

Data Science & ML Course

Lesson #10 Exploratory Data Analysis V

Ivanovitch Silva
October, 2018



Agenda

- Visualizing geographical data
- Working with basemap
- Customizing the plot
- Folium
- Maps with markers
- Marker clusters
- Heatmap

Update from repository

```
git clone https://github.com/ivanovitchm/datascience2machinelearning.git
```

Or

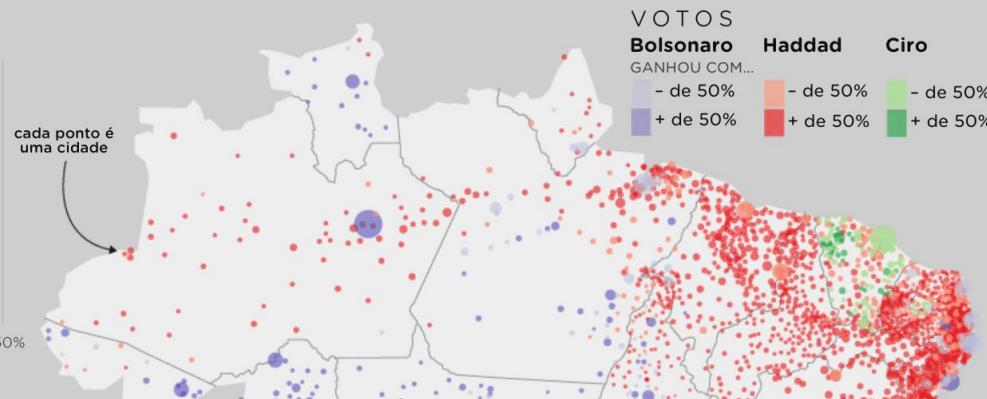
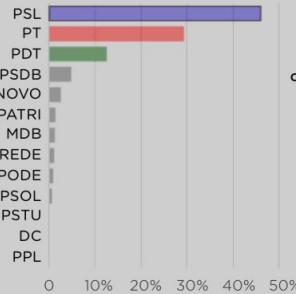
```
git pull
```



motivation

Eleições 2018

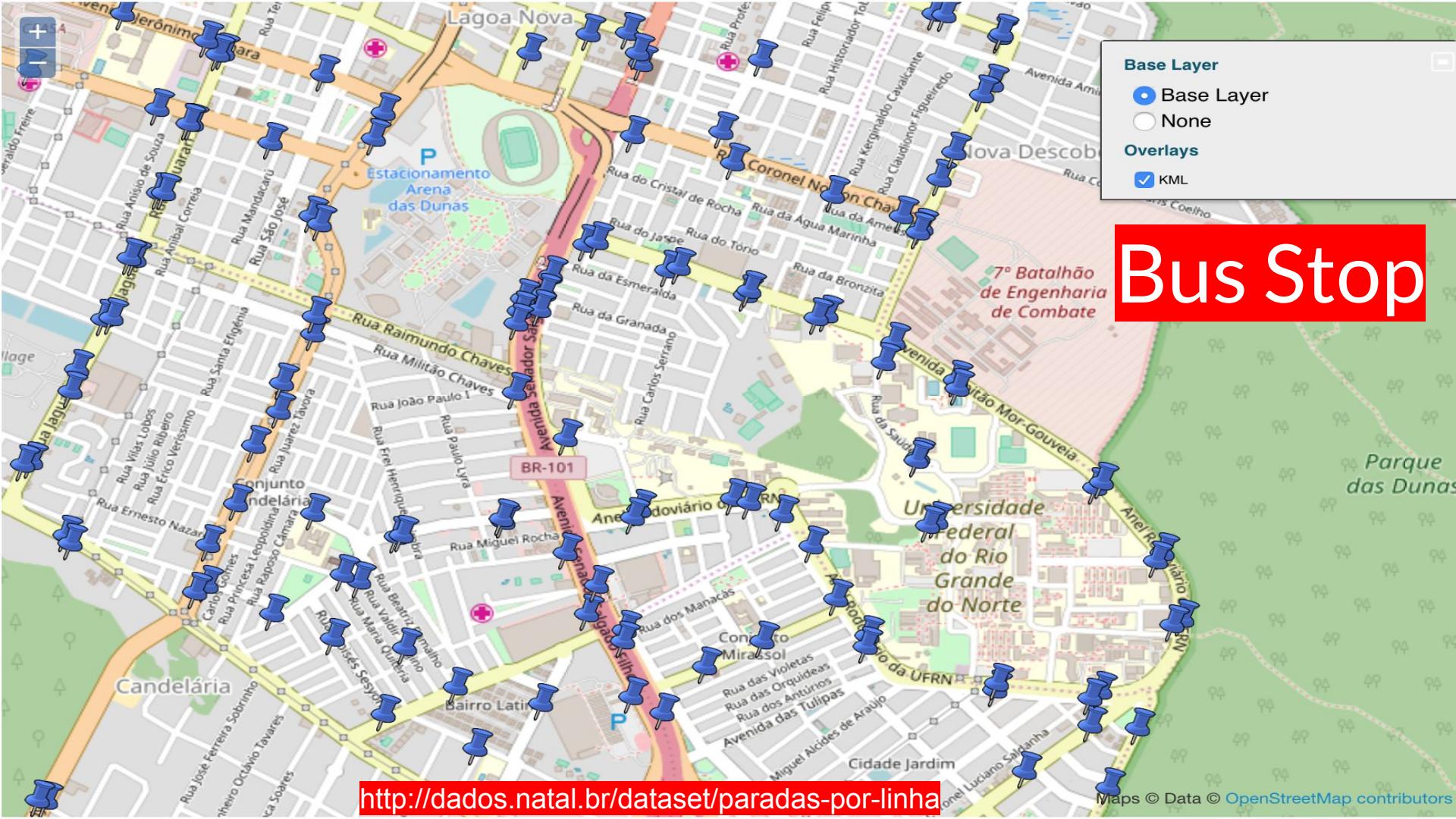
1º TURNO

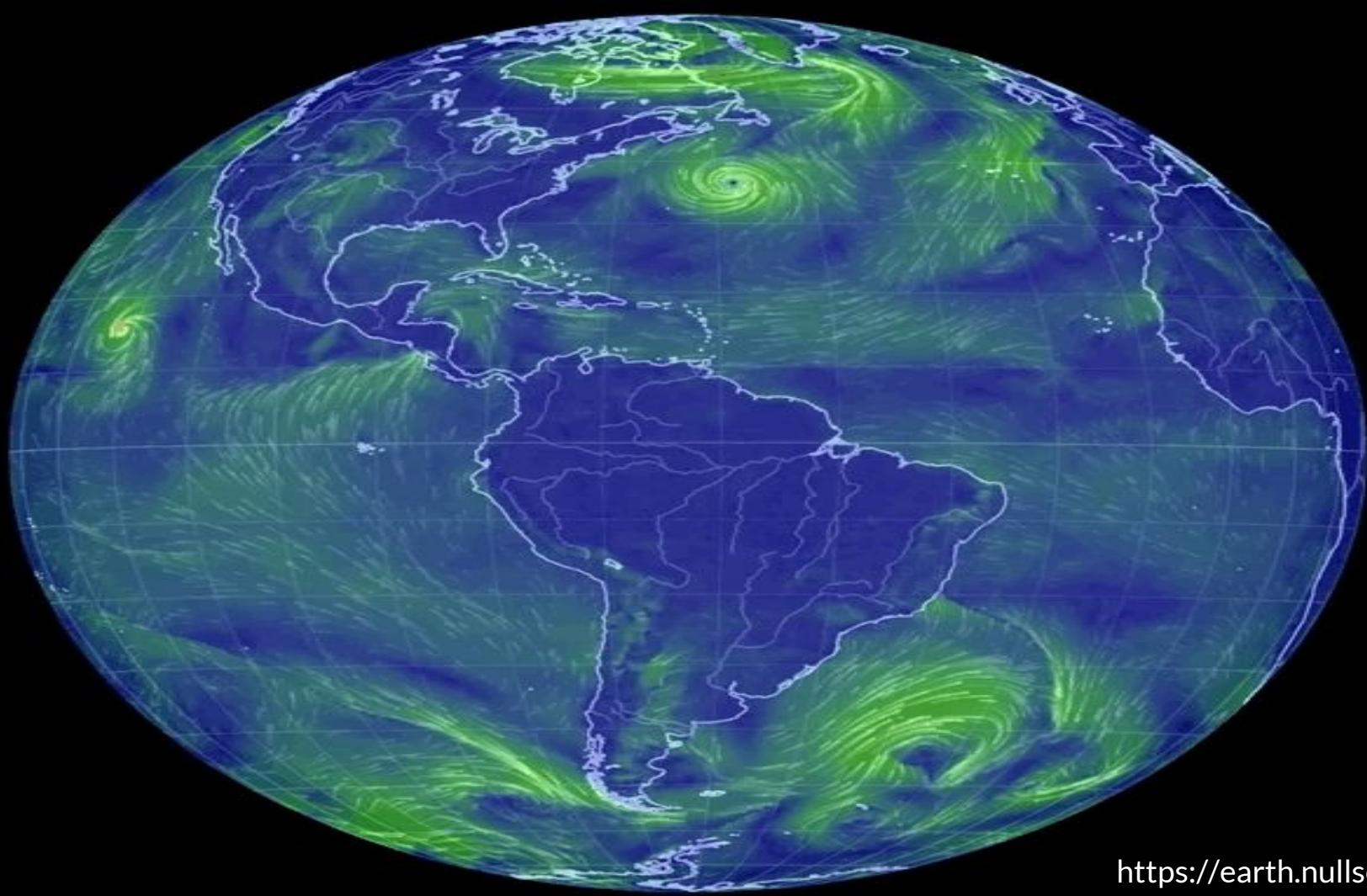


2º TURNO

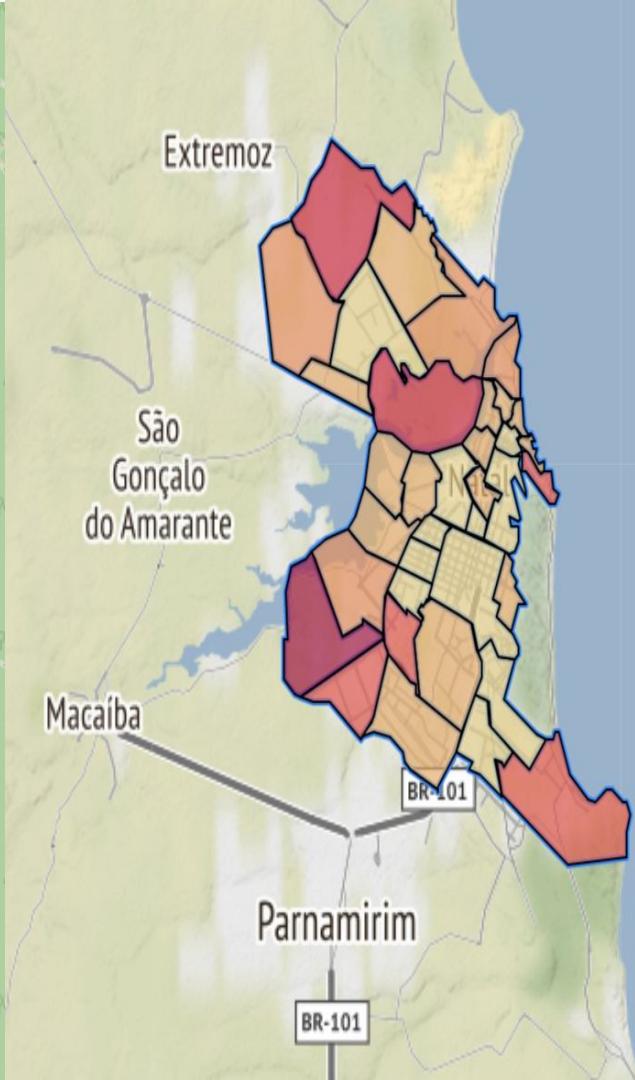
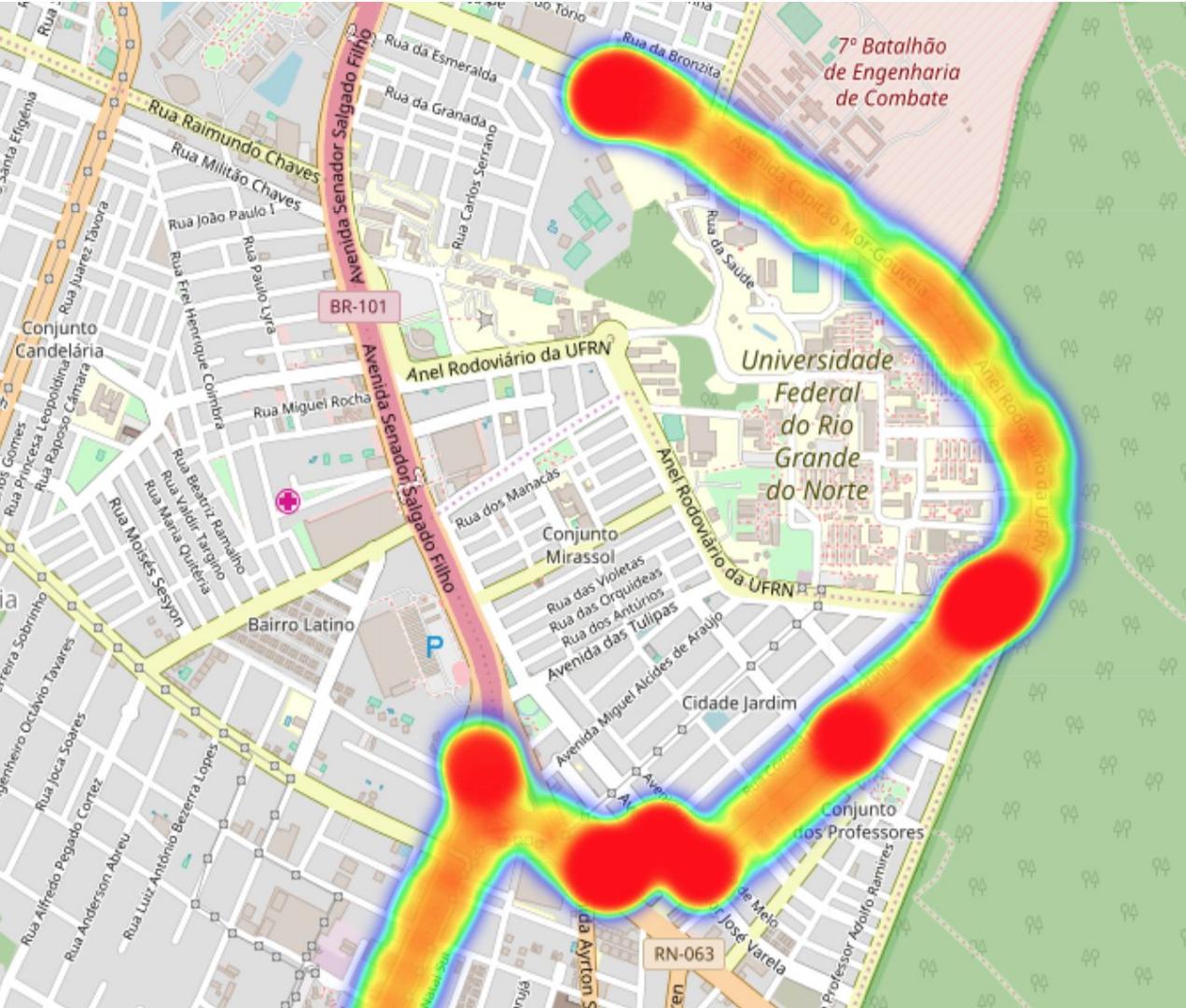
Geographic data is
always present in our
everyday lives

<https://goo.gl/iMCthG>





<https://earth.nullschool.net/>

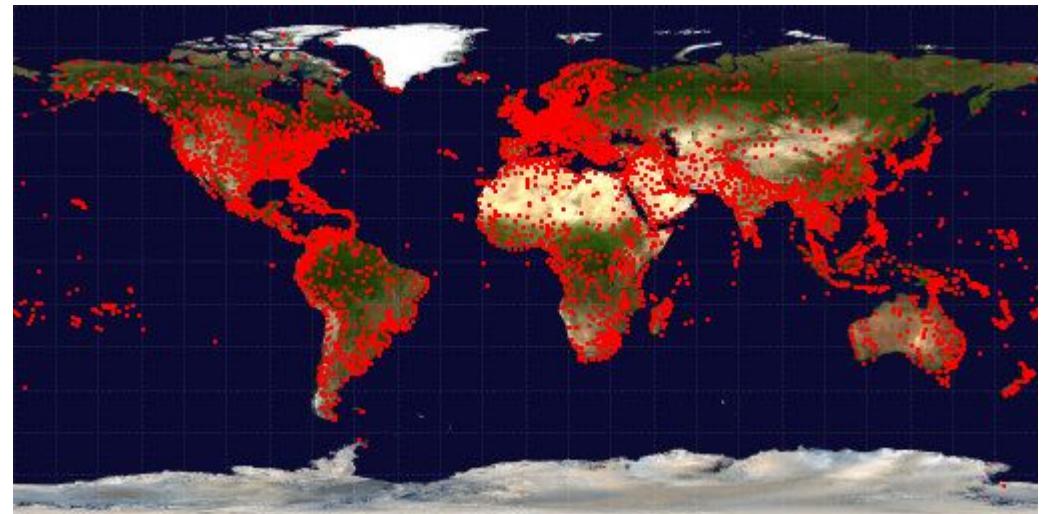


Raw geographic data like **latitudes** and **longitudes** are difficult to understand using the data charts and plots we've discussed so far

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.



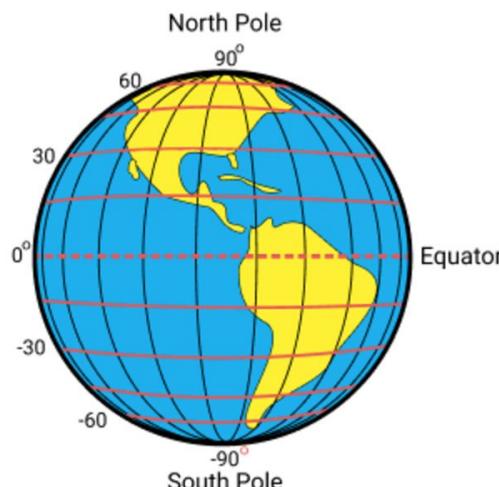
<https://openflights.org/data.html>

Geographic coordinate system

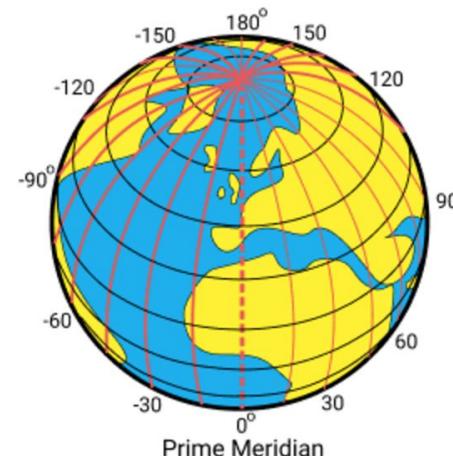
Describes the North-South position

Ranges from -90 to 90 degrees

Latitude



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

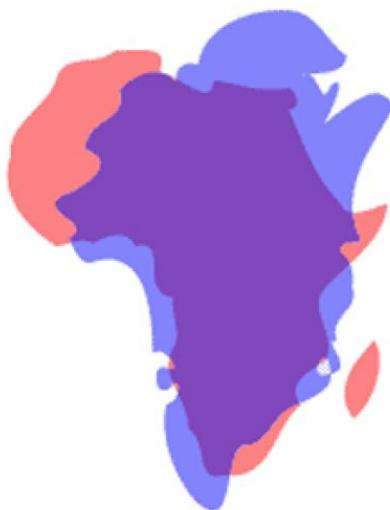
the problem with maps



The world as we know it

<https://goo.gl/ckFKh6>

Greenland is no Africa



Mercator



Actual



The true size of Africa

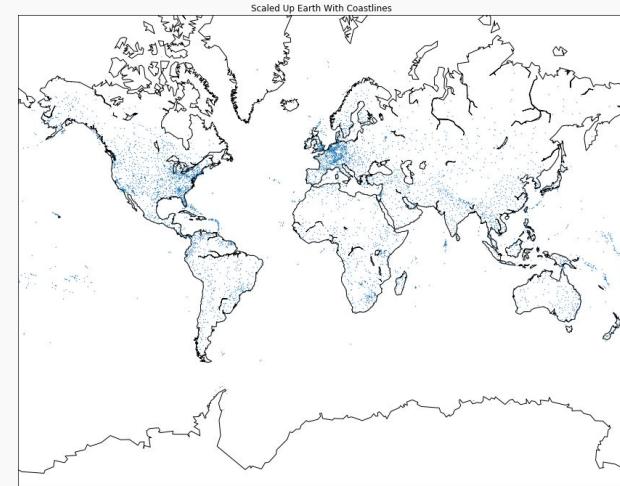
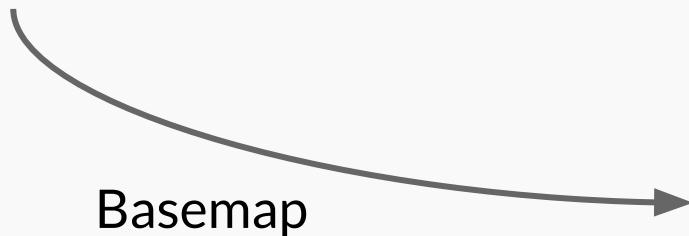
Map Projections

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

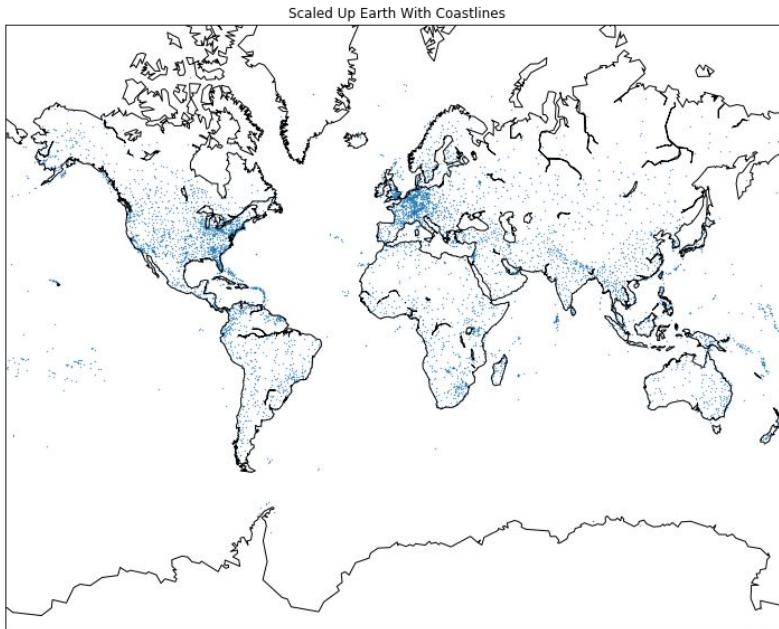


Workflow with basemap

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

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

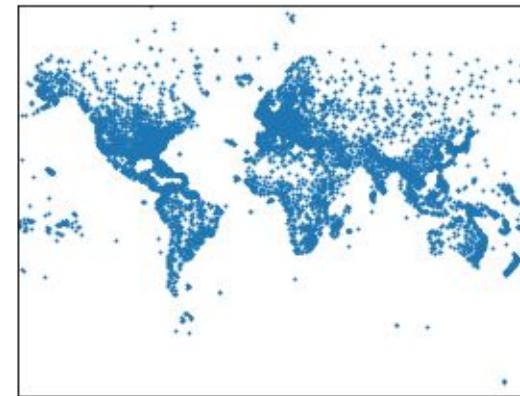
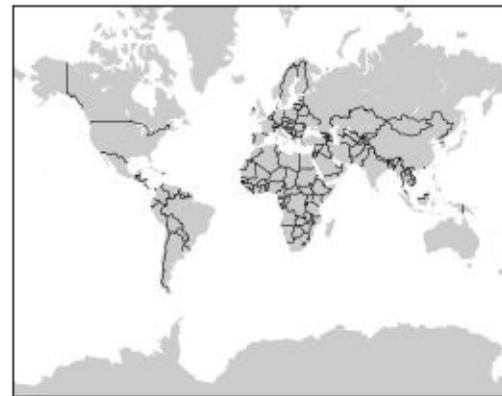
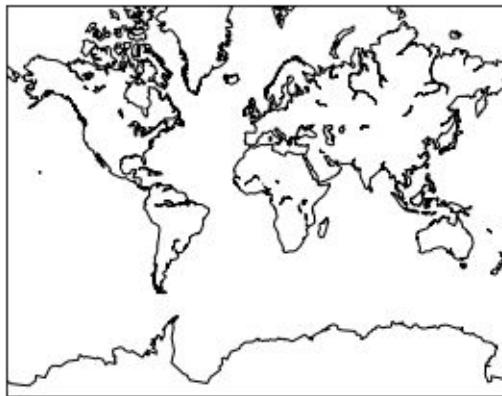
Converting from spherical to cartesian coordinates



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

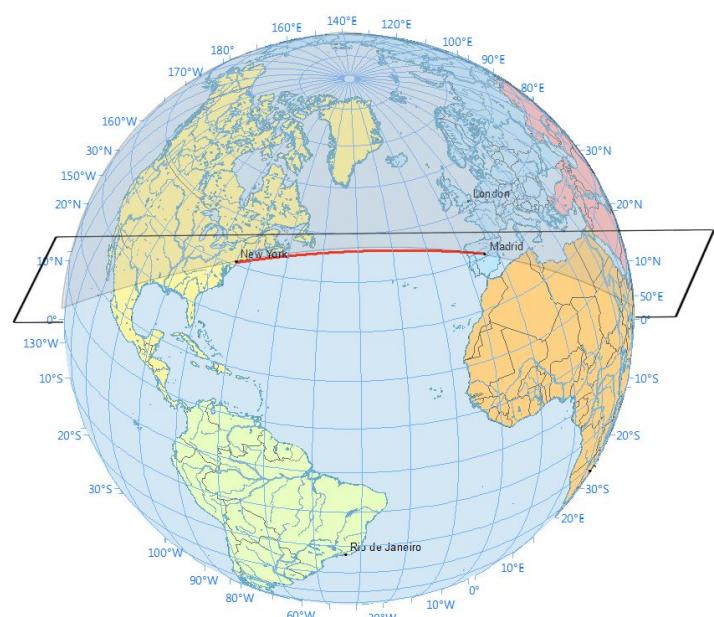
fig, ax = plt.subplots(figsize=(20,10))
plt.title("Scaled Up Earth With Coastlines")
m.scatter(x,y,s=0.1)
m.drawcoastlines()
plt.show()
```

Customizing the plot using Basemap



```
fig, ax = plt.subplots(ncols=3, nrows=1, figsize=(16,6))
m.drawcoastlines(ax=ax[0])
m.fillcontinents(ax=ax[1])
m.drawcountries(ax=ax[1])
m.scatter(x,y,s=1,ax=ax[2])
```

Introduction to great circles



Displaying great circles

lon1 - longitude of the starting point.

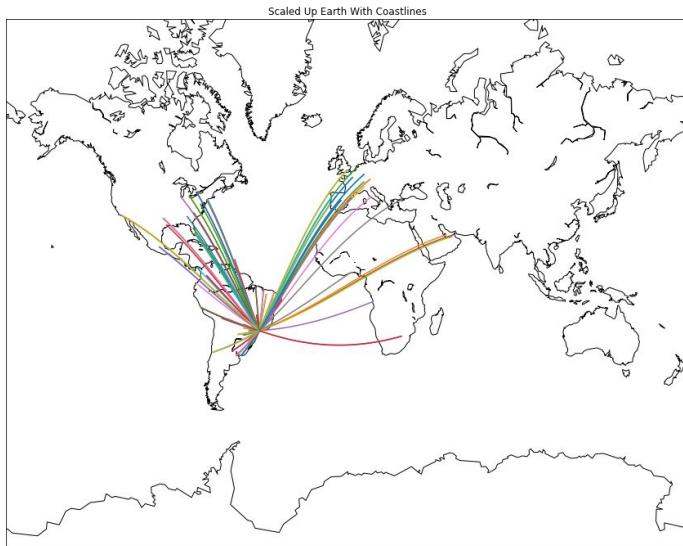
lat1 - latitude of the starting point.

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

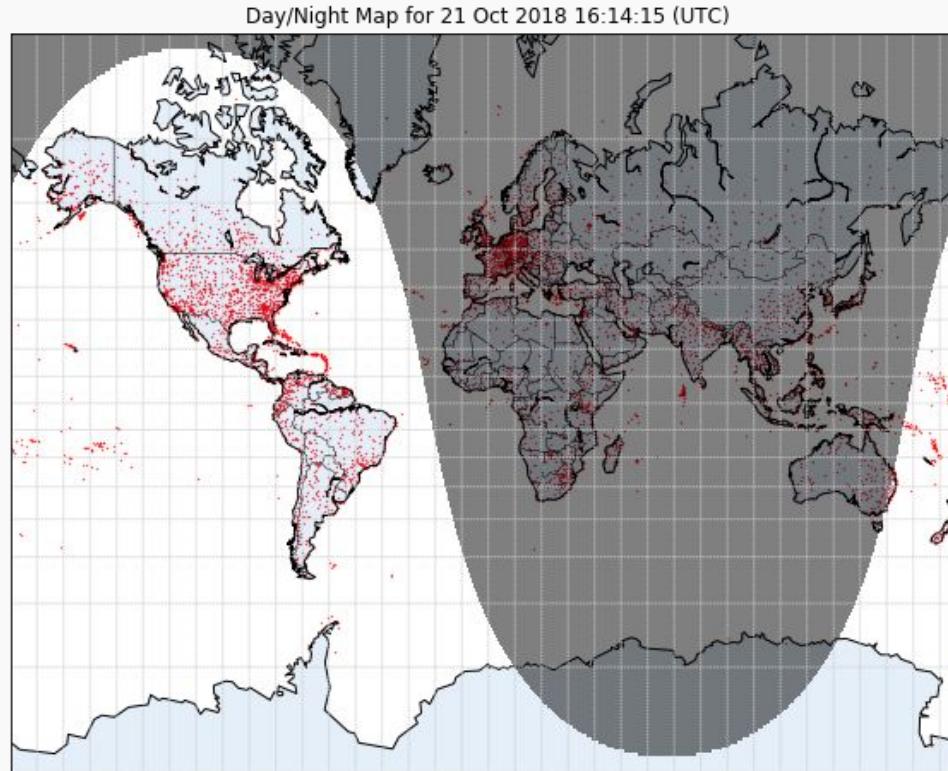


```
def create_great_circles(df):
    for index, row in df.iterrows():
        end_lat, start_lat = row["end_lat"], row["start_lat"]
        end_lon, start_lon = row["end_lon"], row["start_lon"]

        if (abs(end_lat-start_lat) < 180):
            if (abs(end_lon-start_lon) < 180):
                m.drawgreatcircle(start_lon, start_lat, end_lon, end_lat)

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

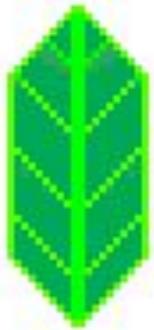
Day-night terminator on map



Lesson 10 - Exploratory Data Analysis V.ipynb

Section 1





Folium

Manipulate your data in Python, then visualize it in on a Leaflet map via Folium.



Creating a map of Brazil



```
import folium  
m = folium.Map(  
    location=[-15.765379,  
             -47.968776],  
    zoom_start=4  
)  
m.save("mapa.html")  
m
```

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



Stamen Toner



OpenStreetMap



Mapbox Control Room



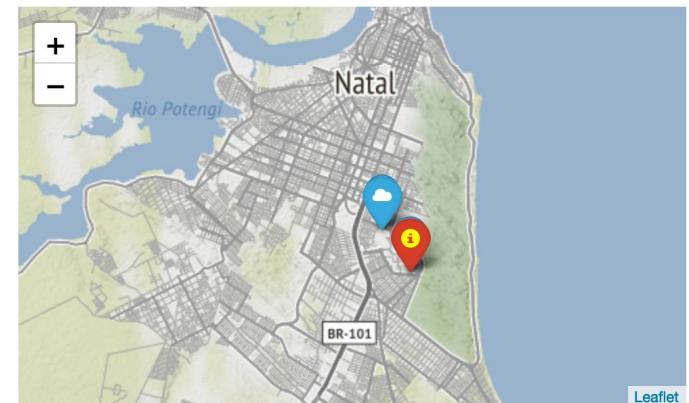
Maps with markers

```
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)
```

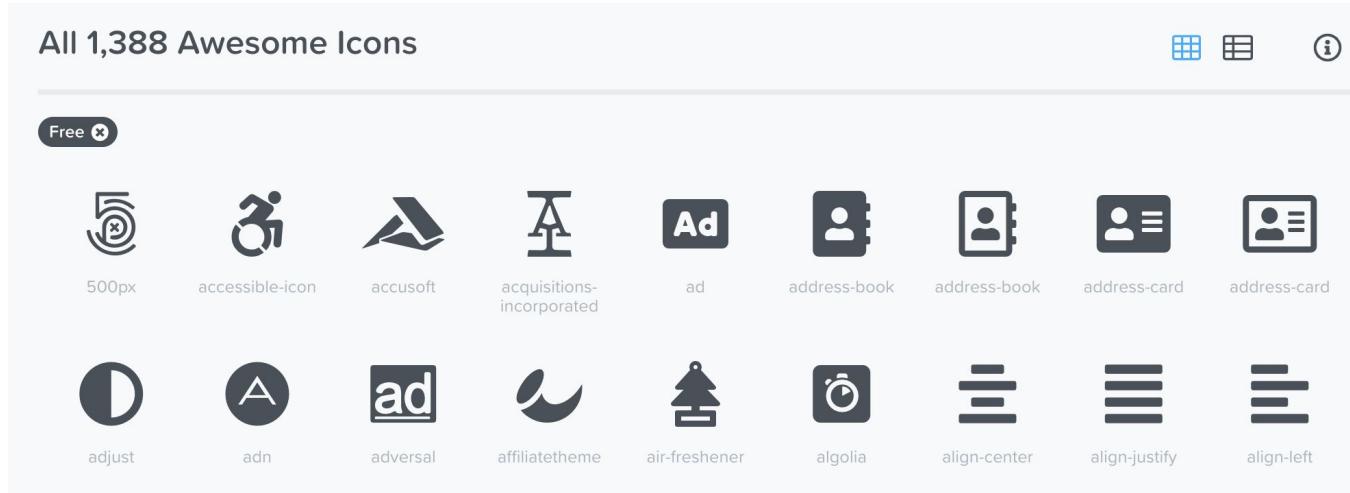


Color and icon types

```
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)
```



Customizing icons



<https://fontawesome.com/icons>

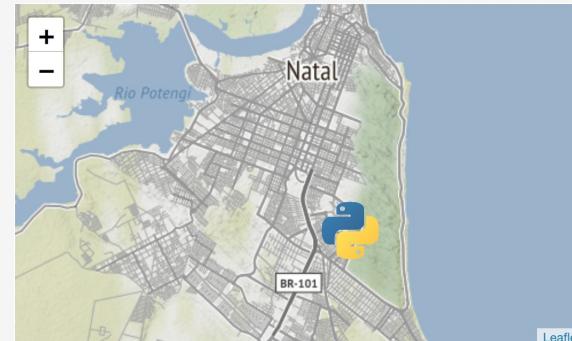
```
icon=folium.Icon(color='red',
                  icon_color='yellow',
                  icon='bicycle',
                  prefix='fa')).add_to(m)
```

Customizing icons

```
url = "https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/\nPython-logo-notext.svg/2000px-Python-logo-notext.svg.png"

icon = folium.features.CustomIcon(url,
                                   icon_size=(50,50))

folium.Marker([-5.842942, -35.198001],
              popup='<b>Centro Tecnológico</b>',
              icon=icon).add_to(m)
```



Customizing icons

```
import base64
from io import BytesIO

encoded = base64.b64encode(open('python.png', 'rb').read())
decoded = base64.b64decode(encoded)
icon_url = BytesIO(decoded)

icon = folium.features.CustomIcon(icon_url,
                                   icon_size=(50,50))

folium.Marker([-5.842942, -35.198001],
              popup='<b>Centro Tecnológico</b>',
              icon=icon).add_to(m)
```

Marker clusters



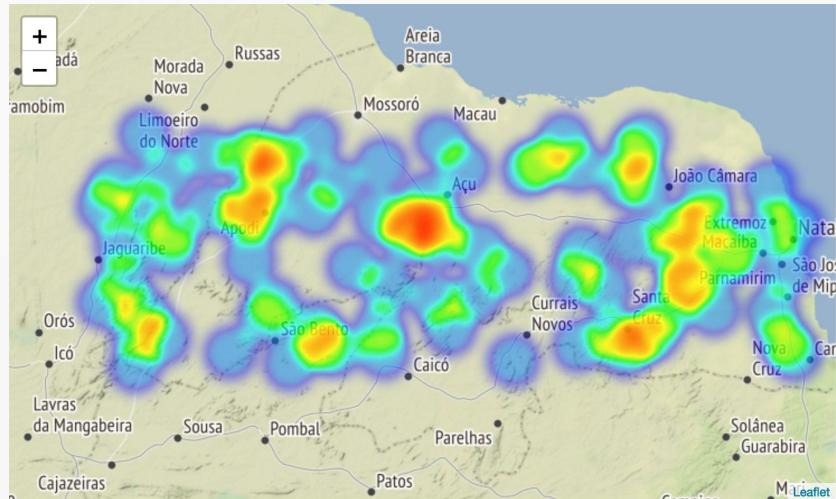
```
# Create a map
m = folium.Map(
    location=[-5.826592, -35.212558],
    zoom_start=12,
    tiles='Stamen Terrain',
    width='75%',
    height='75%'
)

# 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)
```

