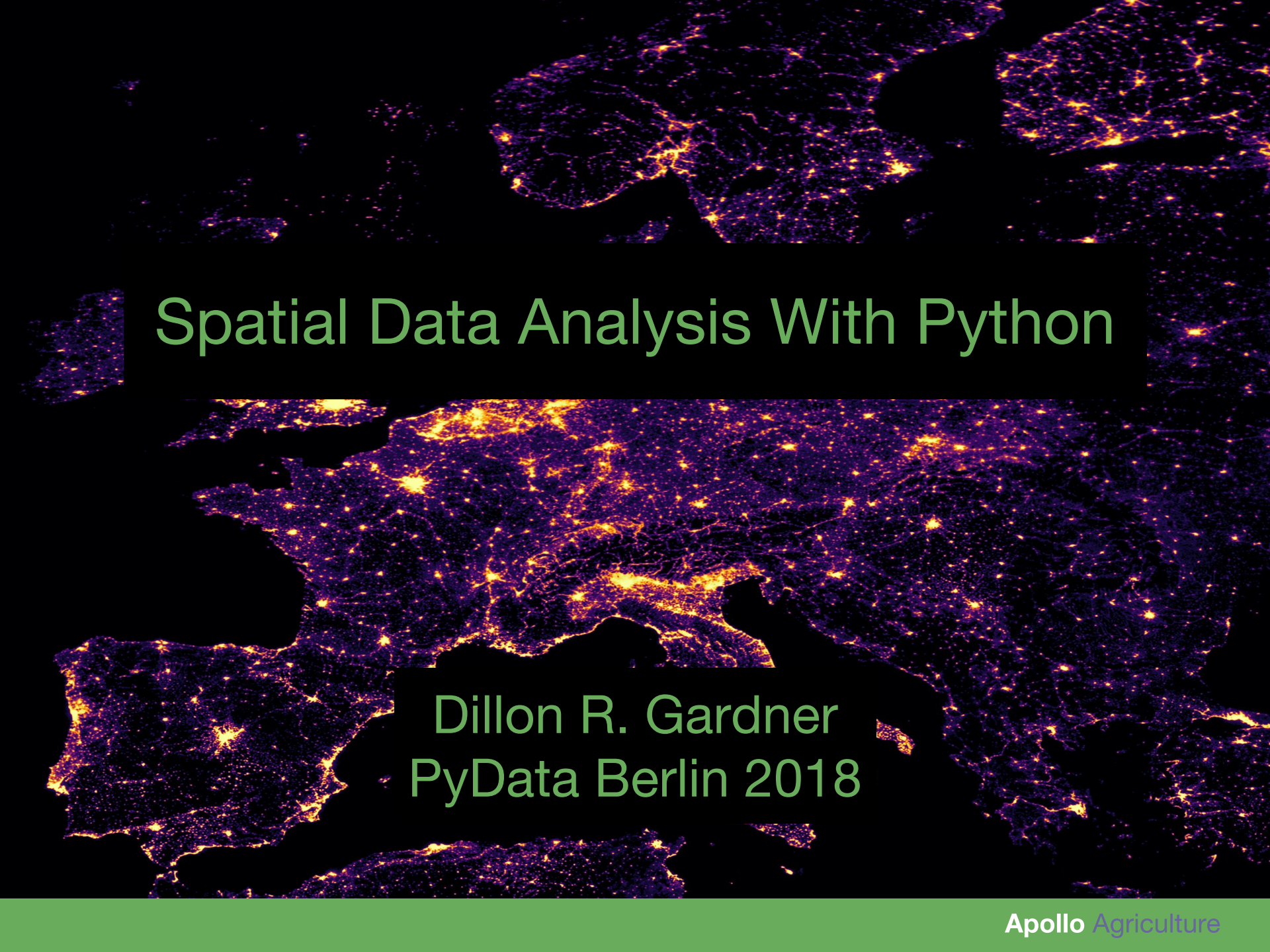
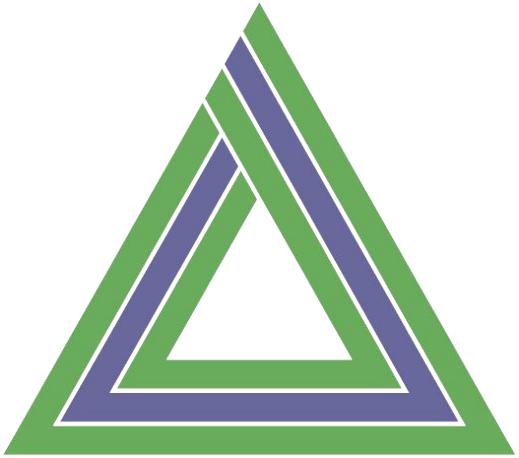


Spatial Data Analysis With Python

Dillon R. Gardner

PyData Berlin 2018

https://github.com/dillongardner/PyDataSpatialAnalysis



APOLLO
AGRICULTURE

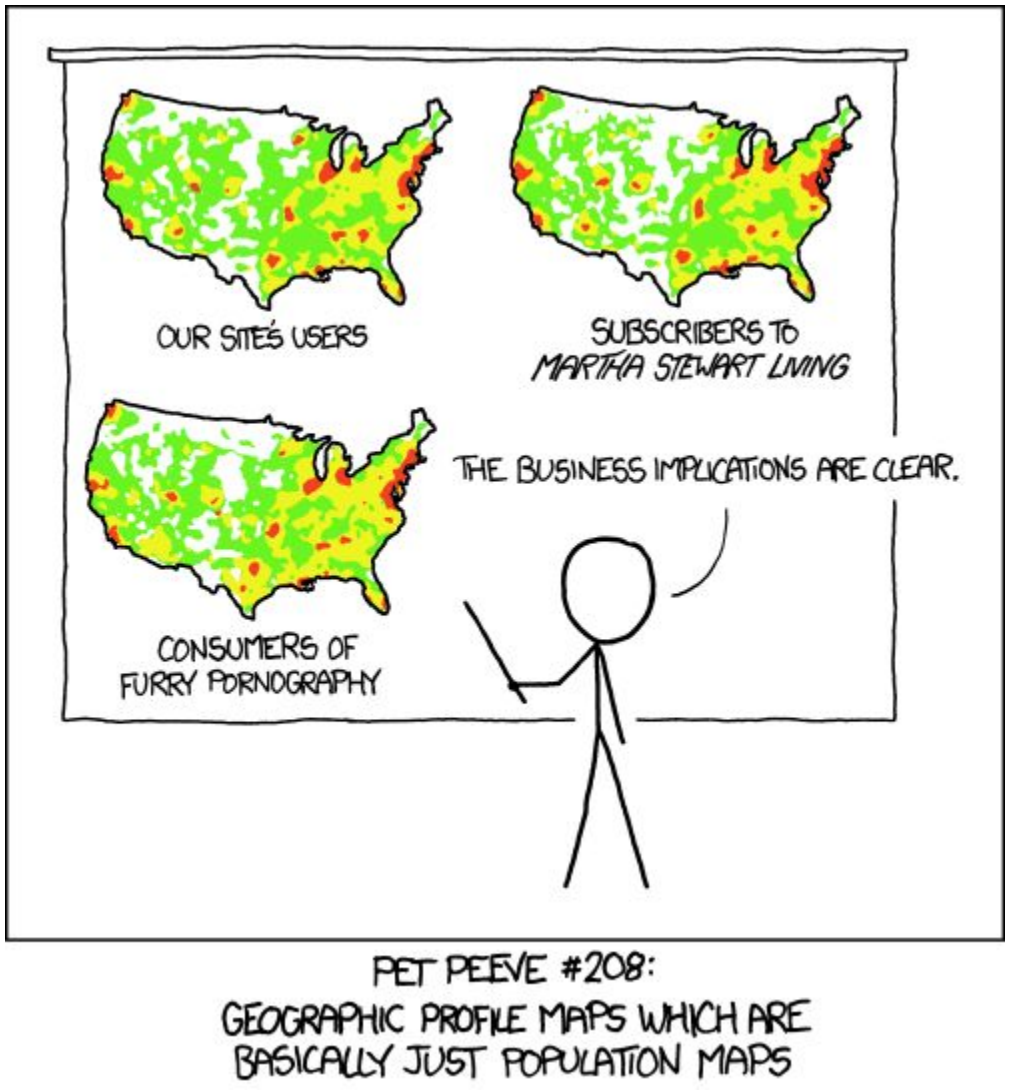
Apollo helps smallholder farmers maximize their profits. We use agronomic machine learning, remote sensing, and mobile phones to deliver input finance and customized advice to smallholder farmers with radical efficiency and scalability.

https://github.com/dillongardner/PyDataSpatialAnalysis

Spatial Data Use Cases

- Credit Modeling
- Fraud Detection
- Analyze Customers Farms
- Route Field Agents

Prerequisite Web Comic

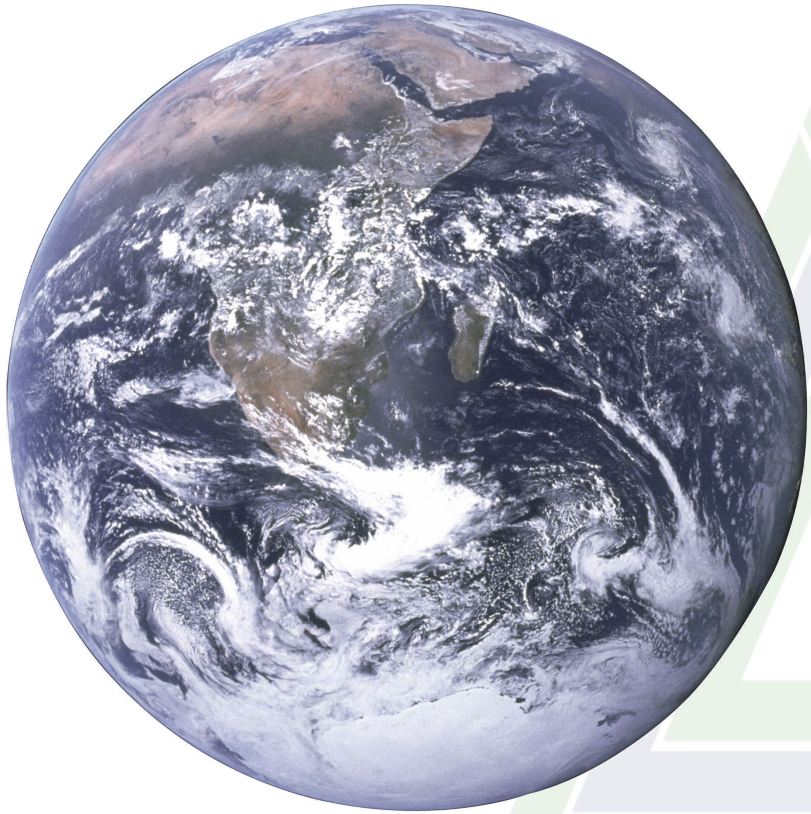


https://www.xkcd.com/1138/

What is geospatial data?

Data that refer to specific locations on the Earth

- Vector Data
  - Point (Current GPS reading)
  - Line (Street)
  - Polygon (Maize Field)
- Raster Data
  - Thematic/Discrete (Land use)
  - Continuous (Satellite imagery)



Vector Data

What it is: Defined region in space with associated properties

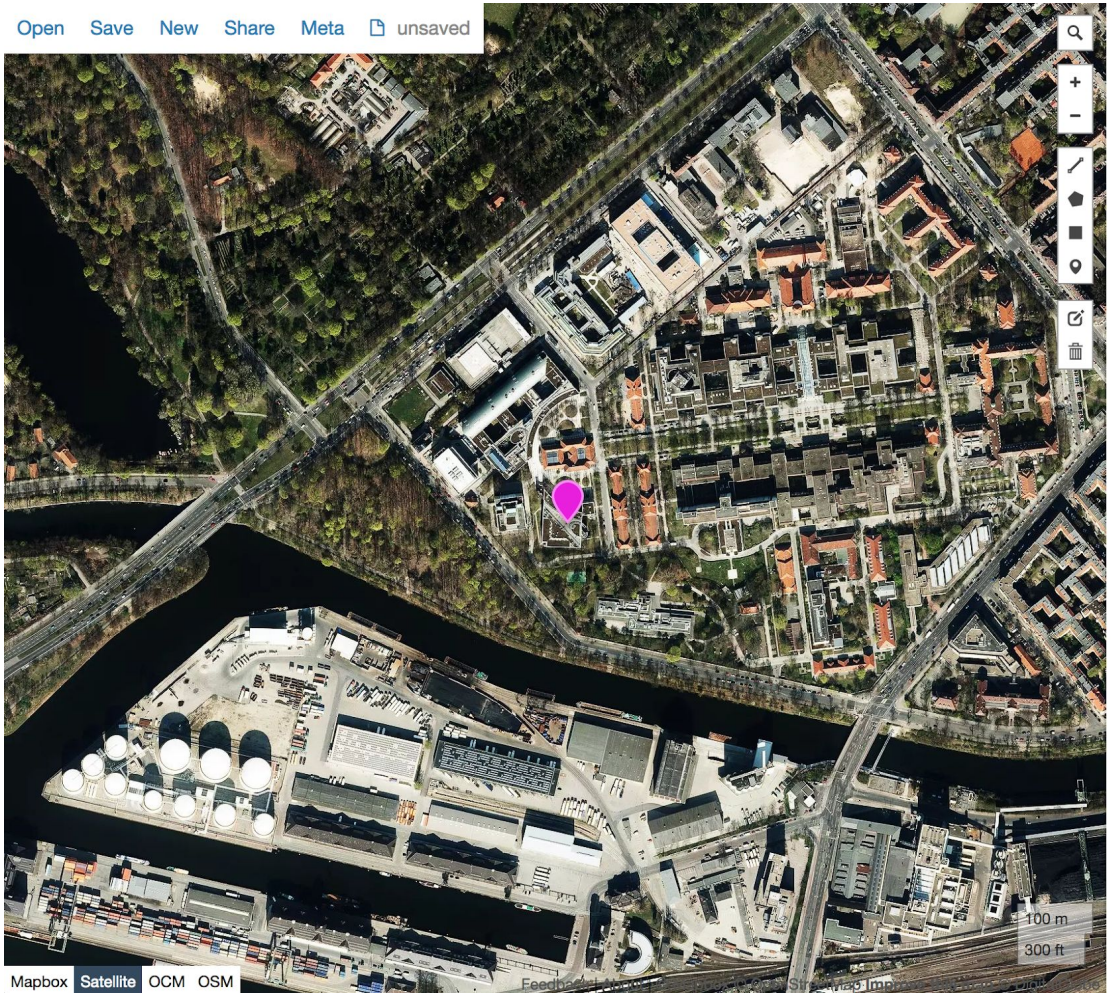
Typical file formats: Shapefiles (.shp), GeoJSON, Database Geometries, Well-Known-Text (WKT), Well-Known-Binary (WKB)

Data Sources: Government and NGO surveys, GPS data, OpenStreetMap

Python Libraries: shapely, fiona, geopandas, GDAL/OGR

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "marker-color": "#e921de",
        "marker-size": "medium",
        "name": "PyData Berlin"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          13.33991289138794,
          52.5409910220435
        ]
      }
    }
  ]
}
```

Vector Data



</> JSON | Table | ? Help

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "marker-color": "#e921de",
        "marker-size": "medium",
        "name": "PyData Berlin"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          13.33991289138794,
          52.5409910220435
        ]
      }
    }
  ]
}
```

http://geojson.io

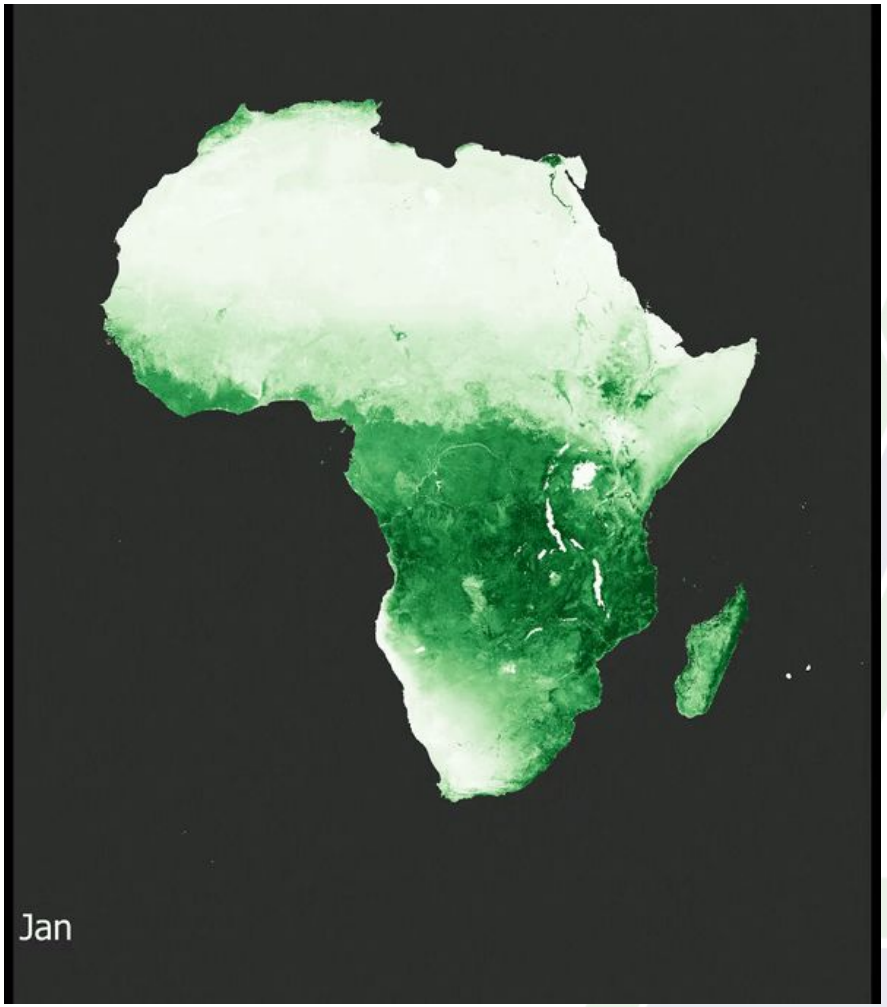
Raster Data

What it is: Raster data (“image”) with associated metadata to translate from pixel location to spacial location

Typical file formats: GeoTiff, JPEG

Data Sources: Satellites, Aerial Photography, Elevations

Useful Libraries: rasterio, GDAL/OGR



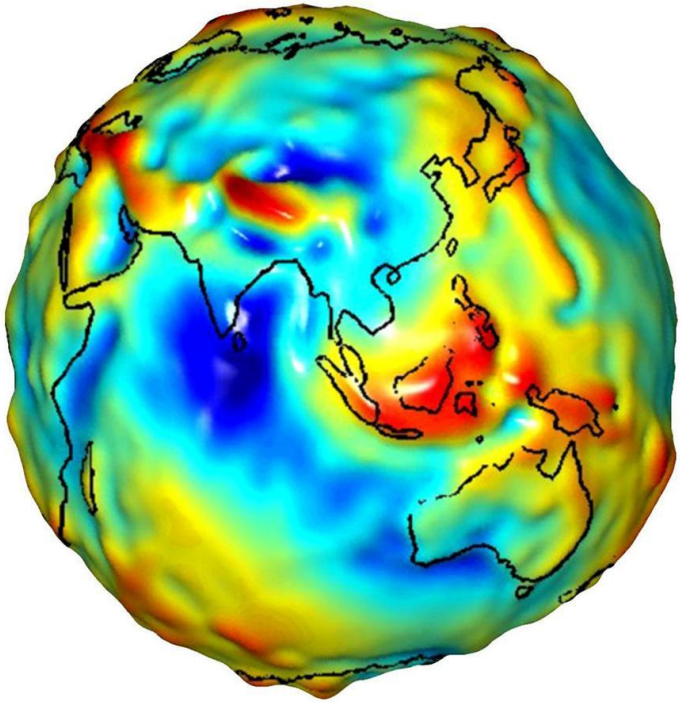
Data : ftp://africagrids.net/250m/MOD13Q1/Version6/NDVI/Monthly/
Visualization:
https://www.reddit.com/r/dataisbeautiful/comments/80o1ah/vegetation_intensity_throughout_the_year_for/duww49s/

“Location” - what does it mean?

- Location always relative to a Coordinate Reference System
- Not all data in the same reference system
- EPSG Geodetic Parameter Set

```
GEOGCS["WGS 84",
    DATUM["WGS_1984",
        SPHEROID["WGS 84",6378137,298.257223563,
            AUTHORITY["EPSG","7030"]],
        AUTHORITY["EPSG","6326"]],
    PRIMEM["Greenwich",0,
        AUTHORITY["EPSG","8901"]],
    UNIT["degree",0.0174532925199433,
        AUTHORITY["EPSG","9122"]],
    AUTHORITY["EPSG","4326"]]
```

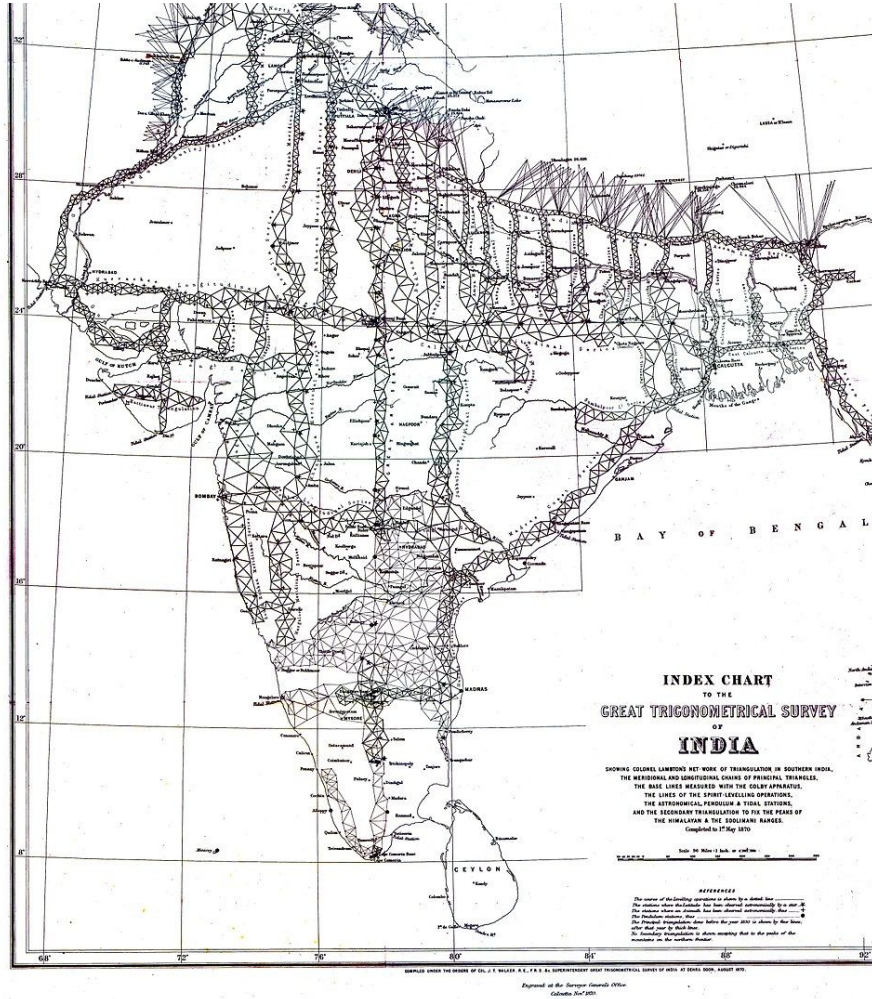
Specifying Location



Nasa/JPL

- Specify the shape of the Earth
  - Ellipsoid - Approximate shape
  - Geoid - Gravitational equipotential
- Datum - Carefully measured network of surveyed points
- Projection - 3D surface on 2D plane

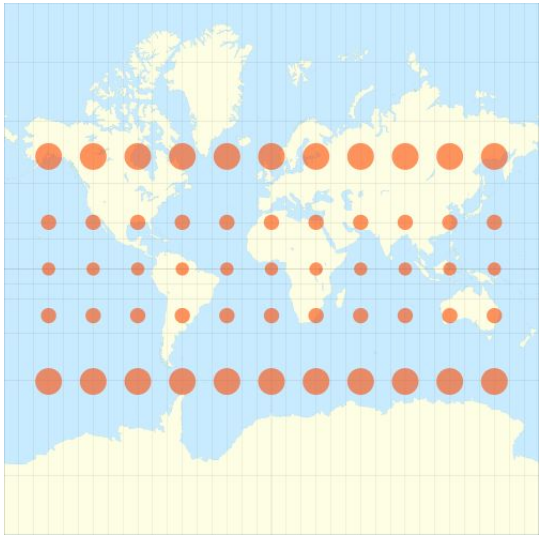
Common Datum



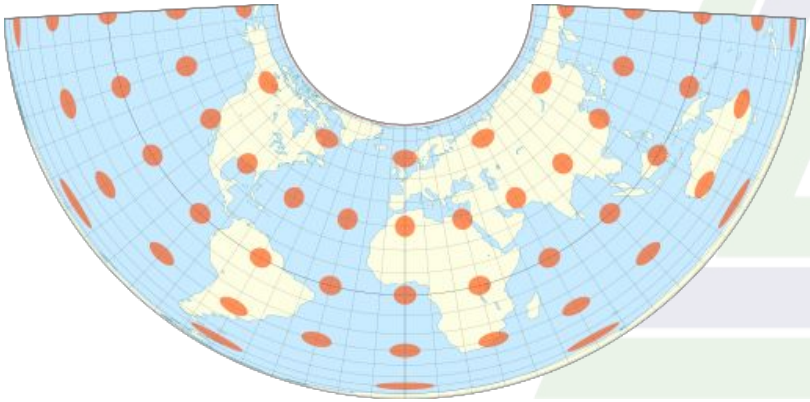
- Datum used varies across the world
- Most Common
  - WGS84 (EPSG 4326)
    - Used by GPS
  - NAD83
  - ED50
- Same location will have different coordinate in different datum

Projections - 3D surface onto 2D plane

- All projections lead to distortions
- Select projection that best matches use case
  - Area
  - Shape
  - Direction
  - Distance
  - Visual properties



By Eric Gaba (Sting - fr:Sting) [GFDL (http://www.gnu.org/copyleft/fdl.html) or CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], from Wikimedia Commons



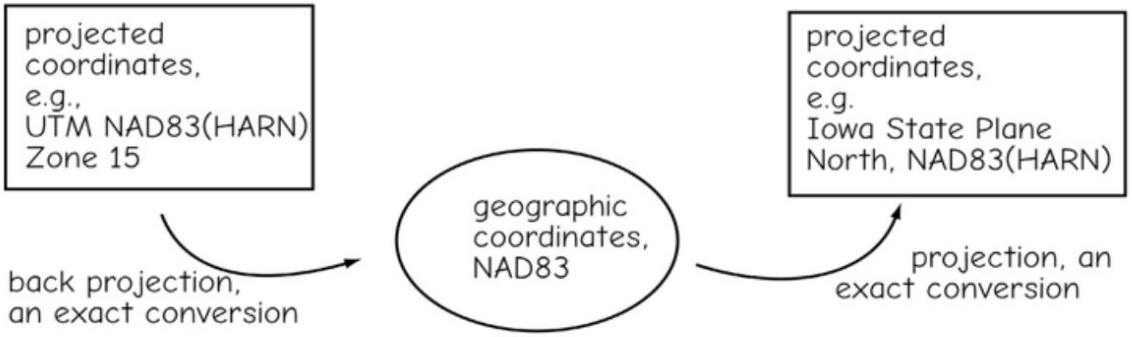
By Justin Kunimune [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)], from Wikimedia Commons

Common Projections

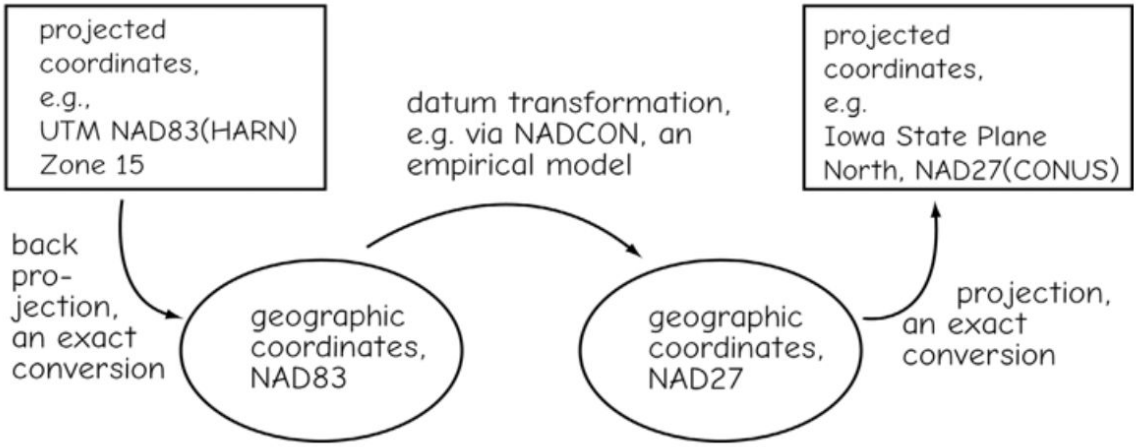
- Web Mercator (EPSG 3857)
  - Used for web maps (Google, Bing, OSM etc.)
  - https://msdn.microsoft.com/en-us/library/bb259689.aspx
- Plate Carrée (EPSG 4326)
  - Used for lots of global raster data
  - Map lon/lat to x-y
- Transverse-mercator
  - Universal Transverse Mercator (UTM)
  - Used for satellite imagery
- Equal Area Projections (e.g Albers)
  - Useful for calculating area of polygons

Conversions

a) From one projection to another - same datum and version



b) From one projection to another - different datums



P. Bolstard *Fundamentals of GIS* Ch. 3

What do you actually *DO?*

- Load
- Transform
  - Use matching coordinate systems
  - Transform to projections that make calculations easy
  - Transform vector data to match raster data
- Combine/Calculate
  - Ex. Mask raster data with a polygon
  - Ex. Intersections of polygons
  - Ex. Area

Taxonomy of Tools

Low Level Geospatial Tools

- GEOS - Vector data manipulation
- GDAL/OGR - Geospatial
  - GDAL - Raster data model
  - OGR - Vector data model
- PROJ.4 - Generic coordinate transformation

Miscellaneous

- GeoPandas - Extends pandas to support geometric operations
- Plotting -
  - Descartes
  - Cartopy

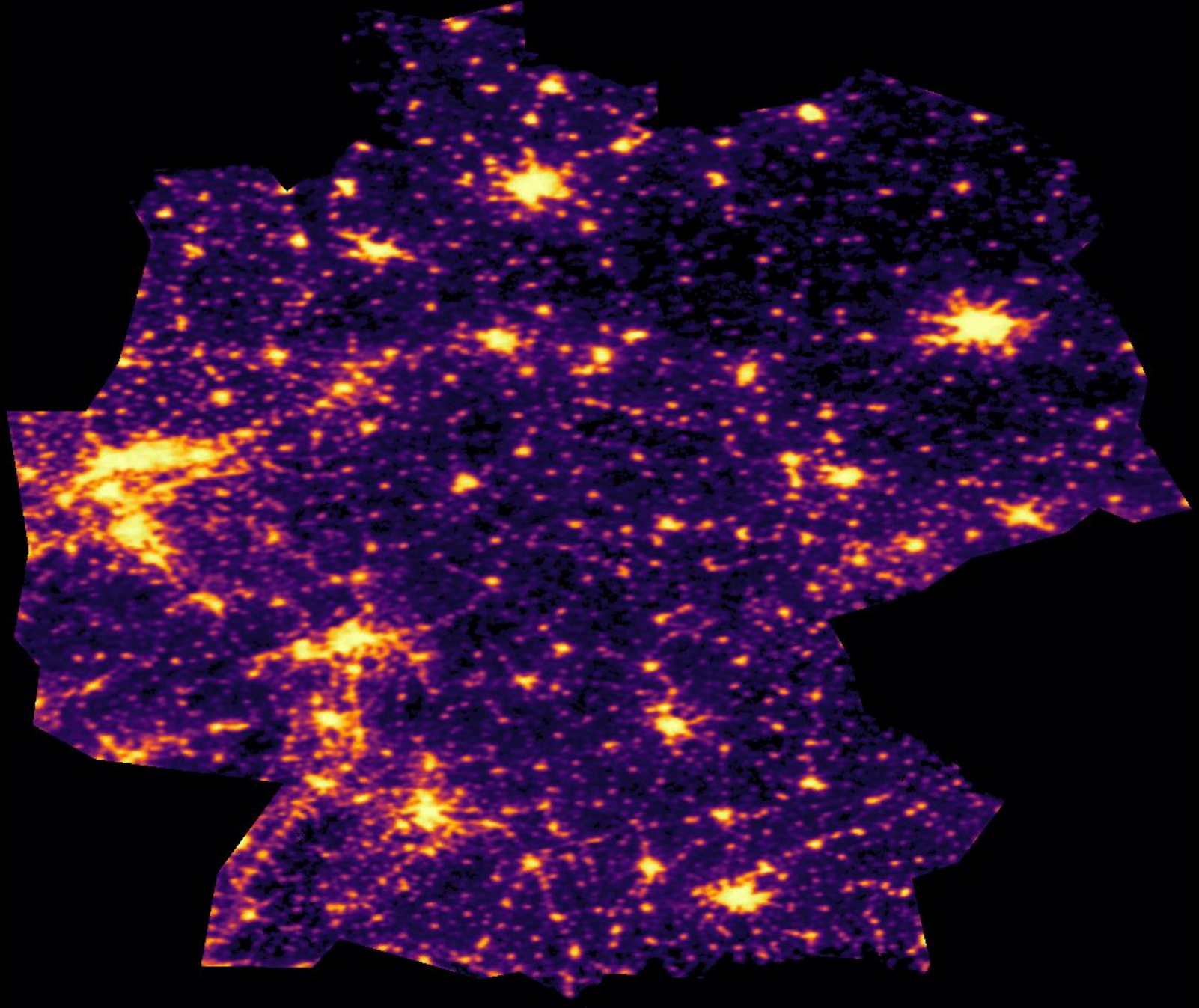
Vector Data

- Shapely
  - Manipulation of geometric objects
  - Based on GEOS
- Fiona
  - API for reading/writing vector data
  - Based on OGR
- Pyproj
  - Coordinate transformations
  - Cython wrapper of PROJ.4

Raster Data

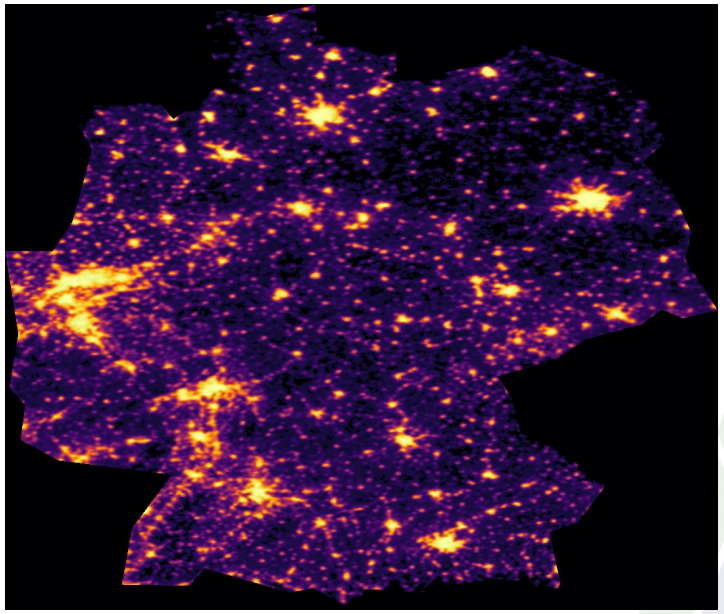
- Rasterio
  - Reading and manipulating raster data
  - Requires on GDAL

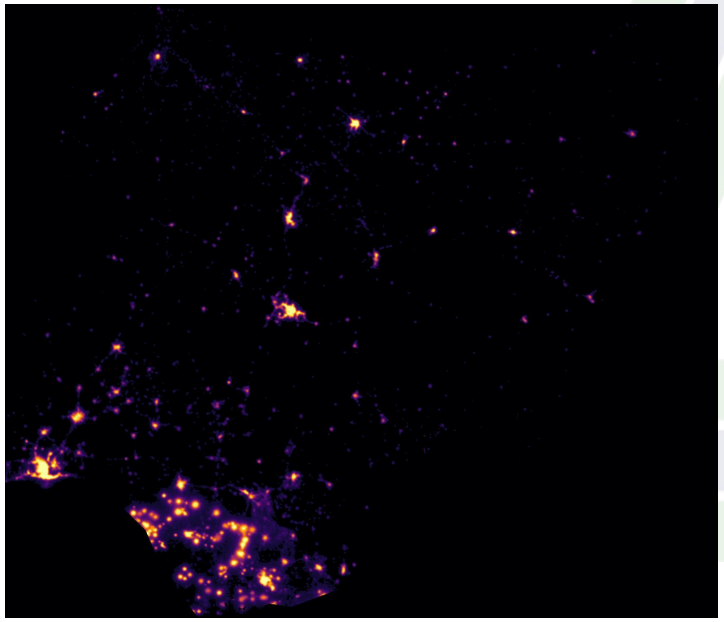
Nighttime Lights and Wealth



Nighttime Lights and Wealth

- Nighttime light data show
  - Human activity
  - Wealth
- As an example of geospatial data, look at community level wealth and night light activity in Africa
  - Demographics and Health Survey
  - DMPS Nighttime Light Data
- See Jean *et al.* **Science** 2016





Assets and Nighttime Lights In Nigeria Steps

- Load survey data - vector data
- Simple exploratory data analysis - vector data
  - Plotting
- Load nighttime lights - raster data
- Simple exploratory data analysis - raster data
- Extract nightlight intensity
  - Transform vector data
  - Mask raster data by vector data
  - Calculate
- Plot

In-depth code available: https://github.com/dillongardner/PyDataSpatialAnalysis

Load Survey Data - GeoPandas

- GeoPandas extends pandas functionality
- Easily load geospatial data files (uses Fiona under the hood)
- Geospatial capabilities based on special “geometry” column
  - Contains Shapely geometric objects
  - Simplifies transformation

```
> import geopandas as gpd
> nigeria_gdf =
gpd.read_file(os.path.join(DATA_PATH,
'formatted_dhs/'))
> nigeria_gdf.head()
```

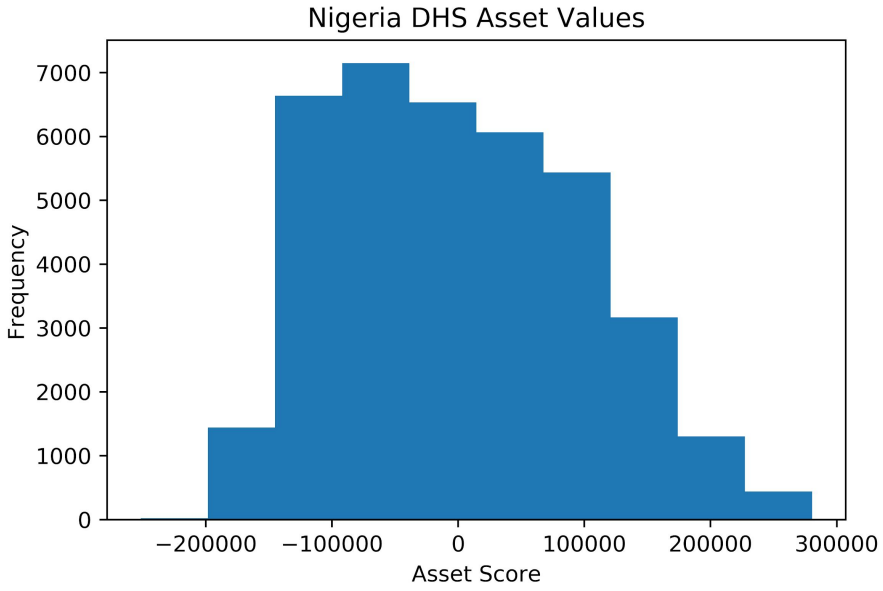
	cluster	assets	geometry
0	1.0	-106269	POINT (8.097115000000001 6.90227)
1	1.0	-89171	POINT (8.097115000000001 6.90227)
2	1.0	-101669	POINT (8.097115000000001 6.90227)
3	1.0	-105983	POINT (8.097115000000001 6.90227)
4	1.0	-89785	POINT (8.097115000000001 6.90227)

```
> print(nigeria_gdf.crs)
{'init': 'epsg:4326'}
```

Exploratory Data Analysis - Vector Data

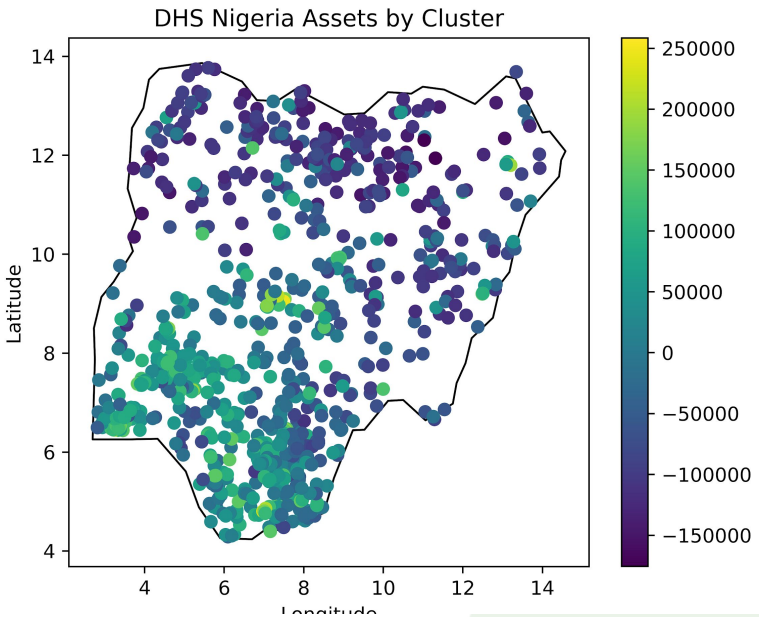
- Base pandas functionality

```
ax = nigeria_gdf.assets.plot(kind='hist')
```



- Cluster level data using pandas split-apply-combine functionality
- GeoPandas plotting

```
> clusters.plot(column='assets',
                legend=True)
```



Load Night Light Raster Data - Rasterio

```
> import rasterio
> nightlights = rasterio.open('/F182013...tif')
> print(nightlights.crs)
```

CRS({'init': 'epsg:4326'})

```
> print(nightlights.bounds)
```

BoundingBox(left=-180.004 bottom=-65.004, right=180.004, top=75.004)

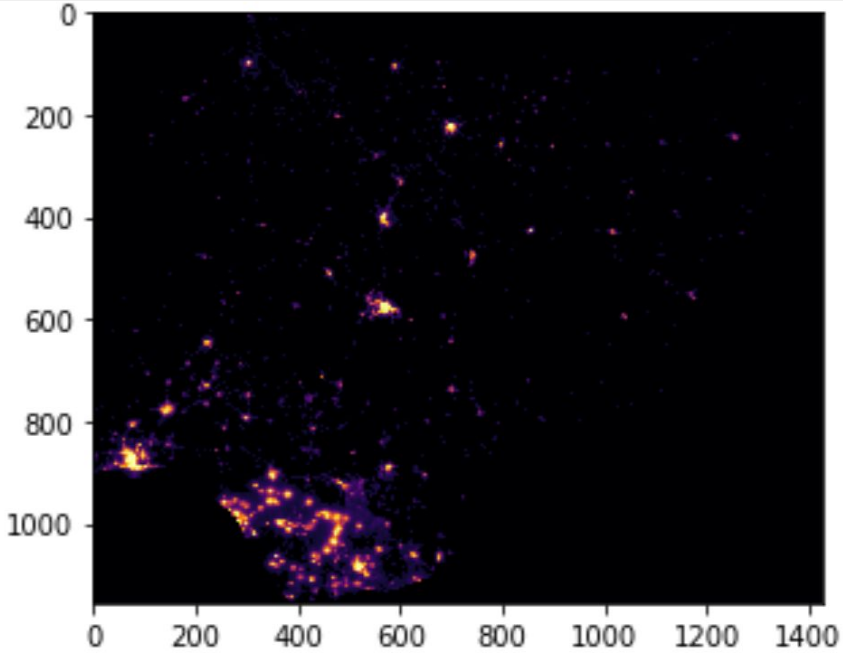
```
> print('Image size: ({}, {})'.format(nightlights.width, nightlights.height))
```

Image size: (43201, 16801)

- Data are a huge image with additional metadata
- Boundary box
  - Units depend on coordinate reference system
  - In this case are lon/lat
- Contains `transform` object that will map pixels to coordinates

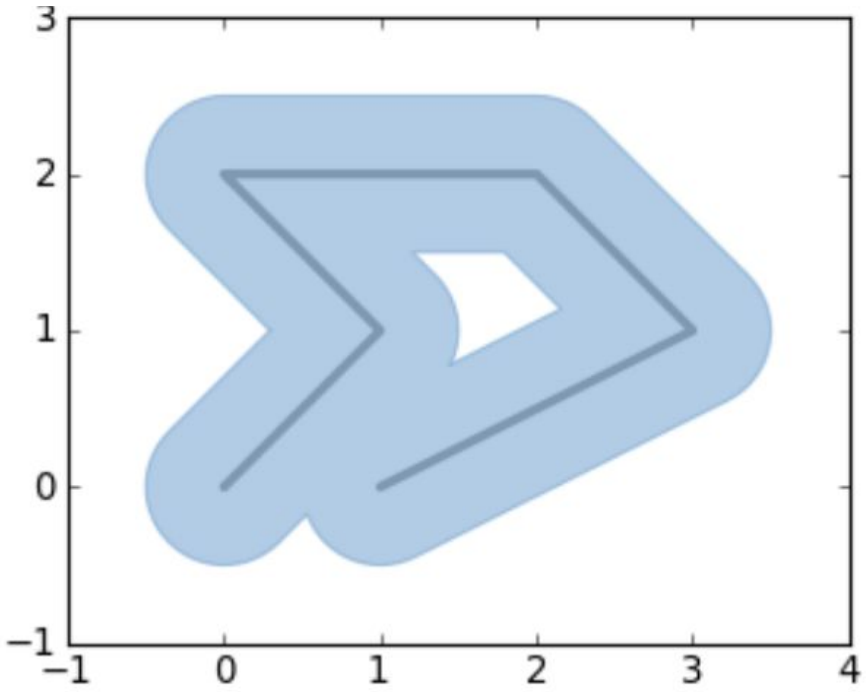
Exploratory Data Analysis

```
> from rasterio.mask import mask
> nigeria_polygon = nigeria.geometry.values[0]
> nigeria_lights, out_transform = mask(nightlights,
                                       [nigeria_polygon],
                                       crop=True)
> plt.imshow(nigeria_lights[0], cmap='inferno')
```



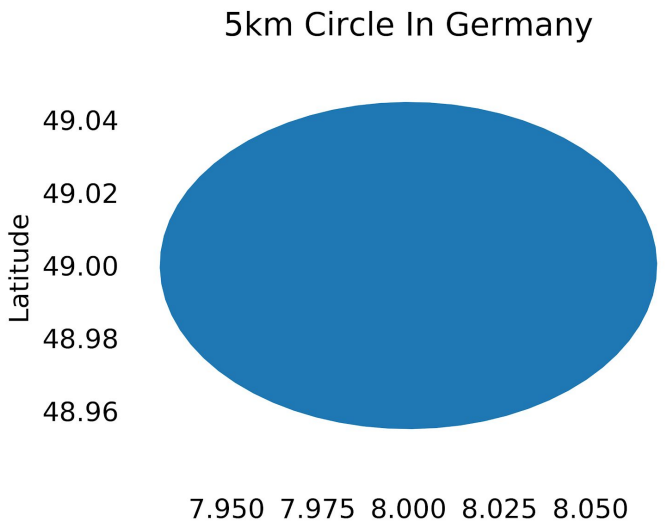
Extract Nighttime Light Intensity

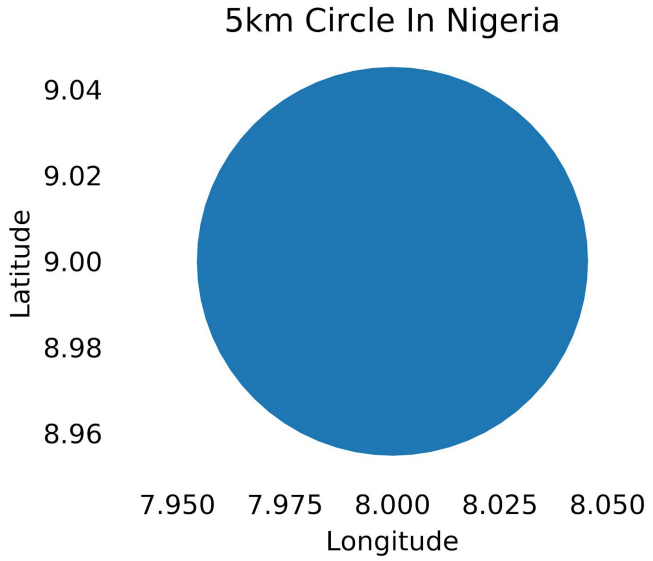
- Result of mask operation is an numpy array
  - Can take the median value
- Want value for each cluster
  - Cannot do on a point
  - Need to buffer point into a polygon



Coordinate Transformations

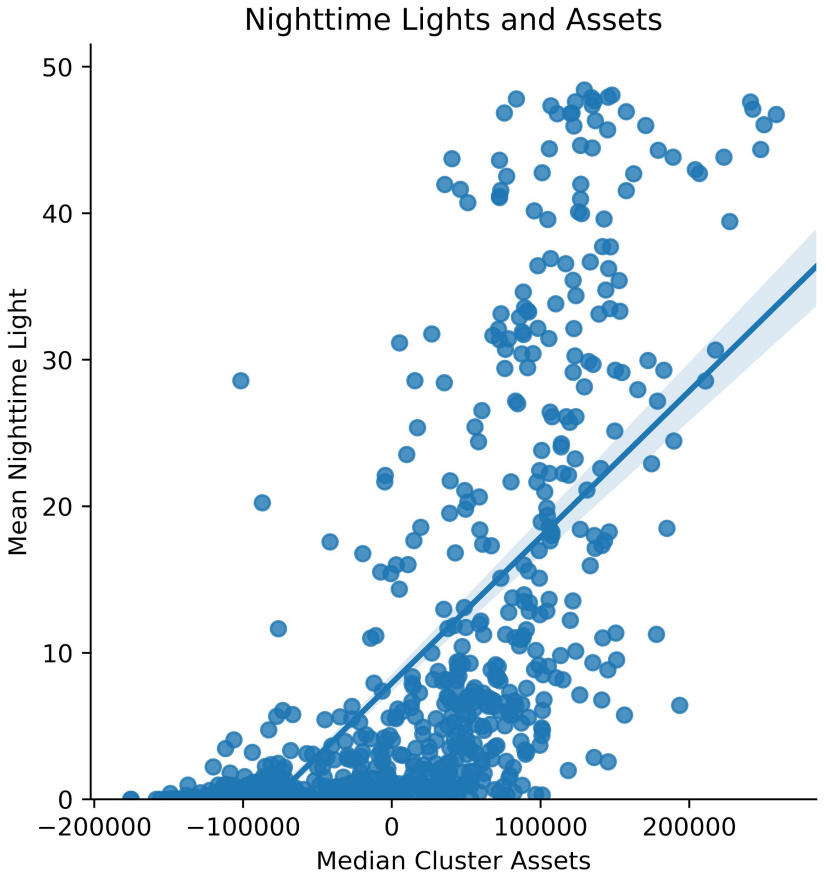
- Buffering in EPSG 4326 is in degrees
- Project into UTM 32 N
  - Or another projection that has small distortion in Nigeria
  - Allows for accurate 5km circles
- Buffer by 5000m
- Project back to EPSG 4326
- Mask nighttime lights and calculate median value





Nighttime Light Intensity vs Asset Score

```
> new_crs = """+proj=utm +zone=32
+ellps=GRS80 +units=m +no_defs"""
> old_crs = "+init=epsg:4326"
> new_projection =
clusters.geometry.to_crs(new_crs)
> buffered = new_projection.buffer(5000)
> original_projection =
buffered.to_crs(old_crs)
> mean_night_lights =
original_projection.map(geom_to_mean_light)
```



Summary

- Geospatial data
  - Powerful
  - Cheap
  - Fun
- General Guidelines
  - Use matching coordinate systems
  - Transform to projections that make calculations easy
  - Transform vector data to match raster data
- Python provides excellent tooling

We are hiring!