

Programação II

+

Estruturas de Dados para Bioinformática

Visualização web

Hugo Pacheco

DCC/FCUP
22/23

Gráficos Web

- Últimas aulas: Gráficos e mapas offline
- Existem muitas bibliotecas para gráficos e mapas web
 - Mais modernas e interativas
 - Menos bem documentadas
 - Mais nuances tecnológicas (servidor/cliente, JavaScript, etc)
- Esta aula: Gráficos e mapas web em *plotly*, *folium* e *Bokeh*



Animações



Interatividade (e.g. botões)

Plotly (bar)

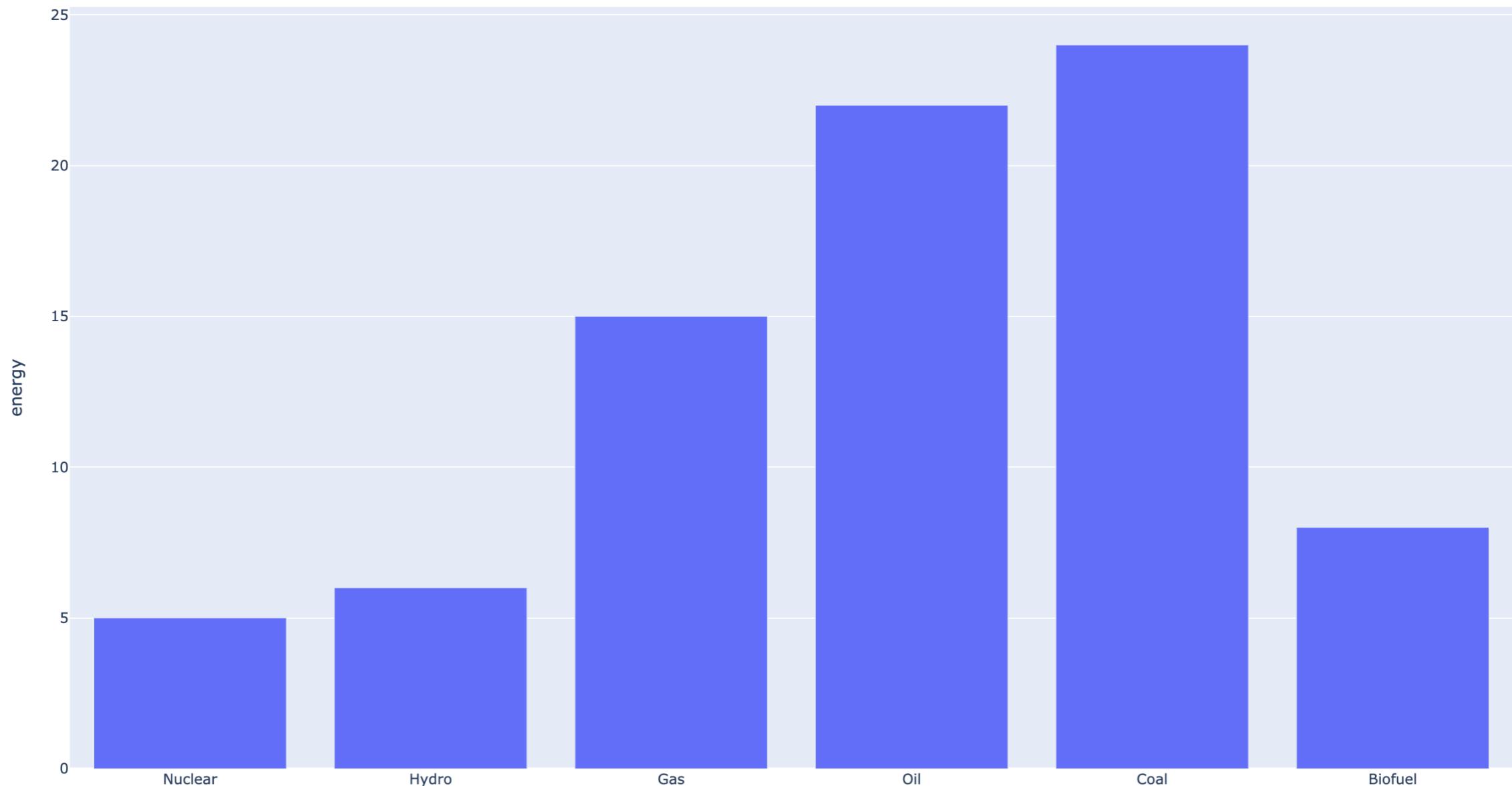
- Como desenhar um gráfico (de barras):
 1. Organizar dados num só DataFrame
 - Coluna X: eixo dos X
 - Coluna Y: eixo dos Y
 2. Uma só chamada para gerar ficheiro HTML

```
import plotly.express as px

fig = px.bar(df, x="x", y="y")
fig.write_html("file.html")
```

Plotly (bar)

- E.g., distribuição de energia por fontes



Plotly (bars)

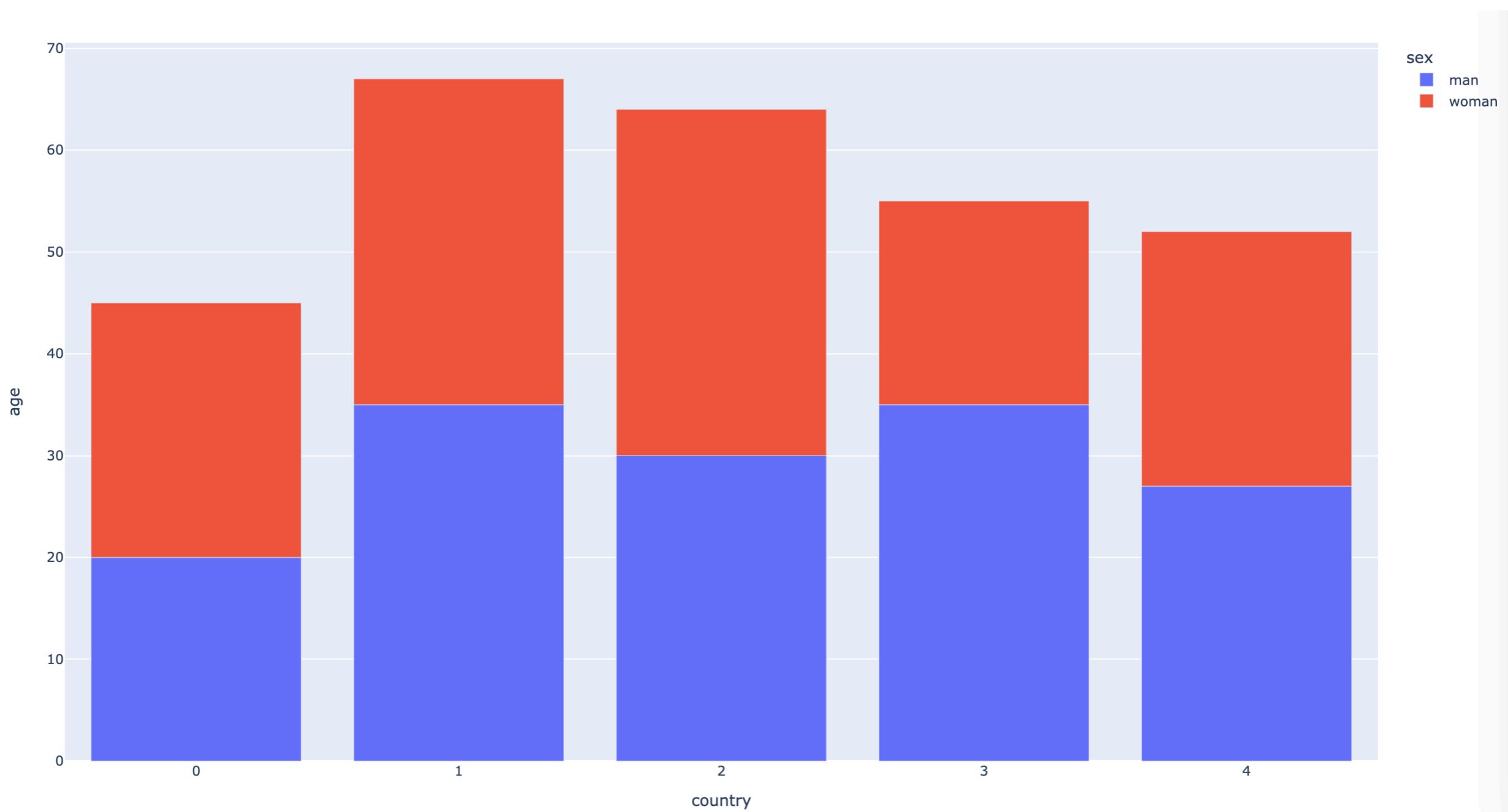
- Como desenhar um gráfico (com múltiplas barras):
 1. Organizar dados num só DataFrame
 - Coluna X: eixo dos X
 - Coluna Y: eixo dos Y
 - Coluna Color: qual a categoria/cor a que os dados (X,Y) pertencem
 2. Uma só chamada para gerar ficheiro HTML

```
import plotly.express as px

fig = px.bar(df, x="X", y="Y", color='Color')
fig.write_html("file.html")
```

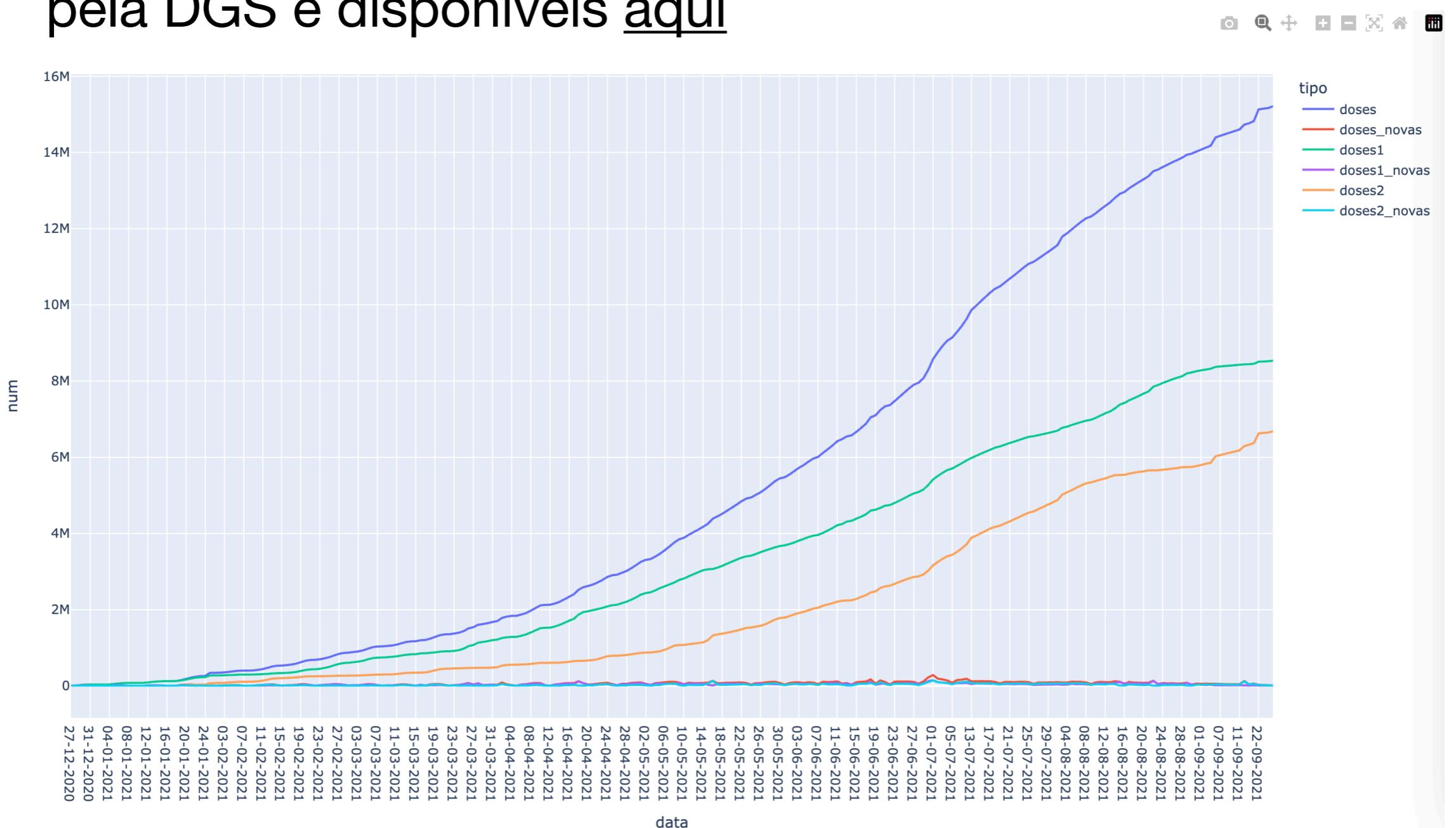
Plotly (bar)

- E.g., demografia por sexo



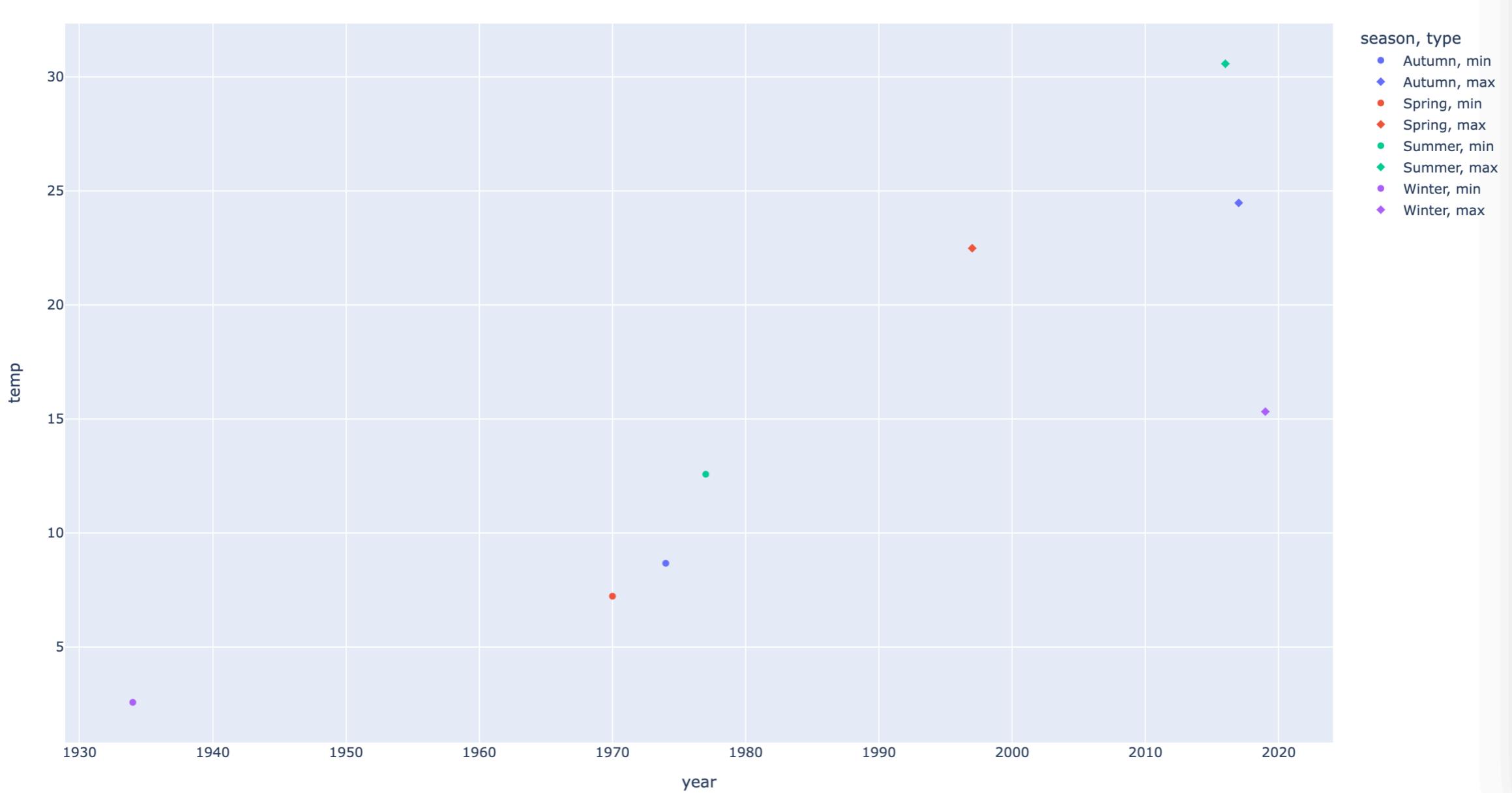
Plotly (lines)

- E.g., dados de vacinação contra a COVID-19 publicados pela DGS e disponíveis [aqui](#)



Plotly (scatter)

- E.g., dados climatéricos anuais de longa duração fornecidos pelo IPMA [aqui](#).

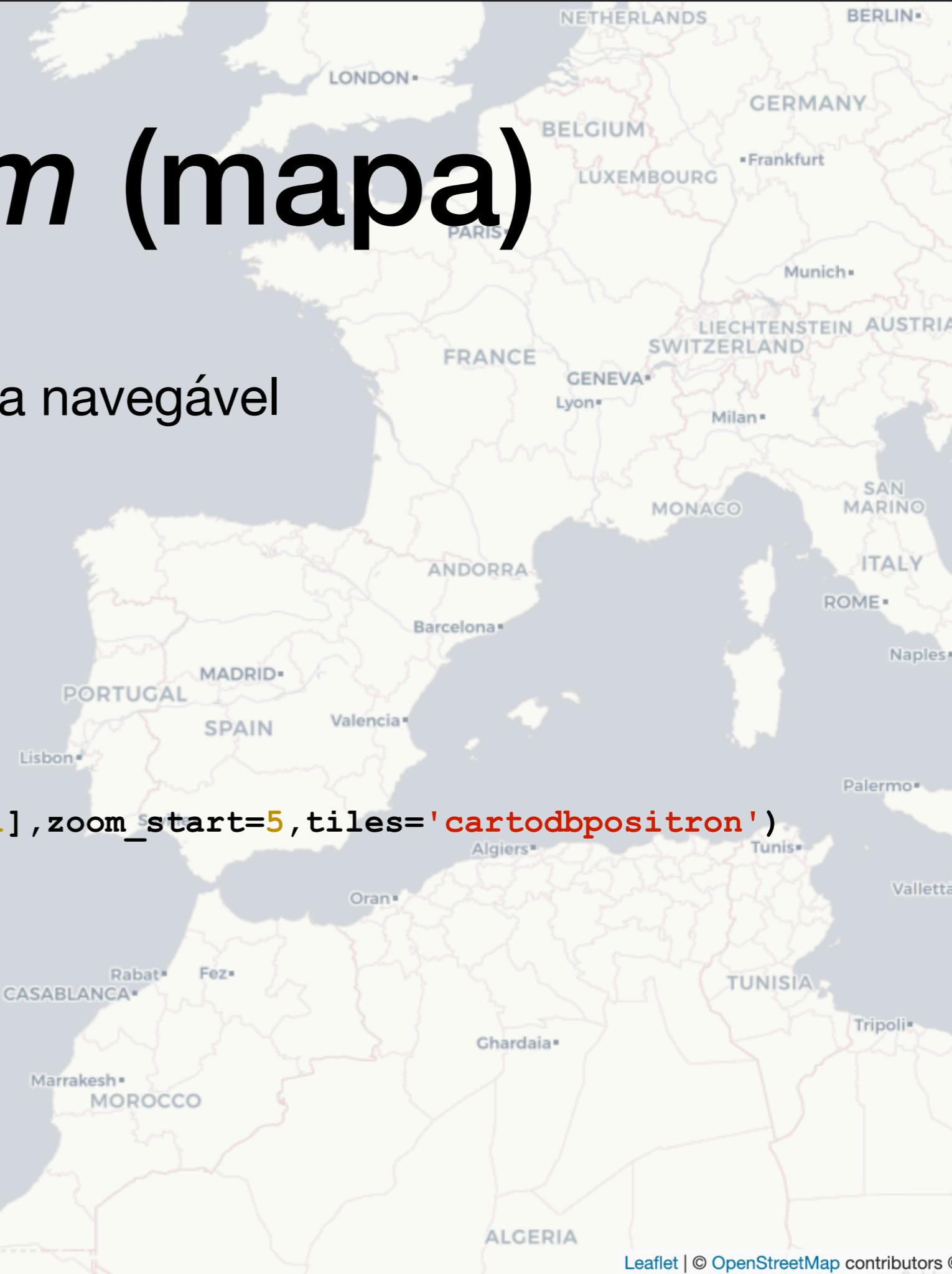


Folium (mapa)

- 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/>

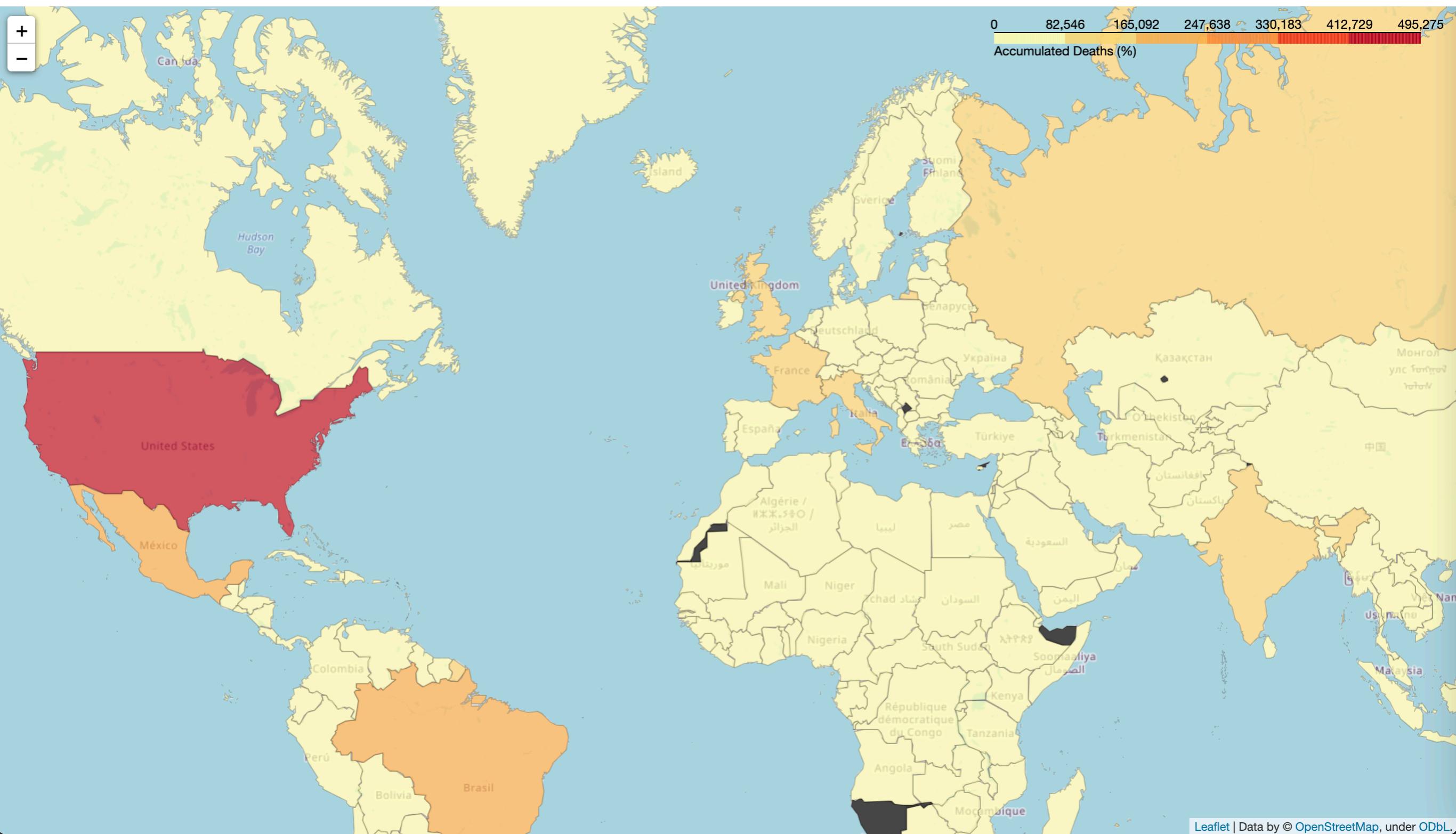
```
with open('countries.geojson', 'r') as f: geojson = json.load(f)

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)



Folium (animação)

- E.g., uma animação do percurso e intensidade do vento de furacões que assolaram a Costa Atlântica dos EUA
 - Dataset HURDAT2 com dados desde 1851 disponibilizado pela NOAA (United States National Oceanic and Atmospheric Administration); versão CSV aqui
1. Animar trajetória como círculos com posição de longitude/latitude e raio proporcional à intensidade do vento
 2. Cada círculo representa uma medição num instante de tempo concreto, e tem uma duração fixa na animação

Folium (animação)

- E.g., uma animação do percurso e intensidade do vento de furações que assolararam a Costa Atlântica dos EUA



Plotly (grafos)

- Podemos desenhar grafos web...



Construir grafo com o *NetworkX*

Calcular layout do grafo com o *NetworkX*



Converter arestas num *DataFrame pandas*

Desenhar arestas (i,j) como um scatter plot usando o *plotly*

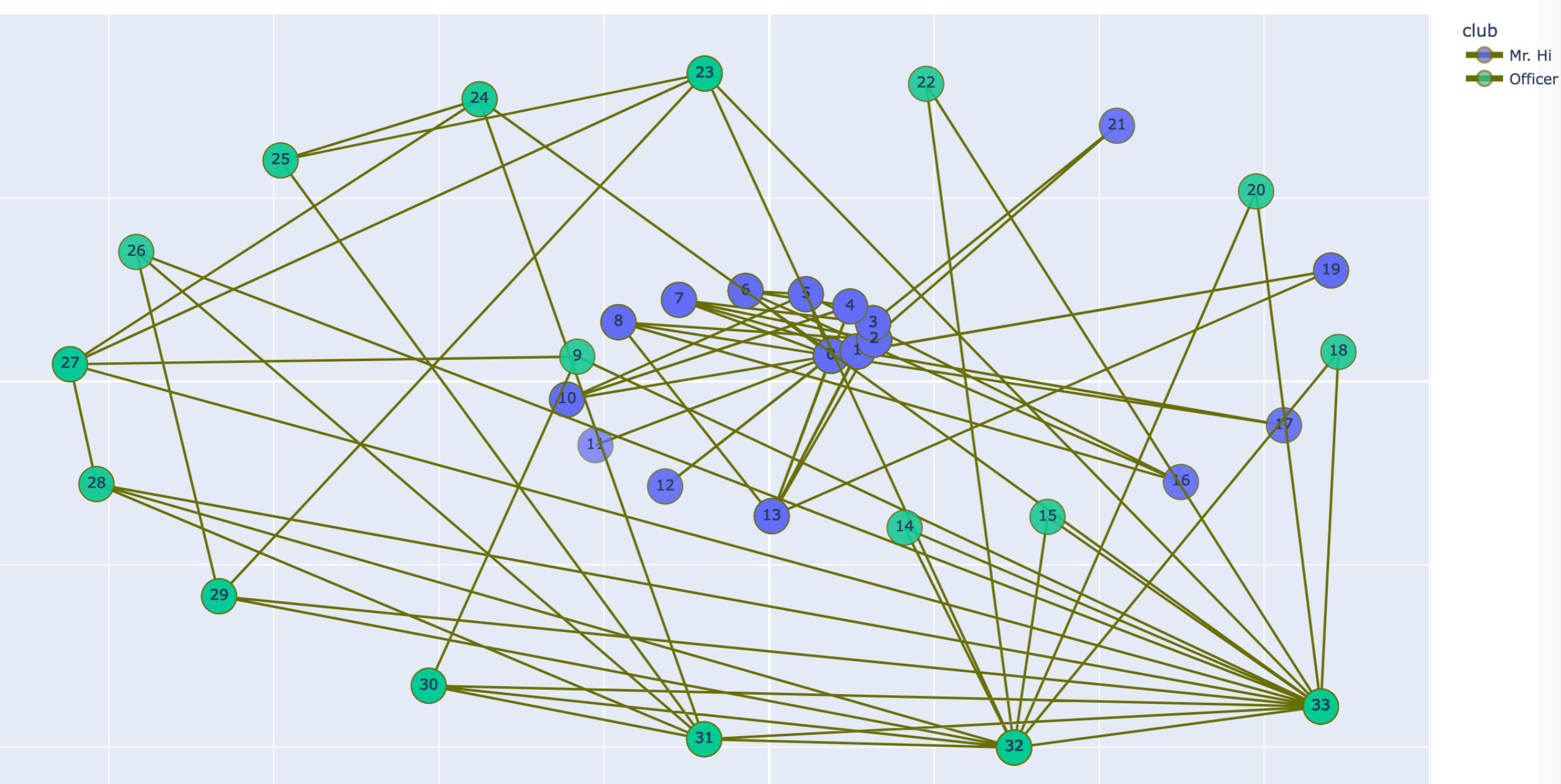
Node	X	Y
i	x_i	y_i
j	x_j	y_j
NaN	NaN	NaN



Várias complicações e limitações, e.g., labels?

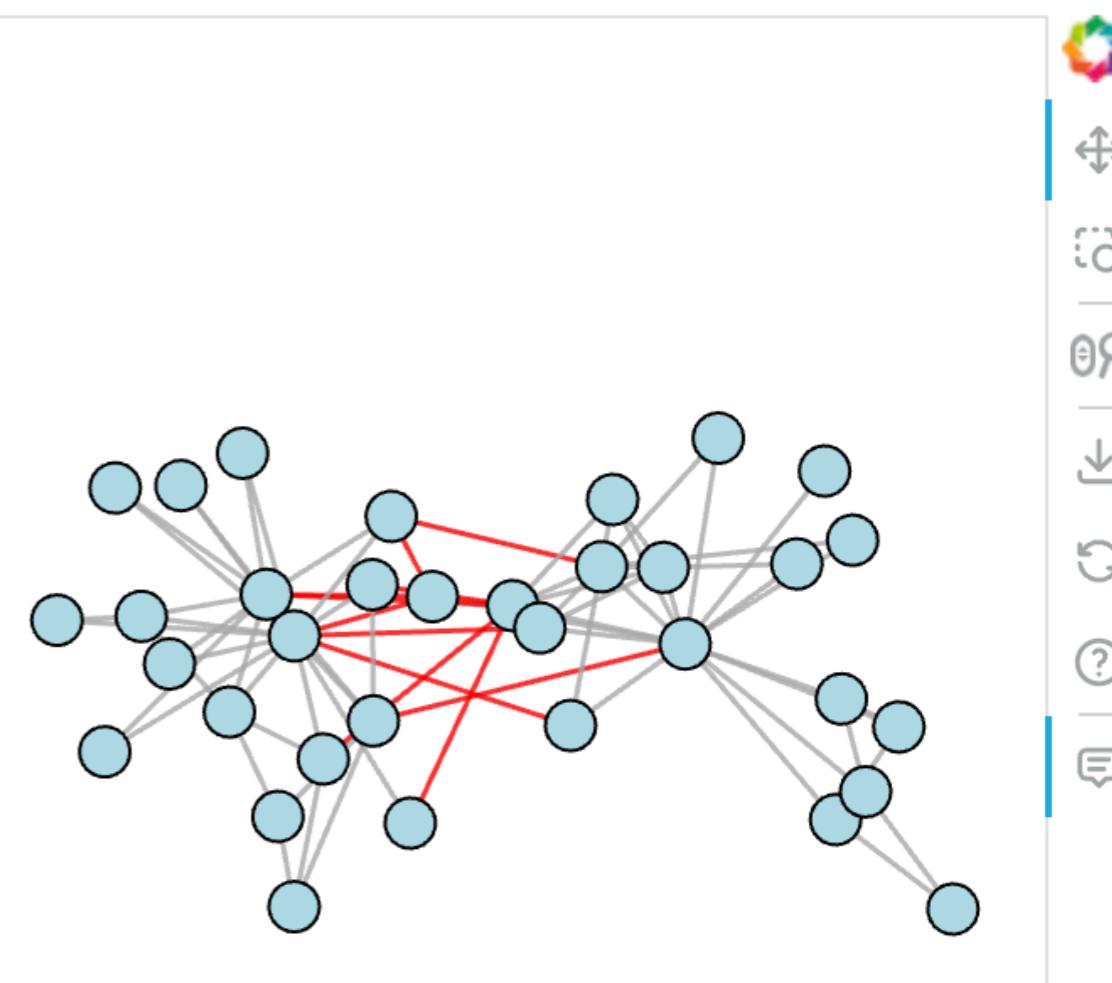
Plotly (grafos)

- E.g., grafo da aula anterior do karate club



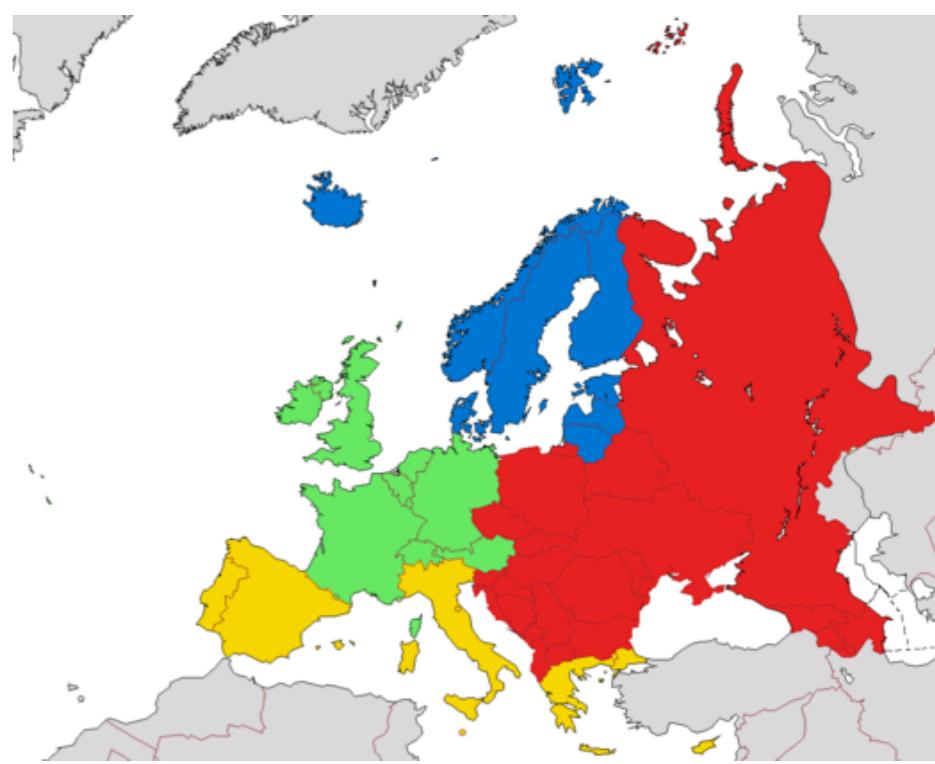
Bokeh (grafos)

- Melhor integração com o *NetworkX*
 - Podemos desenhar diretamente um grafo
- Biblioteca mais complexa
- E.g., grafo da aula anterior do karate club



Bokeh (chord plot)

- Um tipo concreto de visualização circular de grafos
- E.g., correlação de votos com regiões geopolíticas de países na Eurovisão
- Dados de votação retirados [daqui](#)
- Identificação de regiões retirada do [EuroVoc](#)
- Essencialmente um grafo direcionado
 - Nodos = países
 - Arestas = país P1 deu X pontos a país P2



Bokeh (chord plot)

- E.g., correlação de votos com regiões geopolíticas de países na Eurovisão

