UFABC

Visualização de Dados Utilizando o Python com Matplotlib

NHZ3010-15 Física Computacional, Outubro 2024

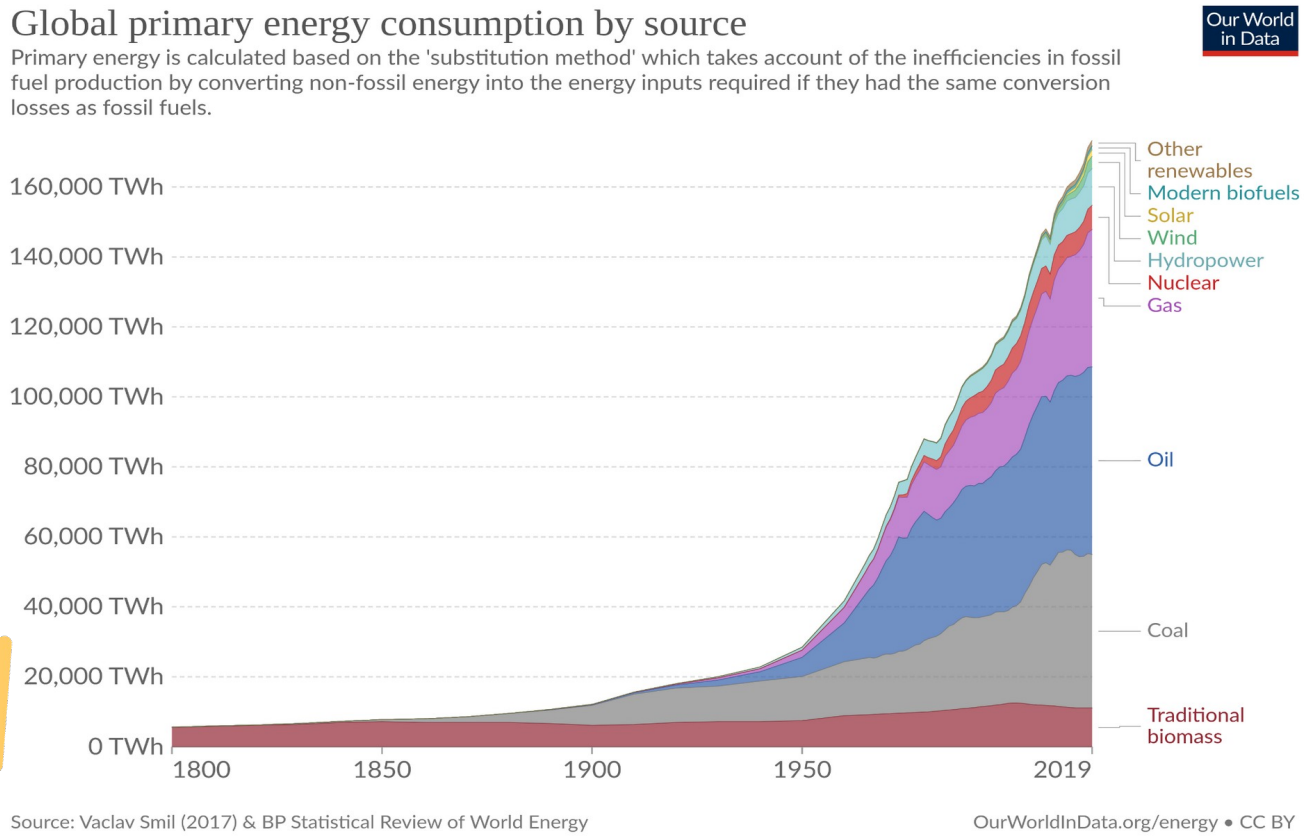
Por que a visualização de dados é importante?

- Explorar a estrutura dos dados, detecção de tendências e anomalias no conjunto, avaliar a saída de modelos matemáticos;
- Revelar características que podem ser difícieis de se identificar por meio de modelos e estatística;
- Apresentar resultados;

Um gráfico: representação de dados numéricos em eixos calibrados (Newman, M.)

Uma imagem vale mais que mil palavras...

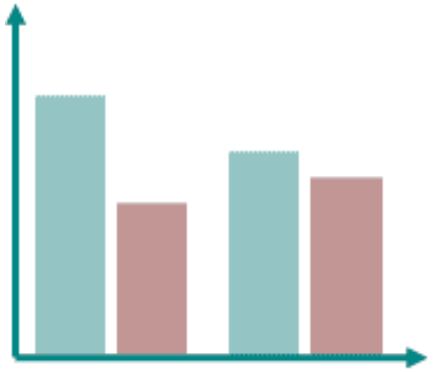
	Area	Variable	Unit	Value
0	Afghanistan	Clean	GW	0.19
1	Afghanistan	Fossil	GW	0.03
2	Afghanistan	Gas and Other Fossil	GW	0.03
3	Afghanistan	Hydro, Bioenergy and Other Renewables	GW	0.19
4	Afghanistan	Renewables	GW	0.19
...	...	...	...	...



Tipos de gráficos:

- Gráfico de dispersão:

- Gráfico de barras:

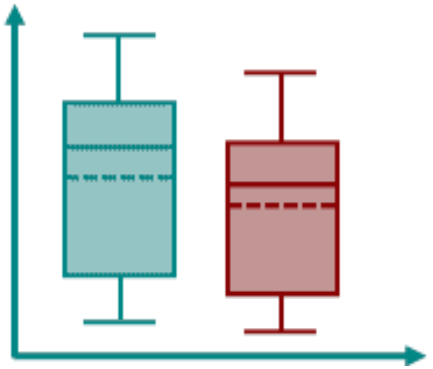


- Gráfico de pizza:

- Gráfico de linhas:

- Histogramas:

- Box plots:



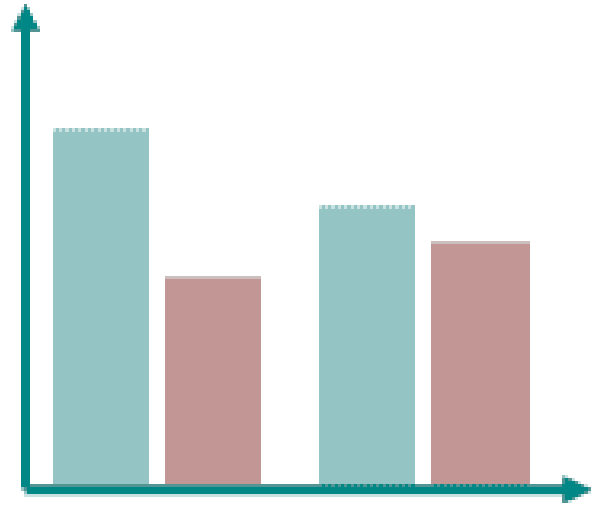
https://datatab.net/tutorial/charts

Qual é a diferença?

Cada tipo de gráfico é utilizado para expressar diferentes características do conjunto de dados:

Correlação

Distribuição de frequência dos dados



Distribuição dos dados

https://datatab.net/tutorial/charts

Bibliotecas para a visualização de dados em Python

https://matplotlib.org/

- ➔ Prós: Biblioteca mais básica; perfeito para gráficos estáticos simples; Alto nível de customização;
- ➔ Cons: Gráficos mais avançados são mais trabalhosos;

https://seaborn.pydata.org/

- ➔ Prós: Contém receitas prontas para gráficos mais avançados; Estilos visuais facilmente customizáveis;
- ➔ Cons: Pouca flexibilidade; Primariamente utilizado em estatística;

https://plotly.com/

- ➔ Prós: Excelente para gráficos interativos; produz gráficos que são facilmente integrados em aplicativos web;
- ➔ Cons: Curva de aprendizado complexa;

Matplotlib

Biblioteca utilizada para criação de gráficos estáticos, animados e iterativos. É o pilar para visualização de dados utilizando python.

➔ Facilmente instalável utilizando gerenciadores de pacotes de python como o pip ou anaconda:

```
pip install matplotlib
```

```
conda create -n <nome-do-ambiente-virtual>
conda activate <nome-do-ambiente-virtual>

conda install -c conda-forge::matplotlib

# Tambem e possivel criar um ambiente que ja possui python e bibliotecas:
conda create -n <nome-do-ambiente-virtual> python=3.x.x numpy matplotlib pandas
```

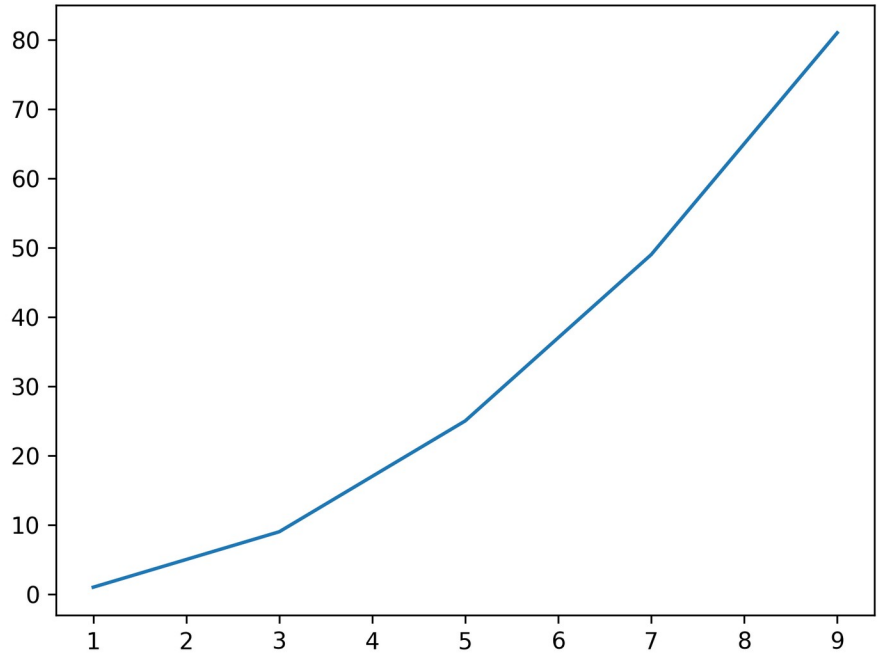
Conceitos básicos do matplotlib

Como fazer um gráfico utilizando o matplotlib?

```
import matplotlib.pyplot as plt

dados_x = [1, 3, 5, 7, 9]
dados_y = [1, 9, 25, 49, 81]

plt.plot(dados_x, dados_y)
plt.show()
```



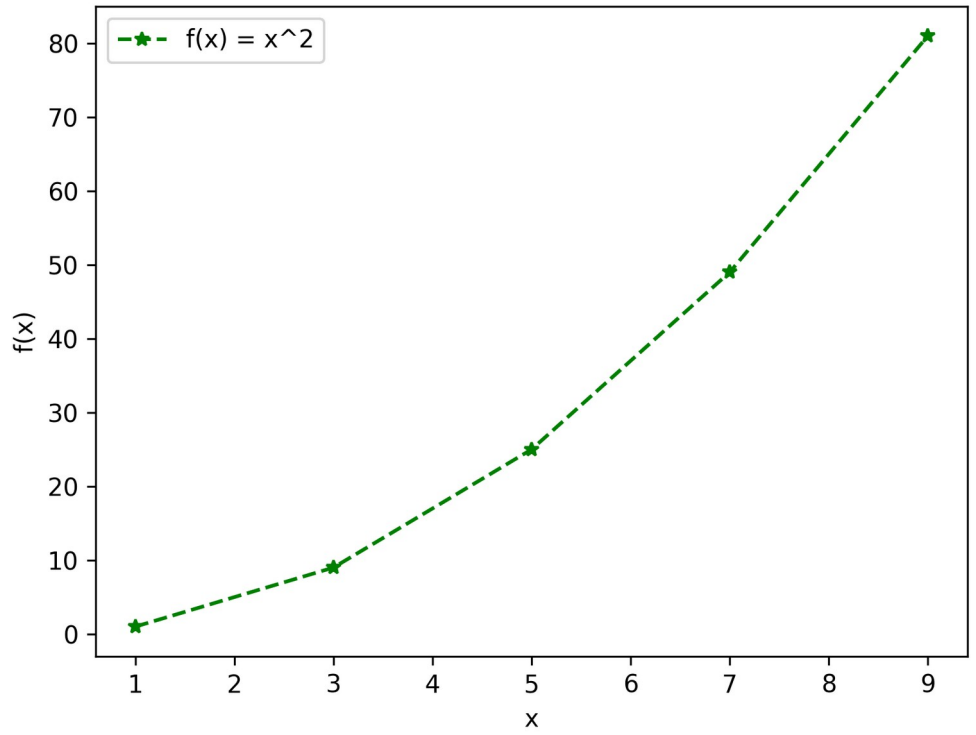
Conceitos básicos do matplotlib

Customizar as cores, legendas, rótulos, etc...

```
plt.plot(dados_x,
         dados_y,
         marker='*',
         color='green',
         linewidth=1.5,
         linestyle='--',
         label="f(x) = x^2")

plt.legend()
plt.xlabel("x")
plt.ylabel("f(x)")

plt.savefig('grafico.png', dpi=300)
plt.show()
```



Tipos de gráficos no matplotlib:

Gráfico de barras:

```
plt.bar(periodo, no_alunos, color='green')
plt.set_xlabel("Ano", fontsize=10)
plt.set_ylabel("Número de alunos", fontsize=10)
```

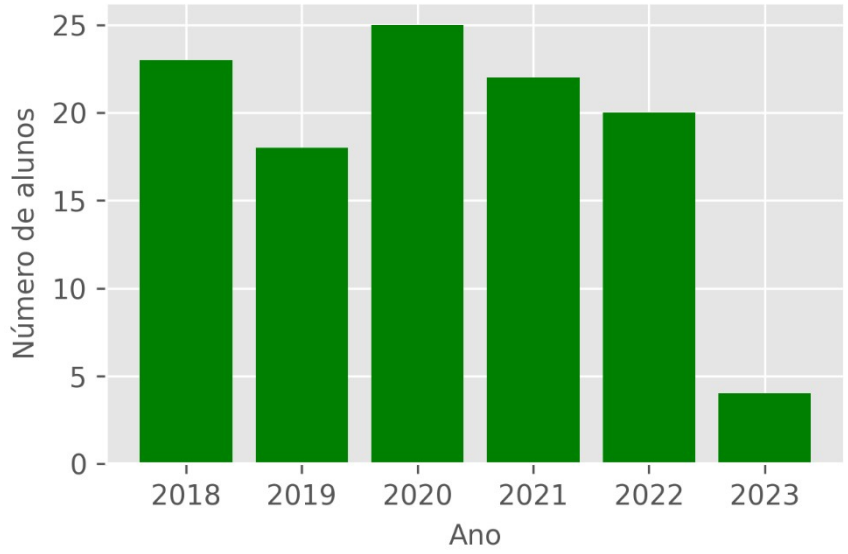
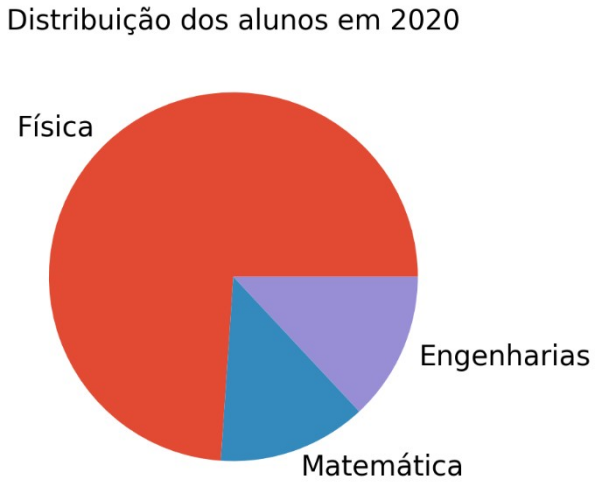


Gráfico de pizza:

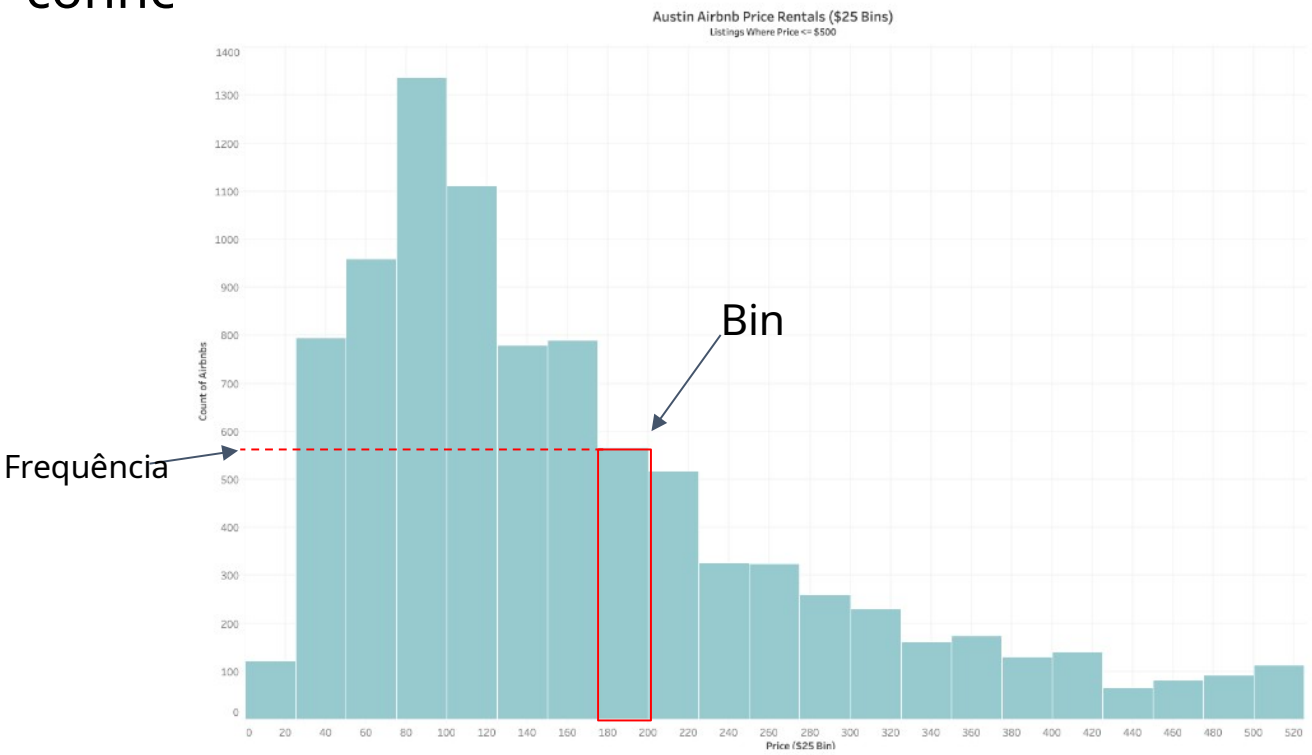
```
plt.set_title("Distribuição dos alunos em 2020", fontsize=10)
plt.pie(percent,
            labels=["Física", "Matemática", "Engenharias"])
```



https://datatab.net/tutorial/charts

Histogramas

São representações da distribuição de frequência de uma variável. Um histograma é construído dividindo os dados em “classes” ou sub-intervalos que são conhe




```
dados = np.loadtxt("student_grades.txt")

plt.hist(dados, 10)
plt.set_xlim([0,10])

plt.xlabel("Notas")
plt.ylabel("Frequencias")
```