

IMD0905 - Data Science I

Lesson #13 - Storytelling from geographic data

Ivanovitch Silva
October, 2018



Agenda

- Case study: open flights
- Geographic coordinate system: problems
- Basemap toolkit

Update the repository

```
git clone https://github.com/ivanovitchm/IMD0905_datascience_one.git
```

Or

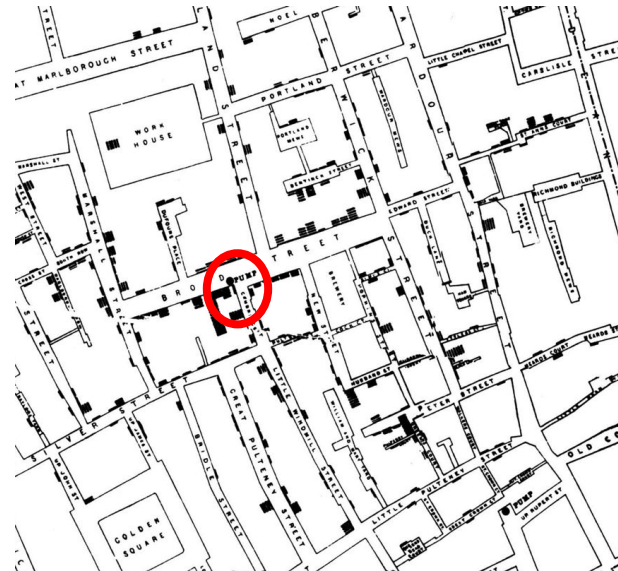
```
git pull
```

motivation

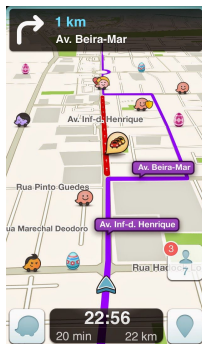
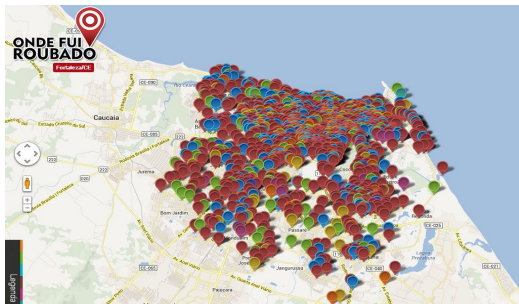
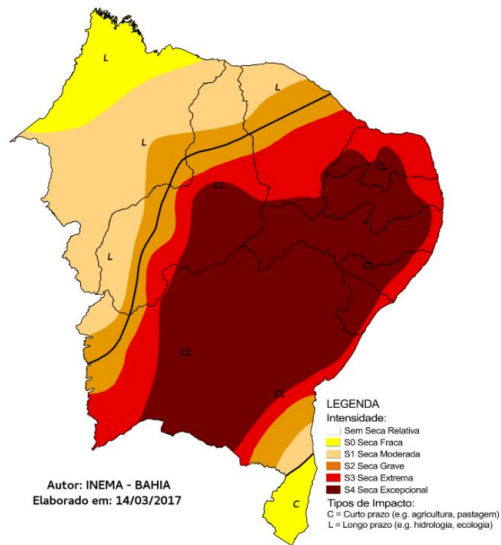
Cholera Outbreak 1854



Maps Save Lives



Geographic data is always present in our everyday lives

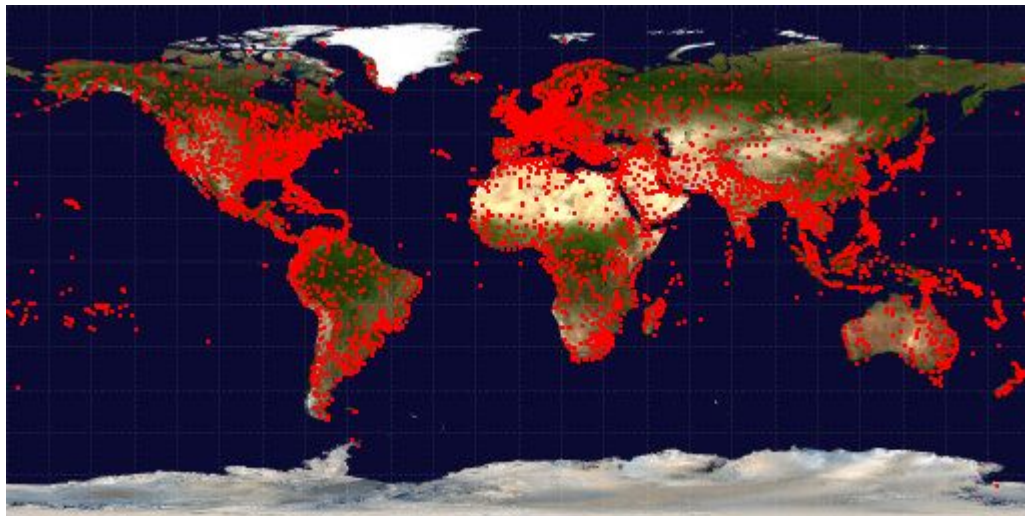


Raw geographic data like **latitudes** and **longitudes** are difficult to understand using the data charts and plots we've discussed so far

Geographic dataset

Airport, airline and route data

- **airlines.csv** - data on each airline.
 - **country** - where the airline is headquartered.
 - **active** - if the airline is still active.
- **airports.csv** - data on each airport.
 - **name** - name of the airport.
 - **city** - the airport is located.
 - **country** - country the airport is located.
 - **code** - unique airport code.
 - **latitude** - latitude value.
 - **longitude** - longitude value.
- **routes.csv** - data on each flight route.
 - **airline** - airline for the route.
 - **source** - starting city for the route.
 - **dest** - destination city for the route.



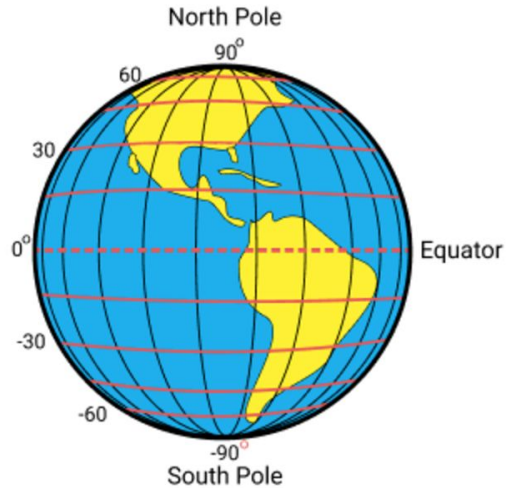
<https://openflights.org/data.html>

Geographic coordinate system

Latitude

Describes the
North-South
position

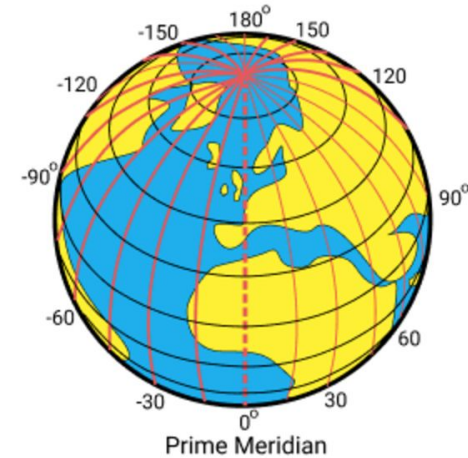
Ranges from
-90 to 90
degrees



Longitude

Describes the
East-West
position

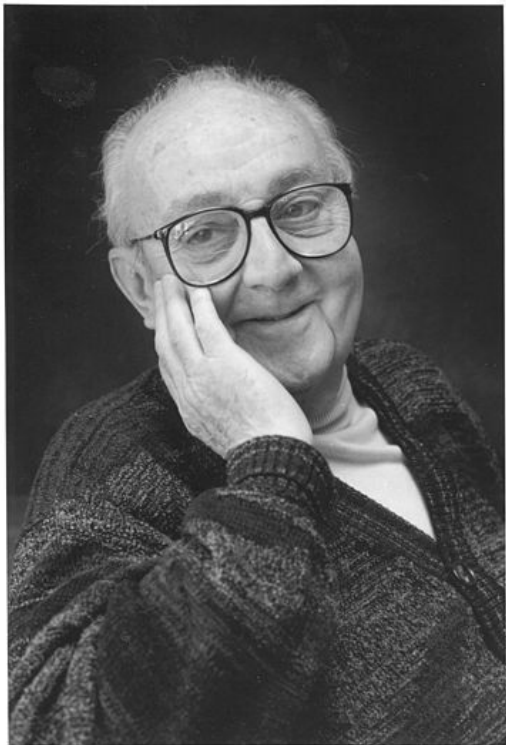
Ranges from
-180 to 180
degrees



Geographic coordinate system

Name	City	State	Latitude	Longitude
White House	Washington	DC	38.898166	-77.036441
Alcatraz Island	San Francisco	CA	37.827122	-122.422934
Instituto Metr�pole Digital	Natal	RN	-5.831997	-35.205415

the problem with maps



"All models are wrong,
but some are useful"

George E. P. Box



The work as we know it

<https://goo.gl/ckFKh6>

Greenland is no Africa



Mercator



Actual



The true size of Africa

Map Projections

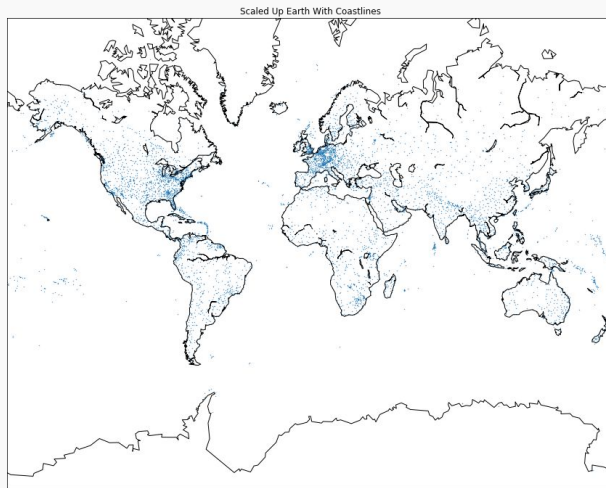
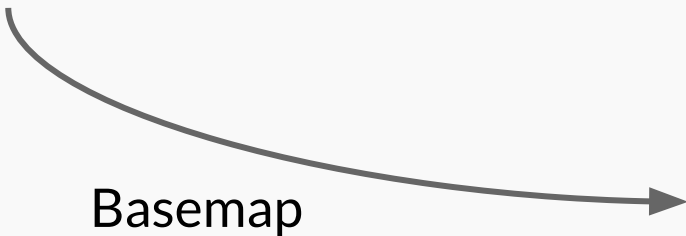
Two types of maps:

- Reference: accuracy is the most important
- Thematic: the data, i.e., getting the story right is the most important

Basemap Toolkit

Basemap is an extension to Matplotlib that makes it easier to work with geographic data

matplotlib

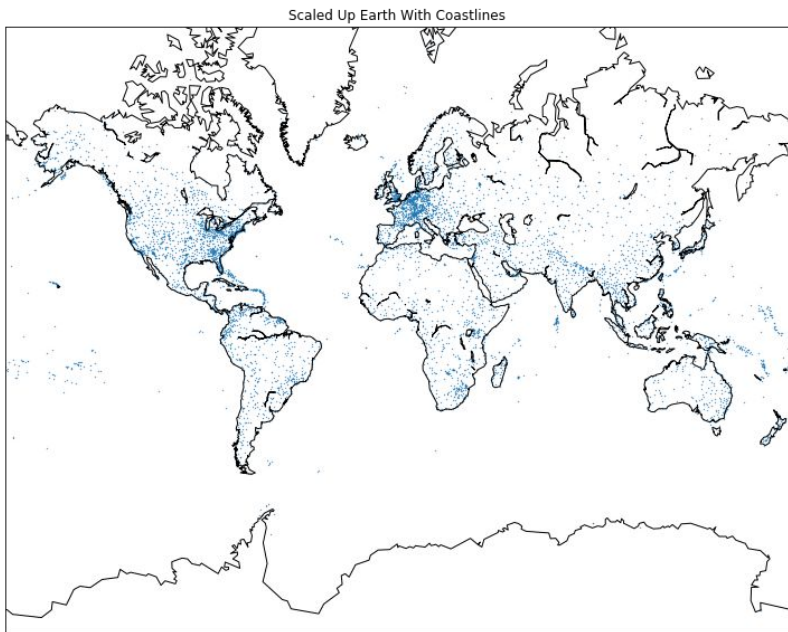


Workflow with basemap

```
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap

m = Basemap(projection='merc',
            llcrnrlat=-80,
            urcrnrlat=80,
            llcrnrlon=-180,
            urcrnrlon=180)
```

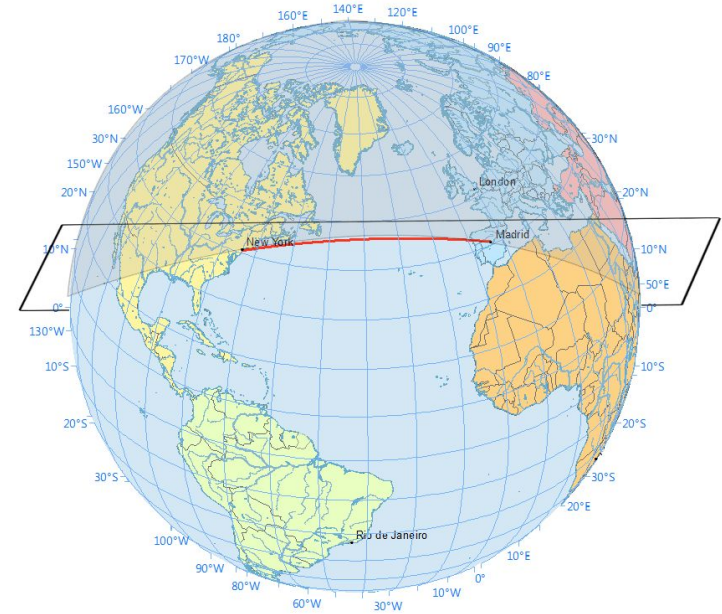
Converting from spherical to cartesian coordinates



```
longitudes = airports["longitude"].tolist()
latitudes = airports["latitude"].tolist()
x, y = m(longitudes, latitudes)

fig, ax = plt.subplots(figsize=(20,10))
plt.title("Scaled Up Earth With Coastlines")
m.scatter(x,y,s=0.1)
m.drawcoastlines()
plt.show()
```

Introduction to great circles



Displaying great circles

lon1 - longitude of the starting point.

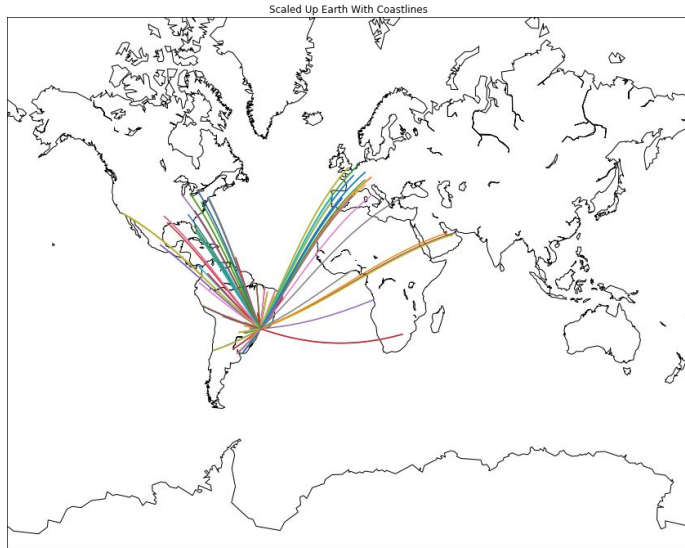
lat1 - latitude of the starting point.

lon2 - longitude of the ending point.

lat2 - latitude of the ending point.

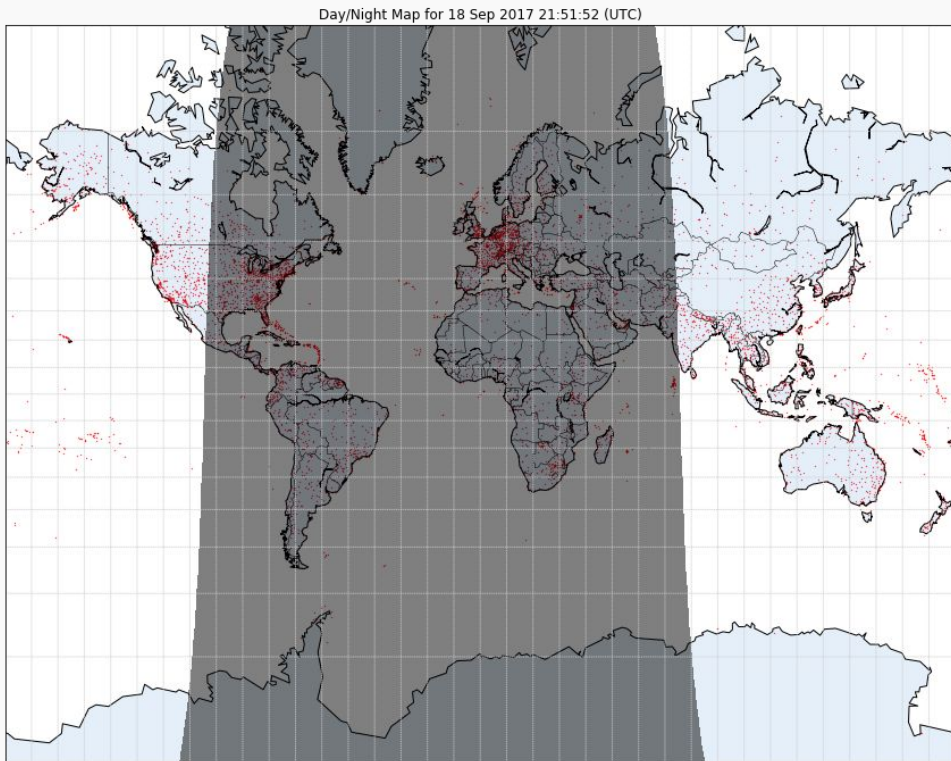
```
m.drawgreatcircle(39.956589, 43.449928, 49.278728, 55.606186)  
m.drawgreatcircle(48.006278, 46.283333, 49.278728, 55.606186)  
m.drawgreatcircle(39.956589, 43.449928, 43.081889, 44.225072)
```

Great circles: case study



```
def create_greate_circles(df):  
    for index,row in df.iterrows():  
        end_lat,start_lat = row["end_lat"],row["start_lat"]  
        end_lon,start_lon = row["end_lon"],row["start_lon"]  
  
        if (abs(end_lat-start_lat) < 180):  
            if (abs(end_lon-start_lon) < 180):  
                m.drawgreatcircle(start_lon,start_lat,end_lon,end_lat).  
  
gru = geo_routes[geo_routes["source"] == "GRU"]  
create_greate_circles(gru)  
m.drawcoastlines()  
plt.show()
```


Day-night terminator on map



Lesson 13 - Visualizing Geographic Data.ipynb

