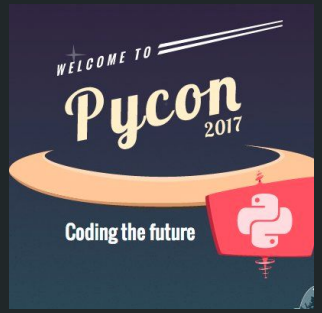
Using Python to Analyze Spatial Data

Juan Carlos Méndez
CTO @gkudos
juan@gkudos.com

kudos
open systems, open ideas



Topics

- Spatial Data?
- Python?
- Visualizing Geodata
- Analyzing Geodata (Basics)

About me

- Juan Carlos Méndez
  - CTO at @gkudos
  - GIS Consultant
  - Software Architect / Programmer / “Data Engineer”
  - https://github.com/dersteppenwolf

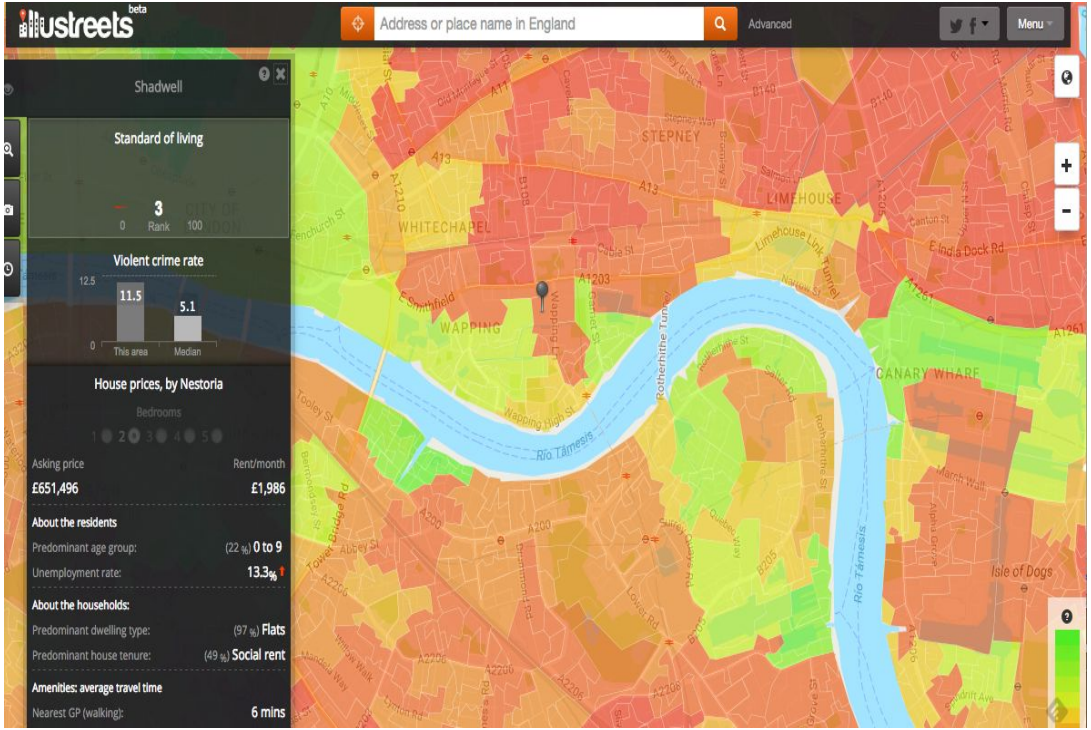
- Education
  - Systems Engineer- Universidad Nacional de Colombia, Bogotá
  - Telematics / eBusiness specialist - Escuela Colombiana de ingeniería
  - Information Engineering (Student) - Universidad de los Andes

Spatial Data?

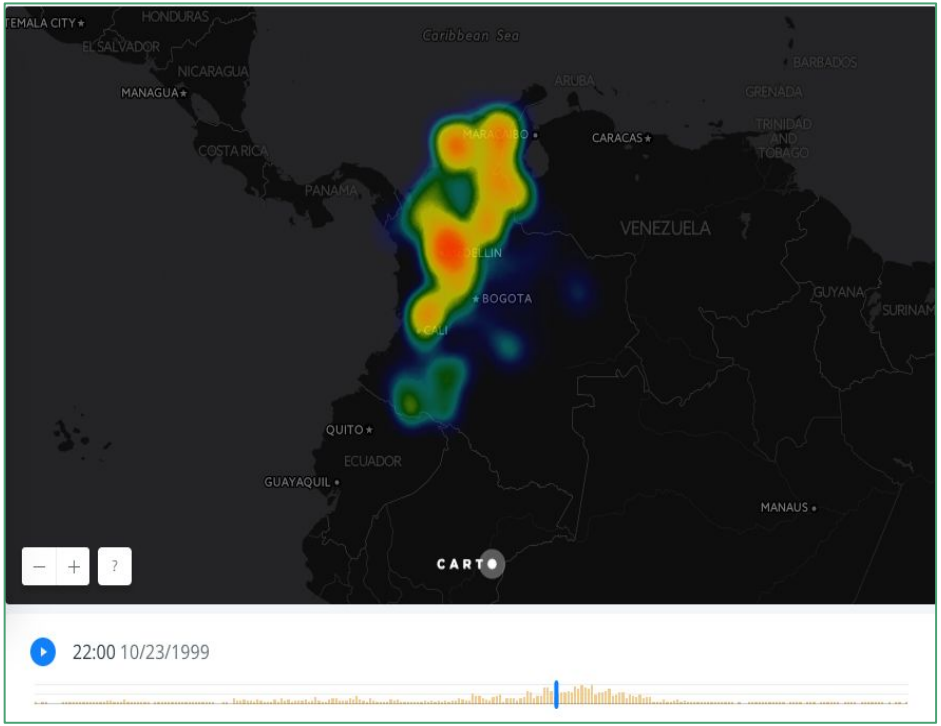
Location, location, location!

“You can buy the right home in the wrong location. You can change the structure, remodel it or alter the home's layout but, ordinarily, you cannot move it. It's attached to the land”

http://bit.ly/2fz2ySD



Spatio-temporal data



"...in Google, about 25 PB of data is being generated per day, and a significant portion of the data falls into the realm of spatio-temporal data..."

Lee, J.-G., & Kang, M. (2015). Geospatial Big Data: Challenges and Opportunities. Big Data Research, 2(2), 74–81. http://doi.org/10.1016/j.bdr.2015.01.003

What for?

30 million AshleyMadison users, mapped

8.5k SHARES

Share on Facebook | Share on Twitter | +

WHAT'S THIS?

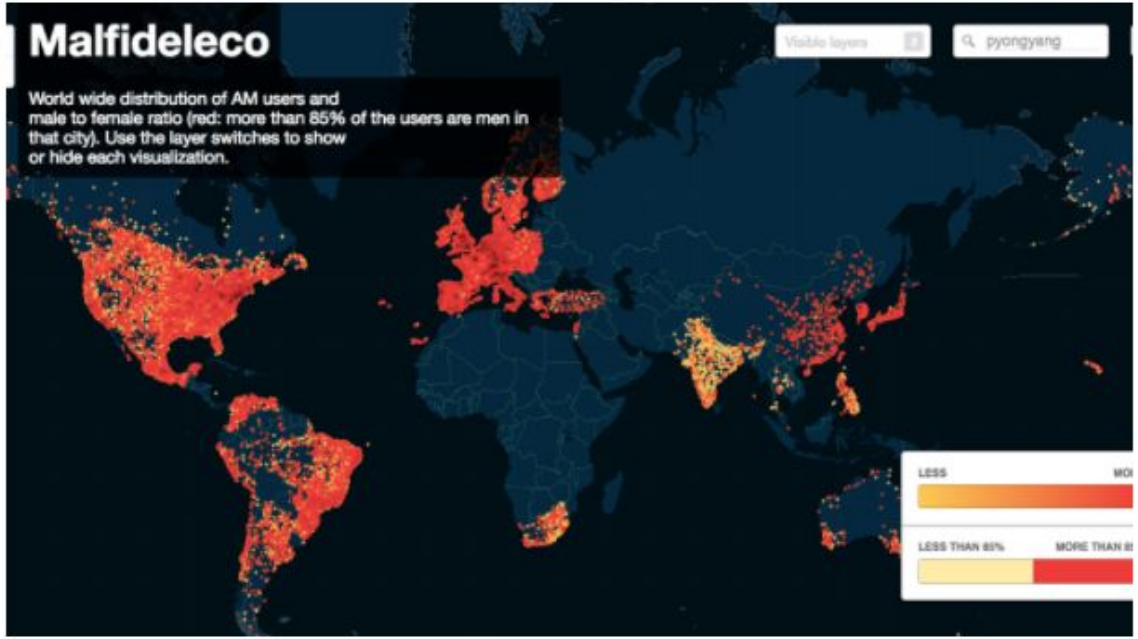


IMAGE: TECNILÓGICA

Obviously someone mapped the hacked personal information of 30 million AshleyMadison users. We're surprised it took this long.

Fechas

10/02/2017 | 13/02/2017 | 1 huésped

Tipo de habitación

- Casa entera
- Habitación privada
- Habitación compartida

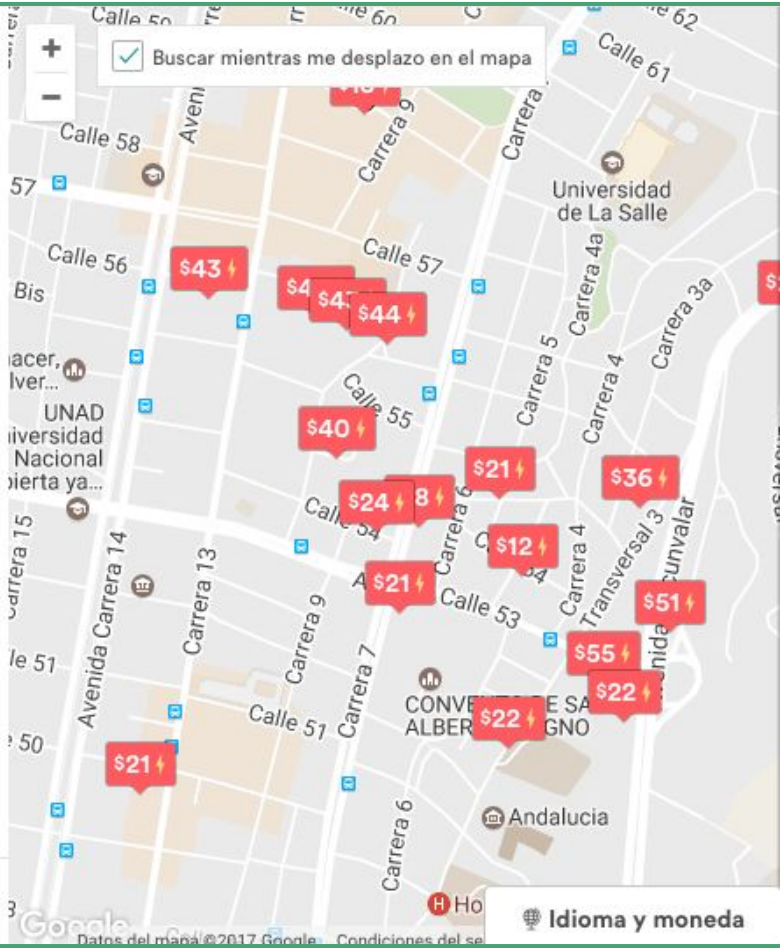
Rango de precios

$10 — Promedio — $1,000+

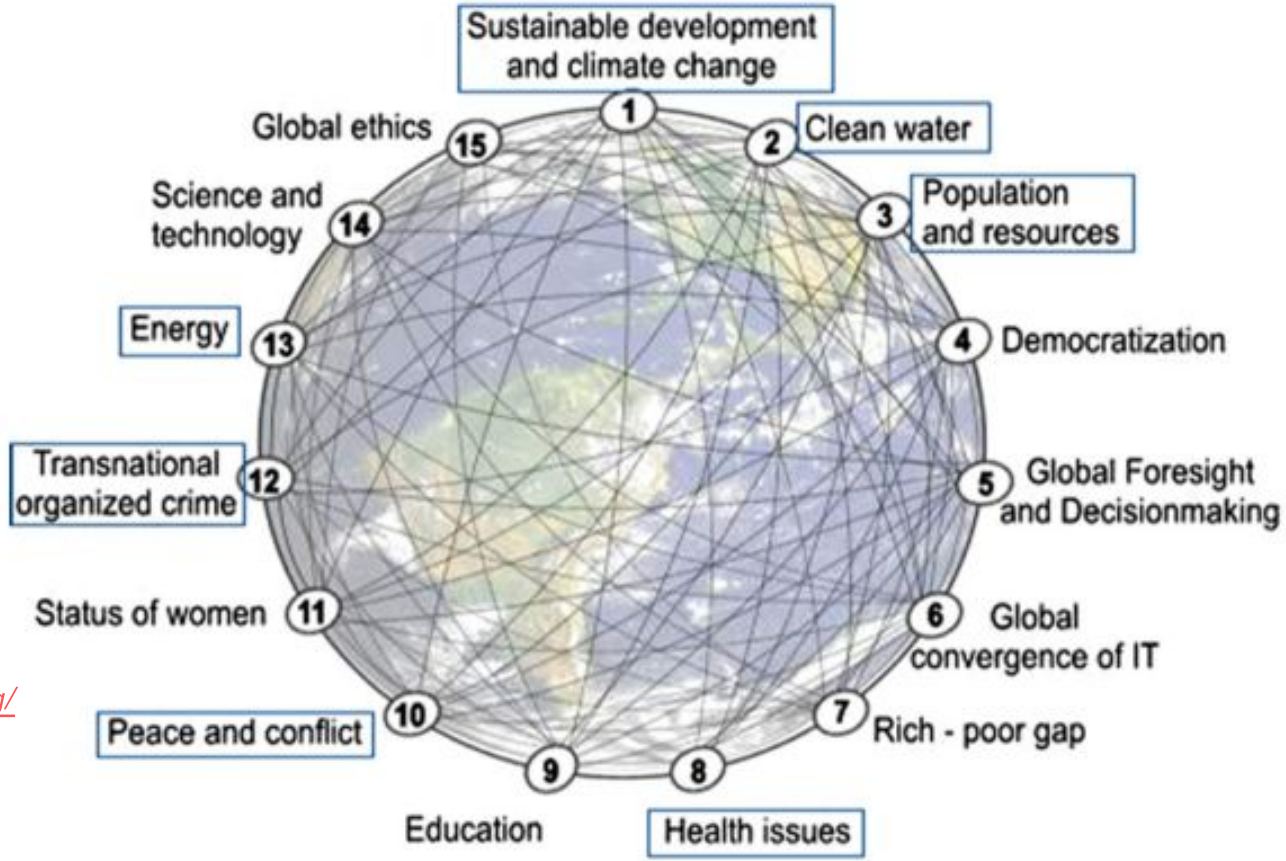
Reserva inmediata

Anuncios que no requieren que el anfitrión acepte la solicitud. Off

Filtros

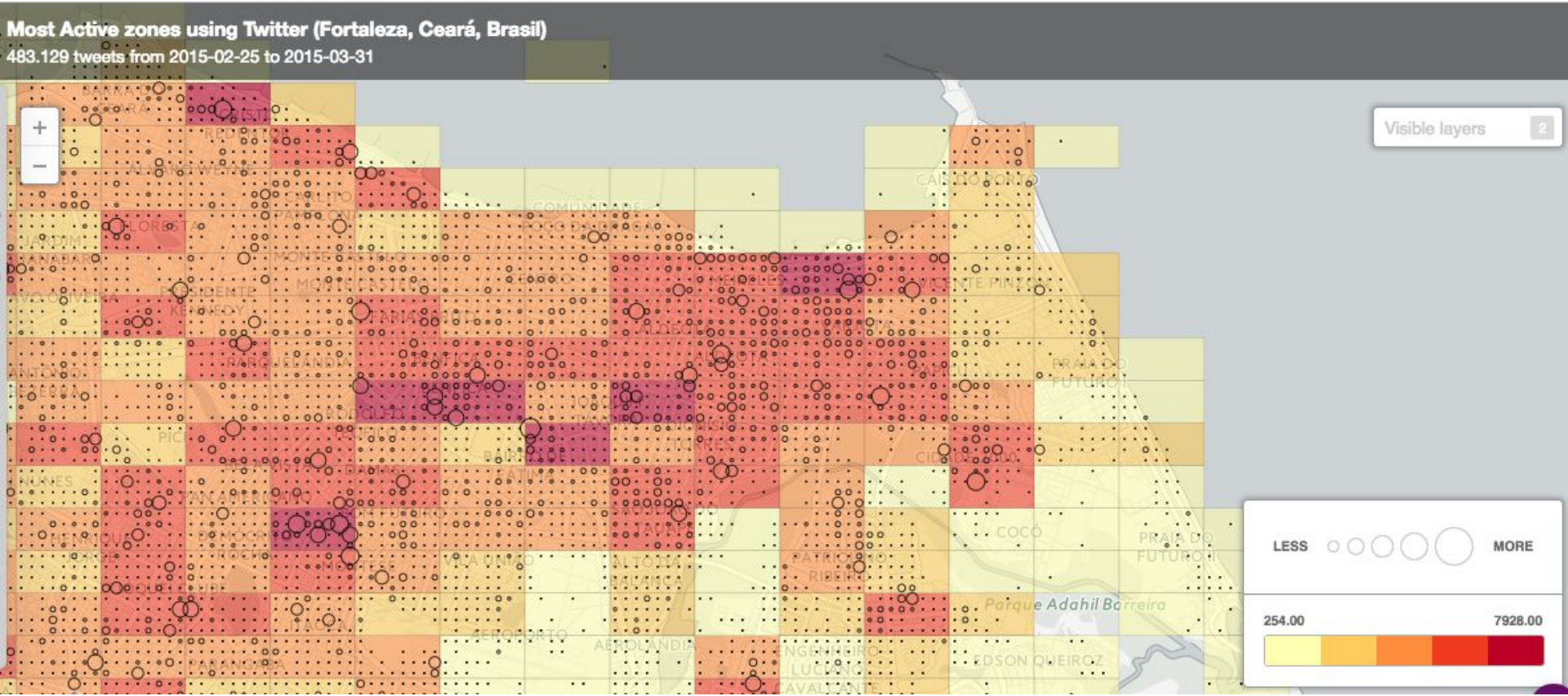


15 Global challenges facing humanity



http://www.millennium-project.org/millennium/challeng.html

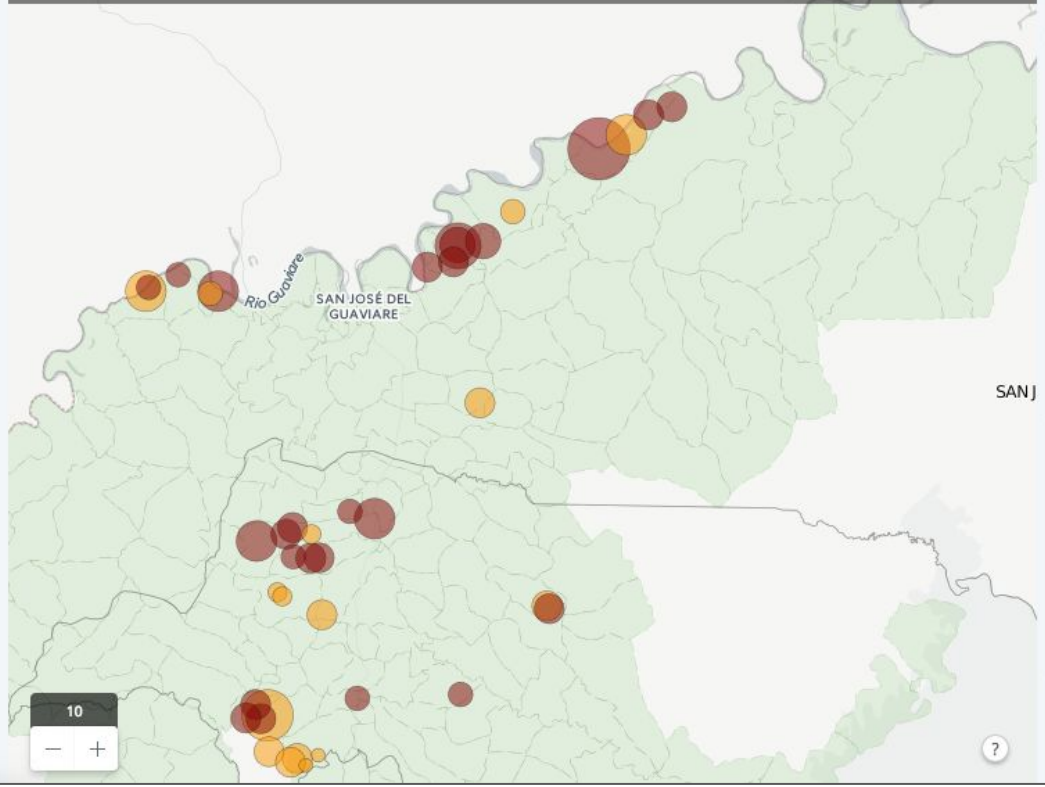
How?


Most Active zones using Twitter (Fortaleza, Ceará, Brasil)
483.129 tweets from 2015-02-25 to 2015-03-31
Visible layers
2
LESS
MORE
254.00
7928.00
Parque Adahil Barreira

People's Movement : Northern Colombia
SANTA MARTA
CIÉNAGA
BARRANQUILLA
TUBARÁ
MALAMBO
Ciénaga Grande de Santa Marta
BARANOA
PALMAR DE VARELA
SABANALARGA
LURUACO
CLEMENCIA
ARACATACA
FUNDACIÓN
REPELÓN
SALAMINA
PIVIJAY
VALLEDUPAR
VILLANUEVA
MANATÍ
CARTAGENA
CAMPO DE LA CRUZ
LA PAZ
TURBACO
SANTA LUCÍA
SAN DIEGO
ARJONA
CALAMAR
ARIGUANÍ
ALGARROBO
EL COPEY

Consolidado de Productores / Proveedores de ASOPROCACAO

Capa de puntos con la ubicación de 57 productores con datos generales de la finca y el cultivo; son datos (o puntos georefenciados) de parte de los más de 300 miembros de la organización Asoprocacao



Total Vendidos 2015

17,109 kg

Diferencia Año Anterior

151.28 %

Municipio

ALL SELECTED

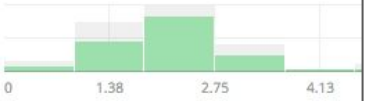
RETORNO

SAN JOSÉ DEL GUAVIARE

Área Cacaotera (Ha)

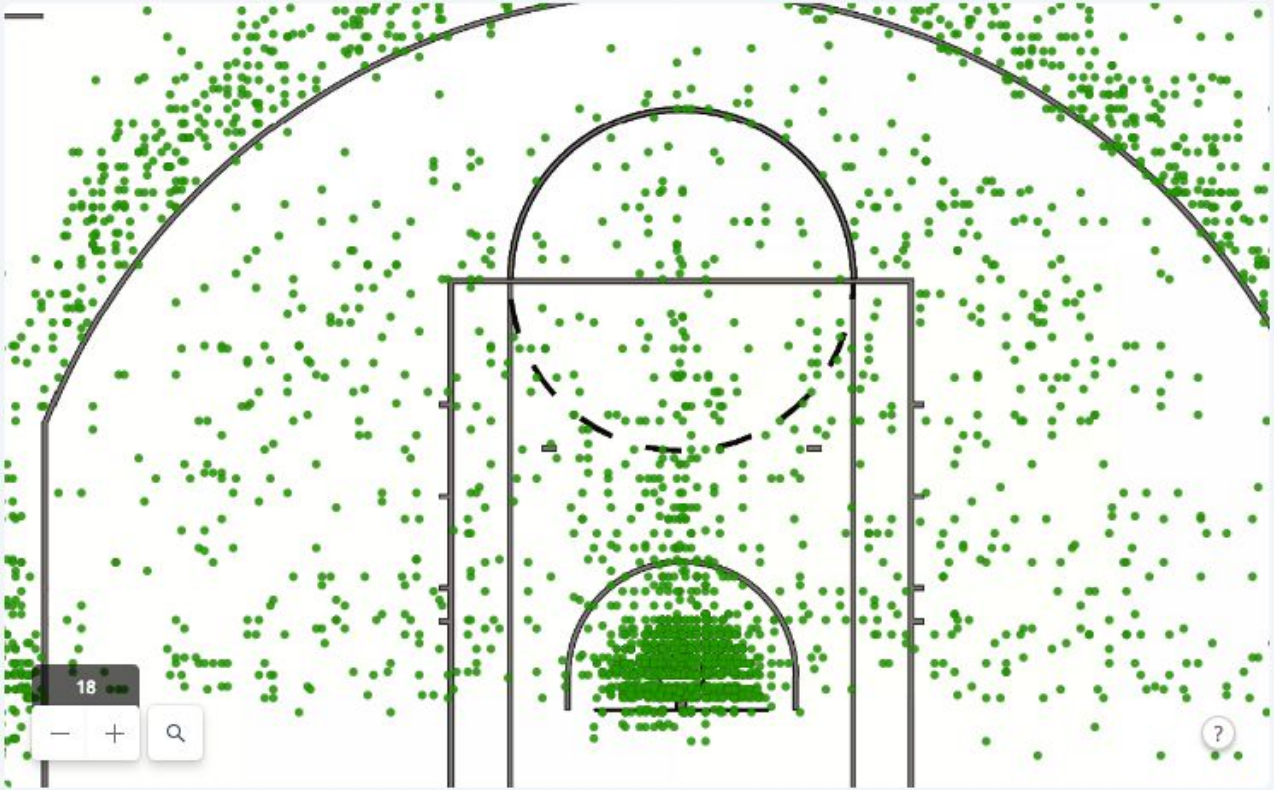
0 NULL ROWS 0 MIN 2.09 AVG 5.50 MAX

41 SELECTED



Precio Promedio por Kg

32 SELECTED



Number of Shots

2,976

Points

6,561

Player

ALL SELECTED

STEPHEN CURRY	609
KLAY THOMPSON	535
DRAYMOND GREEN	357
HARRISON BARNES	241
SHAUN LIVINGSTON	198
OTHER	1.0k

SEARCH IN 16 CATEGORIES

Shot Made

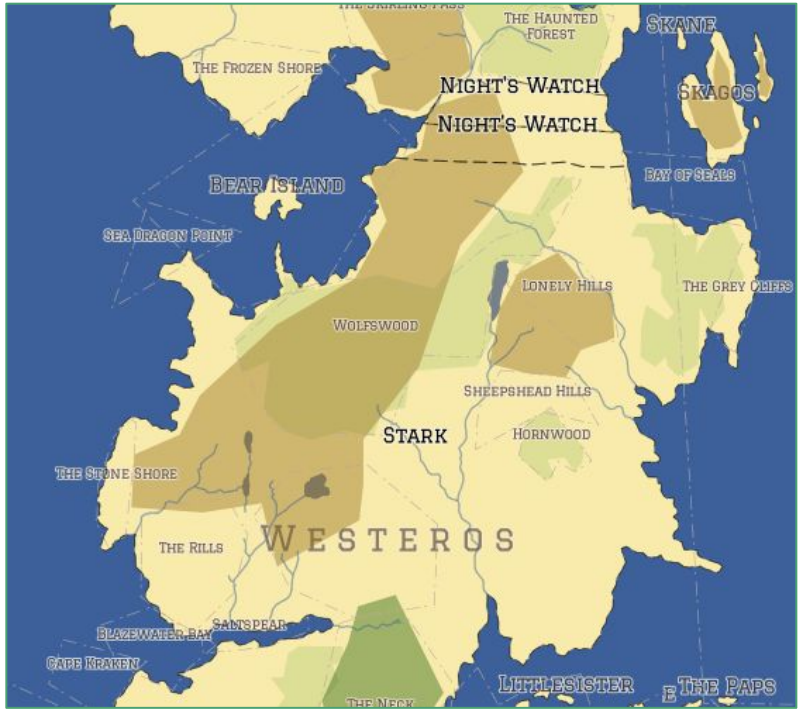
But...
What is Spatial Data?

Spatial Data

- Located on the surface of the earth
- Coordinate Systems

GIS (Geographic Information Systems)

- Body of Knowledge
- Tools
- Science



Spatial is special?

Yes:

- Multidimensional
- Voluminous
- Special methods for analysis
- Updating: Slow, complex and expensive

“Everything is related to everything else, but near things are more related than distant things”
Tobler, W. 1970. A

Spatial is special?

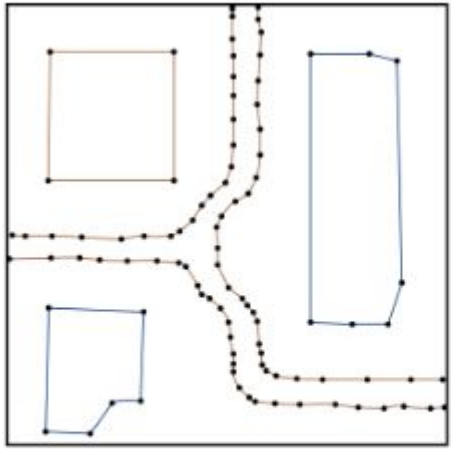
No:

- *“spatial is not special, it’s just another column in the database…” Michael Terner*

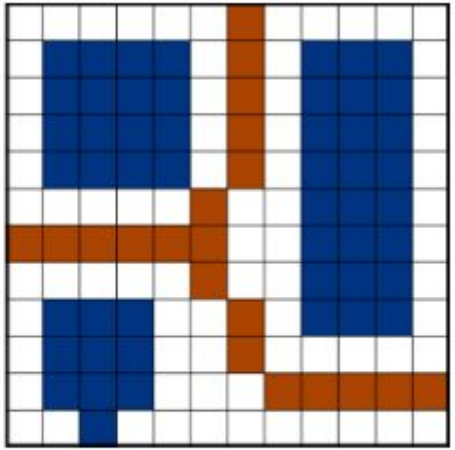
http://bit.ly/2jZAWav

Spatial Data: Vector vs Raster

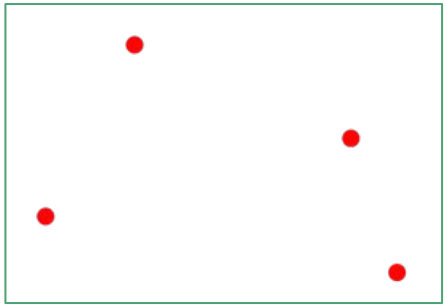
a)

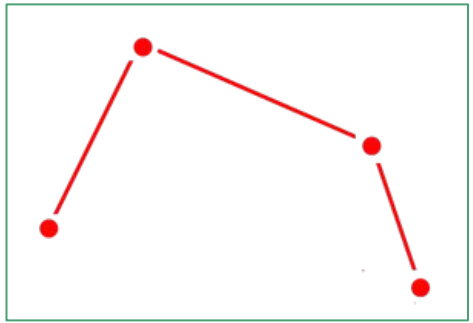


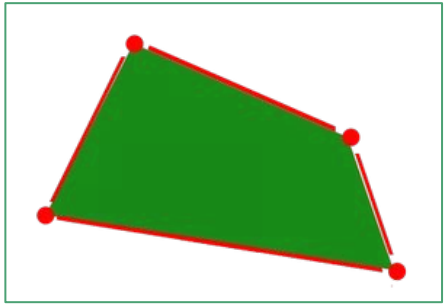
b)



Vector

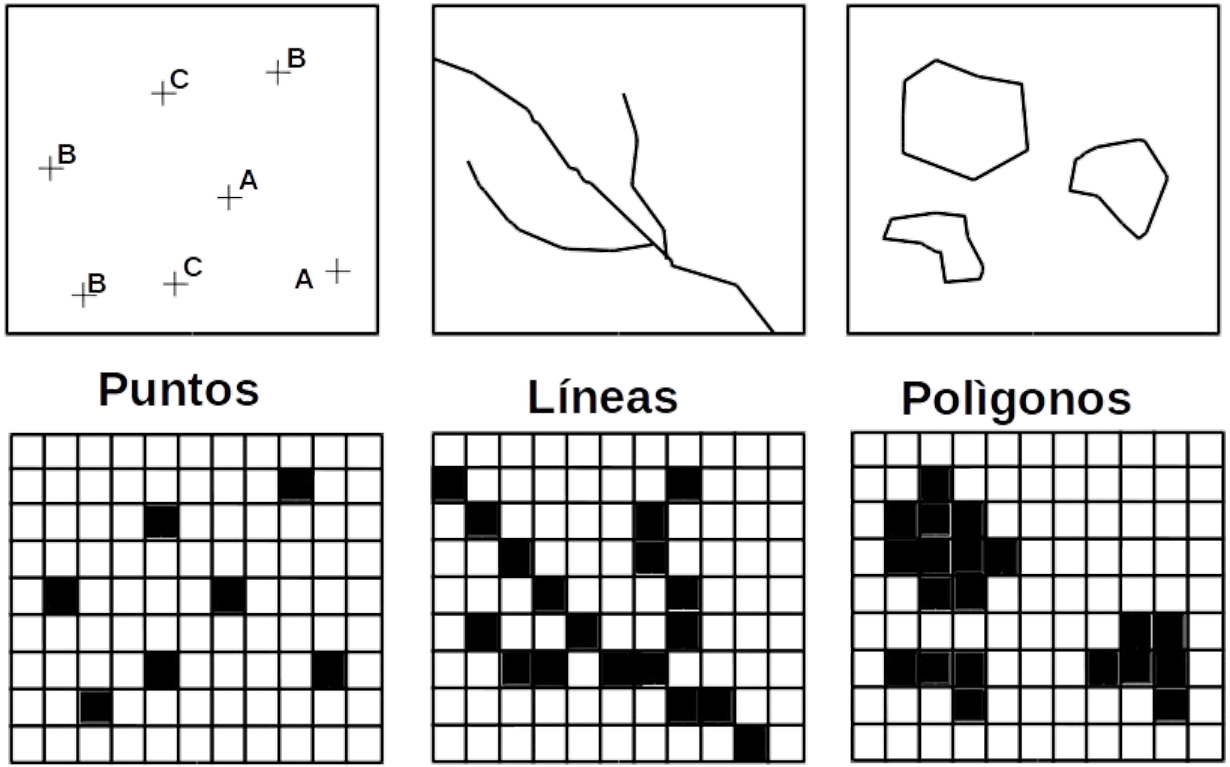






http://gisgeography.com/spatial-data-types-vector-raster/

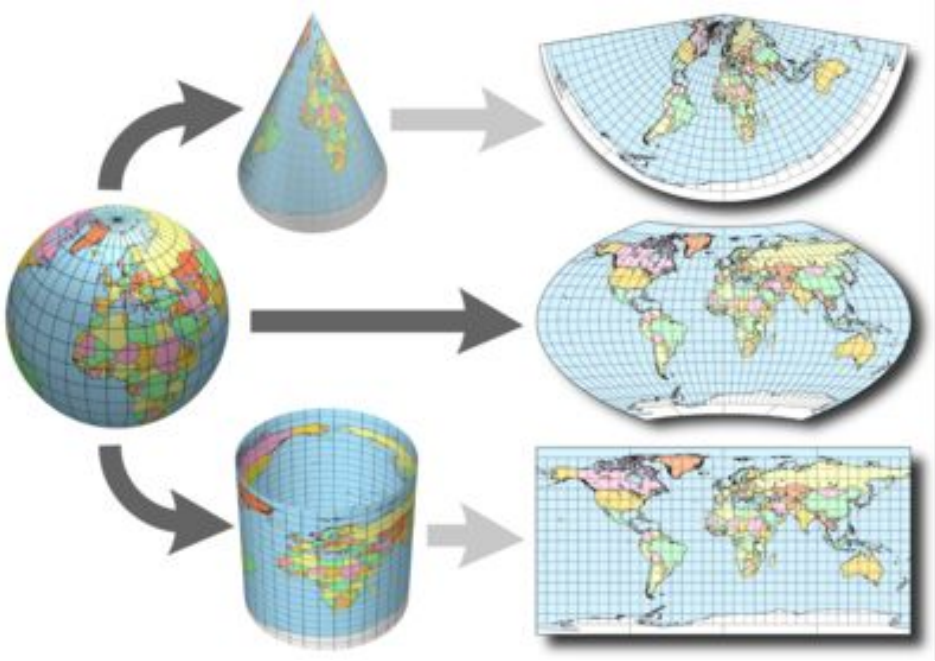
Spatial Data: Vector vs Raster

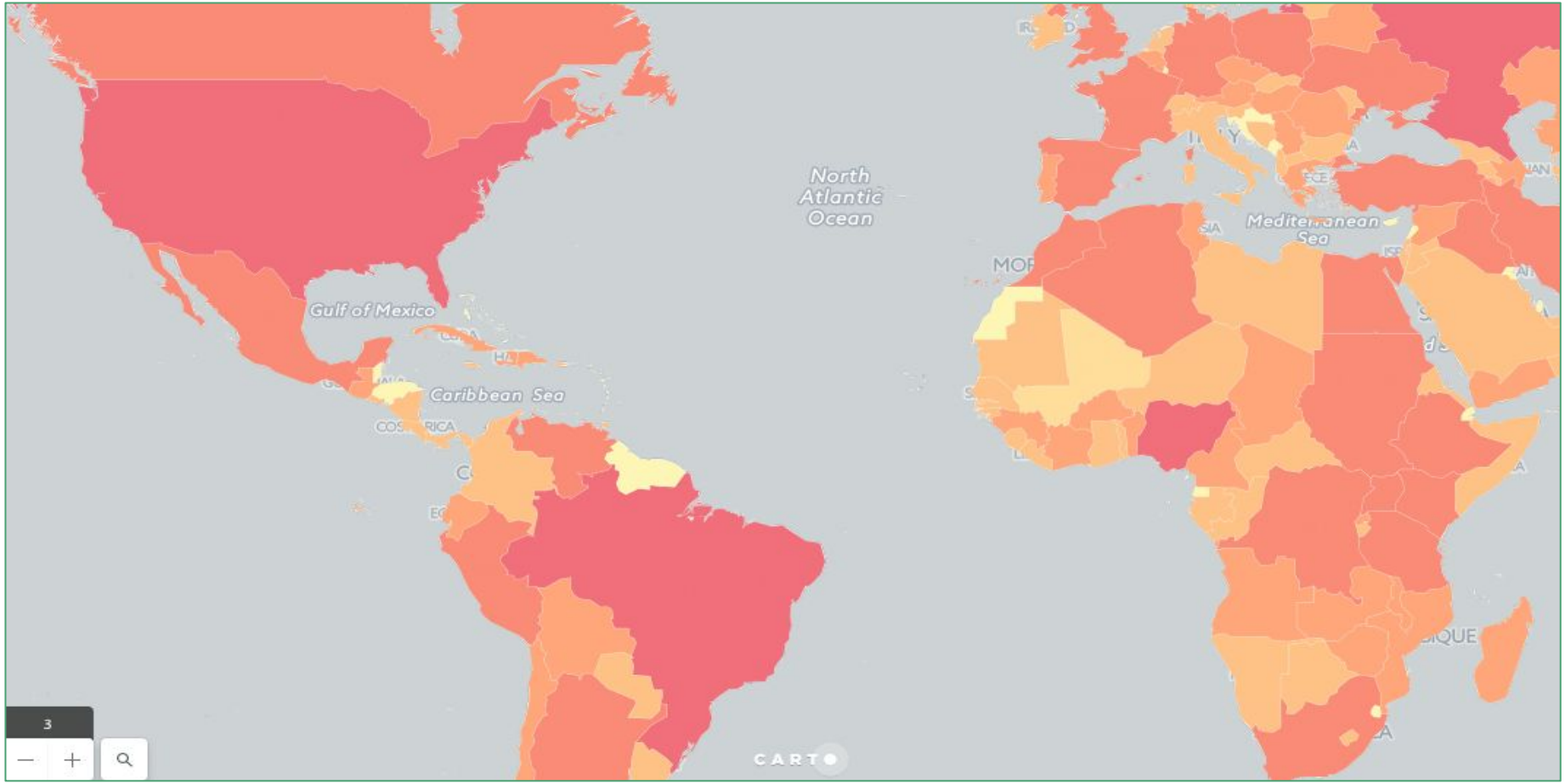


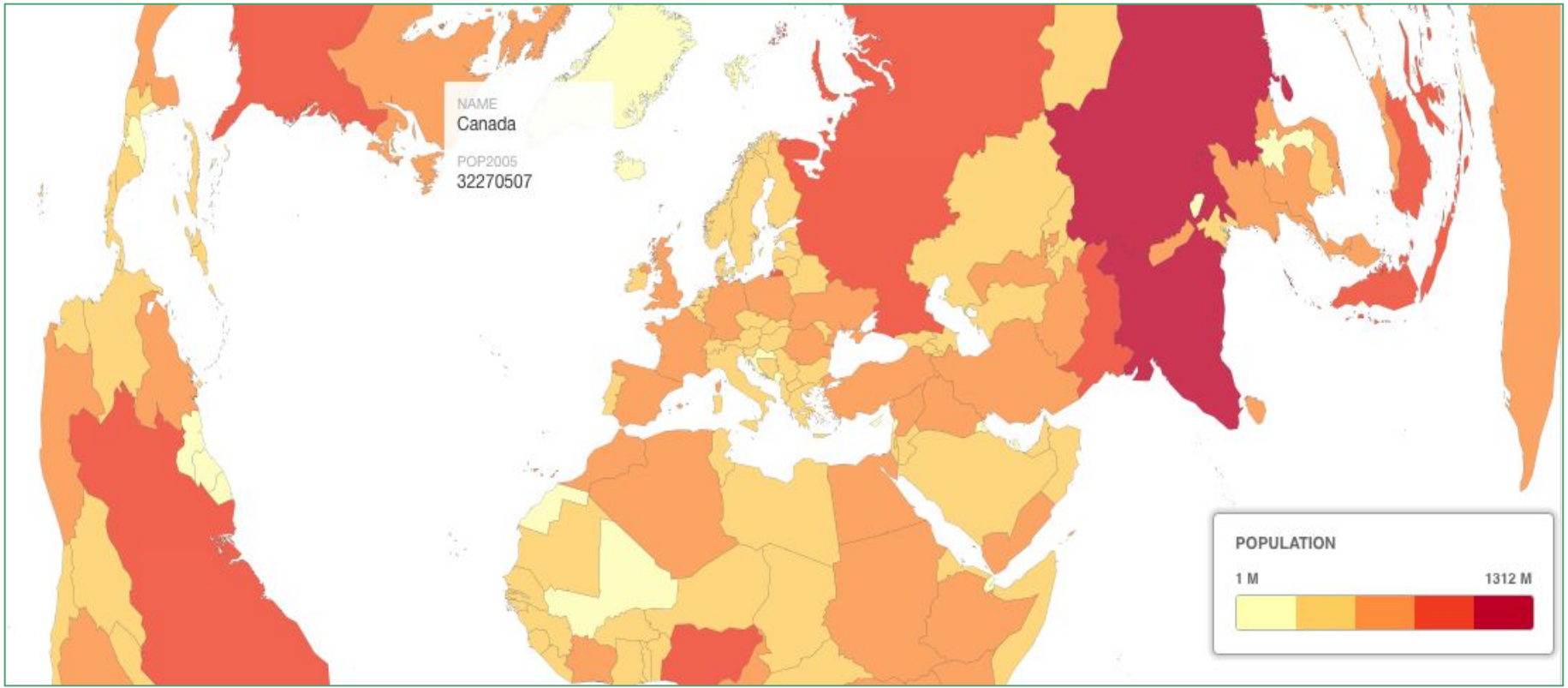
Map projection

"...formal process which converts features between a spherical or ellipsoidal surface and a projection surface, which is often flat..."

http://bit.ly/2bA9Szk




North Atlantic Ocean
Gulf of Mexico
Caribbean Sea
Mediterranean Sea
3
CARTO


NAME
Canada
POP2005
32270507
POPULATION
1 M
1312 M

Spatial Data - File Formats

...lots... (open and proprietary)...

- Shapefile
- Geojson
- KML
- Filegeodatabase
- Geotiff
- Geopackage
- Spatialite
- ... ...

Spatial is special?

...It can be complex...

*... using **python** it will be more fun and easy...*

Python?

Python for Geospatial Data

Lots of...

- Tools / Libraries
- Propietary / Open
- Desktop / Server
- Analysis / Visualization / ETL

Python for Geospatial Data

ESRI

- Arcpy
  - Desktop: Automation / Customization. E.g. CartoDB Toolbox : Import Data From to Carto
  - Server: Geoprocessing as a “Web Service”
- Arcgis Web : ArcGIS API for Python (2017)

QGIS

- Open source desktop GIS Tool written in C++, Python, Qt
- QGIS API E.g. CartoDB Plugin for QGis
- PyQGIS: Scripting using Python
- Server: Qgis Server python plugins

Python for Geospatial Data

- **CKAN** - web-based open source management system for the storage and distribution of **open data** (including geospatial data).
- **GeoDjango** - storing and manipulating geographic data using the Django ORM
- **Geonode** - web-based application and platform for developing geospatial information systems (**GIS**) and for deploying spatial data infrastructures (**SDI**)

Python for Geospatial Data

- pyshp - For reading and writing shapefiles (in pure Python)
- pyproj - For conversions between projections
- shapely - For geometry handling
- fiona - For making it easy to read/write geospatial data formats
- ogr/gdal - For reading, writing, and transforming geospatial data formats *
- Rasterio - reads and writes geospatial raster datasets

Show me the code

https://github.com/dersteppenwolf/pycon

Requirements

- Qgis (http://www.qgis.org/es/site/)

  pip install -r requirements.txt

  brew install mapnik

  brew install spatialindex

Using Python to Analyze Spatial Data - Pycon 2017 - Colombia

New Add topics

14 commits | 1 branch | 0 releases

Branch: master | New pull request | Create new file | Upload

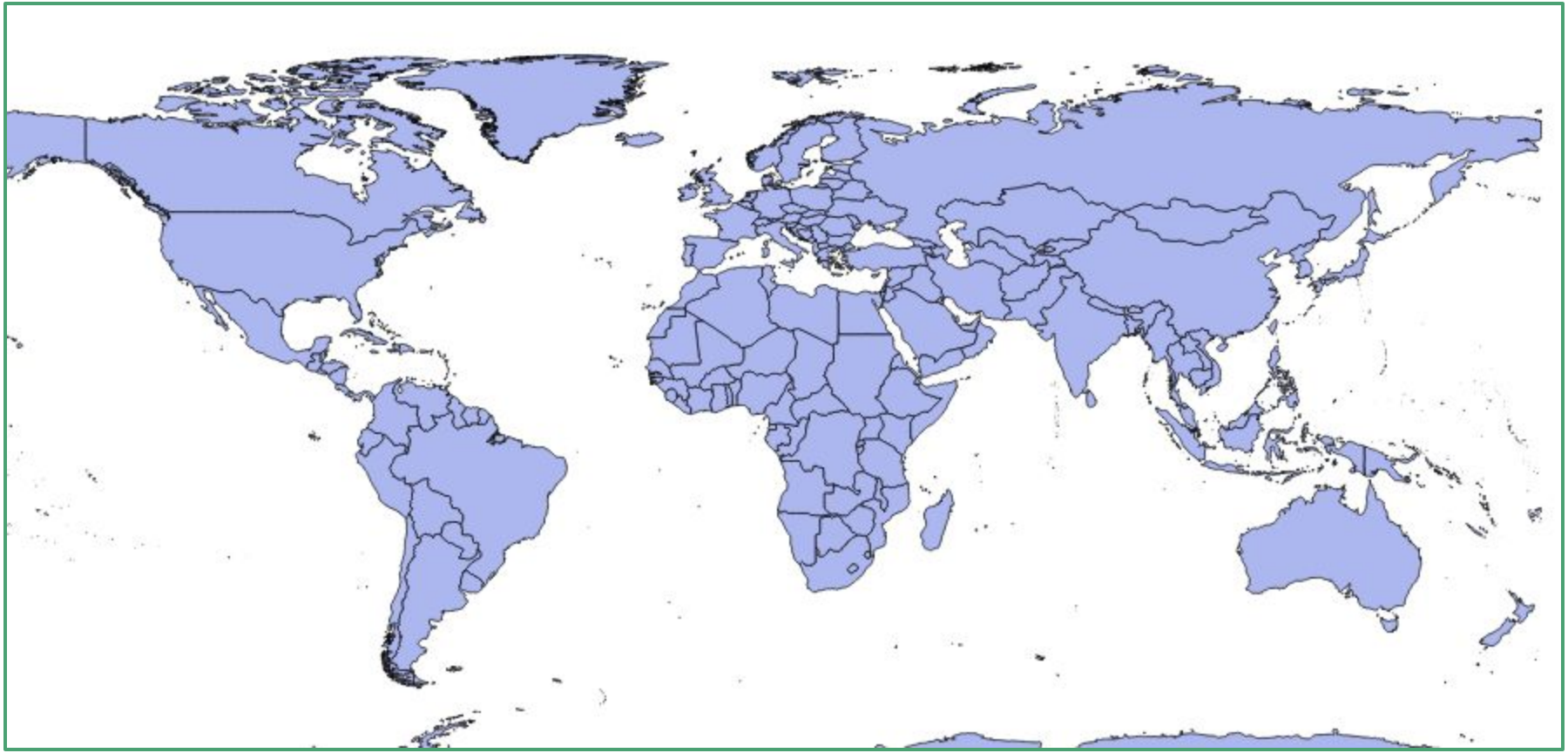
dersteppenwolf Comments

data	Spatial analysis
images	Composite
.gitignore	geopandas - Jupyter
01_describe.py	Fiona
02_transform.py	csv to shp
03_simple_viz.py	First commit
04_geocoding.py	First commit
05_reverse_geocoding.py	First commit
06_simple_mapnik.py	Complex mapnik xml
07_mapnik_from_xml.py	Complex mapnik xml

Describe Data

data/world_borders

TM_WORLD_BORDERS-0.3.dbf	30/7/2008 12:16	25 KB
TM_WORLD_BORDERS-0.3.prj	12/3/2008 13:11	145 bytes
TM_WORLD_BORDERS-0.3.shp	30/7/2008 12:16	6,5 MB
TM_WORLD_BORDERS-0.3.shx	30/7/2008 12:16	2 KB



TM_WORLD_BORDERS-0.3 :: Objetos totales: 246, filtrados: 246, seleccionados: 0

	FIPS	ISO2	ISO3	UN	NAME	AREA	POP2005	REGION	SUBREGION	LON
1	AC	AG	ATG	28	Antigua and …	44	83039	19	29	-61.783
2	AG	DZ	DZA	12	Algeria	238174	32854159	2	15	2.632
3	AJ	AZ	AZE	31	Azerbaijan	8260	8352021	142	145	47.395
4	AL	AL	ALB	8	Albania	2740	3153731	150	39	20.068
5	AM	AM	ARM	51	Armenia	2820	3017661	142	145	44.563
6	AO	AO	AGO	24	Angola	124670	16095214	2	17	17.544
7	AQ	AS	ASM	16	American Sa…	20	64051	9	61	-170.730
8	AR	AR	ARG	32	Argentina	273669	38747148	19	5	-65.167
9	AS	AU	AUS	36	Australia	768230	20310208	9	53	136.189
10	BA	BH	BHR	48	Bahrain	71	724788	142	145	50.562
11	BB	BB	BRB	52	Barbados	43	291933	19	29	-59.559
12	BD	BM	BMU	60	Bermuda	5	64174	19	21	-64.709

Fiona

- https://github.com/Toblerity/Fiona
- Fiona is OGR's neat and nimble API for Python programmers.
- Fiona does reading and writing data formats. For this it uses OGR, the most popular open-source conversion system
- The OGR Simple Features Library is a C++ open source library providing read (and sometimes write) access to a variety of vector file formats including ESRI Shapefiles, S-57, SDTS, PostGIS, Oracle Spatial, and Mapinfo mid/mif and TAB formats.

```python
import fiona
import pprint

data_path = "data/world_borders/TM_WORLD_BORDERS-0.3.shp"
c = fiona.open(data_path)

print c.driver
# 'ESRI Shapefile'

#  coordinate reference system (CRS)
print c.crs
# {'init': u'epsg:4326'}

# number of records
print len(c)
# 246

# minimum bounding rectangle (MBR)
print c.bounds
# (-179.99999999999997, -90.0, 180.0, 83.62359600000008)

pprint.pprint(c.schema)
'''
{'geometry': 'Polygon',
 'properties': OrderedDict([(u'FIPS', 'str:2'), (u'ISO2', 'str:2'),
 (u'ISO3', 'str:3'), (u'UN', 'int:3'), (u'NAME', 'str:50'),
 (u'AREA', 'int:7'), (u'POP2005', 'int:10'), (u'REGION', 'int:3'),
 (u'SUBREGION', 'int:3'), (u'LON', 'float:8.3'), (u'LAT', 'float:7.3')])}
'''
```

Data Format Conversion

From CSV to SHP

Branch: master ▾ pycon / data / earthquakes.csv Find file Copy path

dersteppenwolf First commit 612ee28 a minute ago

1 contributor

2415 lines (2414 sloc) | 194 KB Raw Blame History

Search this file...

1	DateTime	Latitude	Longitude	Depth	Magnitude	MagType	NbStations	Gap	Distance	RMS	Source	EventID
2	2002/01/01 10:39:06.82	-55.2140	-129.0000	10.00	6.00	Mw	78			1.07	NEI	200201014017
3	2002/01/01 11:29:22.73	6.3030	125.6500	138.10	6.30	Mw	236			0.90	NEI	200201014018
4	2002/01/02 14:50:33.49	-17.9830	178.7440	665.80	6.20	Mw	215			1.08	NEI	200201024034
5	2002/01/02 17:22:48.76	-17.6000	167.8560	21.00	7.20	Mw	427			0.90	NEI	200201024041
6	2002/01/03 07:05:27.67	36.0880	70.6870	129.30	6.20	Mw	431			0.87	NEI	200201034024

Shapely

- https://github.com/Toblerity/Shapely
- Manipulation and analysis of geometric objects in the Cartesian plane
- With Shapely, you can do things like buffers, unions, intersections, centroids, convex hulls, and lots more.

```python
import csv
from shapely.geometry import Point, mapping
from fiona import collection
from fiona.crs import from_epsg

schema = { 'geometry': 'Point', 'properties': { 'depth': 'float:7.3', 'magnitude' : 'float:7.3' } }

with collection("data/shp/earthquakes.shp", "w", crs=from_epsg(4326),
                driver='ESRI Shapefile', schema=schema) as output:
    with open('data/earthquakes.csv', 'rb') as f:
        reader = csv.DictReader(f)
        for row in reader:
            point = Point(float(row['Longitude']), float(row['Latitude']))
            output.write({
                'properties': {
                    'depth': row['Depth'],'magnitude': row['Magnitude']
                },
                'geometry': mapping(point)
            })
```

Geocoding

What is geocoding?

- Geocoding is the process of transforming a description of a location—such as a pair of coordinates, an address, or a name of a place—to a location on the earth's surface. http://arcg.is/2kUedk7

GeoPy

- https://github.com/geopy/geopy
- geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources
- geopy includes geocoder classes for the OpenStreetMap Nominatim, ESRI ArcGIS, Google Geocoding API (V3), Baidu Maps, Bing Maps API, Mapzen Search, Yandex, IGN France, GeoNames, NaviData, OpenMapQuest, What3Words, OpenCage, SmartyStreets, geocoder.us, and GeocodeFarm geocoder services.

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from geopy.geocoders import Nominatim
geolocator = Nominatim()
location = geolocator.geocode("carrera 11 #82 - 71, Bogotá, Colombia")

print(location.address)
# Carrera 11, El Nogal, Chapinero, Bogotá, Calle 82 No 10 - 33 Piso 5, Colombia

print((location.latitude, location.longitude))
# (4.6678326, -74.0514441)

print(location.raw)
'''
{u'display_name': u'Carrera 11, El Nogal, Chapinero, Bogot\xe1, Calle 82 No 10 - 33 Piso 5, Colombia',
u'importance': 0.42, u'place_id': u'145766568', u'lon': u'-74.0514441', u'lat': u'4.6678326',
u'osm_type': u'way',
u'licence': u'Data \xa9 OpenStreetMap contributors, ODbL 1.0. http://www.openstreetmap.org/copyright',
u'osm_id': u'375277348', u'boundingbox': [u'4.6678202', u'4.6678741', u'-74.0515172', u'-74.0514222'],
u'type': u'secondary', u'class': u'highway'}
'''
```

Reverse Geocoding

```python
from geopy.geocoders import Nominatim
geolocator = Nominatim()
location = geolocator.reverse("4.638194, -74.0663803")
print(location.address)
# surti libros, Calle 51, Mariscal Sucre, Chapinero, Bogotá, 111311, Colombia
```

Visualize?

Mapnik

- http://mapnik.org/
- the core of geospatial visualization & processing
- mapnik combines pixel-perfect image output with lightning-fast cartographic algorithms, and exposes interfaces in C++, Python, and Node.

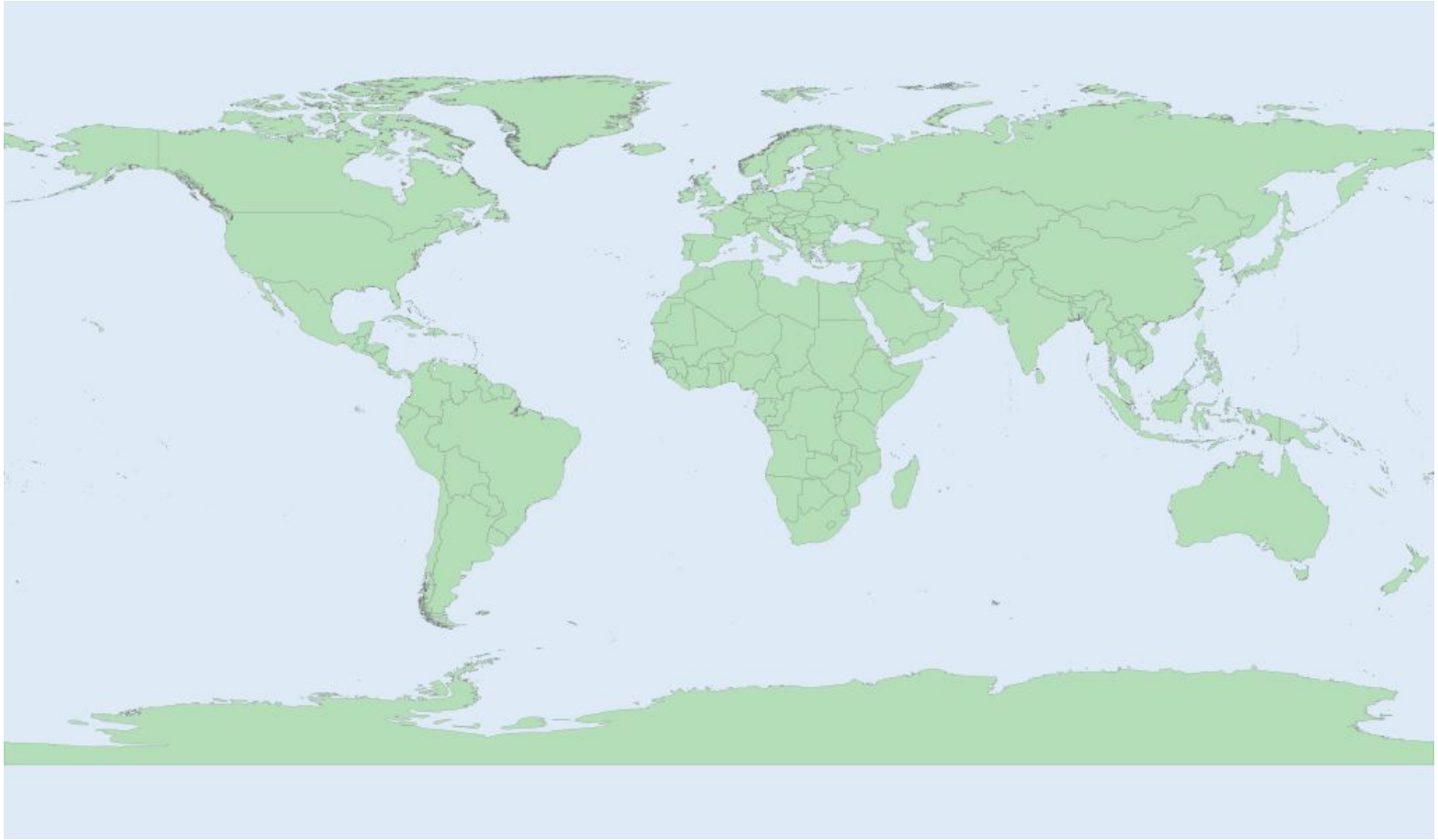
Mapnik

- Installing Mapnik on OS X with Homebrew https://github.com/mapnik/mapnik/wiki/MacInstallation_Homebrew
  - brew install mapnik
- Python bindings for mapnik https://github.com/mapnik/python-mapnik

- Stacks built with mapnik: OpenStreetMap , Mapbox , CartoDB , Stamen , MapQuest , Kosmtik

Mapnik Basics

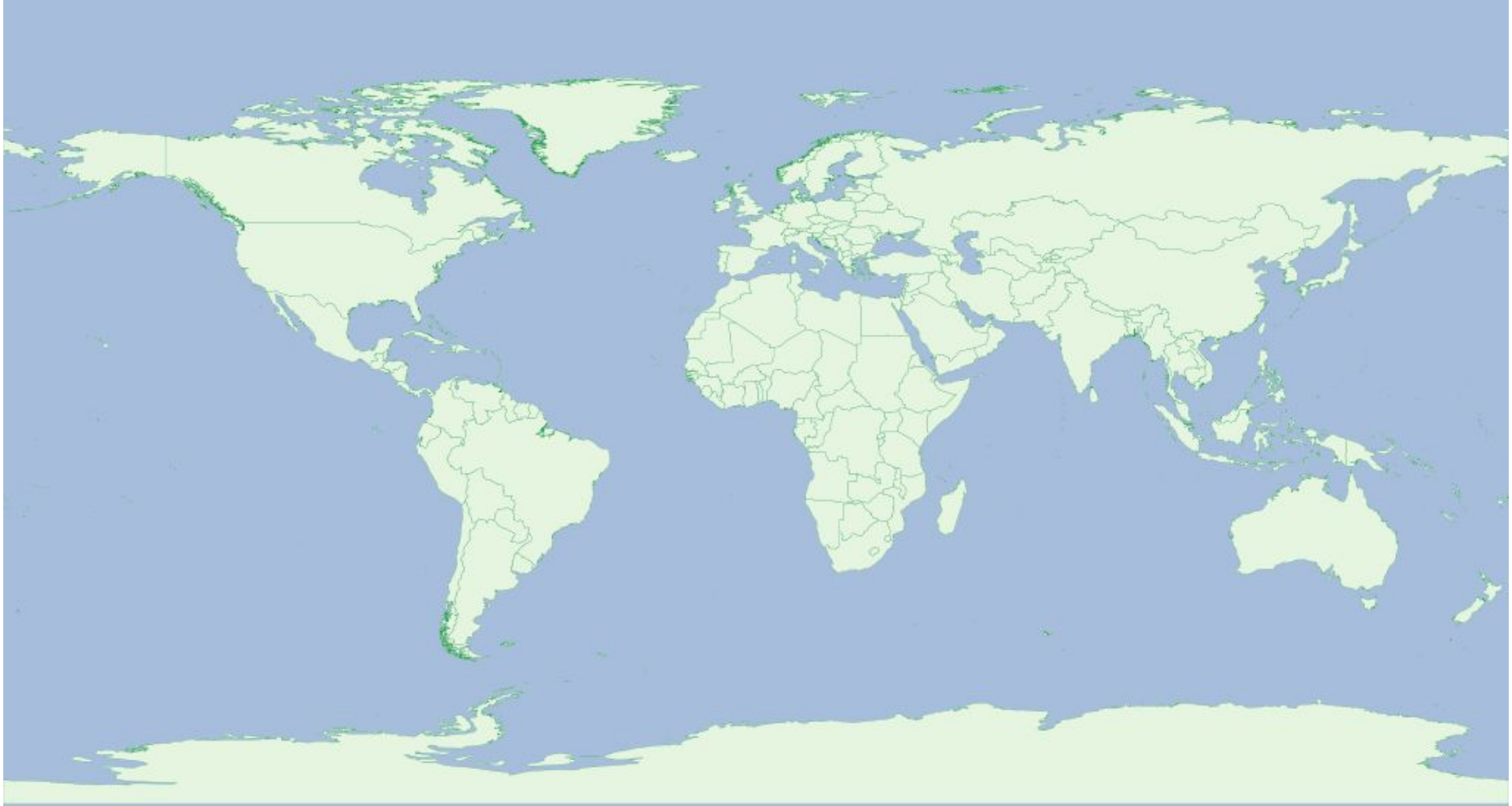
```python
import mapnik

data_path = "data/world_borders/TM_WORLD_BORDERS-0.3.shp"

polygon_symbolizer = mapnik.PolygonSymbolizer()
polygon_symbolizer.fill = mapnik.Color('#a1d99b')
polygon_symbolizer.fill_opacity = 0.7

line_symbolizer = mapnik.LineSymbolizer()
line_symbolizer.stroke = mapnik.Color('rgb(50%,50%,50%)')
line_symbolizer.stroke_width = 0.2

rule = mapnik.Rule()
rule.symbols.append(polygon_symbolizer)
rule.symbols.append(line_symbolizer)

style = mapnik.Style()
style.rules.append(rule)

layer = mapnik.Layer("mapLayer")
layer.datasource = mapnik.Shapefile(file=data_path)
layer.styles.append("mapStyle")

map = mapnik.Map(1024, 600)
map.background = mapnik.Color("#deebf7")
map.append_style("mapStyle", style)
map.layers.append(layer)
map.zoom_all()
mapnik.render_to_file(map, "images/simple_map_1.png", "png")
```



Mapnik - XML

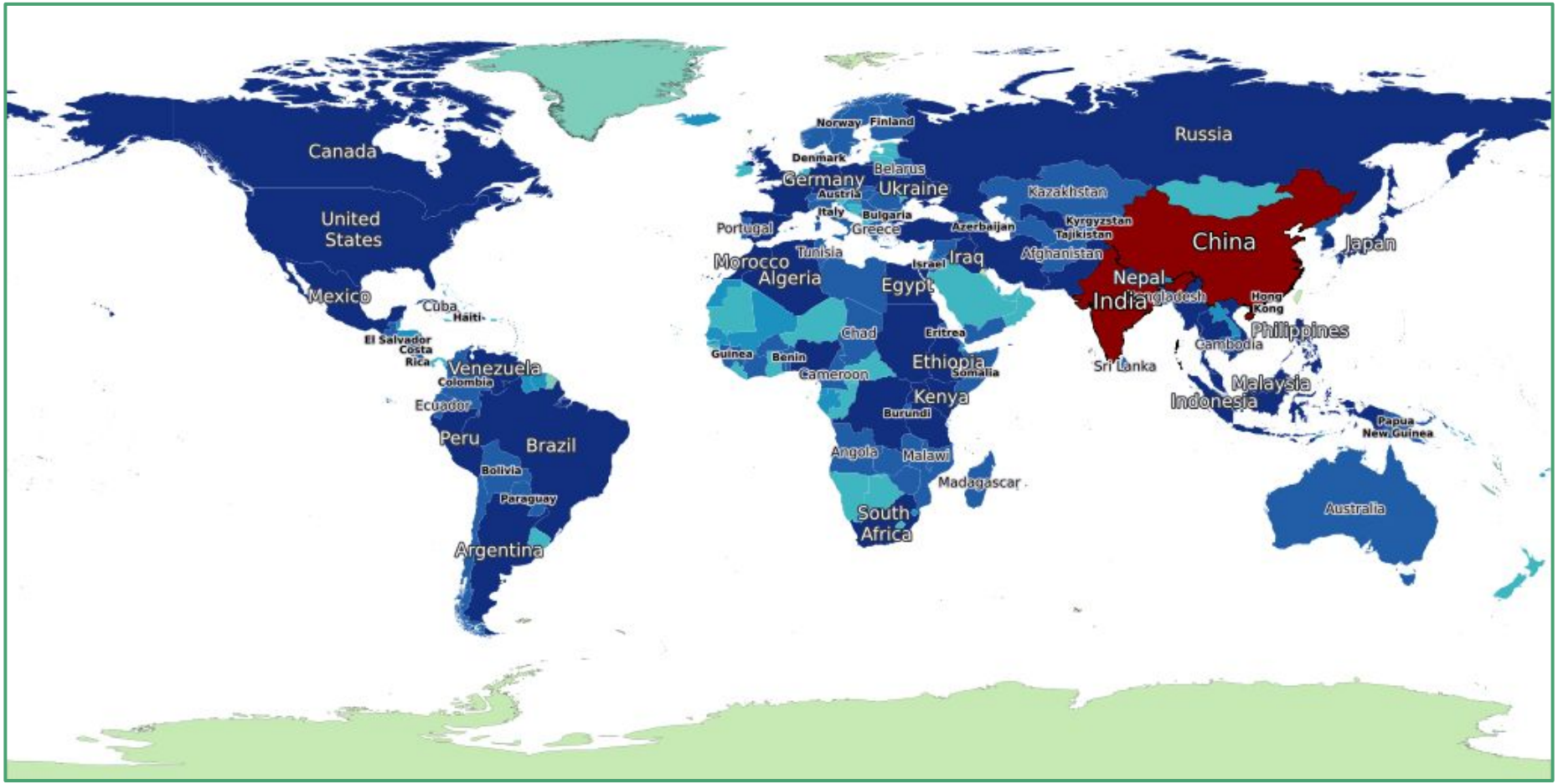
```python
import mapnik
m = mapnik.Map(1024,600)
mapnik.load_map(m, "mapnik_world.xml")
m.zoom_all()
mapnik.render_to_file(m, "images/simple_map_2.png")
```

```xml
<Map background-color="#a6bddb" srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs">

  <Style name="My Style">
    <Rule>
      <PolygonSymbolizer fill="#e5f5e0" />
      <LineSymbolizer stroke="#31a354" stroke-width="0.2" />
    </Rule>
  </Style>

  <Layer name="world" srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs">
    <StyleName>My Style</StyleName>
    <Datasource>
      <Parameter name="type">shape</Parameter>
      <Parameter name="file">data/world_borders/TM_WORLD_BORDERS-0.3.shp</Parameter>
    </Datasource>
  </Layer>
```



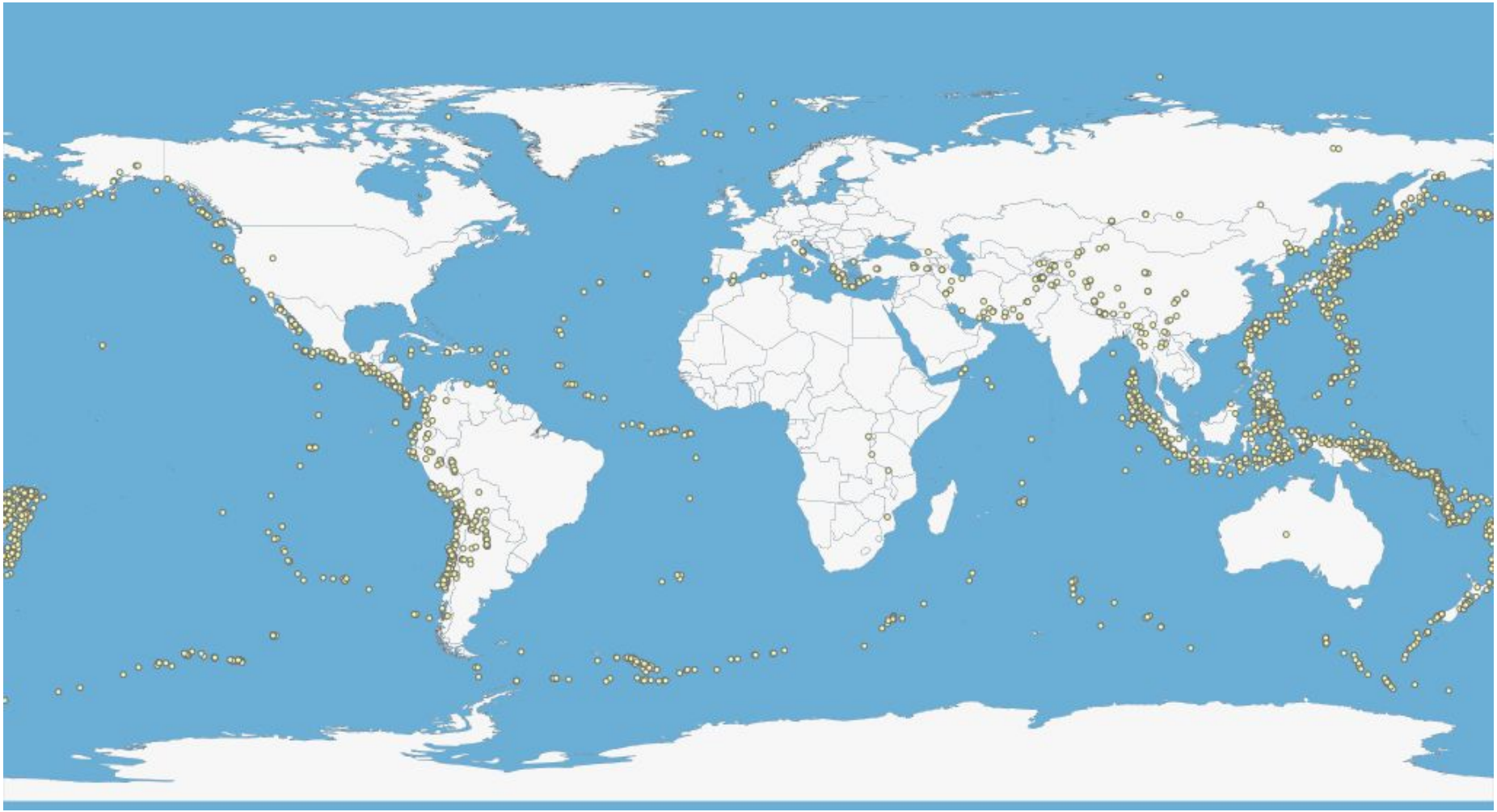
Mapnik Complex Styles

```
7       <Rule>
8        <Filter>[POP2005] &gt; -1 and [POP2005] &lt; 15000</Filter>
9        <PolygonSymbolizer fill="#c7e9b4"/>
10       <LineSymbolizer stroke="black" stroke-width=".1"/>
11      </Rule>
12
13      <Rule>
14       <Filter>[POP2005] &gt;= 15000 and [POP2005] &lt; 255000</Filter>
15       <PolygonSymbolizer fill="#7fcdbb"/>
16       <LineSymbolizer stroke="black" stroke-width=".1"/>
17      </Rule>
18
19      <Rule>
20       <Filter>[POP2005] &gt;= 255000 and [POP2005] &lt; 1300000</Filter>
21       <PolygonSymbolizer fill="#1d91c0"/>
22      </Rule>
23
24      <Rule>
25       <Filter>[POP2005] &gt;= 1300000 and [POP2005] &lt; 4320000</Filter>
26       <PolygonSymbolizer fill="#41b6c3"/>
27      </Rule>
28
29      <Rule>
30       <Filter>[POP2005] &gt;= 4320000 and [POP2005] &lt; 9450000</Filter>
31       <PolygonSymbolizer fill="#225ea8"/>
32      </Rule>
```


Canada
United States
Mexico
Cuba
Haiti
El Salvador
Costa Rica
Venezuela
Colombia
Ecuador
Peru
Brazil
Bolivia
Paraguay
Argentina
Norway
Finland
Denmark
Germany
Austria
Belarus
Ukraine
Italy
Bulgaria
Portugal
Greece
Azerbaijan
Russia
Kazakhstan
Kyrgyzstan
Tajikistan
Afghanistan
China
Japan
Tunisia
Morocco
Algeria
Israel
Iraq
Egypt
Nepal
India
Bangladesh
Hong Kong
Philippines
Cambodia
Chad
Eritrea
Guinea
Benin
Cameroon
Ethiopia
Somalia
Sri Lanka
Malaysia
Indonesia
Kenya
Burundi
Papua New Guinea
Angola
Malawi
Madagascar
South Africa
Australia

Mapnik Markers

```
10    <Style name="earthquake Style">
11      <Rule>
12        <MarkersSymbolizer fill="#ffffb2" opacity=".9"
13          width="4" height="4"
14          stroke="#636363" stroke-width="1"
15          stroke-opacity=".9"
16          placement="point" allow-overlap="true"
17          marker-type="circle"/>
18      </Rule>
19    </Style>
20
21
22    <Layer name="world" srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs">
23      <StyleName>world Style</StyleName>
24      <Datasource>
25        <Parameter name="type">shape</Parameter>
26        <Parameter name="file">data/world_borders/TM_WORLD_BORDERS-0.3.shp</Parameter>
27      </Datasource>
28    </Layer>
29
30    <Layer name="earthquakes" srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs">
31      <StyleName>earthquake Style</StyleName>
32      <Datasource>
33        <Parameter name="type">shape</Parameter>
34        <Parameter name="file">data/shp/earthquakes.shp</Parameter>
35      </Datasource>
36    </Layer>
```



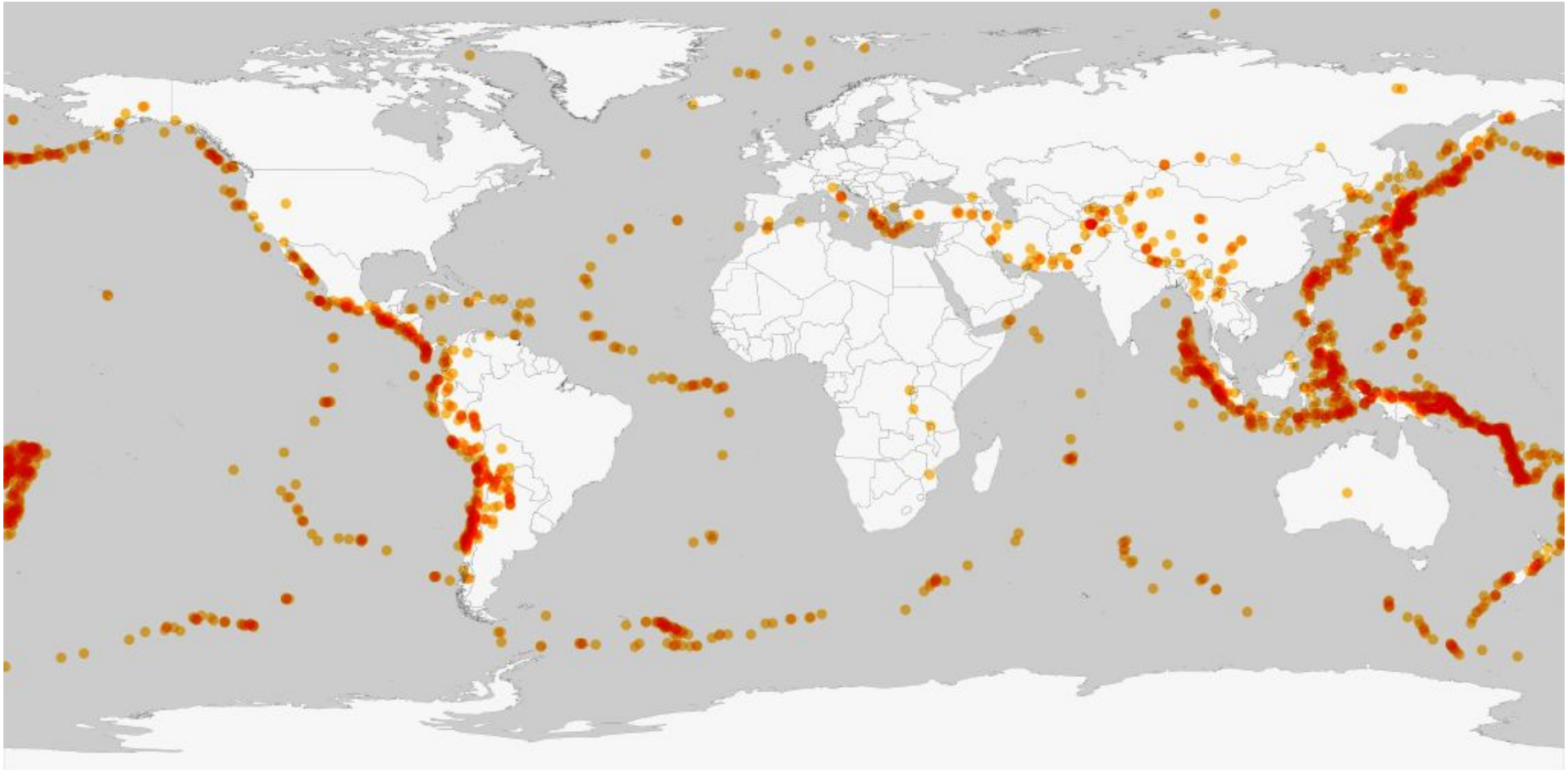
Mapnik Composite

Compositing operations affect the way colors and textures of different elements and styles interact with each other.

Two main categories: color and alpha

E.g. **Multiply** literally multiplies the color of the top layer by the color of each layer beneath, which usually means overlapping areas become darker.

```
10    <Style name="earthquake Style">
11      <Rule>
12        <MarkersSymbolizer
13          fill="#FFB927"
14          opacity="0.9"
15          width="7" height="7"
16          stroke="#636363" stroke-width="0"
17          stroke-opacity="0"
18          placement="point" allow-overlap="true"
19          comp-op="multiply"/>
20      </Rule>
21    </Style>
```



What is Spatial Analysis?

Spatial Analysis

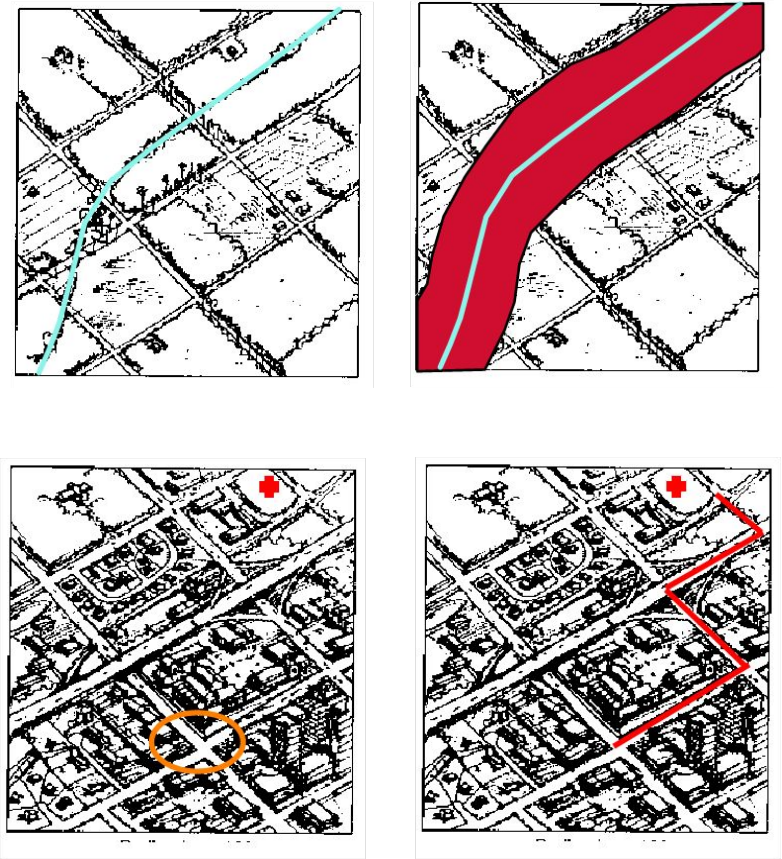
Geospatial data is more than maps!

What is Geoprocessing:?

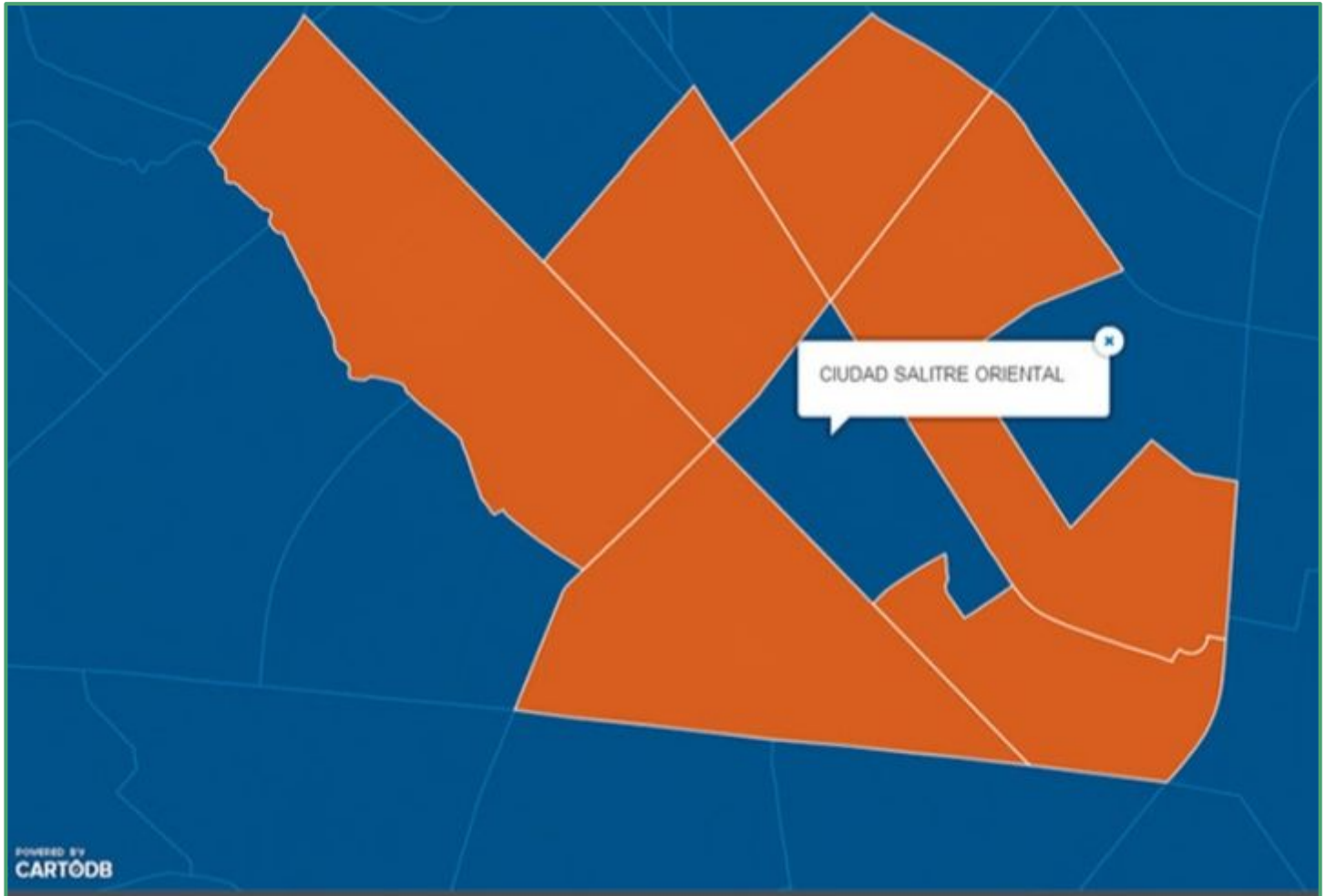
- Geoprocessing is any GIS operation used to **manipulate data**.
- A typical geoprocessing operation takes an **input** dataset, performs an **operation** on that dataset, and returns the result of the operation as an output dataset, also referred to as **derived data.**
- Common geoprocessing operations: geographic feature overlay, feature selection and analysis, topology processing, and data conversion.
- Geoprocessing allows you to define, manage, and analyze geographic information used to make decisions.
- http://bit.ly/2k8I3P8

Spatial Analysis

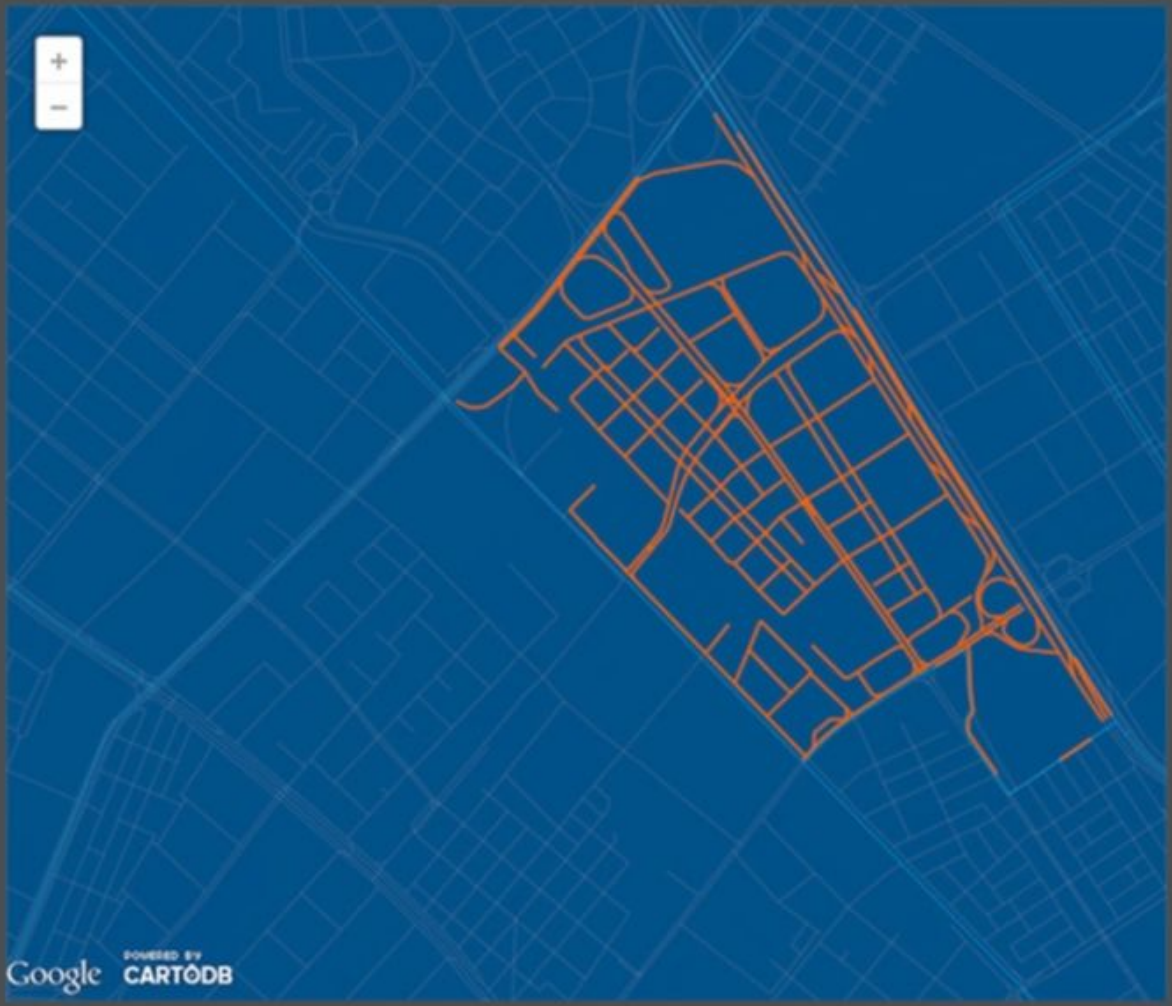
- Spatial analysis includes any of the formal techniques which study entities using their topological, geometric, or geographic properties.



Spatial Analysis



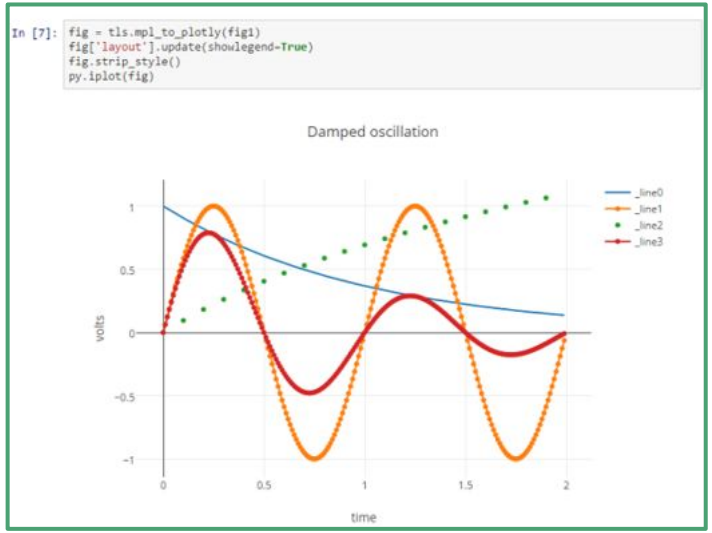
Adyacencia UPZ vecinas de Ciudad Salitre Oriental.



Contenencia: Vías que están dentro de Ciudad Salitre Oriental.

Jupyter

- http://jupyter.org/
- web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text
- Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more



Geopandas

- https://github.com/geopandas/geopandas
- **pandas** is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.
- **GeoPandas** is a project to add support for **geographic data** to **pandas** objects.

PySAL

- https://github.com/pysal/pysal
- PySAL: Python Spatial Analysis Library
- Sub-packages

  pysal.cg – Computational Geometry
  pysal.core — Core Data Structures and IO
  pysal.esda — Exploratory Spatial Data Analysis
  pysal.inequality — Spatial Inequality Analysis
  pysal.region — Spatially Constrained Clustering
  pysal.spatial_dynamics — Spatial Dynamics
  pysal.spreg — Regression and Diagnostics
  pysal.weights — Spatial Weights
  pysal.network — Network Constrained Analysis
  pysal.contrib – Contributed Modules

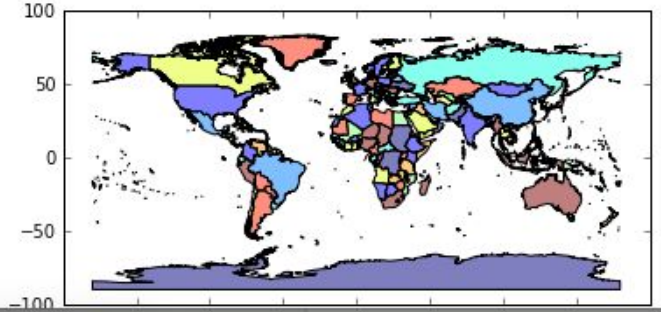
In [6]:
```
import geopandas as gp
%matplotlib inline
pth = "data/world_borders/TM_WORLD_BORDERS-0.3.shp"
world = gp.GeoDataFrame.from_file(pth)
```

In [7]:
```
world.head(3)
```

Out[7]:

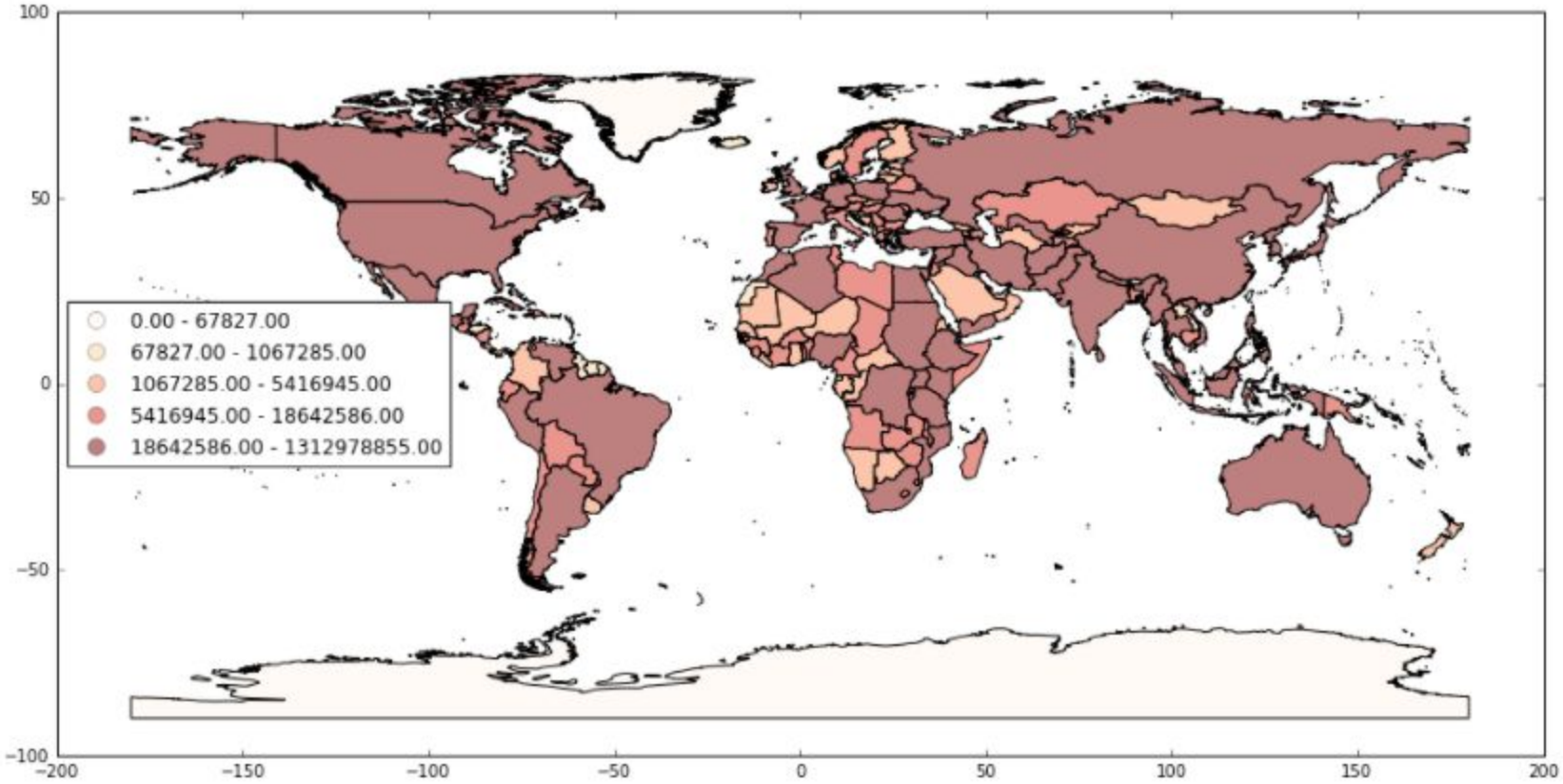
	AREA	FIPS	ISO2	ISO3	LAT	LON	NAME	POP2005	REGION	SUBREGION	UN	geometry
0	44	AC	AG	ATG	17.078	-61.783	Antigua and Barbuda	83039	19	29	28	(POLYGON ((-61.686668 17.02444100000014, -61.7...
1	238174	AG	DZ	DZA	28.163	2.632	Algeria	32854159	2	15	12	POLYGON ((2.96361 36.802216, 2.981389 36.80693...
2	8260	AJ	AZ	AZE	40.430	47.395	Azerbaijan	8352021	142	145	31	(POLYGON ((45.08332100000001 39.76804400000015...

In [3]:
```
world.plot();
```



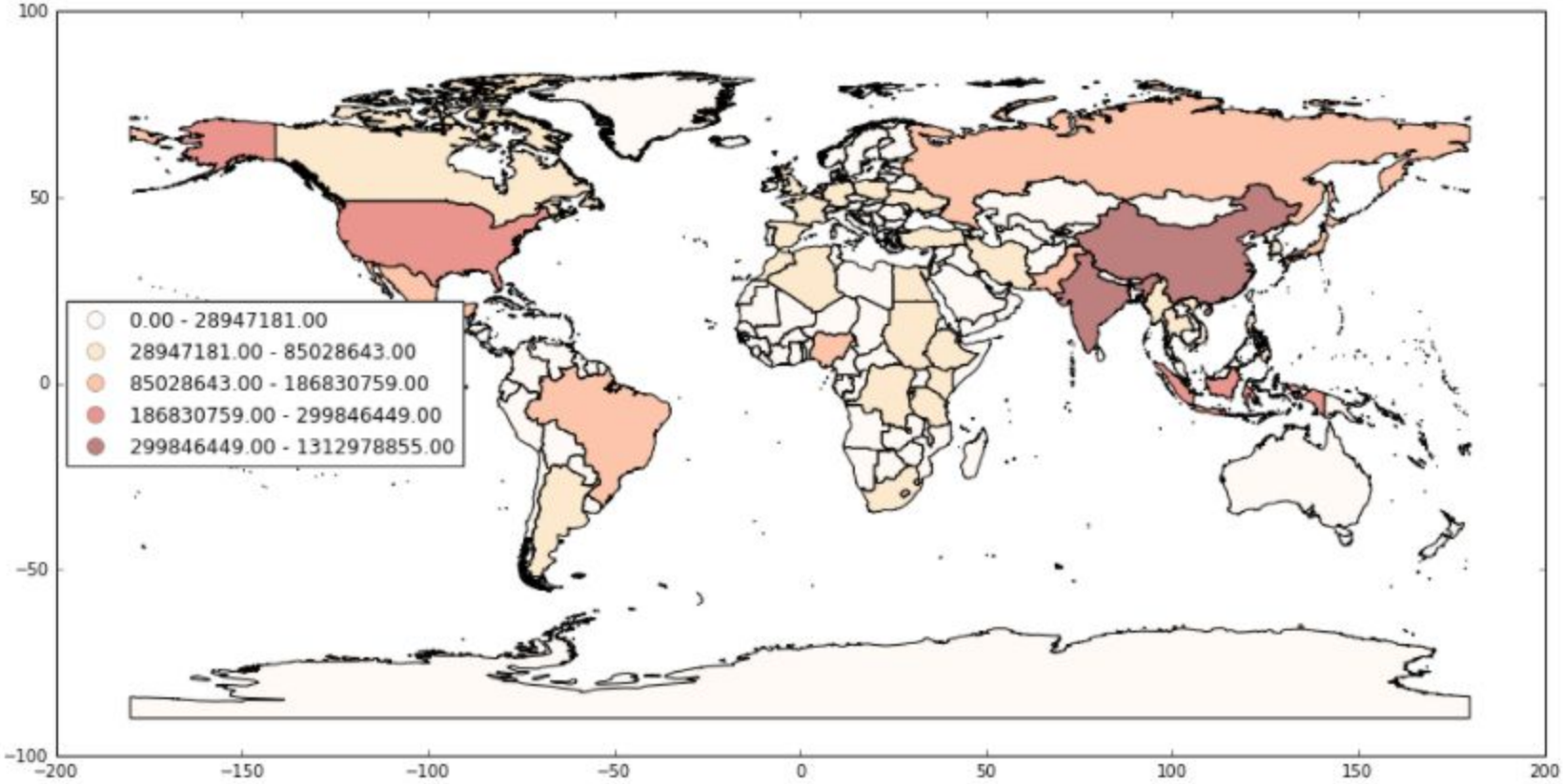
```
In [10]: world.plot(column='POP2005', scheme='QUANTILES', k=5, colormap='OrRd',  legend=True,
                    figsize=(16, 10))
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x160747390>
```



In [9]:
```
world.plot(column='POP2005', scheme='fisher_jenks', k=5, colormap='OrRd',  legend=True,
           figsize=(16, 10))
```

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x14f696850>

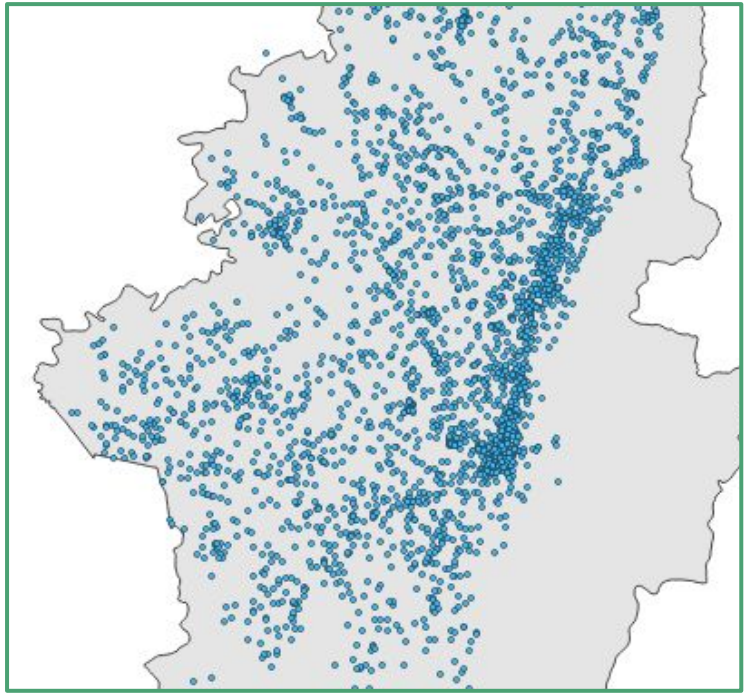


Geoprocessing

Geoprocessing

- Spatial Joins
- Data Clipping
- Data Enrichment
- Data filtering
- Spatial Analysis

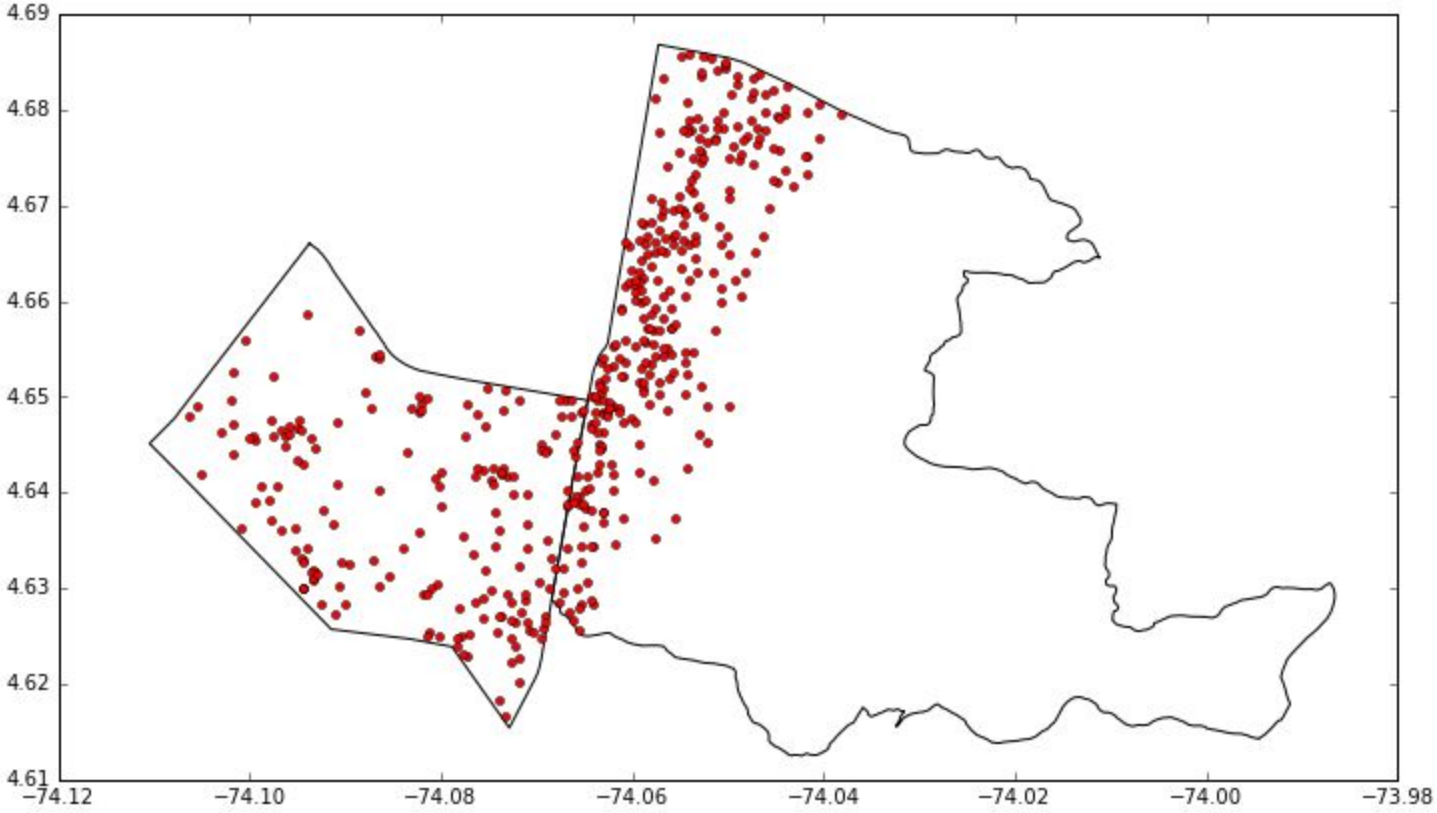
data/ideca/Loca.shp
Localidades, Bogotá



data/ideca/sitios_interes.shp
Sitios de Interés, Bogotá

data/ideca/clasificacion_sitios_interes.csv

In [41]:
```
base = localidades_seleccionadas.plot(color='white')
selected_poi.plot(ax=base, marker='o', color='red', markersize=4);
```



Thanks!!

juan@gkudos.com

http://gkudos.com/
info@gkudos.com