



Curso de Python

01-02-2021

<https://github.com/tsbressan/CursoPython>

Bibliotecas para processamento

- **MatPlotLib:** <https://matplotlib.org/>
- Biblioteca completa para organização e montagem de gráficos em 2D e 3D
- Necessita ser importada antes da utilização:
- `import matplotlib as plt`
- Suporte a criação de gráficos, figuras, gráficos de linhas e multilinhas, colunas, áreas, pizza, Scatter, spectrum, ..., muitos outros gráficos conforme o tipo de dados

Bibliotecas para processamento

- **MatPlotLib:** <https://matplotlib.org/>
- Método: pyplot (método principal para criar gráficos)
- `import matplotlib.pyplot as plt`
- ou
- `import matplotlib.pyplot`

(importante destacar que as demais bibliotecas-numpy,scipy,pandas-também são utilizadas em conjunto.)

Bibliotecas para processamento

- **Matplotlib:** Exemplos (gráfico em linha)

```
import matplotlib
```

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

```
import numpy as np
```

```
labels = ['P1', 'P2', 'P3', 'P4', 'P5']
```

```
Valor1 = [20, 34, 30, 35, 27]
```

```
Valor2 = [25, 32, 34, 20, 25]
```

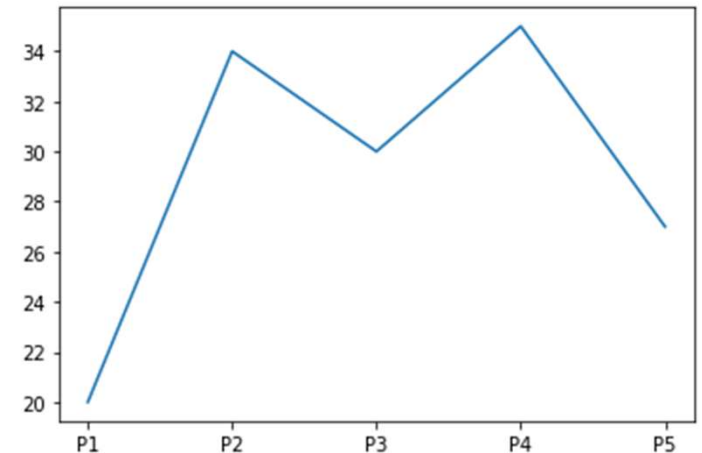
Bibliotecas para processamento

- **Matplotlib:** Exemplos

#plot – desenha um gráfico de linha

matplotlib.pyplot.plot(labels, Valor1) #labels = x, Valor1 = y

matplotlib.pyplot.show() #mostrar na tela



Bibliotecas para processamento

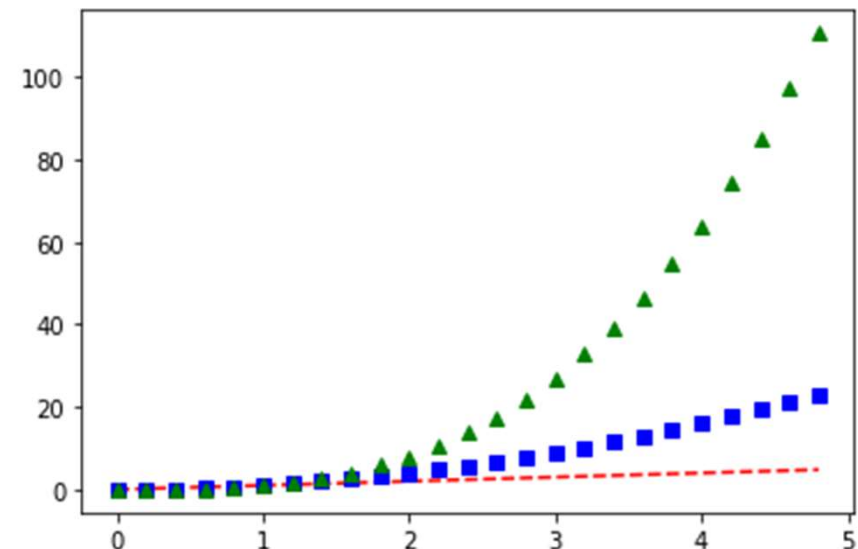
- **Matplotlib:** Exemplos

#plot – configuração

```
t = np.arange(0.0, 5.0, 0.2)
```

```
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
```

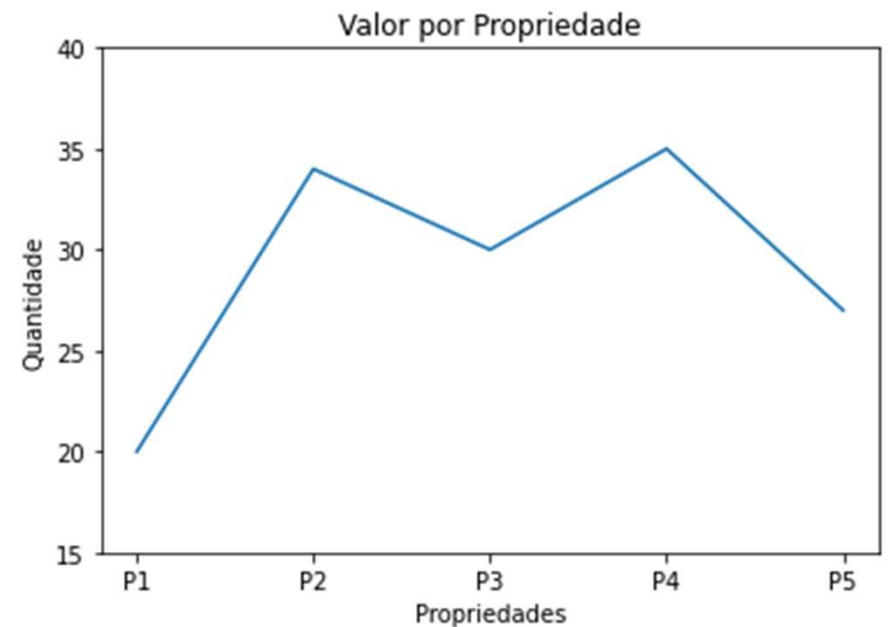
```
plt.show()
```



Bibliotecas para processamento

- **Matplotlib:** Exemplos

```
matplotlib.pyplot.title('Valor por Propriedade') #adicionar título ao gráfico  
matplotlib.pyplot.plot(labels, Valor1) #labels = x, Valor1 = y  
matplotlib.pyplot.ylim(15, 40) #intervalo de valores ao eixo y  
matplotlib.pyplot.xlabel('Propriedades') #título ao eixo x  
matplotlib.pyplot.ylabel('Quantidade') #título ao eixo y  
matplotlib.pyplot.show()
```



Bibliotecas para processamento

- **Matplotlib:** Exemplos - subplots

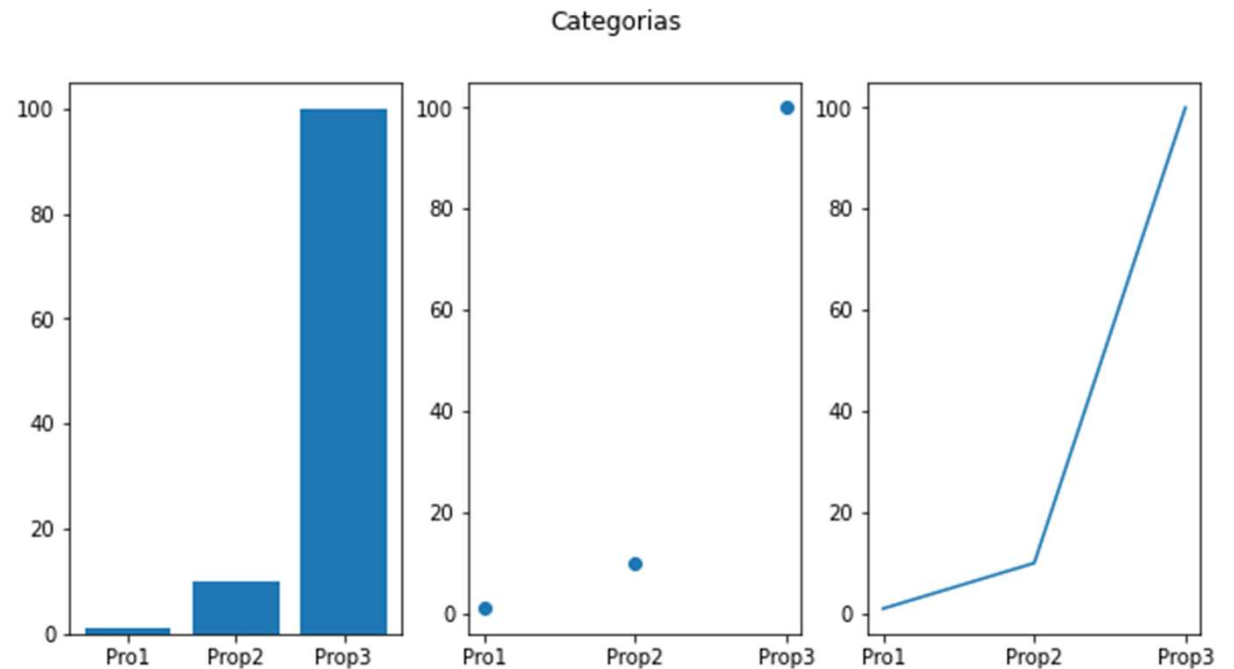
```
names = ['Pro1', 'Prop2', 'Prop3']  
values = [1, 10, 100]
```

```
plt.figure(figsize=(10, 5))
```

```
plt.subplot(131)  
plt.bar(names, values)  
plt.subplot(132)
```

```
plt.scatter(names, values)  
plt.subplot(133)
```

```
plt.plot(names, values)  
plt.suptitle('Categorias')  
plt.show()
```



https://matplotlib.org/3.3.3/api/_as_gen/matplotlib.pyplot.subplot.html

Bibliotecas para processamento

- **Matplotlib:** Exemplos – subplots – outra forma 1(subplot com axes)

```
names = ['Pro1', 'Prop2', 'Prop3']
```

```
values = [1, 10, 100]
```

```
plt.figure(figsize=(10, 5))
```

```
ax1=plt.subplot(1, 3, 2)
```

```
plt.bar(names, values)
```

```
ax2=plt.subplot(1, 3, 1)
```

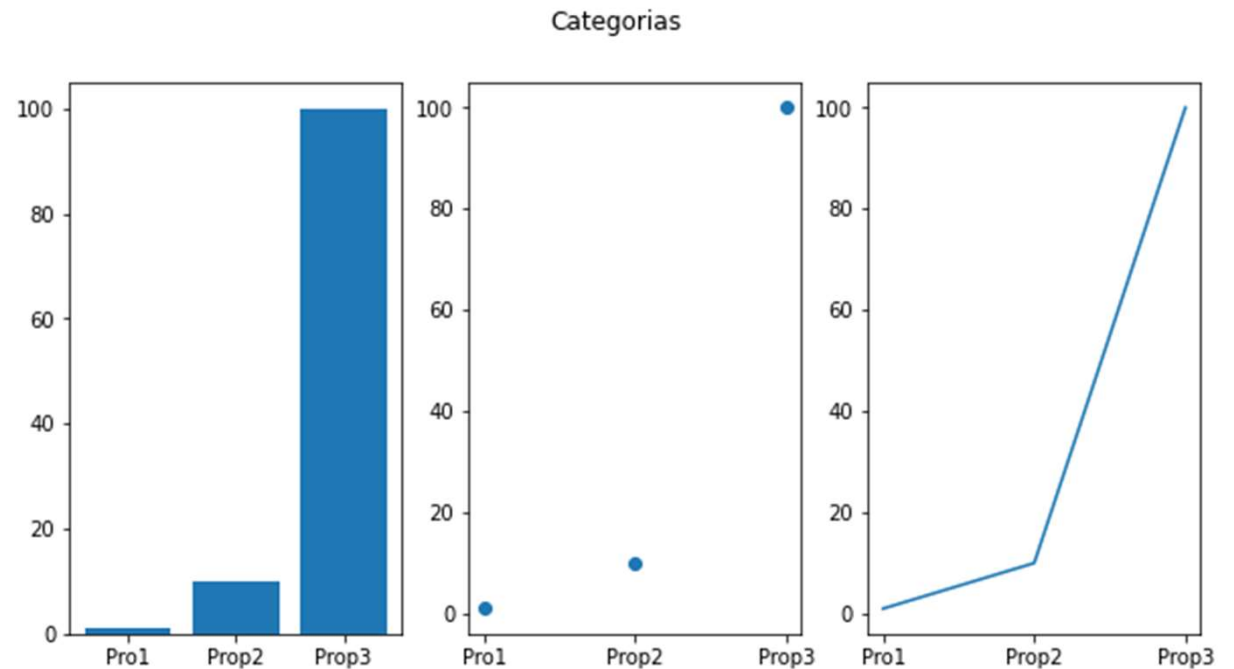
```
plt.scatter(names, values)
```

```
ax2=plt.subplot(1, 3, 3)
```

```
plt.plot(names, values)
```

```
plt.suptitle('Categorias')
```

```
plt.show()
```



https://matplotlib.org/3.3.3/api/_as_gen/matplotlib.pyplot.subplot.html

Bibliotecas para processamento

- **Matplotlib:** Exemplos – subplots – com 2 plt.figure

```
names = ['Pro1', 'Prop2', 'Prop3']
```

```
values = [1, 10, 100]
```

```
#-----
```

```
plt.figure(1, figsize=(10, 5))
```

```
ax1=plt.subplot(1, 3, 2)
```

```
plt.bar(names, values)
```

```
ax2=plt.subplot(1, 3, 1)
```

```
plt.scatter(names, values)
```

```
ax2=plt.subplot(1, 3, 3)
```

```
plt.plot(names, values)
```

```
plt.suptitle('Categorias')
```

```
plt.show()
```

```
#-----
```

```
plt.figure(2, figsize=(10, 5))
```

```
ax1=plt.subplot(1, 3, 2)
```

```
plt.bar(names, values)
```

```
ax2=plt.subplot(1, 3, 1)
```

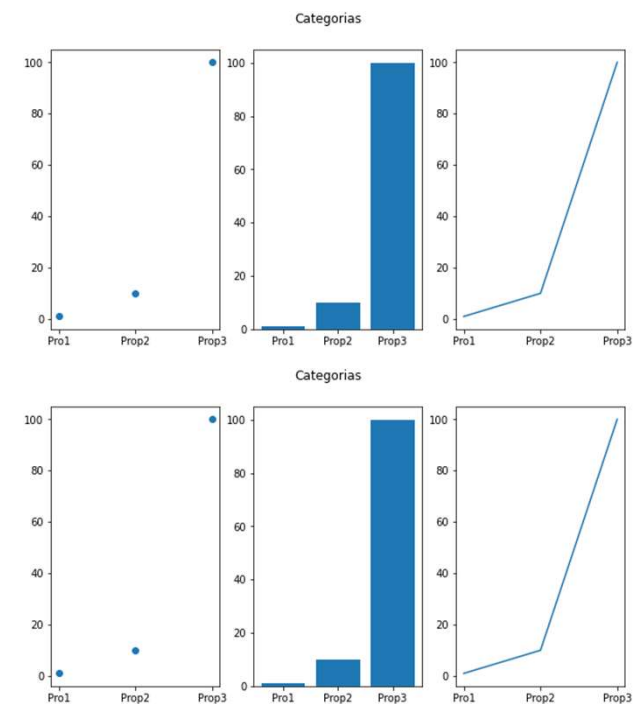
```
plt.scatter(names, values)
```

```
ax2=plt.subplot(1, 3, 3)
```

```
plt.plot(names, values)
```

```
plt.suptitle('Categorias')
```

```
plt.show()
```



https://matplotlib.org/3.3.0/api/_as_gen/matplotlib.pyplot.figure.html#matplotlib.pyplot.figure

Bibliotecas para processamento

- **Matplotlib:** Exemplos (gráfico em barra)

```
x = np.arange(len(labels))
```

```
width = 0.35
```

```
fig, ax = plt.subplots()
```

```
ax.bar(x - width/2, Valor1, width, label='Valor1')
```

```
ax.bar(x + width/2, Valor2, width, label='Valor2')
```

```
ax.set_ylabel('Quantidade')
```

```
ax.set_xlabel('Propriedade')
```

```
ax.set_title('Valor por Propriedade')
```

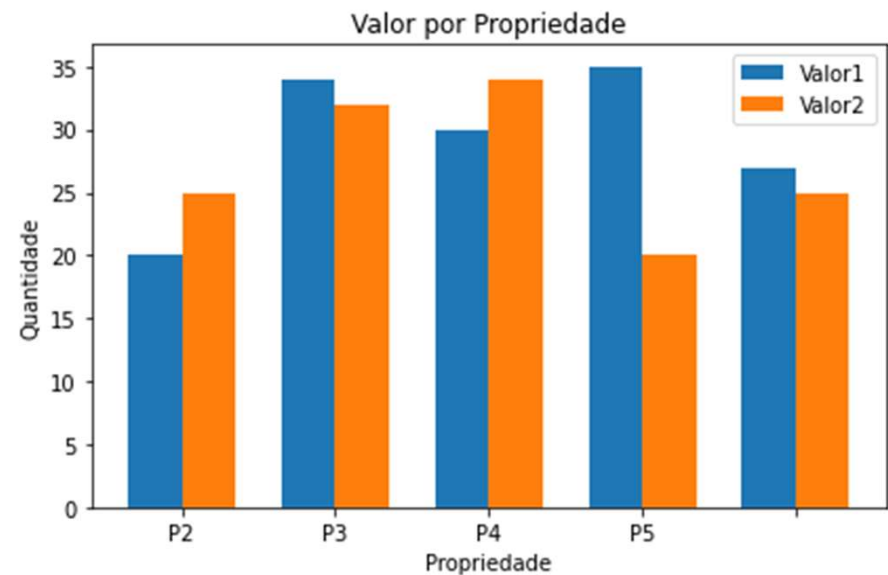
```
ax.set_xticks(x)
```

```
ax.set_xticklabels(labels)
```

```
ax.legend()
```

```
fig.tight_layout()
```

```
plt.show()
```



Bibliotecas para processamento

- **Matplotlib:** Exemplos (gráfico em barra-axes)

```
x = np.arange(len(labels))
```

```
width = 0.35
```

```
fig, ax = plt.subplots(2) #vertical ou horizontal
```

```
ax[0].bar(x - width/2, Valor1, width, label='Valor1')
```

```
ax[0].bar(x + width/2, Valor2, width, label='Valor2')
```

```
ax[0].set_ylabel('Quantidade')
```

```
ax[0].set_xlabel('Propriedade')
```

```
ax[0].set_title('Valor por Propriedade')
```

```
ax[0].set_xticks(x)
```

```
ax[0].set_xticklabels(labels)
```

```
ax[0].legend()
```

```
ax[1].bar(x - width/2, Valor11, width, label='Valor1')
```

```
ax[1].bar(x + width/2, Valor22, width, label='Valor2')
```

```
ax[1].set_ylabel('Quantidade')
```

```
ax[1].set_xlabel('Propriedade')
```

```
ax[1].set_title('Valor por Propriedade')
```

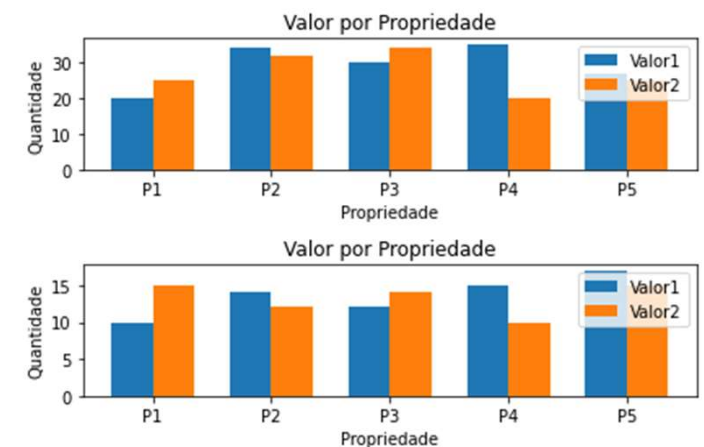
```
ax[1].set_xticks(x)
```

```
ax[1].set_xticklabels(labels)
```

```
ax[1].legend()
```

```
fig.tight_layout()
```

```
plt.show()
```



Bibliotecas para processamento

- **Matplotlib:** Exemplos
- Exemplos com dados do DataFrame

```
file = "csv_merge_properties.csv"
```

```
#ler arquivos .csv com delimitador informado, neste caso a ,
```

```
df_importcsv_merge = pd.read_csv(file,sep=",")
```

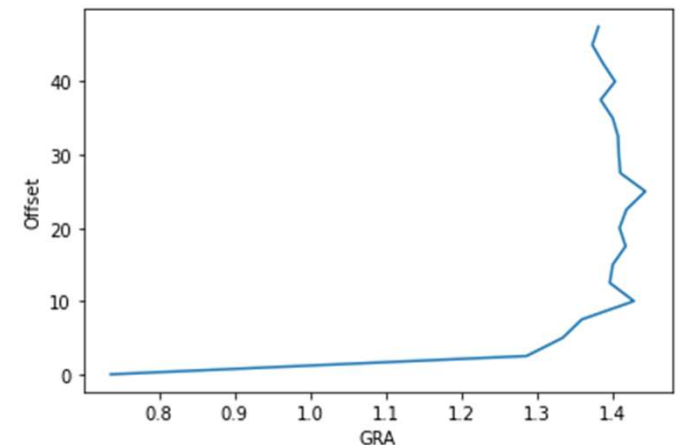
```
df_importcsv_merge1 = df_importcsv_merge[0:20]
```

```
matplotlib.pyplot.plot(df_importcsv_merge1["Bulkdensity(GRA)"], df_importcsv_merge1["Offset(cm)"])
```

```
matplotlib.pyplot.xlabel('GRA')
```

```
matplotlib.pyplot.ylabel('Offset')
```

```
matplotlib.pyplot.show()
```



Bibliotecas para processamento

- **Matplotlib:** Exemplos
- Exemplos com dados do DataFrame

```
df_importcsv_merge1 =  
df_importcsv_merge1.sort_values(by='Offset(cm)', ascending=False)  
fig, ax = plt.subplots(1,2,figsize=(6, 10))
```

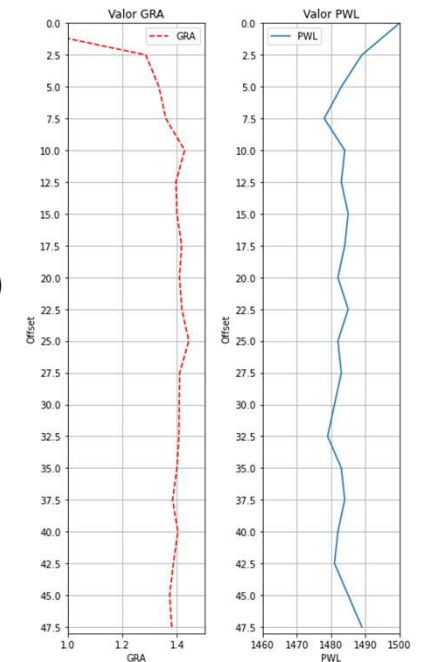
```
ax[0].plot(df_importcsv_merge1["Bulkdensity(GRA)"],  
df_importcsv_merge1["Offset(cm)"], 'r--', label="GRA")  
ax[0].set_ylabel('Offset')  
ax[0].set_xlabel('GRA')  
ax[0].set_title('Valor GRA')  
ax[0].set_xlim(1.0, 1.5)  
ax[0].set_ylim([48, 0])  
ax[0].locator_params(axis='y', nbins=20)
```

```
ax[0].grid()  
ax[0].legend()
```

```
ax[1].plot(df_importcsv_merge1["P-wavevelocity"],  
df_importcsv_merge1["Offset(cm)"], label="PWL")
```

```
ax[1].set_ylabel('Offset')  
ax[1].set_xlabel('PWL')  
ax[1].set_title('Valor PWL')  
ax[1].set_xlim(1460, 1500)  
ax[1].set_ylim([48, 0])  
ax[1].locator_params(axis='y', nbins=20)  
ax[1].grid()  
ax[1].legend()
```

```
fig.tight_layout()  
plt.show()
```



Bibliotecas para processamento

- **Matplotlib:** Exemplos
- Exemplos prático em código python:
- U1480E – 1H– Depth: 0.0 – 50.0
- Mostrar valor original e valor interpolado (a cada 1 cm).