

Aula 14 - Técnicas de visualização de dados: Matplotlib e Seaborn.

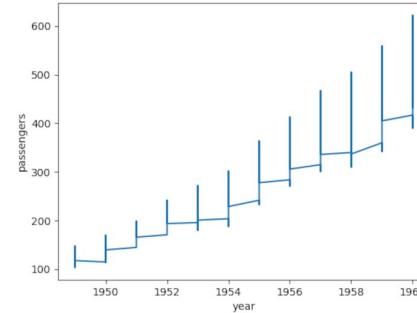
Profa. Gabrielly Queiroz

Matplotlib e Seaborn

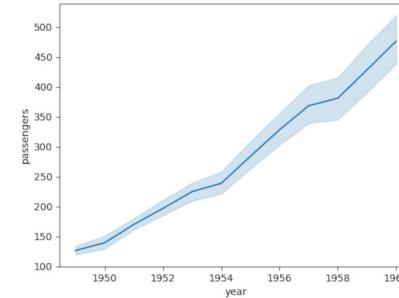
Matplotlib: Biblioteca básica para criar gráficos em Python. É altamente personalizável, mas pode ser um pouco "manual".

Seaborn: Biblioteca mais avançada, construída sobre o Matplotlib. Facilita a criação de gráficos bonitos e com análises estatísticas.

```
● ● ●  
import matplotlib.pyplot as plt  
import pandas as pd  
  
flights = pd.read_csv('flights.csv')  
  
plt.plot(flights['month'], flights['passengers'])  
plt.xlabel('year')  
plt.ylabel('passengers')  
plt.show()
```



```
● ● ●  
import matplotlib.pyplot as plt  
import seaborn as sns  
import pandas as pd  
  
flights = pd.read_csv('flights.csv')  
  
sns.lineplot(x='year', y='passengers', data=flights)  
plt.show()
```



Instalar as bibliotecas

```
pip install matplotlib seaborn
```

Matplotlib

Quando você precisa de controle completo sobre todos os elementos do gráfico.

Exemplo de gráficos disponíveis:

- `plot()`: Gráficos de linha.
- `scatter()`: Gráficos de dispersão.
- `bar()`: Gráficos de barras.
- `hist()`: Histogramas.
- `boxplot()`: Gráficos de caixa (distribuição).

```
import matplotlib.pyplot as plt
```

```
# Criar dados
```

```
x = [1, 2, 3, 4, 5]
```

```
y = [2, 4, 6, 8, 10]
```

```
# Criar gráfico de linha
```

```
plt.plot(x, y) # Plota os dados
```

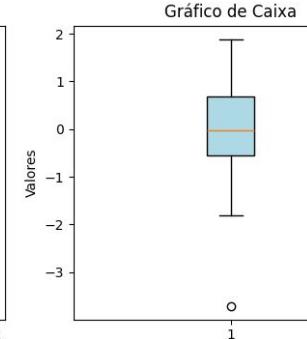
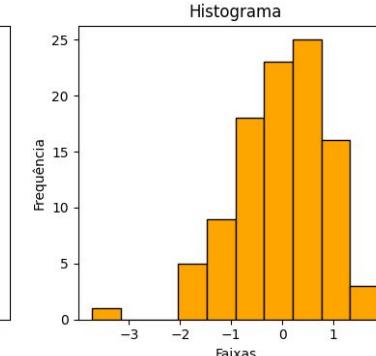
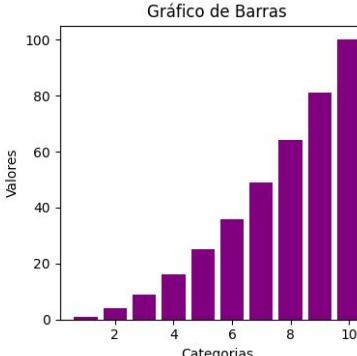
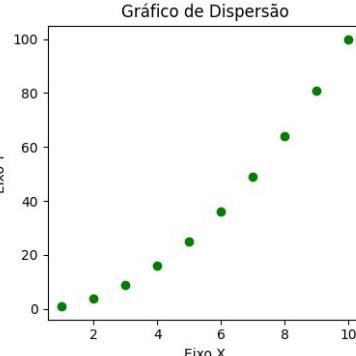
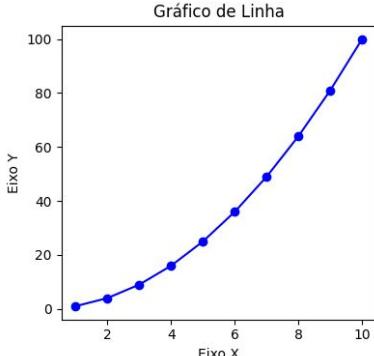
```
plt.title("Meu Gráfico") # Adiciona um título
```

```
plt.xlabel("Eixo X") # Adiciona rótulo ao eixo X
```

```
plt.ylabel("Eixo Y") # Adiciona rótulo ao eixo Y
```

```
plt.grid(True) # Adiciona uma grade
```

```
plt.show() # Exibe o gráfico
```



Quando utilizar cada tipo de gráfico

- `plot()`: Gráficos de linha. Ideal para **visualizar tendências ou variações ao longo do tempo** ou de uma sequência, como crescimento de vendas mensal ou variação de temperatura diária.
- `scatter()`: Gráficos de dispersão. Útil para **mostrar a relação ou correlação entre duas variáveis**. Por exemplo, altura versus peso ou horas de estudo versus desempenho. Perfeito para **comparar categorias ou grupos distintos**. Exemplo: vendas por região, população por cidade ou desempenho de diferentes times.
- `bar()`: Gráficos de barras. Perfeito para **comparar categorias ou grupos distintos**. Exemplo: vendas por região, população por cidade ou desempenho de diferentes times.
- `hist()`: Histogramas. Melhor para **analisar a distribuição de uma variável contínua**. Por exemplo, frequência de idades em uma amostra ou distribuição de notas de alunos.
- `boxplot()`: Gráficos de caixa (distribuição). Indicado para **resumir e comparar a distribuição de dados** com foco em mediana, quartis e valores atípicos (outliers). Exemplo: comparar salários entre diferentes setores.

Seaborn

```
import seaborn as sns  
  
import matplotlib.pyplot as plt
```

Configurando estilo

```
sns.set(style="darkgrid") # Estilos:  
white, dark, whitegrid, darkgrid, ticks
```

scatterplot: Dispersão (relação entre duas variáveis).

lineplot: Linhas (tendências ao longo de uma variável, como tempo).

barplot: Barras (valores agregados, como médias, por categoria).

histplot: Histogramas (distribuição de uma variável contínua).

kdeplot: Densidade (curva para distribuição de probabilidade).

boxplot: Boxplot (distribuição e outliers de uma variável).

violinplot: Violino (distribuição detalhada com densidade e quartis).

heatmap: Mapa de calor (relações em matrizes, como correlações).

pairplot: Múltiplos gráficos (relações entre todas as variáveis numéricas).

Seaborn

Use um countplot quando você quiser **visualizar quantas ocorrências existem de cada categoria** em uma coluna específica. Por exemplo:

```
import seaborn as sns  
  
import matplotlib.pyplot as plt  
  
# Contar quantos alunos são de cada situação (Aprovado ou Em Risco)  
sns.countplot(x='situacao', data=df, palette='Set2')  
  
plt.title("Quantidade de Alunos por Situação")  
  
plt.xlabel("Situação Acadêmica")  
  
plt.ylabel("Número de Alunos")  
  
plt.show()
```

Documentação Matplotlib e Seaborn

Matplotlib: <https://matplotlib.org/stable/index.html>

Seaborn: <https://seaborn.pydata.org/>

https://colab.research.google.com/drive/1BjGYWf26PU8_NSYCu7A-4hZ5gbIsK2-M?usp=sharing

Atividade

Utilizando o conjunto de dados disponível no link

<https://archive.ics.uci.edu/dataset/320/student%2Bperformance>, baixe os dois arquivos

CSV fornecidos: student-mat.csv (dados de Matemática) e student-por.csv (dados de Português). Carregue os dados usando o pandas e selecione colunas relevantes para visualizar a relação entre o tempo de estudo semanal (studytime) e a nota final (G3). Para isso, produza um gráfico de dispersão com Matplotlib para visualizar essa relação, mapa de calor e relação entre as variáveis. Em seguida, use a biblioteca Seaborn para criar um gráfico de boxplot que mostre a distribuição da nota final (G3) de acordo com a frequência de consumo de álcool durante a semana (Dalc).

Interprete visualmente se o consumo de álcool parece afetar o desempenho.