

Programação II

Mapas web

Hugo Pacheco

DCC/FCUP
21/22

Mapas Web

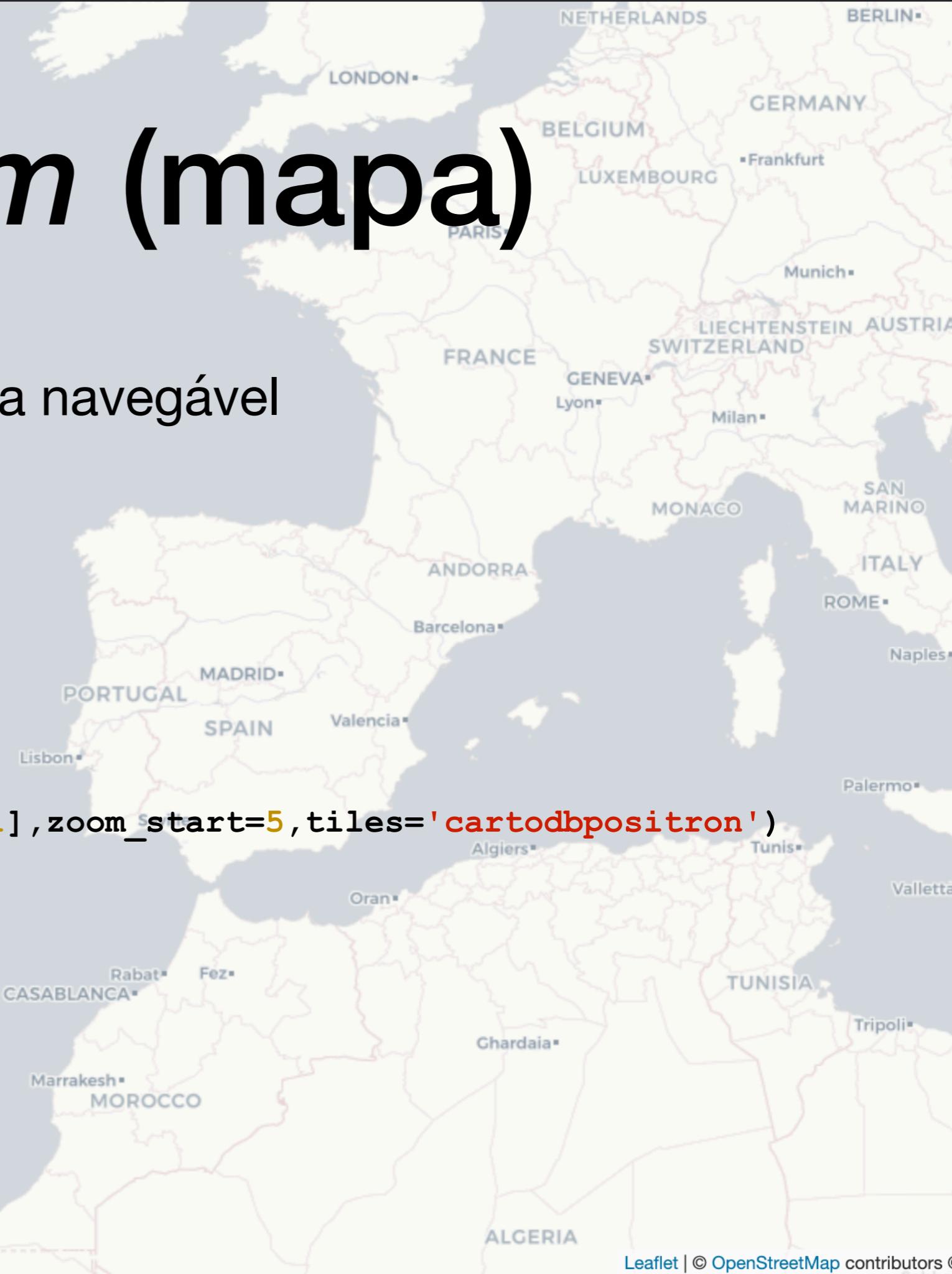
- Últimas aulas:
 - Gráficos e mapas offline usando *geopandas* + *contextily* + *matplotlib*
- Existem muitas bibliotecas para gráficos e mapas web
 - Mais modernas e interativas
 - Menos bem documentadas
 - Mais nuances tecnológicas (servidor/cliente, JavaScript, etc)
- Esta aula (**valorização Projeto 3**):
 - Mapas web em *folium*
 - Gerar mapas web navegáveis (HTML)
 - Adaptação de alguns exemplos anteriores

Folium (mapa)

- Podemos criar um mapa navegável

```
import folium

# sequencia de lat,lon
mapa =
folium.Map([41.1579,-8.6291],zoom_start=5,tiles='cartodbpositron')
mapa.save("mapa.html")
```



Folium (GeoJSON)

- Podemos criar um mapa a partir de um ficheiro GeoJSON, de forma similar ao que faz o geojson.io
- E.g., regiões de Portugal

```
# centrado no Porto
mapa = folium.Map([41.1579,-8.6291],
                  zoom_start=5,
                  tiles='cartodbpositron')

with open('portugal.geojson','r') as f: geojson = json.load(f)
folium.GeoJson(geojson).add_to(mapa)
mapa.save("mapa.html")
```

Folium (GeoJSON)

- Podemos criar um mapa a partir de um ficheiro GeoJSON, de forma similar ao que faz o geojson.io
- E.g., estações metereológicas ([IPMA](#))

```
# centrado no Porto
mapa =
folium.Map([41.1579,-8.6291],
zoom_start=6,tiles='stamenterrain')

with open('obs-surface.geojson','r') as f:
geojson = json.load(f)
folium.GeoJson(geojson).add_to(mapa)
mapa.save("mapa.html")
```



Folium (marcadores)

- Podemos criar e costumizar marcadores
- E.g., estado do tempo



Folium (marcadores)

- Carregar dados com `geopandas`
- Adicionar marcadores um a um (lista de ícones)

```
mapa = folium.Map([41.1579,-8.6291],zoom_start=8,tiles='stamenterrain')

gdf = gpd.read_file('obs-surface.geojson')
gdf['time'] = pd.to_datetime(gdf['time'])
latest_time = gdf['time'].max()
gdf = gdf[gdf['time'] == latest_time]

for i, row in gdf.iterrows():
    if row['temperatura'] >= 20: co='red';
    elif row['temperatura'] >= 10: co='orange';
    elif row['temperatura'] >= 0: co='blue'
    else: co='gray';
    icn = folium.Icon(color=co, icon='glyphicon-cloud')

folium.Marker(location=(row['geometry'].y, row['geometry'].x),icon=icn,tooltiptext=row['localEstacao'],popup=str(row)).add_to(mapa)
mapa.save("mapa.html")
```

Folium (mapas cores)

- E.g., dados de mortes totais por COVID-19 da OMS, código de exemplo anterior
- Processar mortes totais (dados de cores)

```
codes = pd.read_csv("country-codes.csv",usecols=['ISO3166-1-Alpha-3','ISO3166-1-Alpha-2'])
codes = codes.rename(columns={'ISO3166-1-Alpha-3':'ISO_A3','ISO3166-1-Alpha-2':'ISO_A2'})
df = pd.read_csv("WHO-COVID-19-global-data.csv")
df = df.groupby(by='Country').tail(1)
df =
df.rename(columns={'Country_code':'ISO_A2','Cumulative_deaths':'Deaths'})
df = df[['ISO_A2','Deaths']]
gdf = pd.merge(codes,df,how='inner')
```

Folium (mapas cores)

- Carregar coordenadas *geojson* e dados de cores à parte
- Código de palete de cores, e.g.. <https://colorbrewer2.org/>

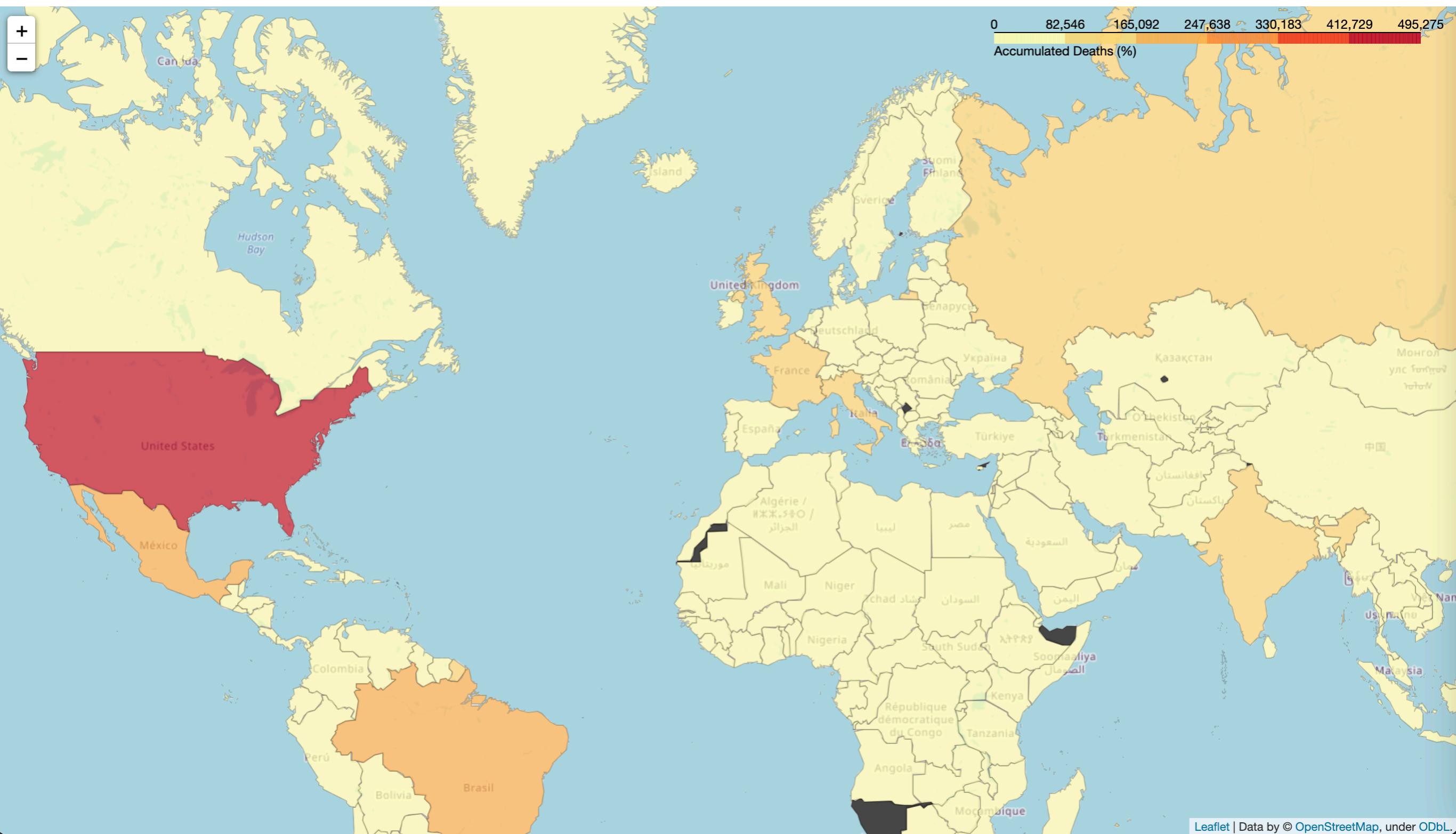
```
with open('countries.geojson', 'r') as f: geojson = json.load(f)

mapa = folium.Map([41.1579, -8.6291], zoom_start=3)

folium.Choropleth(
    geo_data=geojson,
    data=gdf,
    columns=["ISO_A3", "Deaths"],
    key_on="feature.properties.ISO_A3",
    fill_color="YlOrRd",
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name="Accumulated Deaths (%)",
).add_to(mapa)

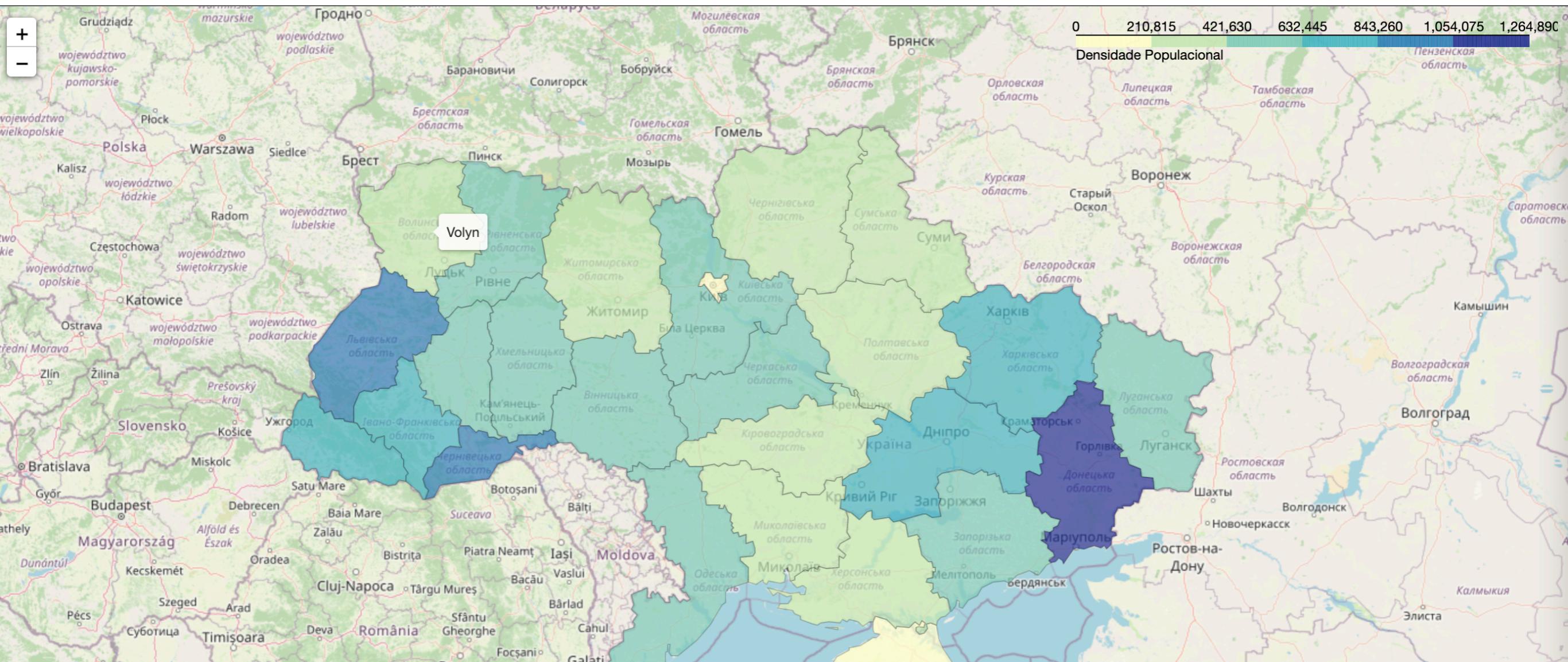
mapa.save("mapa.html")
```

Folium (mapas cores)



Projeto 3: Tarefa 3

1. Construir um *DataFrame* com duas colunas [*regiao,densidade*])
2. Gerar um mapa *Chloropeth* (**Grátis!**)
3. Acrescentar *tooltips* e *popups* sobre cada região, usando outras colunas (**Grátis!**)



Folium (mapas cores + slider)

- E.g., dados de mortes diárias por COVID-19 da OMS, código de exemplo anterior
- Processar dados de mortes (para atribuir cores do mapa)

```
codes = pd.read_csv("country-codes.csv",usecols=['ISO3166-1-Alpha-3','ISO3166-1-Alpha-2'])
codes = codes.rename(columns={'ISO3166-1-Alpha-3':'ISO_A3','ISO3166-1-Alpha-2':'ISO_A2'})
df = pd.read_csv("WHO-COVID-19-global-data.csv")
df = df.rename(columns={'Country_code':'ISO_A2'})
gdf = pd.merge(codes,df,how='inner'); gdf = gdf.dropna()
gdf['Date_reported'] = pd.to_datetime(gdf['Date_reported'])
# data em milésimos de segundo
gdf['Date']=(gdf['Date_reported'].astype(int)// 10**9).astype('U10')
# máximo de mortes diário
gdaily = gdf.groupby(by='Date')[['New_deaths']].max()
gdaily.rename(columns={'New_deaths':'max_deaths'},inplace=True)
```

Folium (mapas cores + slider)

- Dados de cores devem ser um dicionário com o formato
{ id ↠ { data ↠ { 'color' ↠ hex, 'opacity' ↠ [0..1] } } }
- Atribuir cores manualmente: vermelhos RGB (código hexadecimal) em função do máximo de mortes

```
def sort_df(df):  
    df = df.set_index('Date')[['New_deaths']]  
    df = pd.merge(df,gdaily,on='Date')  
    #print(df)  
    df['color'] = df['New_deaths'] * 255 / df['max_deaths']  
    df.fillna(0,inplace=True)  
    del df['New_deaths']  
    del df['max_deaths']  
    df['color'] = df['color'].apply(lambda r :  
        '#%02x%02x%02x' % (round(r),0,0))  
    df['opacity'] = 0.5  
    return df.to_dict(orient='index')  
  
geodict = { iso3 : sort_df(df) for iso3,df in  
gdf.groupby(by='ISO_A3') }
```

Folium (mapas cores + slider)

- Carregar *geojson* com dados geográficos
- Desenhar o mapa passando coordenadas e cores à parte

```
import folium.plugins

mapa = folium.Map([0,0],zoom_start=2)

geodf = gpd.read_file('countries.geojson')
geodf.set_index('ISO_A3', inplace=True)
del geodf['ADMIN']
geojson = geodf.to_json()

folium.plugins.TimeSliderChoropleth(
    geojson,
    styledict=geodict
).add_to(mapa)

mapa.save("mapa.html")
```

Folium (mapas cores + slider)

Thu Jan 14 2021

