

▼ Project #1

Data Visualization (LECD) 2023/24

Student name: Rodrigo Miguel Rodrigues Santos

Instructions

- Do not modify the structure of this notebook!
- Fill-in the corresponding cells with the code or text depending on the objective
- You can use this template as a working notebook, or copy&paste your final code here
- It is recommended that you export and submit the rendered version of the notebook
- Submit the notebook not later than the date specified on the Infoestudante webpage of the course

▼ Data

O Dataset que escolhi foi "Life Expectancy Data", que apresenta diversas informações sobre não só a life expectancy de alguns países desde 2000 até 2015, como também o estatuto do país (desenvolvido ou em desenvolvimento) e outras informações sobre mortalidade e doenças. Quanto a atributos categóricos temos: Country e Status; Quanto a atributos quantitativos temos: Year, Life expectancy, Adult Mortality, Infant deaths, Alcohol, Percentage expenditure, Hepatitis B, Measles, BMI, Under-five deaths, Polio, Total expenditure, Diphtheria, HIV/AIDS, GDP, Population, Thinness 1-19 years, Thinness 5-9 years, Income composition of resources, Schooling. Para o meu projeto, apenas incidi na life expectancy de cada país no ano de 2015. #falar sobre atributos quantitativos e categoricos

Quanto à limpeza dos dados, removi os espaços em branco no início e no final dos nomes das colunas, passei os espaços em branco entre palavras para underscore e todas as letras para minúsculas, isto tudo para ser mais fácil manipular os dados. Com a ajuda da biblioteca "country-converter", criei uma nova coluna no dataset chamada "continent" onde, para a linha de cada país, é adicionado o seu respetivo continente (por exemplo, na linha onde o país é Portugal, vai ser adicionado na coluna "continent"- Europe).

```
!pip install country-converter
```

```
Collecting country-converter
  Downloading country_converter-1.0.0-py3-none-any.whl (44 kB)
    44.5/44.5 kB 1.3 MB/s eta 0:00:00
Requirement already satisfied: pandas>=1.0 in /usr/local/lib/python3.10/dist-packages (from country-converter) (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0->country-conve
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0->country-converter) (202
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0->country-converter) (1.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas>=1.0->cou
Installing collected packages: country-converter
Successfully installed country-converter-1.0.0
```

```
# import your dataset in this section (cell)
import pandas as pd
import plotly.graph_objects as go
import plotly.express as px

df = pd.read_csv("/content/Life Expectancy Data.csv")

orig_cols = list(df.columns)
new_cols = []
for col in orig_cols:
    new_cols.append(col.strip().replace(' ', '_').replace('-', '_').lower())
df.columns = new_cols

from country_converter import CountryConverter
cc = CountryConverter()

df['continent'] = df['country'].apply(lambda x: cc.convert(names=x, to='continent'))
df_2015 = df[df['year'] == 2015]
```

▼ Visualization model #1

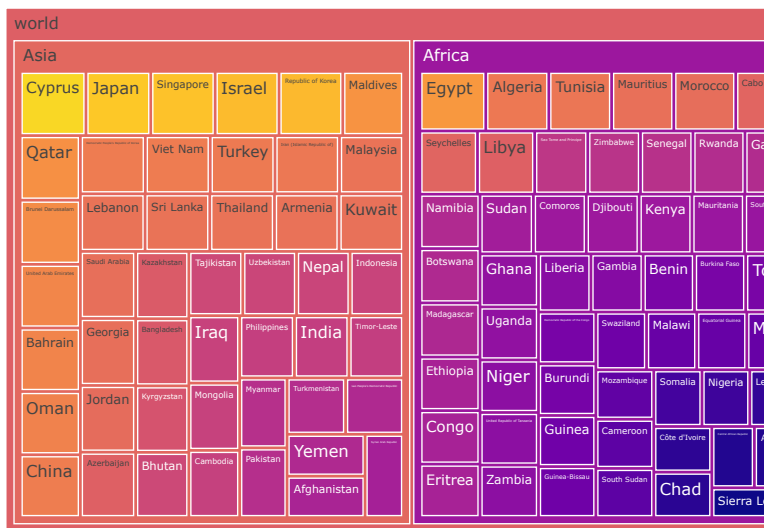
```
#implementation of the viz 1 in this cell
```

```
def treemap(df):
    fig = px.treemap(df_2015, path=[px.Constant('world'), 'continent', 'country'], values='life_expectancy', color='life_expectancy')
```

```
fig.update_layout(title='Life Expectancy by Country in 2015',coloraxis_colorbar=dict(title='Life Expectancy'))
fig.show()
```

```
treemap(df)
```

Life Expectancy by Country in 2015



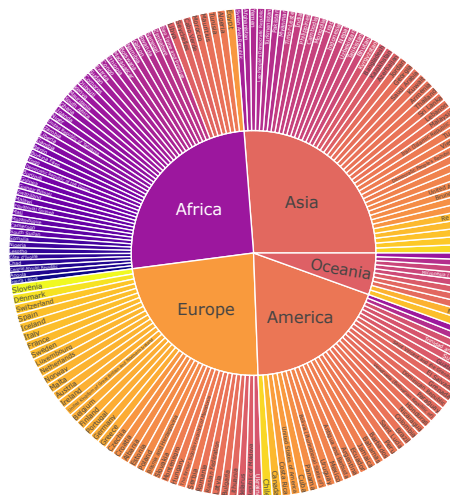
Visualization model #2

```
#implementation of the viz 2 in this cell
```

```
def sunburst(df):
    fig = px.sunburst(df_2015, color = 'life_expectancy', values = 'life_expectancy', path = ['continent','country'], hover_name='country')
    fig.update_layout(title='Life Expectancy by Country in 2015',coloraxis_colorbar=dict(title='Life Expectancy'))
    fig.show()
```

```
sunburst(df)
```

Life Expectancy by Country in 2015



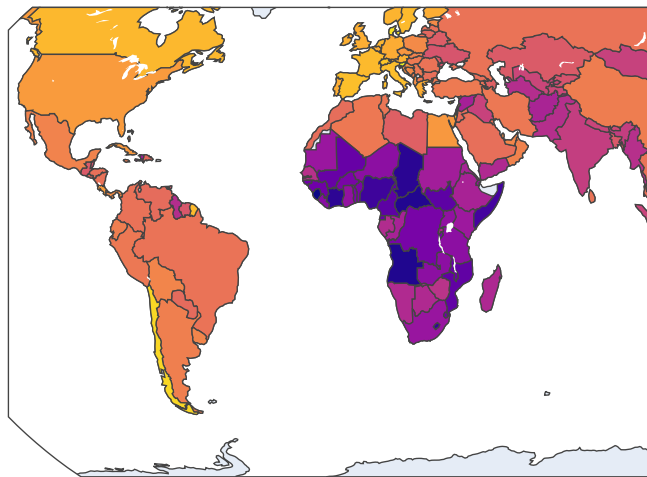
Visualization model #3

#implementation of the viz 3 in this cell

```
def map_life_expectancy(df):
    fig = go.Figure(data=go.Choropleth(
        locations=df_2015["country"],
        locationmode='country names',
        z=df_2015["life_expectancy"],
        hovertext=df_2015["country"] + ' -\nLife Expectancy: ' + df_2015["life_expectancy"].astype(str),
        hoverinfo="text",
        colorbar=dict(title="Life Expectancy")
    ))
    fig.update_layout(
        title_text="Life Expectancy by Country in 2015",
        geo=dict(
            showframe=True,
            showcoastlines=True,
            projection_type='natural earth'
        )
    )
    fig.show()

map_life_expectancy(df)
```

Life Expectancy by Country in 2015



Visualisation model #4

#implementation of the viz 4 in this cell

```
def graph3d(df):
    fig = go.Figure(data=go.Scatter3d(
        x=df_2015['year'],
        y=df_2015['country'],
        z=df_2015['life_expectancy'],
        text='Status: ' + df_2015["status"] + ' -\nLife Expectancy: ' + df_2015["life_expectancy"].astype(str),
        mode='markers',
        marker=dict(
            color=df_2015['life_expectancy'],
            opacity=0.8,
            size=5,
            colorbar=dict(title="Life Expectancy"),
        )
    ))
    fig.update_layout(scene=dict(
        xaxis_title='Year',
        yaxis_title='Country',
        zaxis_title='Life Expectancy',
        yaxis=dict(showticklabels=False),
    ), title_text="Life Expectancy by Country in 2015",)
    fig.show()
```

Life Expectancy by Country in 2015

Critical reflection

Como pretendido implementei 4 visualizações diferentes para os meus dados, tendo estas 4 sido: 1) Tree Map; 2) Sunburst; 3) Choropleth; 4) Scatter 3D.

Das 4 visualizações, aquela que na minha opinião nos dá uma melhor visão dos dados é a Tree Map, visto que estão diretamente explícitos os nomes e as cores consoante a Life Expectancy e, mesmo podendo, não precisamos de clicar nas células para expandir, ao contrário da Sunburst (pois o nome dos países não é muito bem legível). A visualização mais memorável é a Choropleth pois é uma representação do mapa do mundo ao qual já estamos bastante habituados e familiarizados.

A visualização que me vem primeiro à cabeça é a Scatter 3D pois é aquela mais fácil de perceber e com menos "poluição" (não é tão pesada ao olhar). Das 4 visualizações, penso que a melhor para comunicação seria a Sunburst dado a sua capacidade de ampliar o continente (interação possível com um público, por exemplo) e ver os países ordenados por Life Expectancy; para exploração ou análise, a Choropleth, uma vez que, sendo o mapa mundo, é possível sobrepor muitas outras informações e ter uma análise mais aprofundada seja do mundo em geral, dos continentes ou até mesmo dos países. Cada uma das visualizações pode ser atribuída a várias tarefas, tendo por exemplos:

- Tree Map – Visualização de estruturas e as suas infraestruturas;
- Sunburst – Dados relacionados com hierarquias;
- Choropleth – Representar dados geográficos;
- Scatter 3D – Comparar 3 variáveis em vez de 2;

References

All the sources you have used in this project should be cited and credited in this section.

<https://plotly.com/python/plotly-express/>

<https://plotly.com/python/treemaps/>

<https://plotly.com/python/sunburst-charts/>

<https://plotly.com/python/3d-scatter-plots/>

<https://plotly.com/python/colormaps/>

<https://www.youtube.com/watch?v=FpCgG85g2Hw>

<https://chart-studio.plotly.com/~prirala/0/life-expectancy-across-the-world/#/code>

<https://chart-studio.plotly.com/~zeet/8/examining-population-and-life-expectancy-over-time/#/plot>

<https://www.kaggle.com/code/philbowman212/life-expectancy-exploratory-data-analysis/notebook>

<https://www.kaggle.com/code/imdevskp/covid-19-analysis-visualization-comparisons/notebook>