Data Science Foundation

Lesson #9 - Storytelling from Geographic Data

Ivanovitch Silva

September, 2017

Agenda

- Case study I: open flights
- Geographic coordinate system: problems
- Basemap toolkit
- Folium
- Maps styles, markers, color and icon types
- Marker clusters
- Heatmap
- Popups
- Case study II: open data Natal-RN

Update the repository

```
git clone https://github.com/ivanovitchm/EEC2006.git
```

Or ....

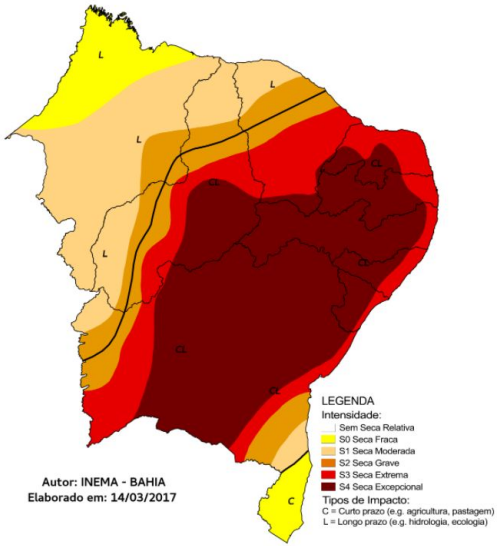
```
git pull
```

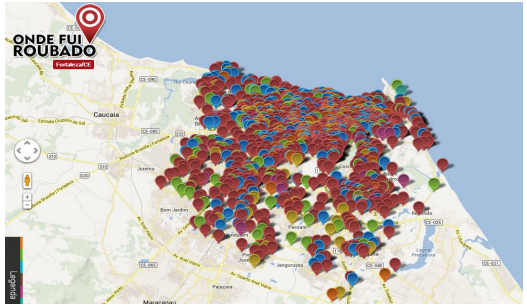
motivation

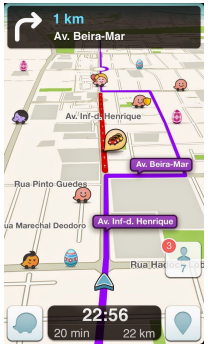
Cholera Outbreak 1854

Maps Save Lives

Geographic data is always present in our everyday lives





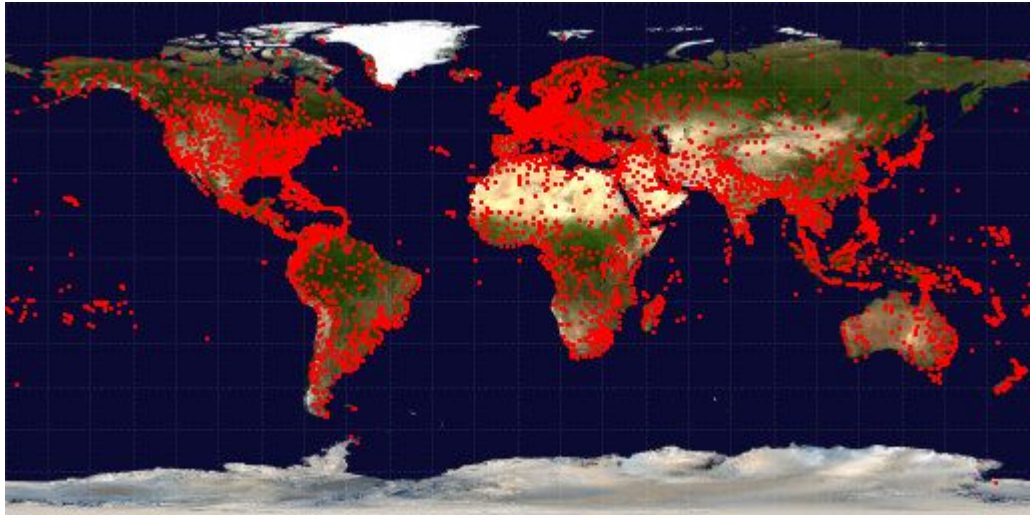


Raw geographic data like latitudes and longitudes are difficult to understand using the data charts and plots we've discussed so far

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.



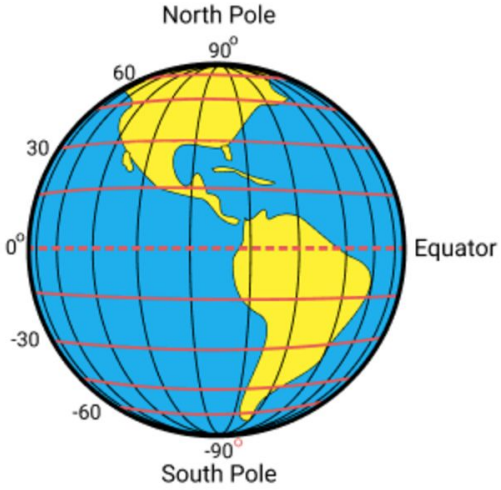
openflights.org

Geographic coordinate system

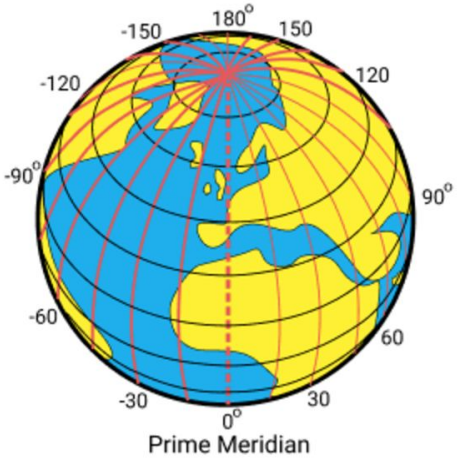
Latitude

Describes the North-South position

Ranges from -90 to 90 degrees



Longitude



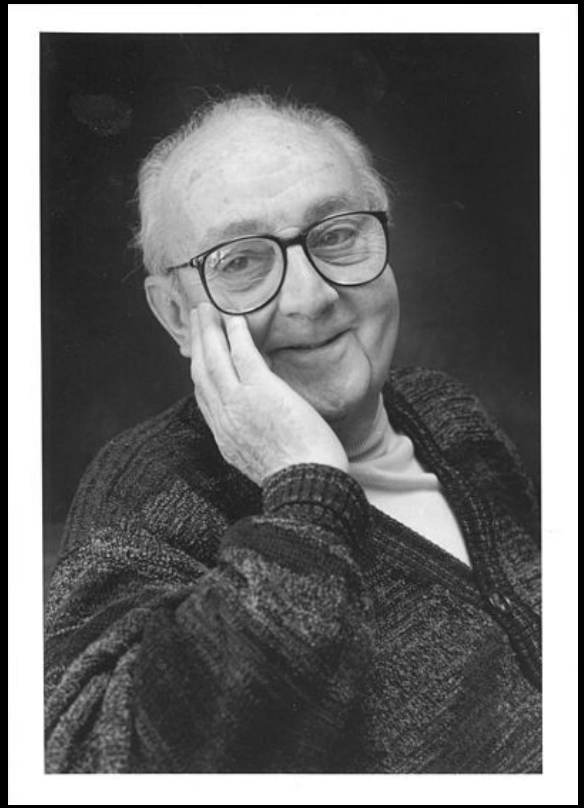
Describes the East-West position

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 Metrópole Digital	Natal	RN	-5.831997	-35.205415

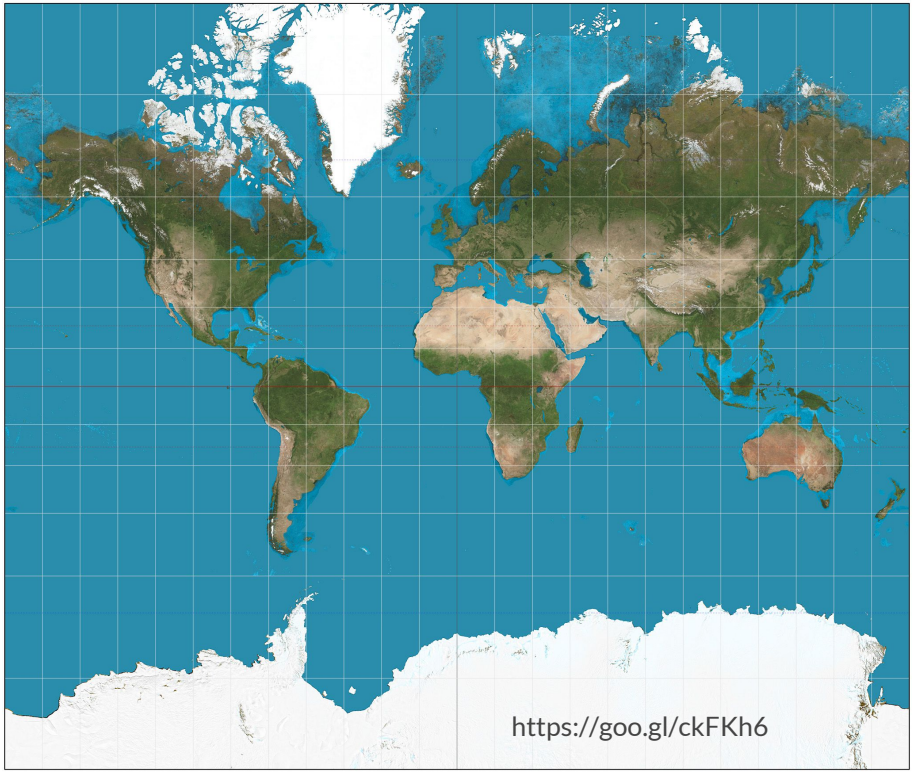
the problem with maps



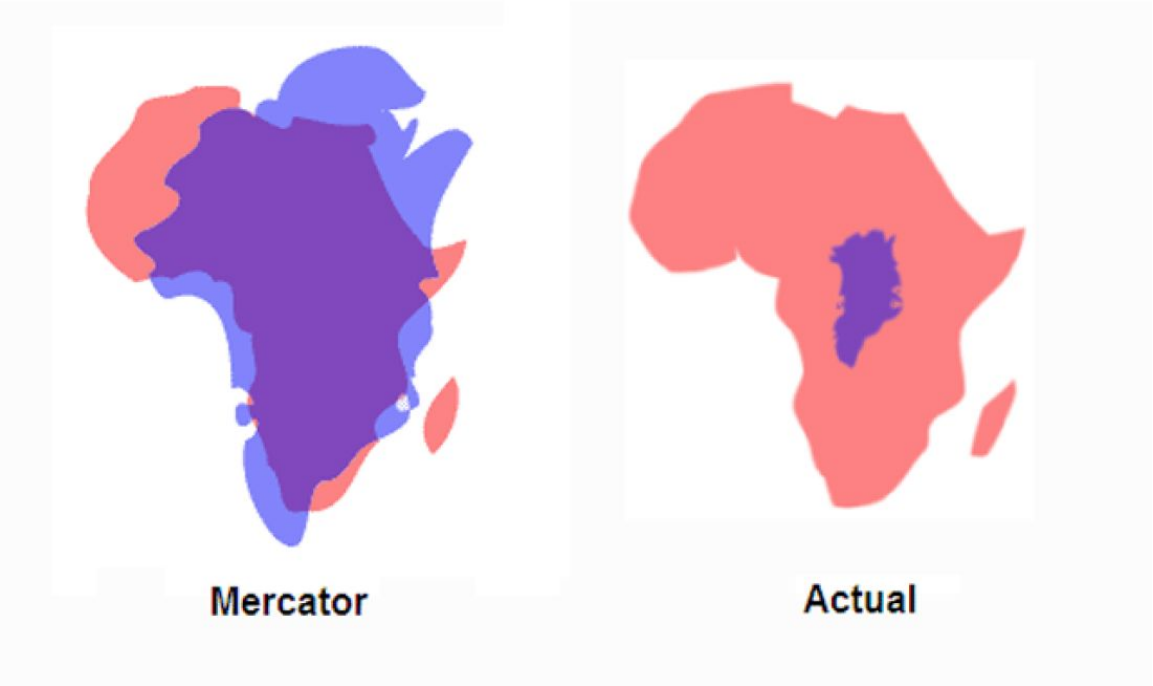
"All models are wrong, but some are useful"

George E. P. Box

The world as we know it



Greenland is no Africa



The true size of Africa

http://kai.sub.blue/en/africa.html

Map Projections

Two types of maps:

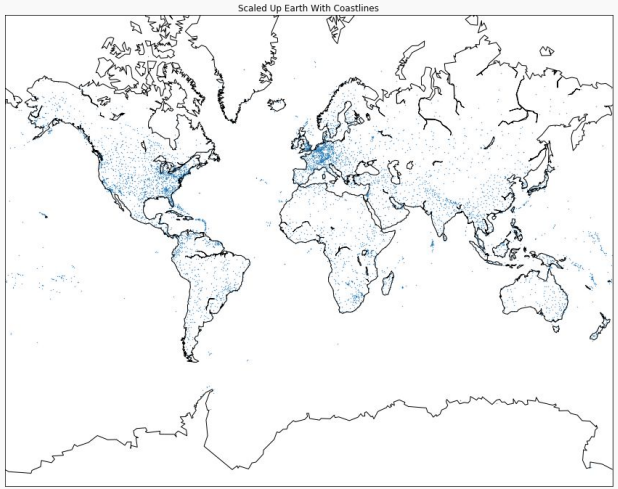
- Reference: accuracy is the most important
- Thematic: the data, i.e., getting the story right is the most important

Basemap Toolkit

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

matplotlib

Basemap



Installing basemap

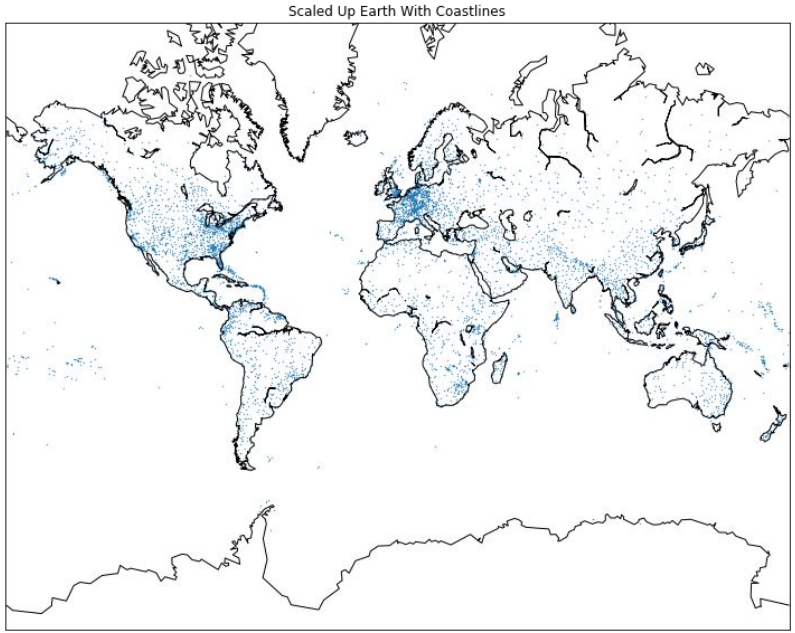
```
conda install -c conda-forge basemap
```

```
from mpl_toolkits.basemap import Basemap
```

Workflow with basemap

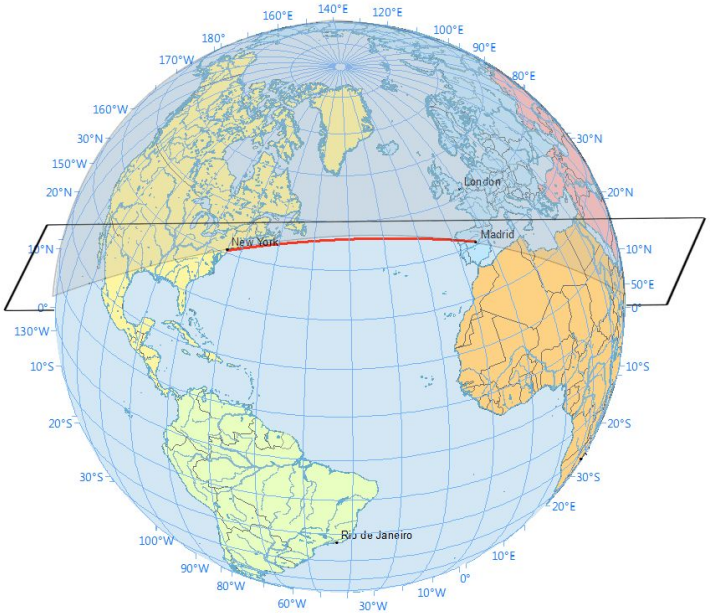
```
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap

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

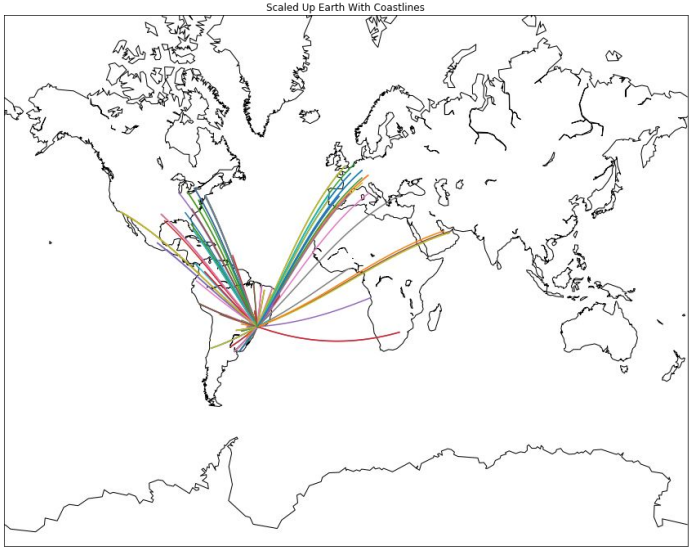


Displaying great circles

lon1 - longitude of the starting point.
lat1 - latitude of the starting point.
lon2 - longitude of the ending point.
lat2 - latitude of the ending point.

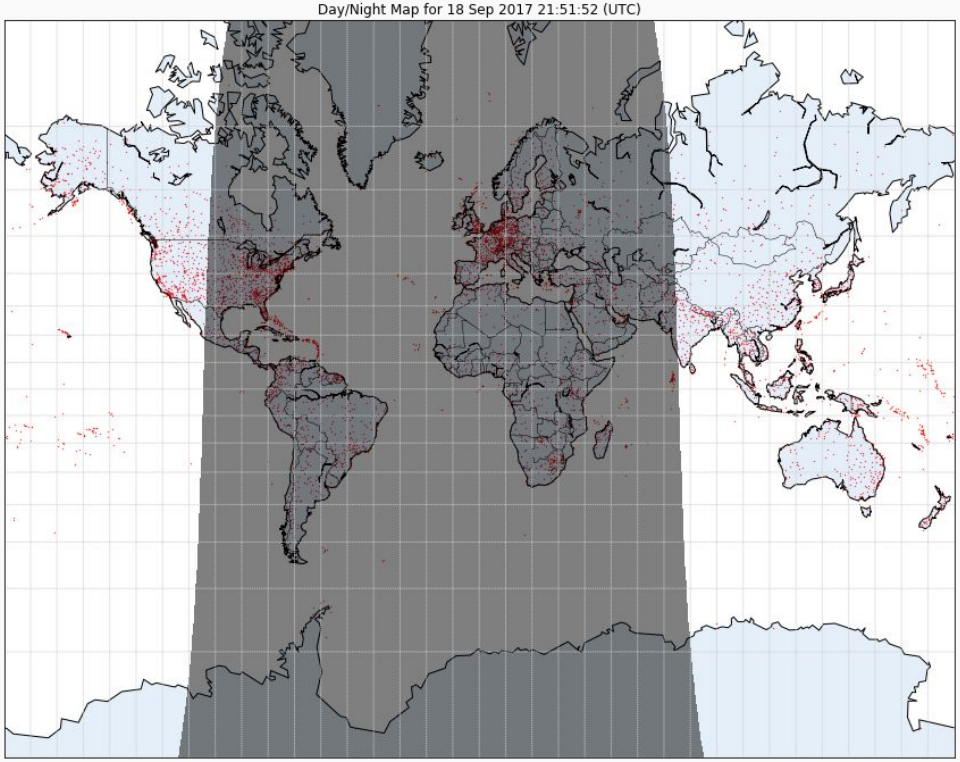
```
m.drawgreatcircle(39.956589, 43.449928, 49.278728, 55.606186)
m.drawgreatcircle(48.006278, 46.283333, 49.278728, 55.606186)
m.drawgreatcircle(39.956589, 43.449928, 43.081889 , 44.225072)
```

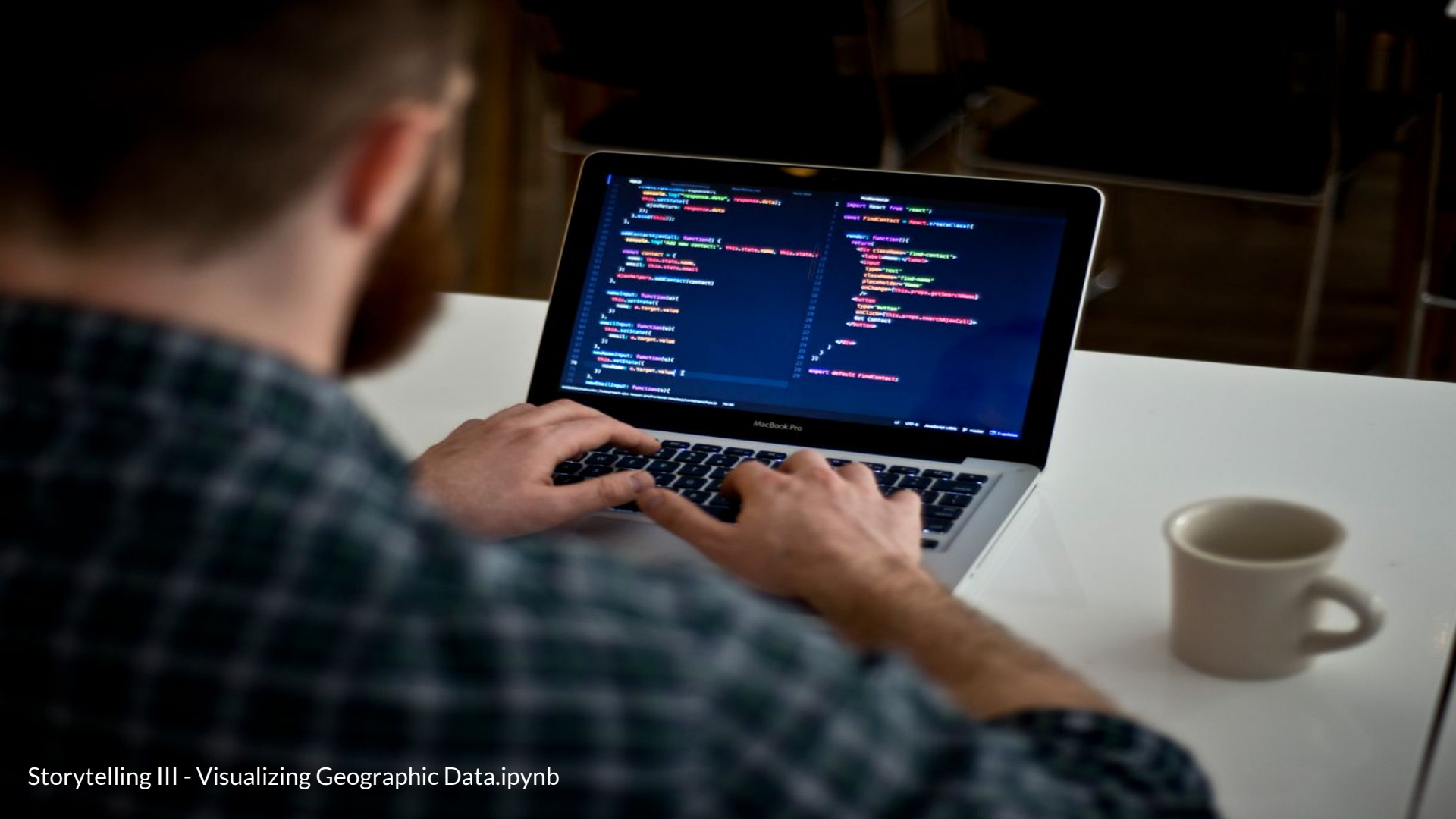
Great circles: case study



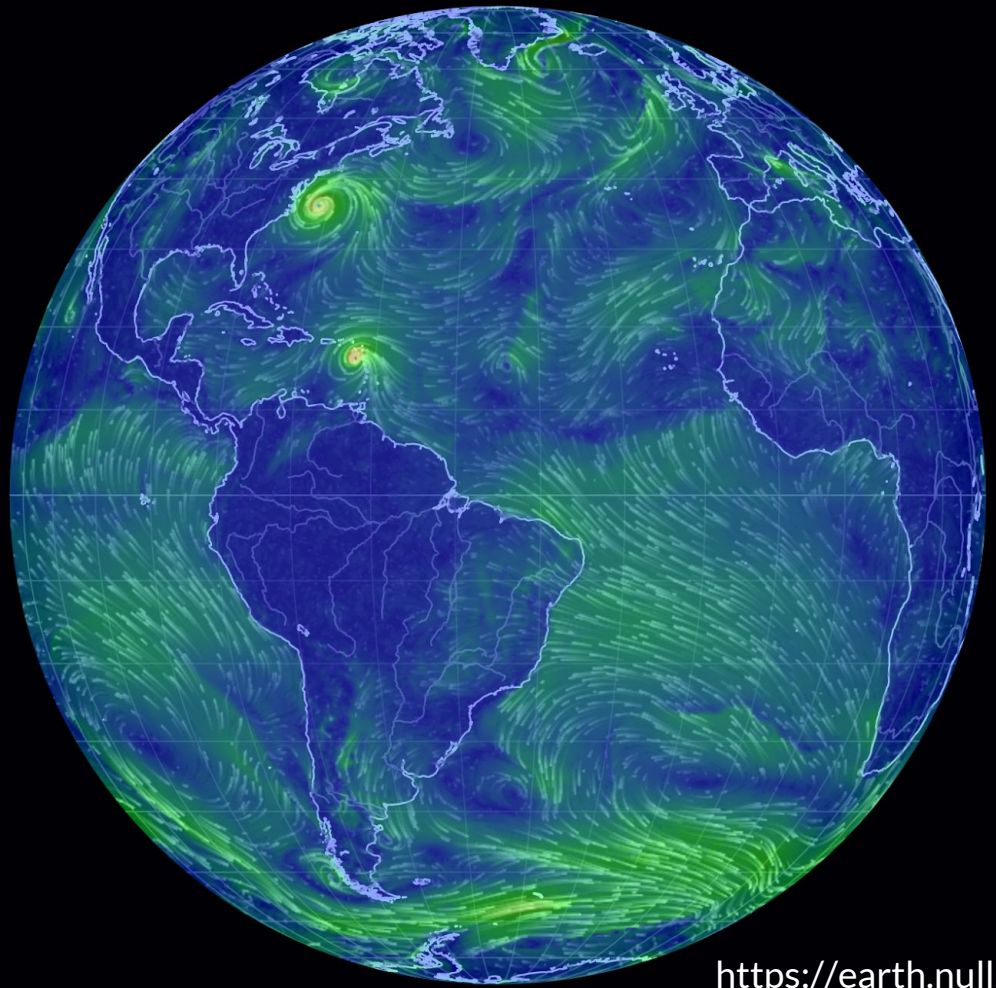

```
fig, ax = plt.subplots(figsize=(15,20))
plt.title("Scaled Up Earth With Coastlines")

def create_great_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)

dfw = geo_routes[geo_routes['source'] == "GRU"]
create_great_circles(dfw)
m.drawcoastlines()
plt.show()
```

Day-night terminator on map





Storytelling III - Visualizing Geographic Data.ipynb



earth ≡

https://earth.nullschool.net/

community
EarthWindMap

Folium

Folium builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the Leaflet.js library. Manipulate your data in Python, then visualize it in on a Leaflet map via Folium.

```
conda install folium -c conda-forge
```

Creating a world map

```
import folium
m = folium.Map()
m.save('index.html')
m
```

Creating a map of Brazil

```
m = folium.Map(
    location=[-15.765379, -47.968776],
    zoom_start=4
)
```

Maps styles

```
m = folium.Map(
    location=[-15.765379, -47.968776],
    tiles='Stamen Toner',
    zoom_start=4
)
```

Leaflet Custom Tileset

```
m = folium.Map(
    location=[-15.765379, -47.968776],
    zoom_start=12,
    tiles='https://api.mapbox.com/styles/v1/mapbox/outdoors-v10/tiles/256/{z}/{x}/{y}?access_token=y
our_token',
    attr='Imagery © <a href="http://mapbox.com">MapBox</a>'
```

Maps with markers

```
m = folium.Map(
    location=[-5.826592, -35.212558],
    zoom_start=12,
    tiles='Stamen Terrain',
    width='50%',
    height='50%'
)
folium.Marker([-5.832187, -35.205432], popup='<i>Instituto Metrópole Digital</i>').add_to(m)
folium.Marker([-5.842942, -35.198001], popup='<b>Centro Tecnológico</b>').add_to(m)
```

Color and icon types

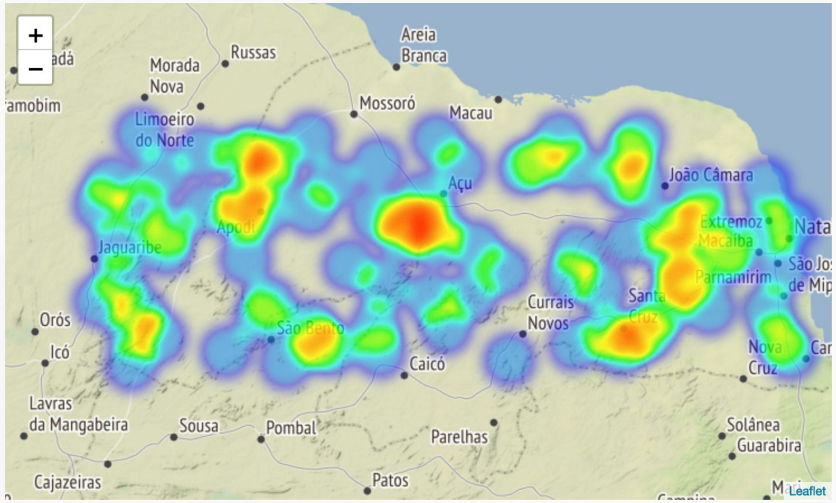
```
folium.Marker([-5.832187, -35.205432],
              popup='<i>Instituto Metrópole Digital</i>',
              icon=folium.Icon(icon='cloud')).add_to(m)
folium.Marker([-5.842942, -35.198001],
              popup='<b>Centro Tecnológico</b>',
              icon=folium.Icon(color='red',
                               icon_color='yellow',
                               icon='info-sign')).add_to(m)
```

Marker clusters

```
# Create a map
m = folium.Map(
    location=[-5.826592, -35.212558],
    zoom_start=12,
    tiles='Stamen Terrain',
    width='75%',
    height='75%'
)

# Make a cluster
plugins.MarkerCluster(data).add_to(m)
```

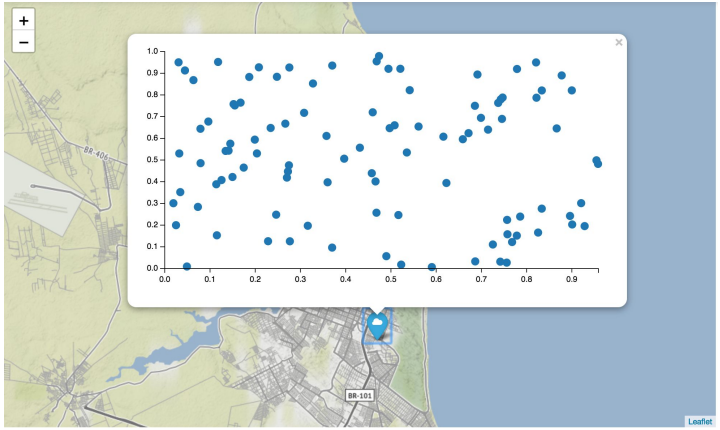
Heatmap

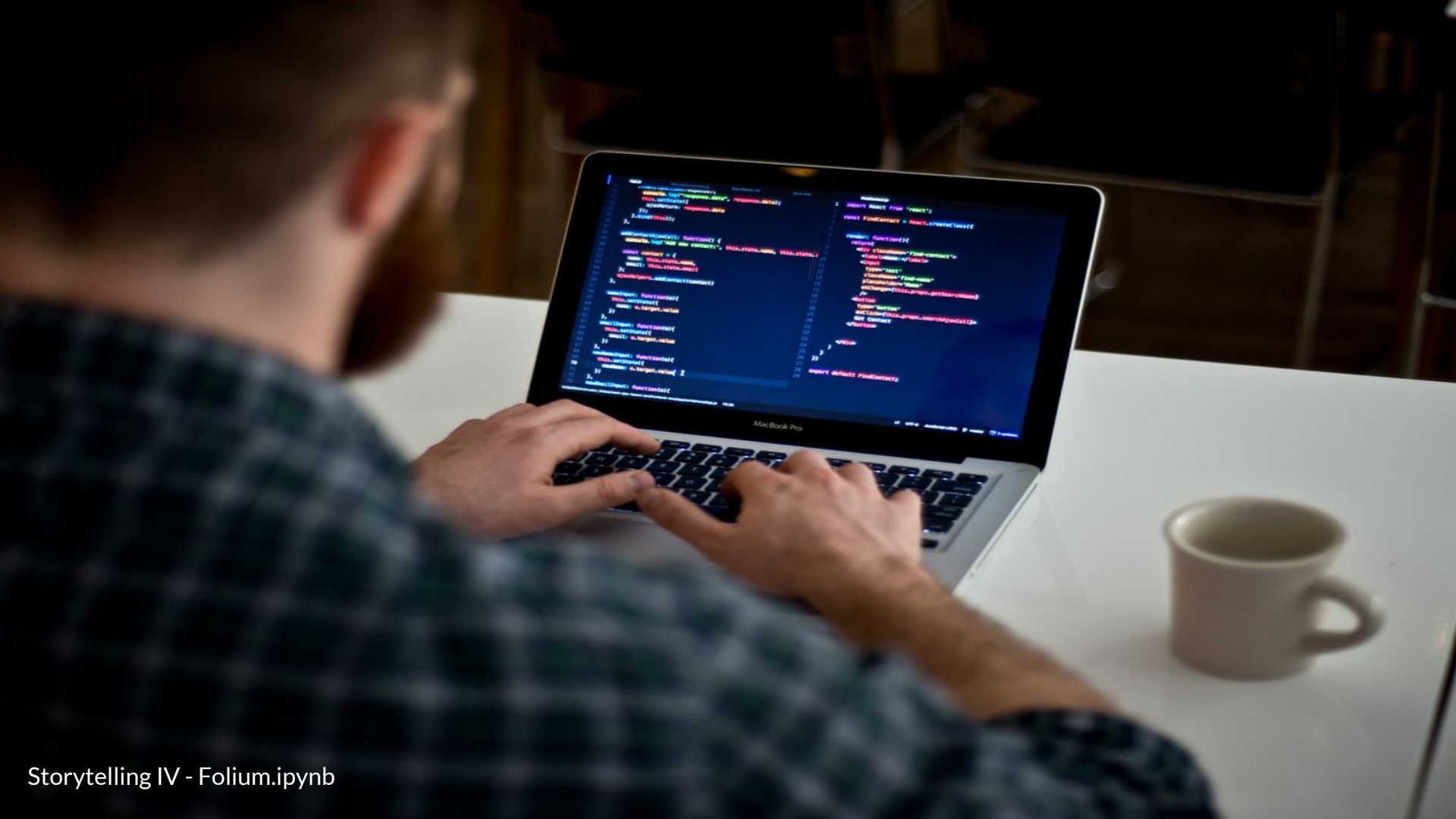
```
# Create a map
m = folium.Map(
    location=[-5.826592, -35.212558],
    zoom_start=12,
    tiles='Stamen Terrain',
    width='75%',
    height='75%'
)

# data : list of points of the form [lat, lng] or [lat, lng, weight]
HeatMap(data).add_to(m)
```



Create popups

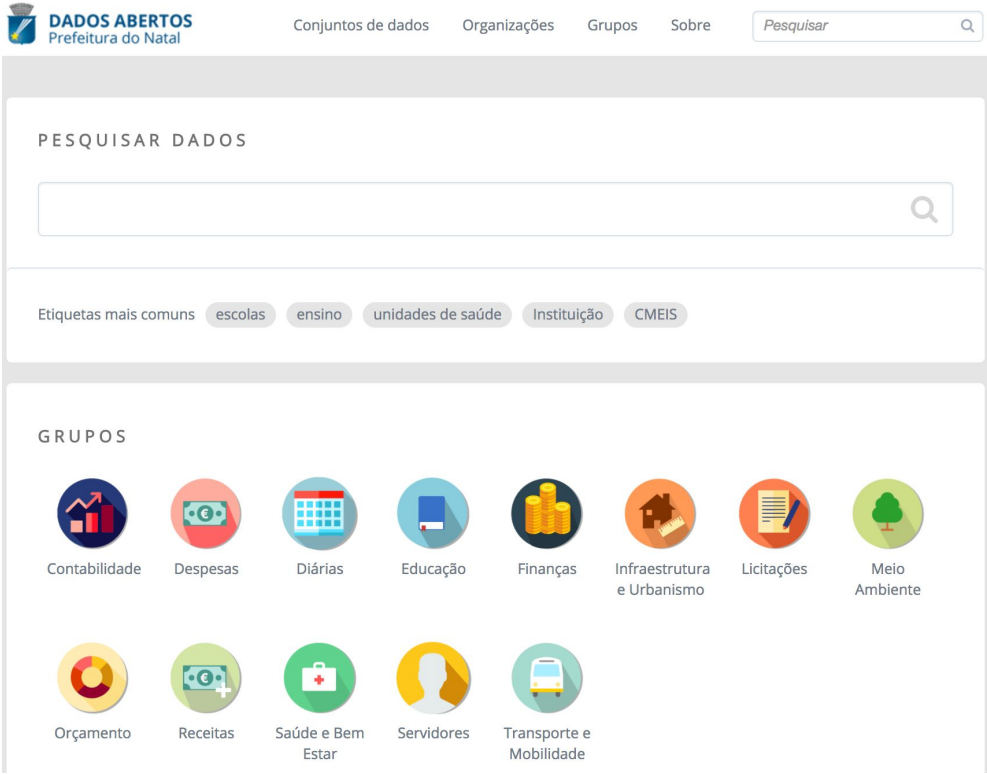
```
# Create an object popup and adding a graph for it
popup = folium.Popup(max_width=650)
folium.Vega(scatter_chart, height=350, width=650).add_to(popup)

# Print a icon on map
folium.Marker([-5.832187, -35.205432],
              icon=folium.Icon(icon='cloud'),
              popup=popup).add_to(m)
```





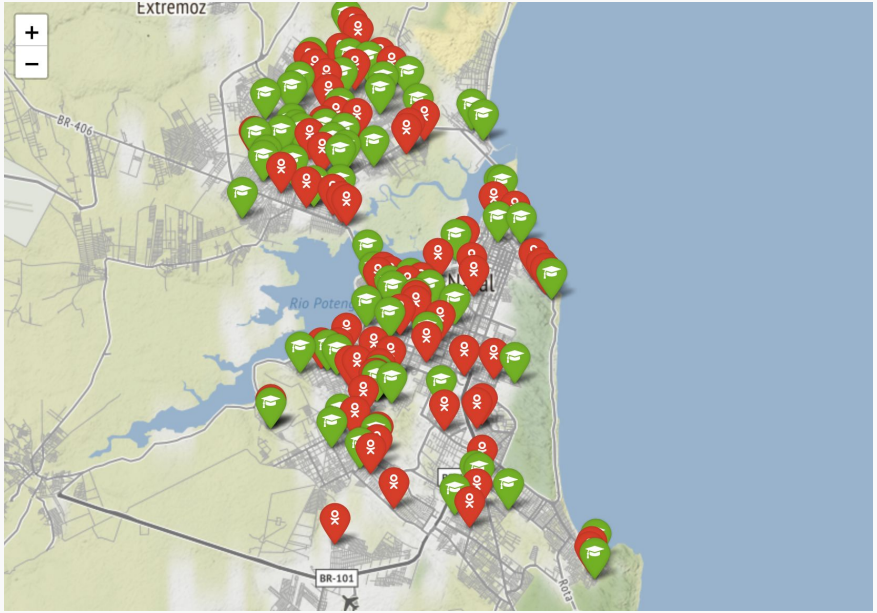
Storytelling IV - Folium.ipynb

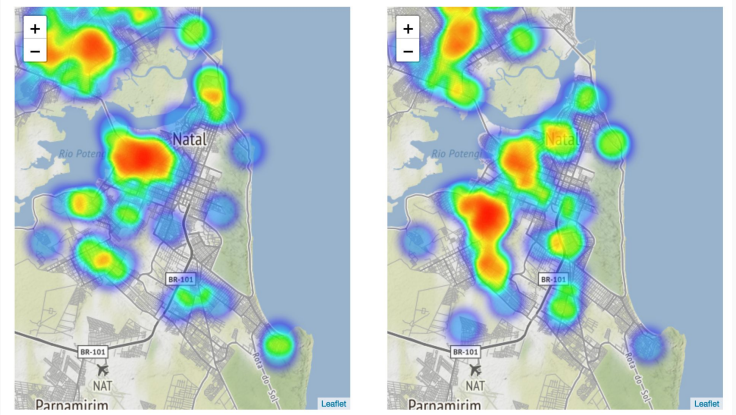
Case study: open data Natal



- Collect
- EDA
- Cleaning
- Preparing
- Analysis
- Geolocalization
- Education (elementary schools and kindergarten)

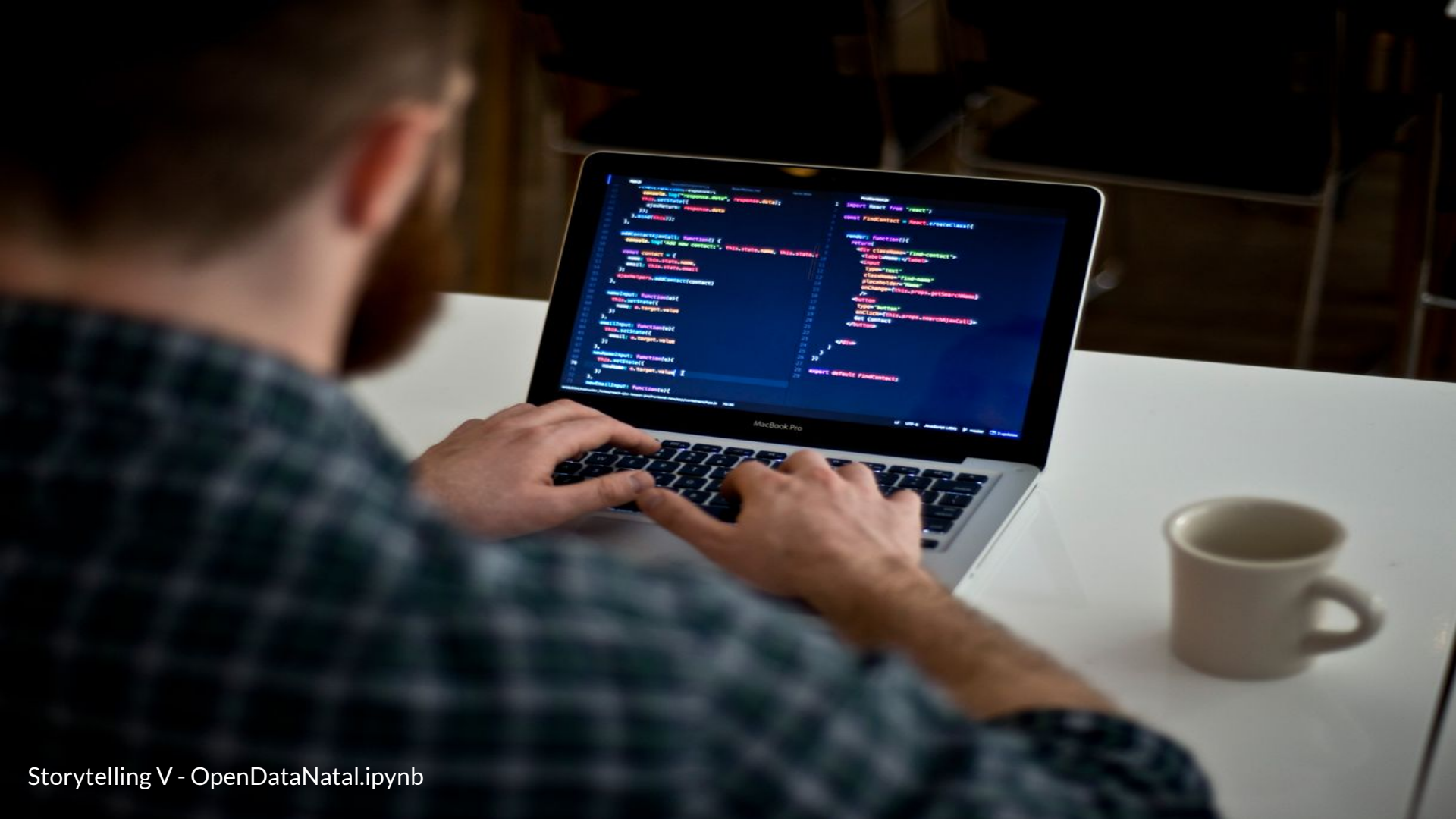
Case study: open data Natal





Elementary School

Kindergarten



Storytelling V - OpenDataNatal.ipynb