

# INTEGRA ASTRO

1ª semana de workshop da ASTRONOMIA

Como deixar seus gráficos em Python  
visualmente melhores

Por Lais Borbolato

# Como deixar uma figura visualmente melhor?

---

01

Borda, eixos, título,  
grade

02

Tamanho e formato  
dos pontos

03

Espessura e estilo  
das linhas

04

Legendas

05

Cores

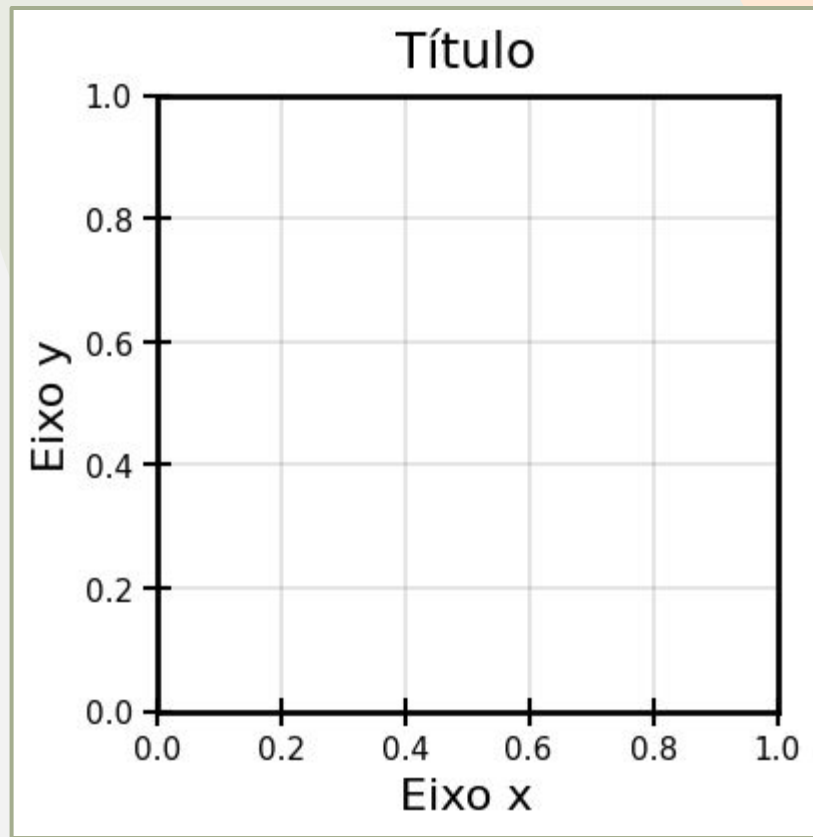
06

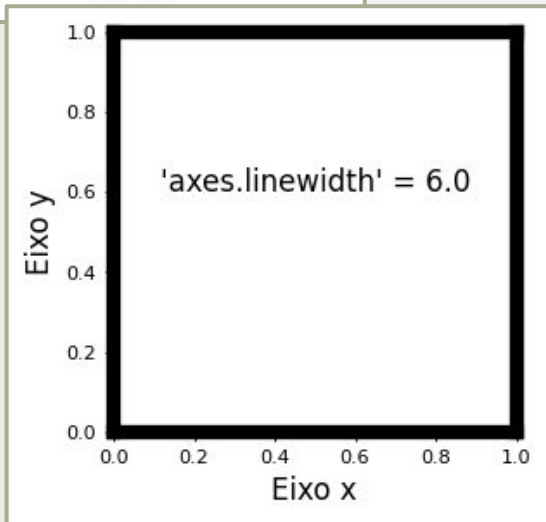
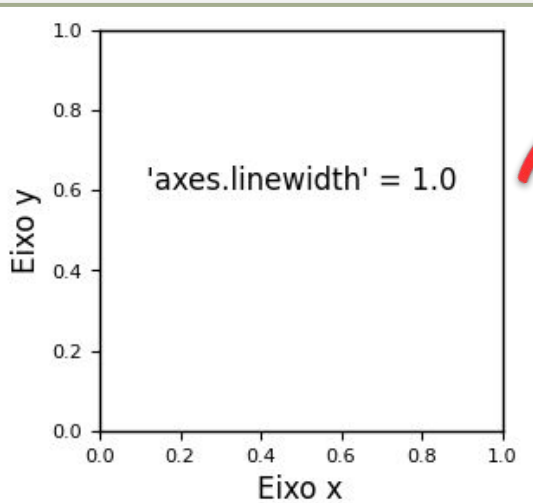
Outros

01

# Borda, eixos, título e grade

Partes básicas do gráfico





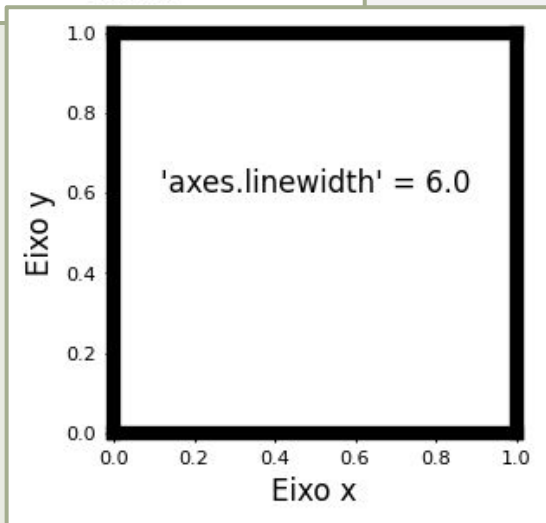
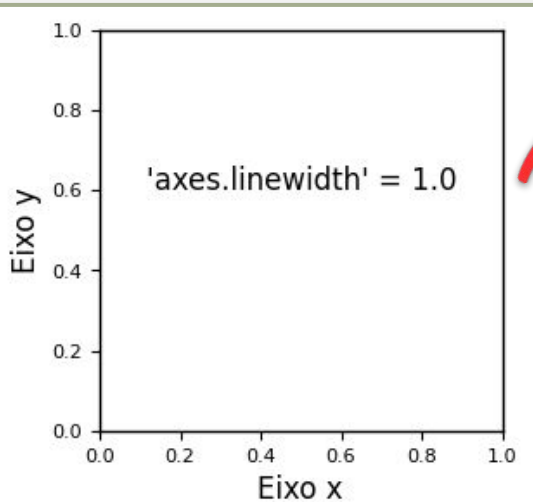
É possível mudar a espessura da borda

- Isso é uma escolha pessoal e fica a critério de cada pessoa.

```
#Biblioteca necessária:  
import matplotlib as mpl
```

```
#Definindo a espessura da borda  
mpl.rcParams['axes.linewidth'] = 1.0
```

Valor padrão



É possível mudar a espessura da borda

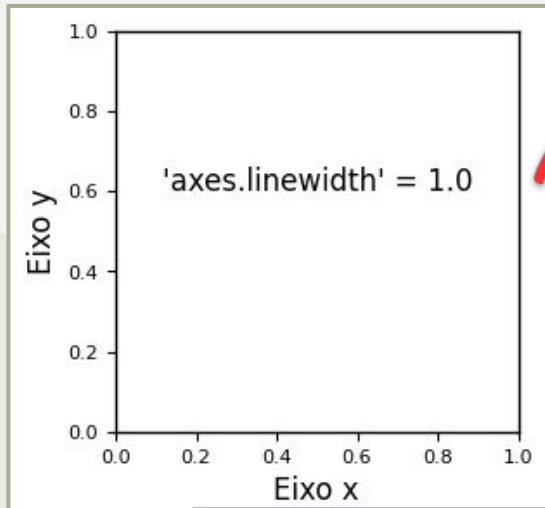
- Isso é uma escolha pessoal e fica a critério de cada pessoa.

```
#Biblioteca necessária:  
import matplotlib as mpl  
  
#Definindo a espessura da borda  
mpl.rcParams['axes.linewidth'] = 1.0
```

Valor padrão

Dica!

Se você definir isso no começo do código, todos os seus gráficos irão ficar com a borda padronizada



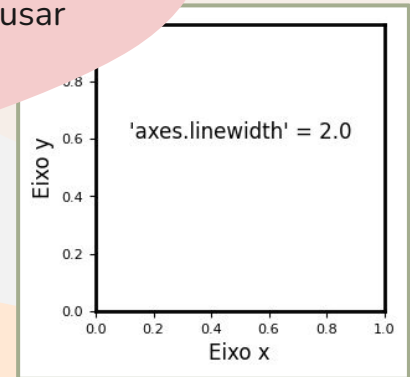
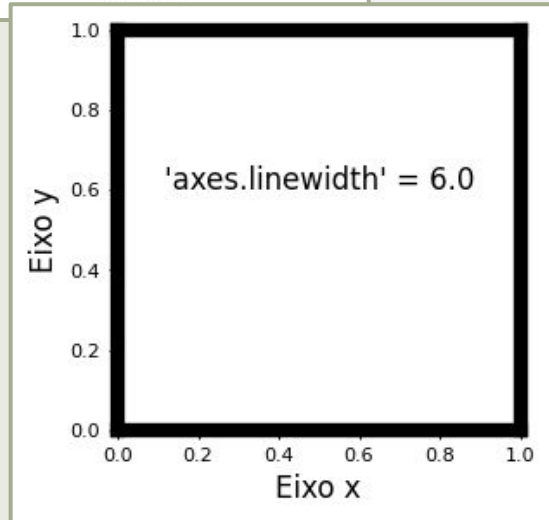
É possível mudar a espessura da borda

- Isso é uma escolha pessoal e fica a critério de cada pessoa.

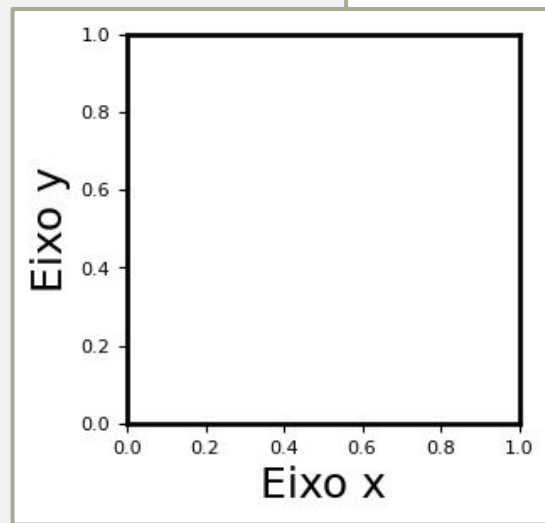
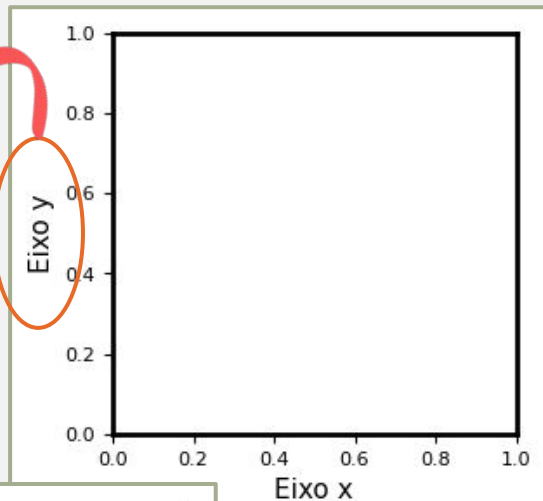
Dica!

Para trabalhos acadêmicos, prefira bordas mais finas, sem chamar muita atenção.

Eu, por exemplo, gosto de usar `axes.linewidth = 2.0`



Ajuste o  
tamanho da  
fonte do nome  
dos eixos



Há várias formas de fazer isso:

→ Pode usar o **rcParams**.

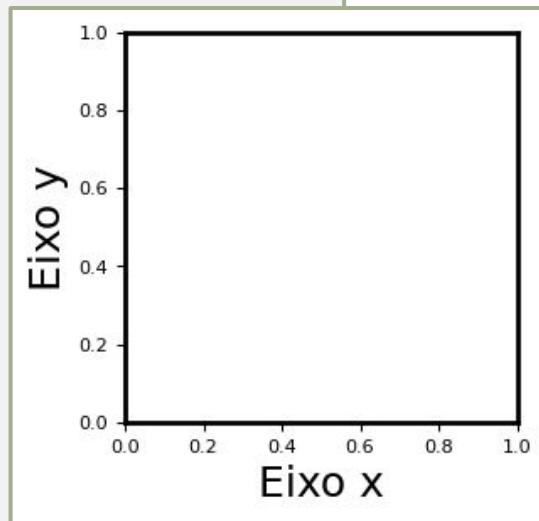
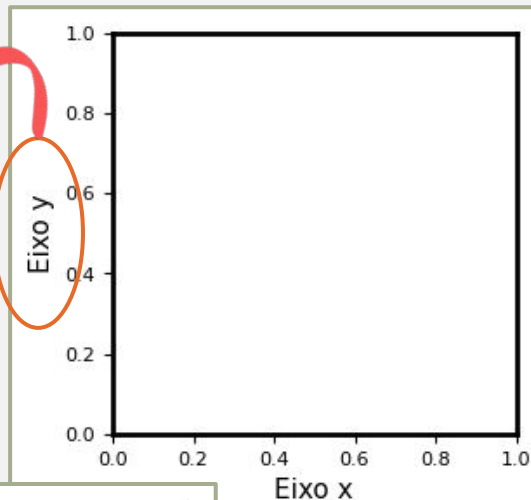
Aqui serve a mesma dica da borda: Se definir no começo do código, todos os gráficos saem padronizados

```
mpl.rcParams['axes.labelsize'] = 16
```

→ Pode usar o **fontsize** na hora de definir o nome dos eixos

```
plt.xlabel('Eixo x', fontsize = 12)  
plt.ylabel('Eixo y', fontsize = 12)
```

Ajuste o  
tamanho da  
fonte do nome  
dos eixos





Há várias formas de fazer isso:

→ Pode usar o **rcParams**.

Aqui serve a mesma dica da borda: Se definir no começo do código, todos os gráficos saem padronizados

```
mpl.rcParams['axes.labelsize'] = 16
```

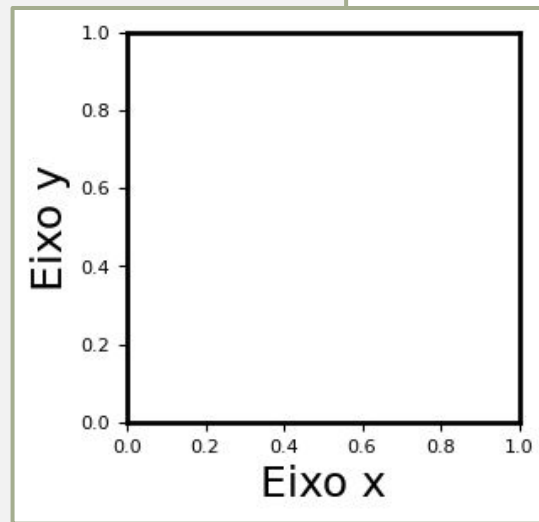
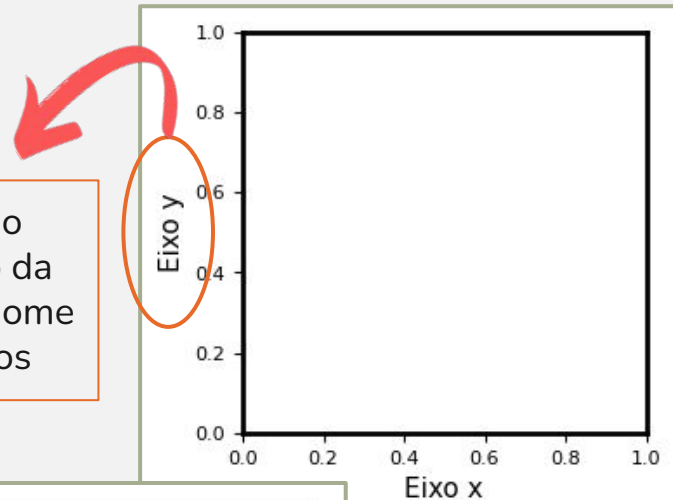
→ Pode usar o **fontsize** na hora de definir o nome dos eixos

```
plt.xlabel('Eixo x', fontsize = 12)  
plt.ylabel('Eixo y', fontsize = 12)
```

Importante!

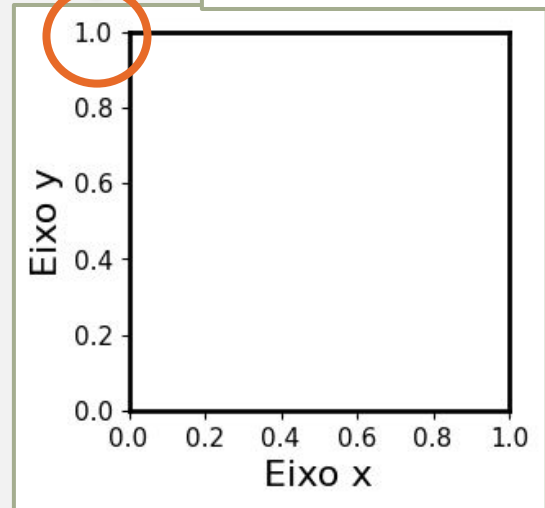
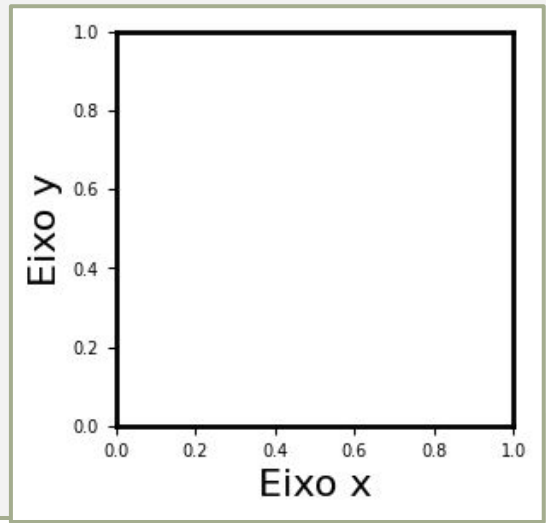
Lembre-se que os eixos devem ser fáceis de ler, porque são informações importantes do gráfico

Ajuste o tamanho da fonte do nome dos eixos



**NUNCA  
APRESENTE  
GRÁFICOS SEM  
O NOME NOS  
EIXOS  
!!!!!!!!!!!!!!**

Para tornar a  
visualização dos  
valores dos eixos  
melhor, você  
pode alterar o  
tamanho



→ Pode usar o **rcParams**.

Aqui também serve a mesma dica: Se definir no começo do código, todos os gráficos saem padronizados

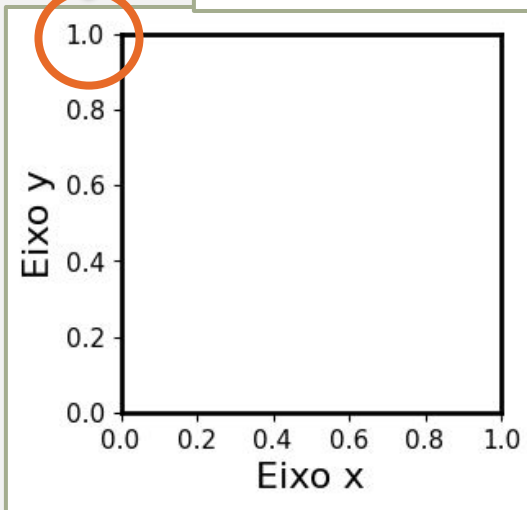
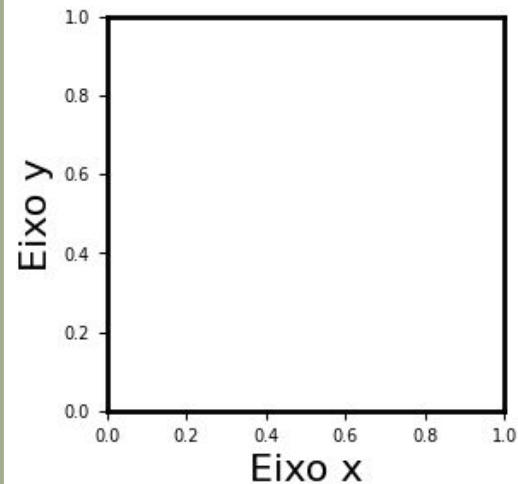
```
mpl.rcParams['xtick.labelsize'] = 12  
mpl.rcParams['ytick.labelsize'] = 12
```

\*Precisa colocar para os dois eixos

→ Pode usar o **plt.tick\_params**

```
plt.tick_params(labelsize = 12)
```

Para tornar a visualização dos valores dos eixos melhor, você pode alterar o tamanho



→ Pode usar o **rcParams**.

Aqui também serve a mesma dica: Se definir no começo do código, todos os gráficos saem padronizados

```
mpl.rcParams['xtick.labelsize'] = 12  
mpl.rcParams['ytick.labelsize'] = 12
```

\*Precisa colocar para os dois eixos

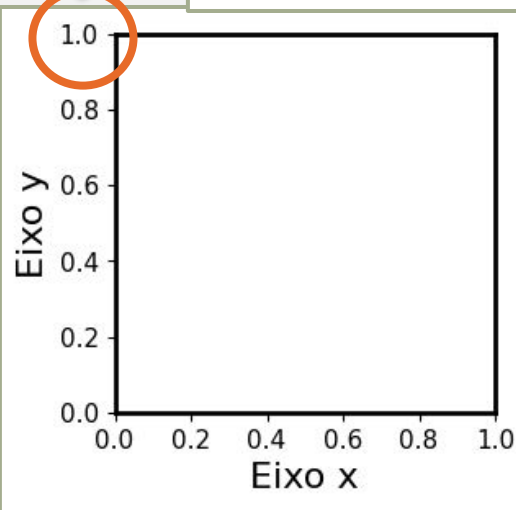
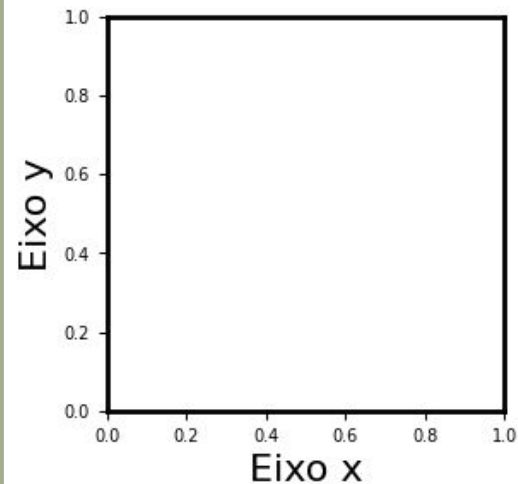
→ Pode usar o **plt.tick\_params**

```
plt.tick_params(labelsize = 12)
```

Lembrando que você pode definir os valores máximo e mínimo dos eixos usando:

```
plt.xlim(0, 1)  
plt.ylim(0, 1)
```

Para tornar a visualização dos valores dos eixos melhor, você pode alterar o tamanho



[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.xlim.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xlim.html)

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.tick\\_params.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.tick_params.html)

→ Pode usar o **rcParams**.

Aqui também serve a mesma dica: Se definir no começo do código, todos os gráficos saem padronizados

```
mpl.rcParams['xtick.labelsize'] = 12  
mpl.rcParams['ytick.labelsize'] = 12
```

\*Precisa colocar para os dois eixos

→ Pode usar o **plt.tick\_params**

```
plt.tick_params(labelsize = 12)
```

Lembrando que você pode definir os valores máximo e mínimo dos eixos usando:

```
plt.xlim(0, 1)  
plt.ylim(0, 1)
```

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.xlim.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xlim.html)

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.tick\\_params.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.tick_params.html)

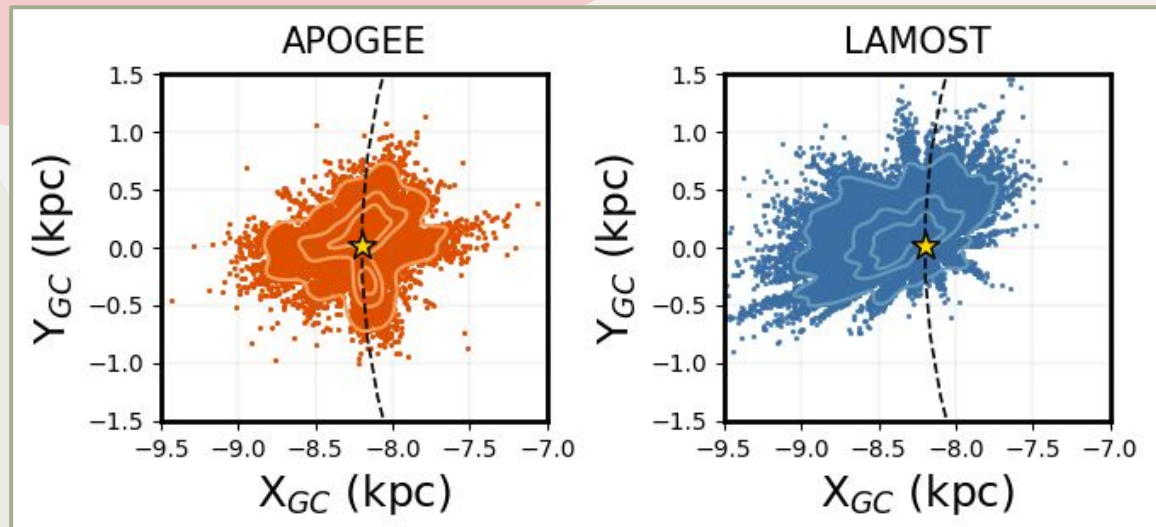
## Observação

As funções de **mpl.rcParams** podem ser definidas apenas uma vez no início do código. As demais funções que começam como **plt.**, como `plt.tick_params`, devem ser definidas para cada figura

Dica!

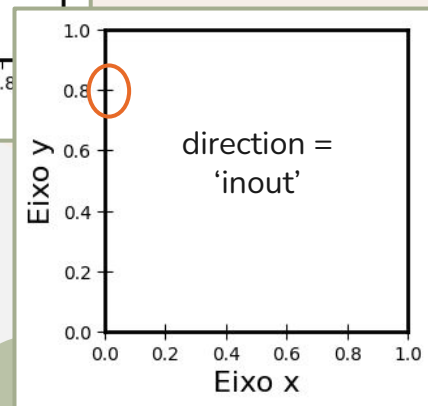
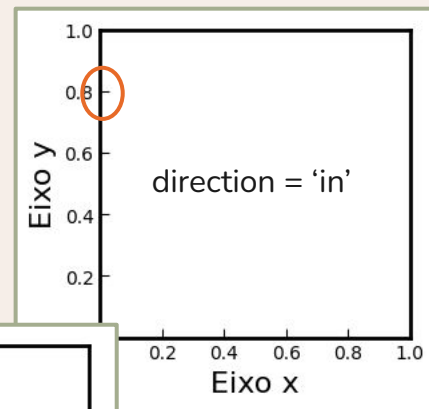
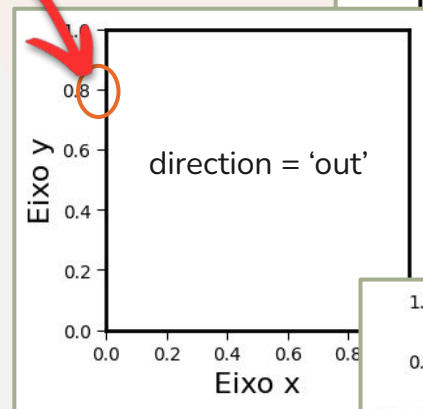
Se você estiver mostrando uma comparação entre dois gráficos de eixos iguais, mantenha os valores máximo e mínimo também iguais. Isso facilita a visualização.

Exemplo:



Ainda falando da função **`plt.tick_params`**, é possível ajustar as “marcações” dos eixos

```
plt.tick_params(labelsize = 10, direction = 'inout')
```



Ainda falando da função **`plt.tick_params`**, é possível ajustar as “marcações” dos eixos

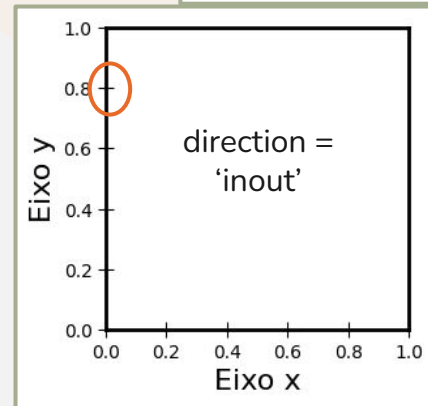
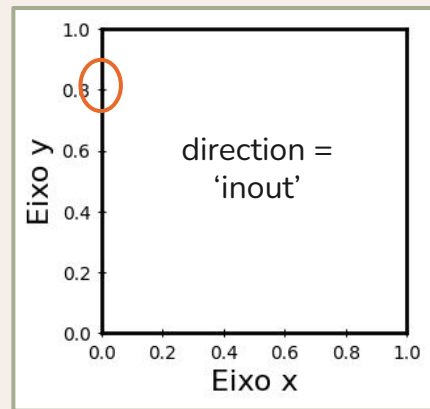
```
plt.tick_params(labelsize = 10, direction = 'inout')
```

Se adicionar *length*, é possível mudar o comprimento das marcações

```
plt.tick_params(labelsize = 10, direction = 'inout', length = 10)
```

Tem outros parâmetros que vocês podem alterar. Olhem em:

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.tick\\_params.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.tick_params.html)



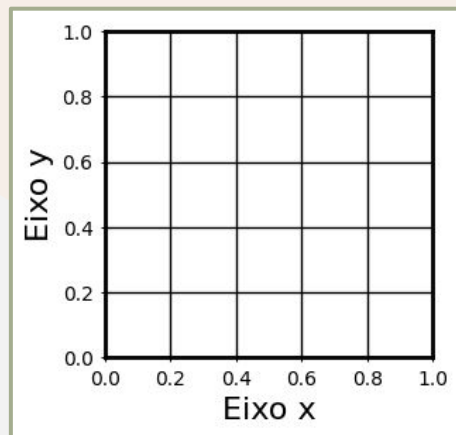


Vamos falar sobre a **grade de fundo...**

A grade facilita a leitura do gráfico, por isso pode ser uma ferramenta legal para ser inclusa (eu, por exemplo, coloco em todos os meus gráficos)

O comando para adicionar a grade é

```
plt.grid()
```



Vamos falar sobre a **grade de fundo**

A grade facilita a leitura do gráfico, por isso pode ser uma ferramenta legal para ser inclusa (eu, por exemplo, coloco em todos os meus gráficos)

O comando para adicionar a grade é

```
plt.grid()
```

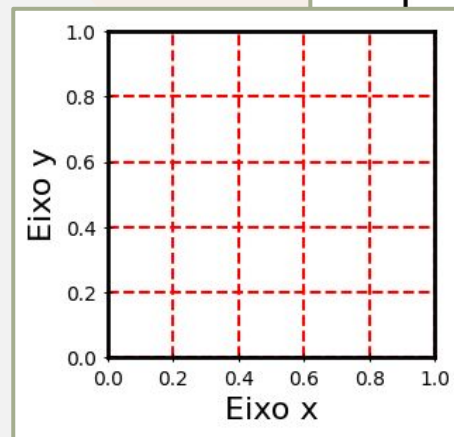
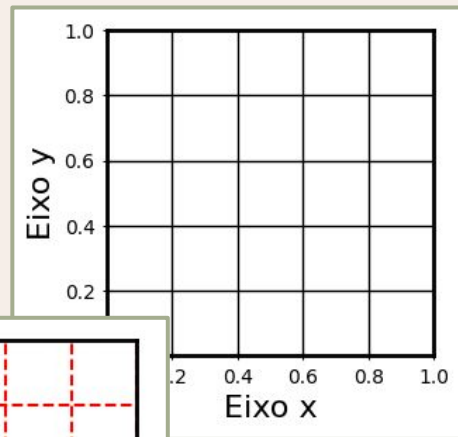
É possível mudar a cor da linha, a espessura e o estilo

```
plt.grid(color='red', linestyle='--', linewidth=1.5)
```

Muda a cor

Muda o  
estilo

Muda a  
espessura

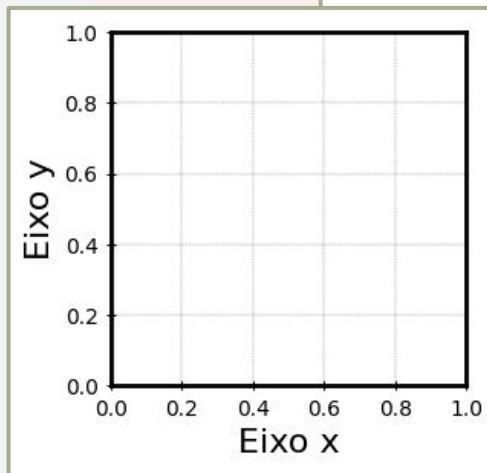
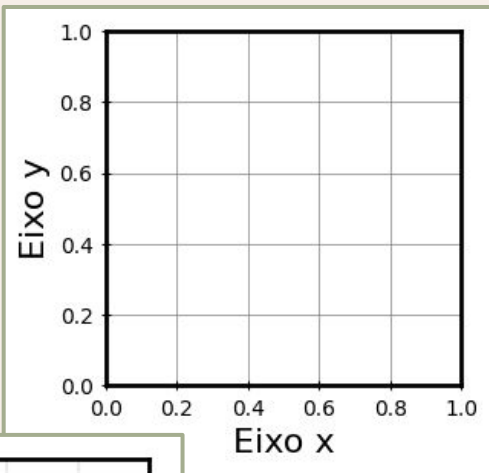


Dica!

Prefira uma grade mais discreta para não chamar muita atenção.

Ela é uma ferramenta para ajudar a ler as informações, não para chamar atenção.

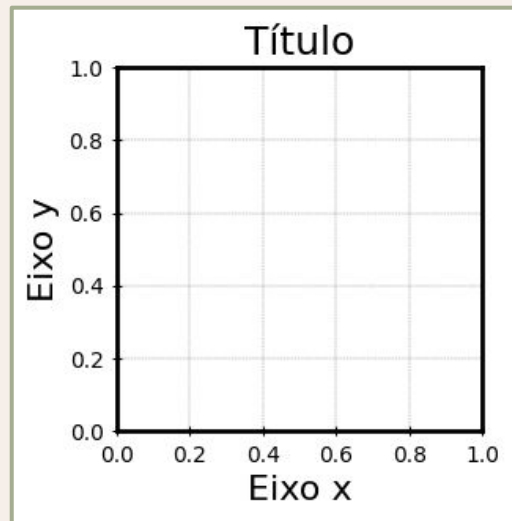
Prefira, por exemplo:



Por fim, podemos colocar título nos gráficos

```
plt.title("Título", fontsize = 18)
```

Define o tamanho da fonte



Por fim, podemos colocar título nos gráficos

```
plt.title("Título", fontsize = 18)
```

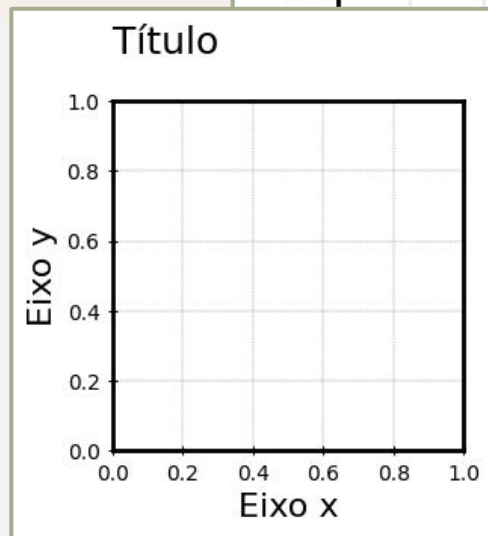
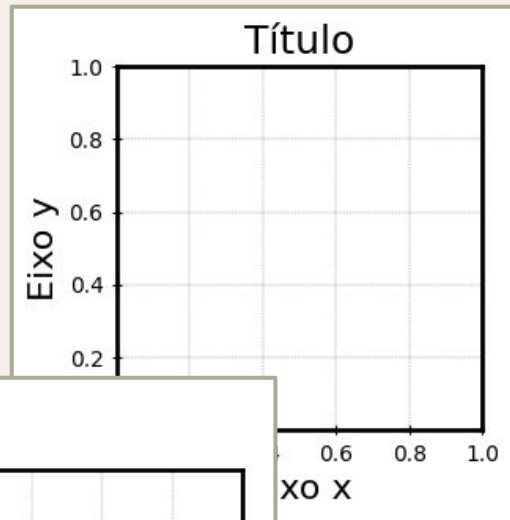
Define o tamanho da fonte

É possível alterar a posição do título com os parâmetros *y* e *loc*

```
plt.title("Título", fontsize = 18, y = 1.5, loc = 'left')
```

Altera a altura do título em relação à borda superior do gráfico. Se nada for especificado, é assumido que *y* = 1.0

Muda horizontalmente a posição do título. Pode ser 'left', 'center' ou 'right'. Se nada for especificado, é sempre assumido *loc* = 'center'



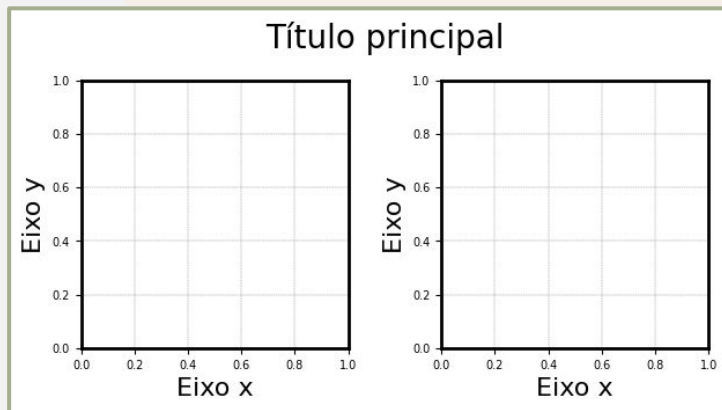
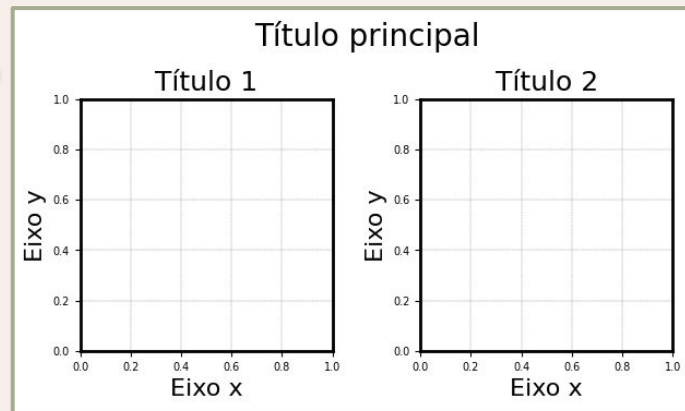
Se você desejar colocar um título entre painéis, pode utilizar a função **`plt.suptitle()`**

```
fig = plt.figure(figsize=(7,3))

ax1 = fig.add_subplot(1,2,1)
ax1.set_xlim(0, 1)
ax1.set_ylim(0, 1)
ax1.set_xlabel('Eixo x', fontsize = 16)
ax1.set_ylabel('Eixo y', fontsize = 16)
ax1.set_title("Título 1",fontsize = 18)
ax1.grid(color='gray', linestyle='--', linewidth=0.3)

ax2 = fig.add_subplot(1,2,2)
ax2.set_xlim(0, 1)
ax2.set_ylim(0, 1)
ax2.set_xlabel('Eixo x', fontsize = 16)
ax2.set_ylabel('Eixo y', fontsize = 16)
ax2.set_title("Título 2",fontsize = 18)
ax2.grid(color='gray', linestyle='--', linewidth=0.3)

plt.suptitle("Título principal", fontsize = 20, y = 1.05)
plt.subplots_adjust(hspace=0.3, wspace = 0.35)
```



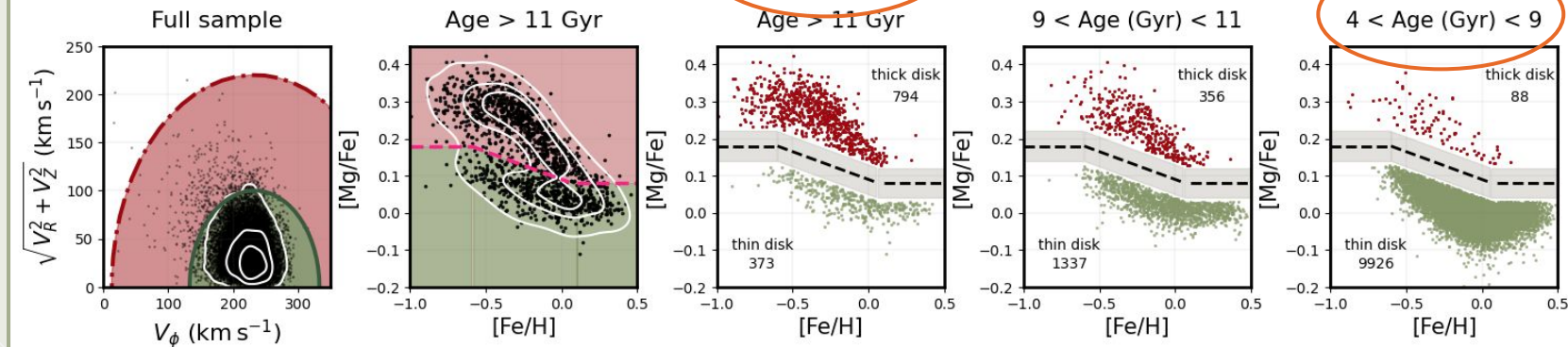
Dica!

Use os títulos a seu favor  
para ajudar a identificar  
amostras, ou trazer uma  
informação adicional

Levantamento  
espectroscópico que  
forneceu os dados de  
abundâncias químicas  
(porque havia mais de  
uma base de dados no  
trabalho)

Intervalo de idades  
das estrelas  
mostradas neste  
painel

APOGEE



# Como deixar o gráfico visualmente melhor?



Borda, eixos, título,  
grade



Tamanho e formato  
dos pontos



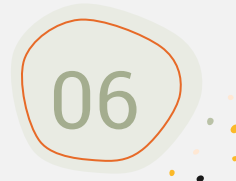
Espessura e estilo  
das linhas



Legendas



Cores



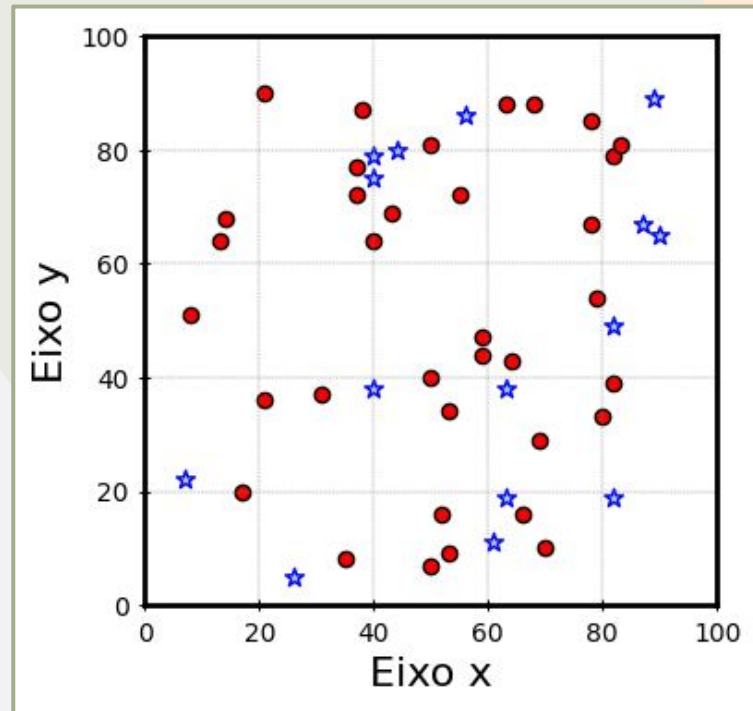
Outros



02

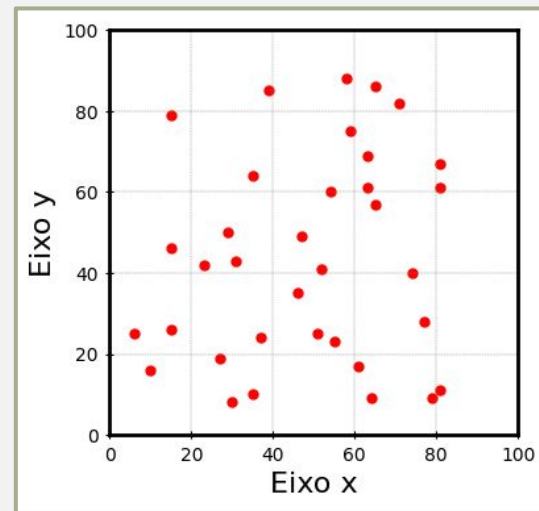
## Tamanho e formato dos pontos

Visualização de gráficos de pontos



Já vimos que ***plt.scatter()*** é uma forma de colocar os pontos no gráfico

```
plt.scatter(x, y, marker = 'o', color = 'red', s = 25)
```

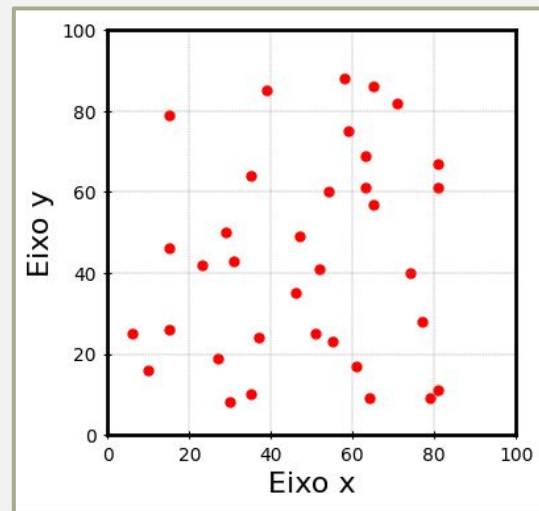
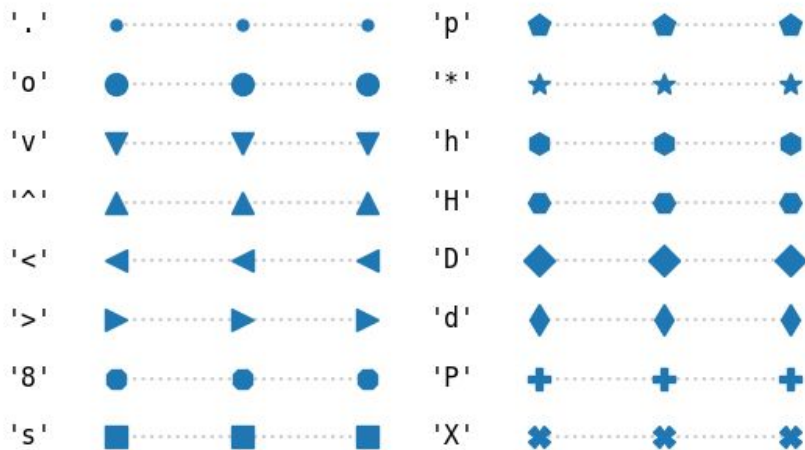


Já vimos que **`plt.scatter()`** é uma forma de colocar os pontos no gráfico

```
plt.scatter(x, y, marker = 'o', color = 'red', s = 25)
```

É possível mudar o formato dos pontos

Por exemplo:

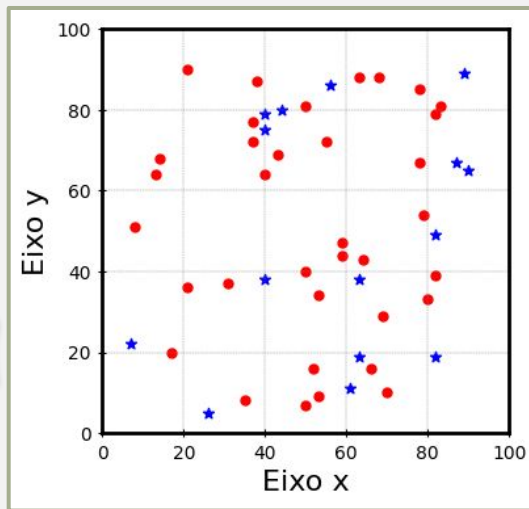
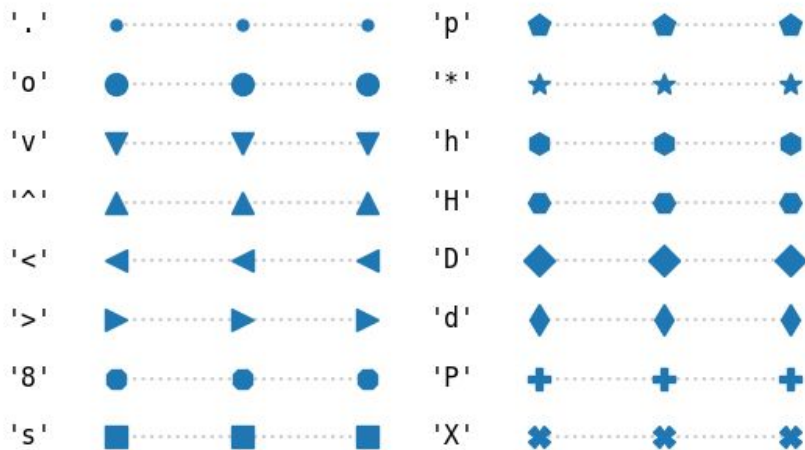


Já vimos que **plt.scatter()** é uma forma de colocar os pontos no gráfico

```
plt.scatter(x, y, marker = 'o', color = 'red', s = 25)
```

É possível mudar o formato dos pontos

Por exemplo:



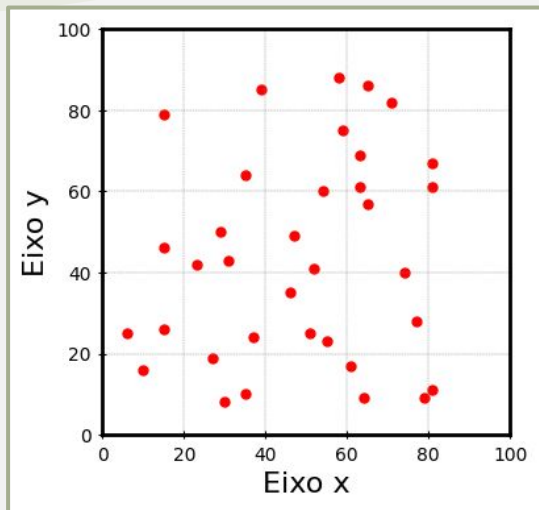
Dica!

Use diferentes formatos de pontos e cores se estiver trabalhando com diferentes grupos no mesmo gráfico. Isso torna a visualização melhor. Evite cores e formatos semelhantes entre as amostras

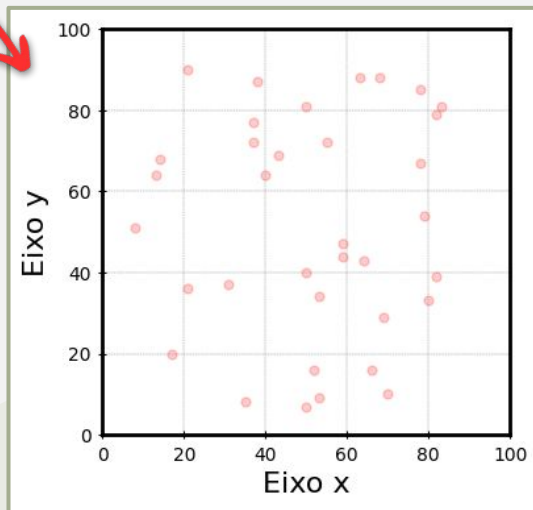
É possível modificar o **tamanho** e a **transparência** dos pontos

```
plt.scatter(x, y, marker = 'o', color = 'red', s = 25, alpha = 0.2)
```

Tamanho      Transparência  
alpha < 1.0



alpha = 1.0



alpha = 0.2

Cuidado para não  
“apagar” seus  
dados, eles  
precisam estar  
visíveis

É possível modificar o **tamanho** e a **transparência** dos pontos

```
plt.scatter(x, y, marker = 'o', color = 'red', s = 25, alpha = 0.2)
```

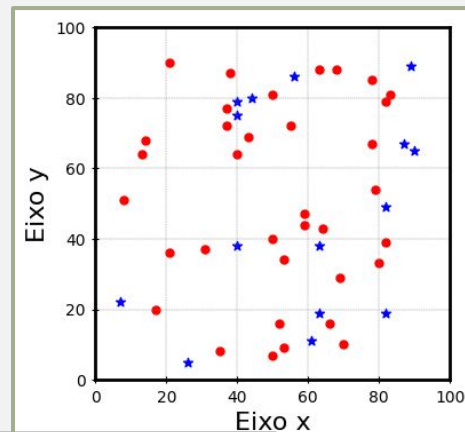
Tamanho      Transparência  
alpha < 1.0

É possível colocar **borda** nos pontos para ajudar na visualização

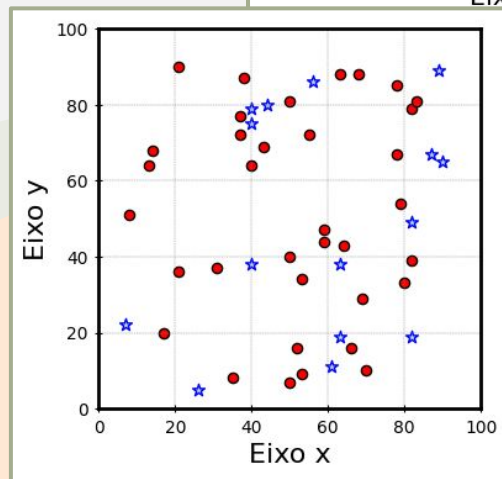
```
plt.scatter(x, y, marker = 'o', color = 'red', edgecolor = 'black', s = 25)
```

Vale para todos os formatos de marcadores

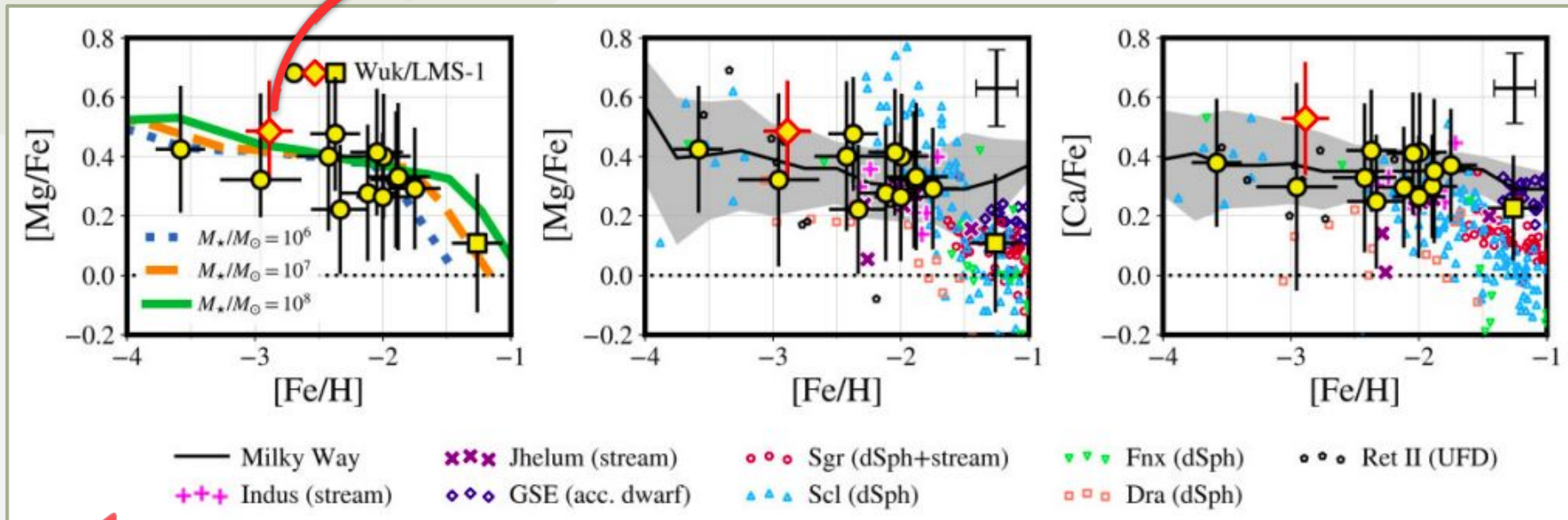
Sem bordas:



Com bordas:

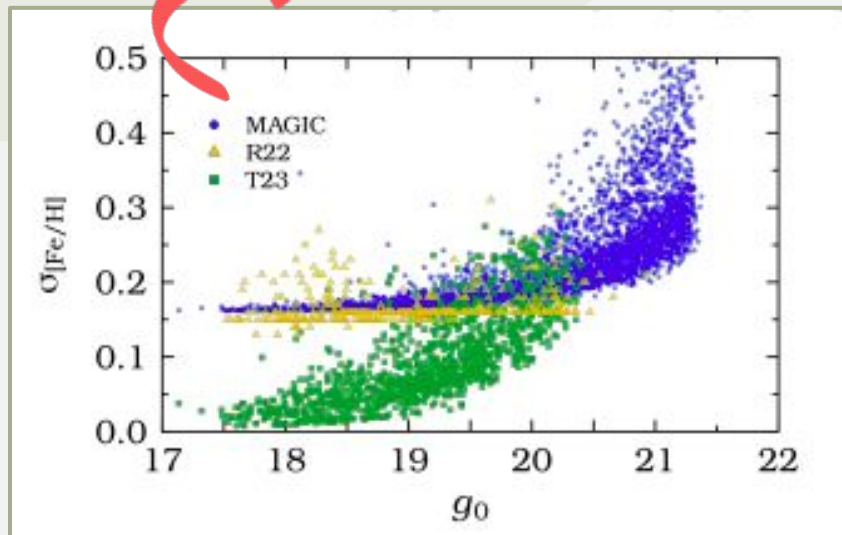


A estrela em destaque tem a mesma cor das outras, mas uma borda de cor diferente

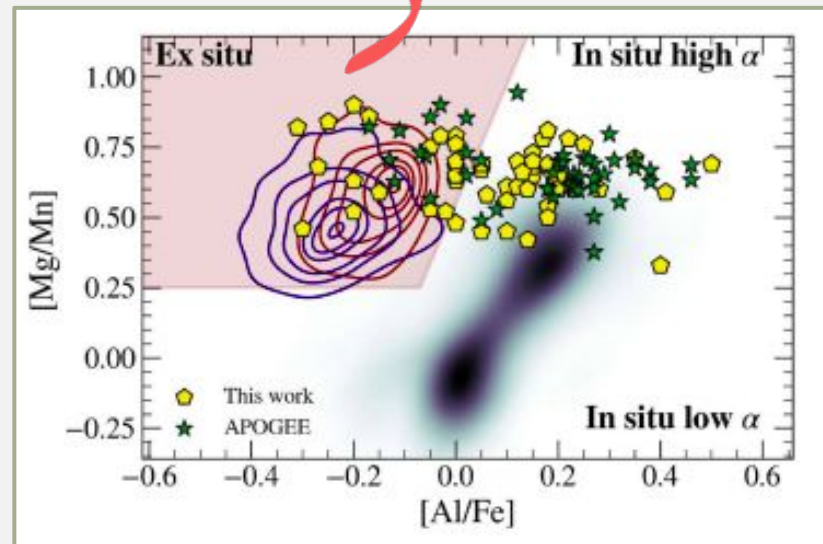


Cada estrutura representada ao fundo é identificada por um símbolo e uma cor diferente

Diferentes amostras com símbolos e cores diferentes



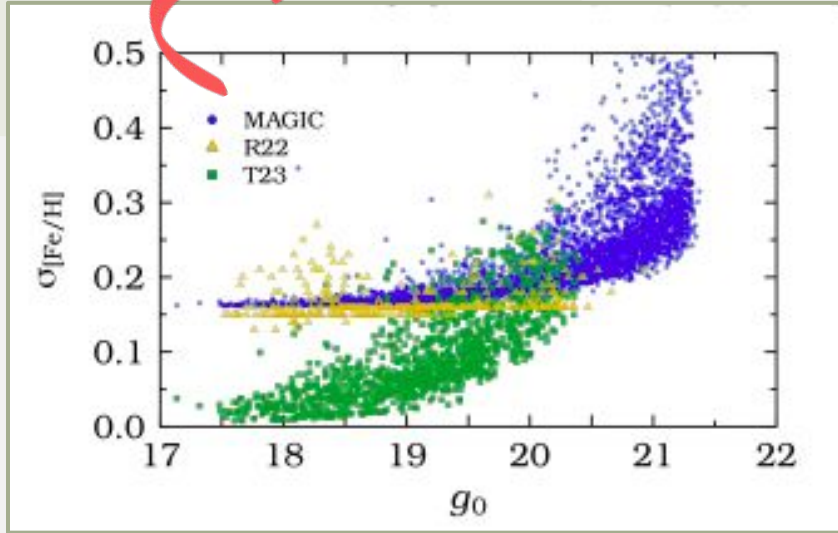
Barbosa et al. (2025)



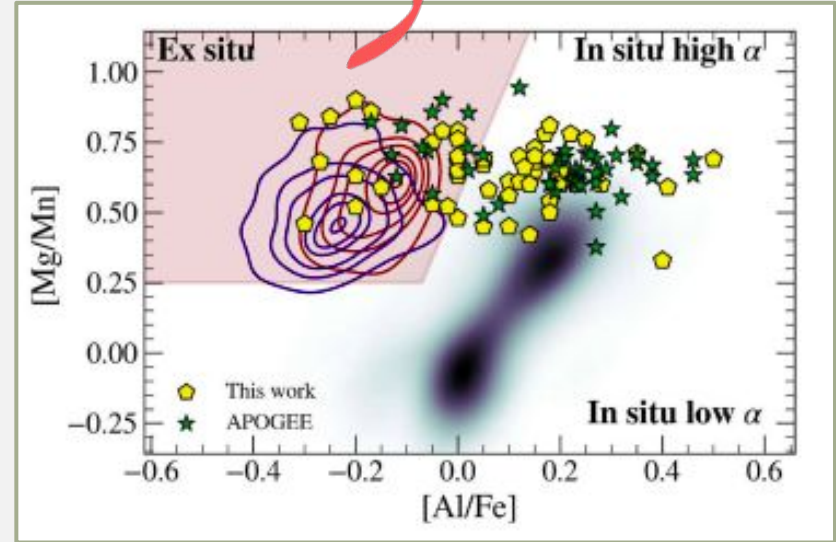
Barbuy et al. (2024)



Diferentes amostras com símbolos e cores diferentes



Barbosa et al. (2025)



Barbuy et al. (2024)

Dica!

Se você optar por usar o mesmo símbolo e apenas diferenciar as cores, teste diferentes tamanhos e transparência nos pontos.

# Como deixar o gráfico visualmente melhor?



Borda, eixos, título,  
grade



Tamanho e formato  
dos pontos



Espessura e estilo  
das linhas



Legendas



Cores

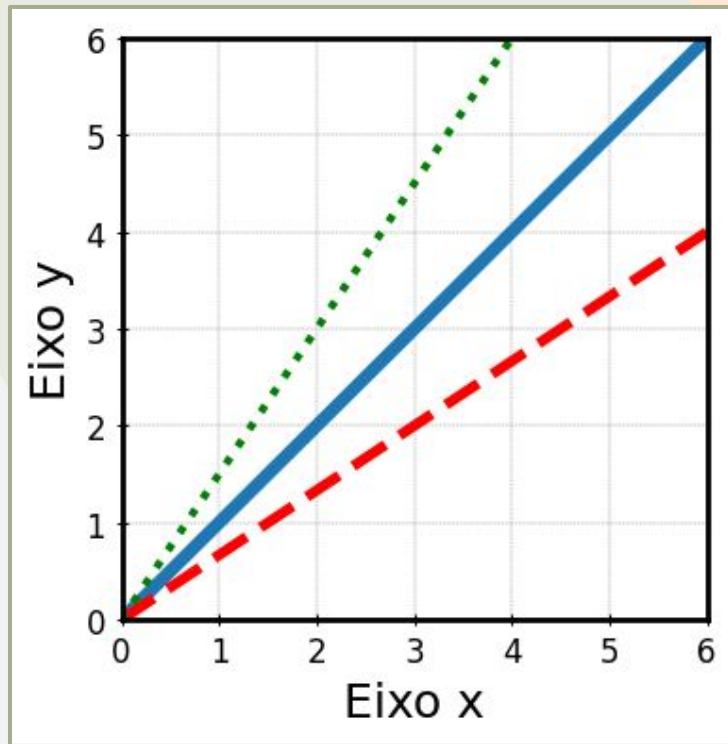


Outros

03

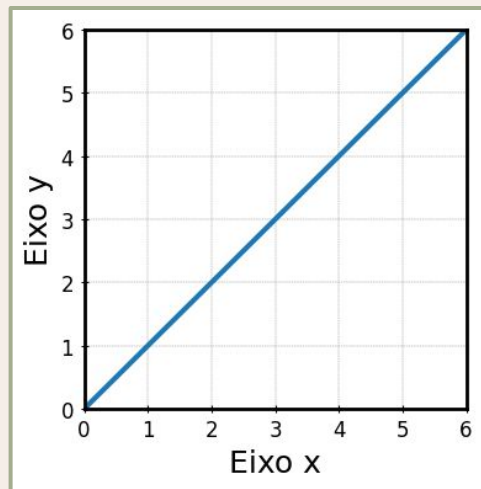
# Espessura e estilo de linhas

Visualização de gráficos de linha



Já vimos que ***plt.plot()*** é uma forma de fazer gráfico de linhas

```
plt.plot(x,y, linestyle = '-', linewidth = 3, color = 'blue')
```



Já vimos que **plt.plot()** é uma forma de fazer gráfico de linhas

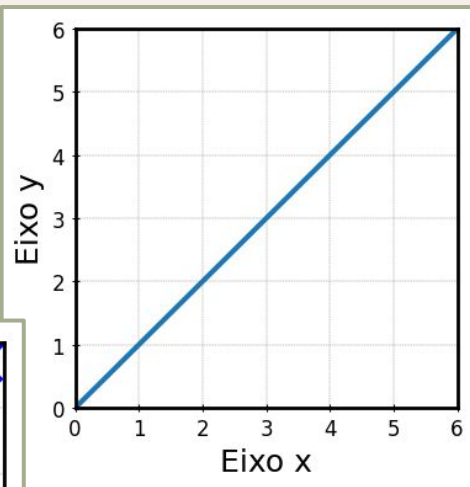
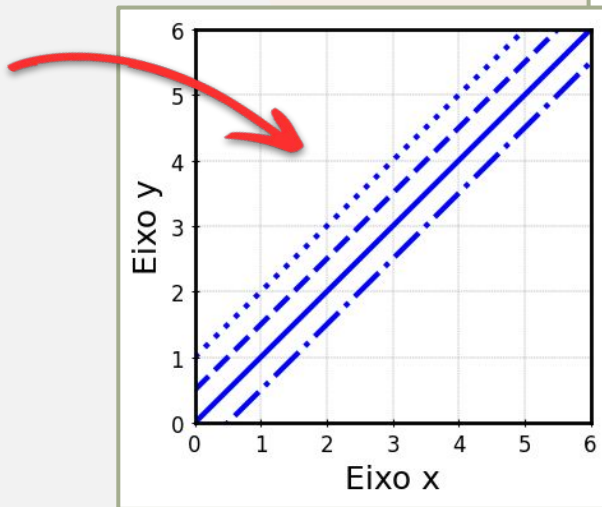
```
plt.plot(x,y, linestyle = '-', linewidth = 3, color = 'blue')
```

Muda o estilo da linha

pode usar apenas *ls* no lugar de *linestyle*

Opções de estilo de linhas:

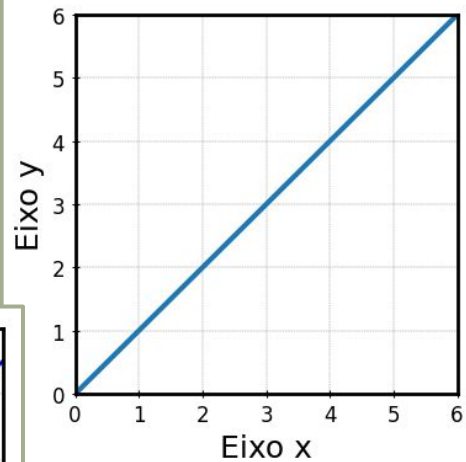
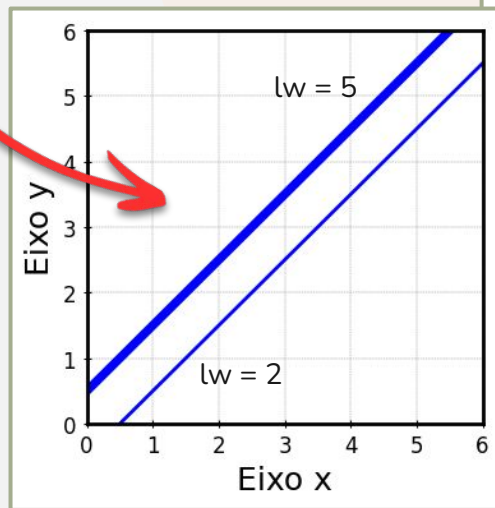
- '.' (linha pontilhada)
- ' (linha tracejada),
- '-' (linha sólida),
- . ' (linha tracejada e pontilhada)



Já vimos que ***plt.plot()*** é uma forma de fazer gráfico de linhas

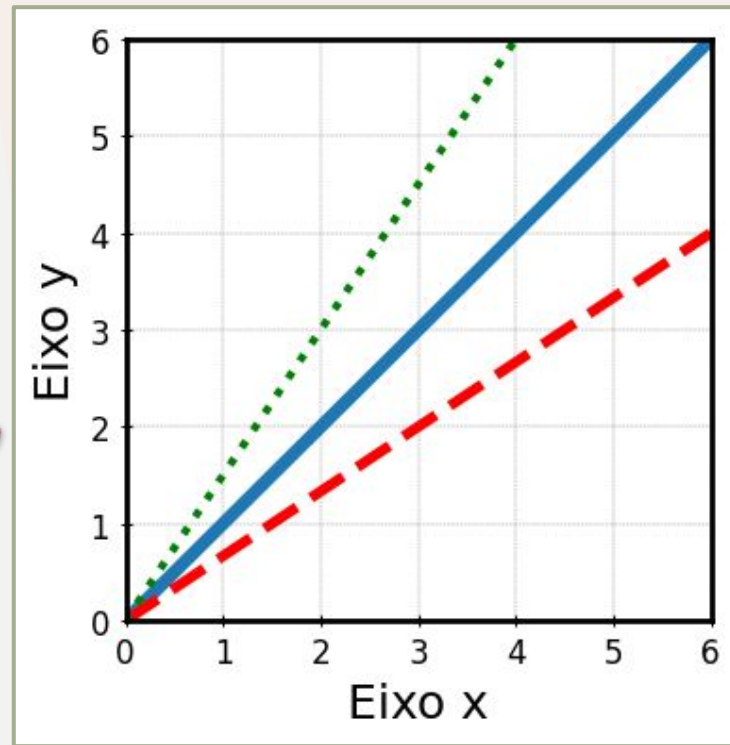
```
plt.plot(x,y, linestyle = '-', linewidth = 3, color = 'blue')
```

Muda a espessura da linha  
pode usar apenas *lw* no lugar de *linewidth*



Dica!

Use diferentes estilos de linhas, combinados com diferentes cores e espessuras, para facilitar a visualização de diferentes curvas



Já vimos que **`plt.plot()`** é uma forma de fazer gráfico de linhas

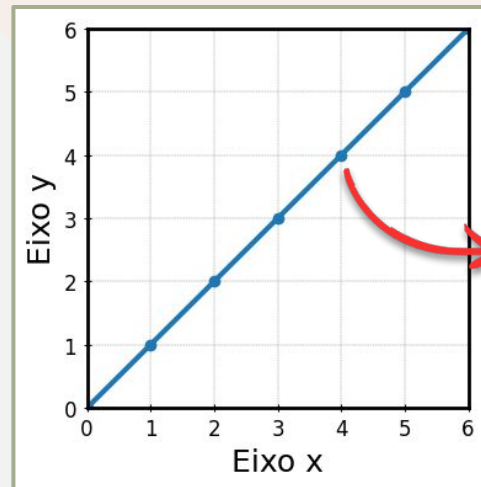
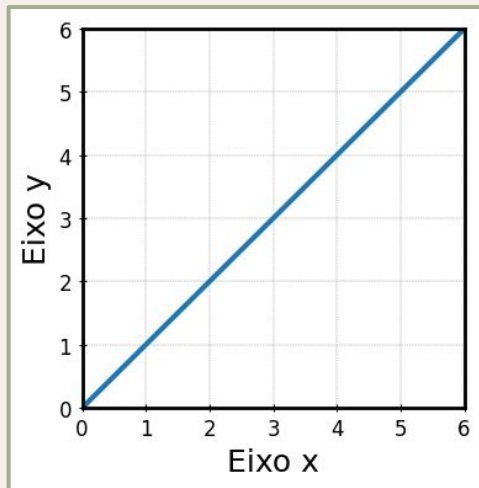
```
plt.plot(x,y, linestyle = '-', linewidth = 3, color = 'blue')
```

Se você quiser adicionar a marcação dos pontos, use

```
plt.plot(x,y, marker = 'o', linestyle = '-', linewidth = 3, color = 'blue')
```

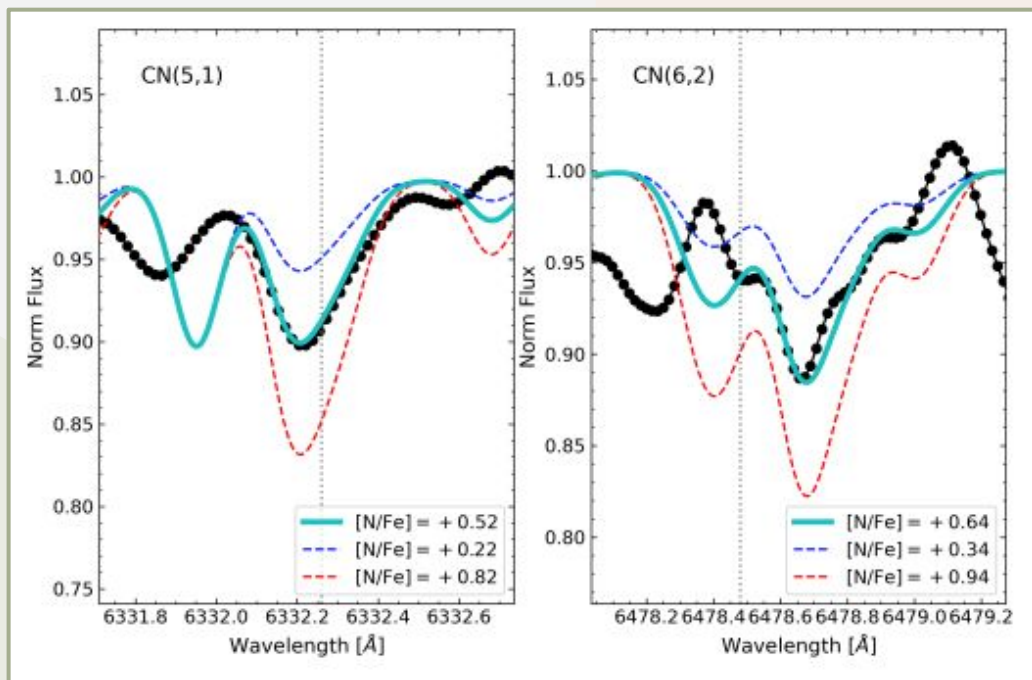


Aqui vale os mesmos marcadores que discutimos na parte de gráficos de pontos.



Pode manter ou não os pontos





Souza et al. (2021)

# Como deixar o gráfico visualmente melhor?



Borda, eixos, título,  
grade



Tamanho e formato  
dos pontos



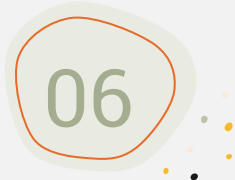
Espessura e estilo  
das linhas



Legendas



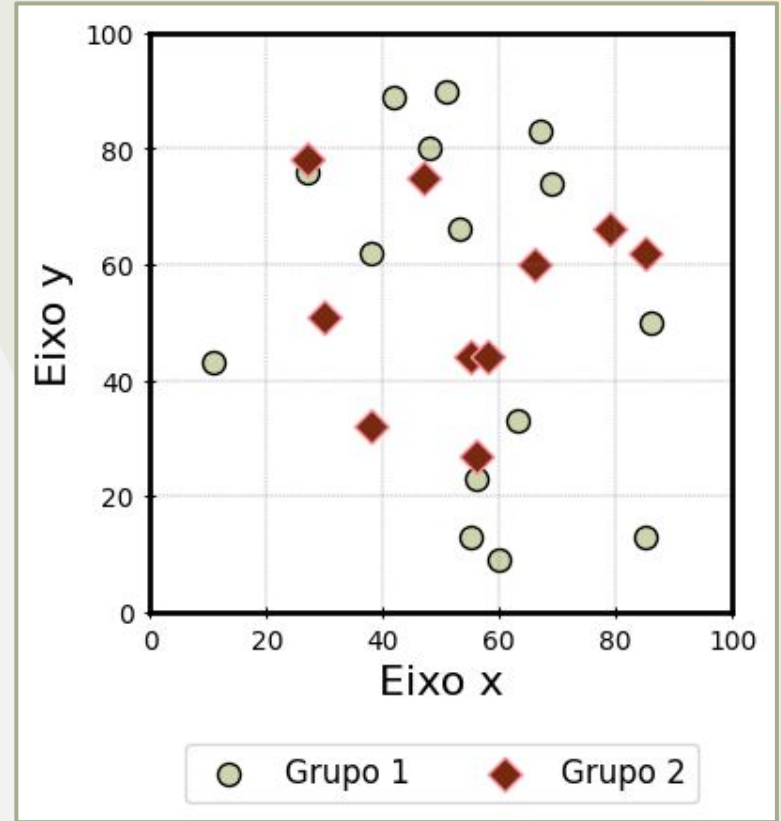
Cores



Outros

04

# Legendas

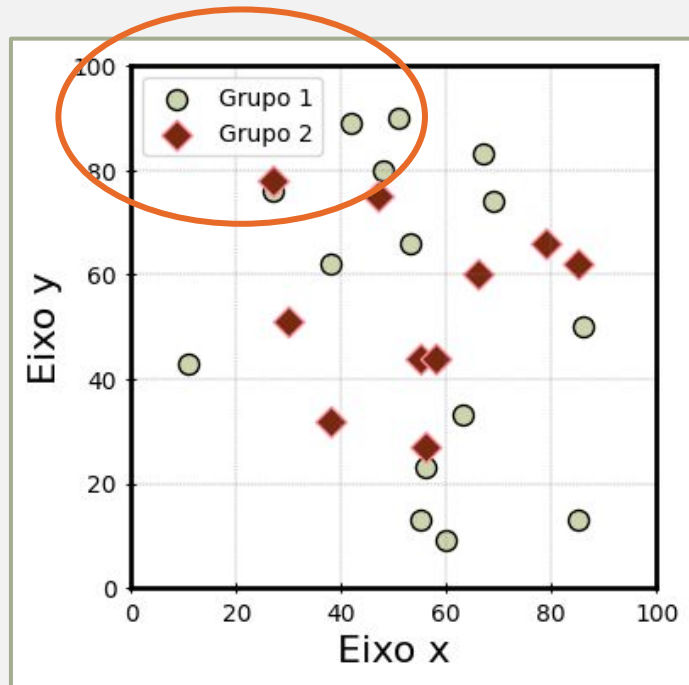


As legendas servem para explicar o que os símbolos e linhas no gráfico representam. Para isso, você pode usar o comando:

```
plt.legend()
```

Detalhe importante: você irá adicionar a descrição na linha de comando que você faz os pontos ou linhas usando o parâmetro **label**. No exemplo ao lado, foi definido da seguinte forma:

```
plt.scatter(x1, y1, marker = 'o', color = '#ccd5ae', edgecolor = 'black',  
            s = 75, label = 'Grupo 1')  
plt.scatter(x2, y2, marker = 'D', color = '#78290f', edgecolor = '#ff8fa3',  
            s = 85, label = 'Grupo 2')  
plt.legend()
```

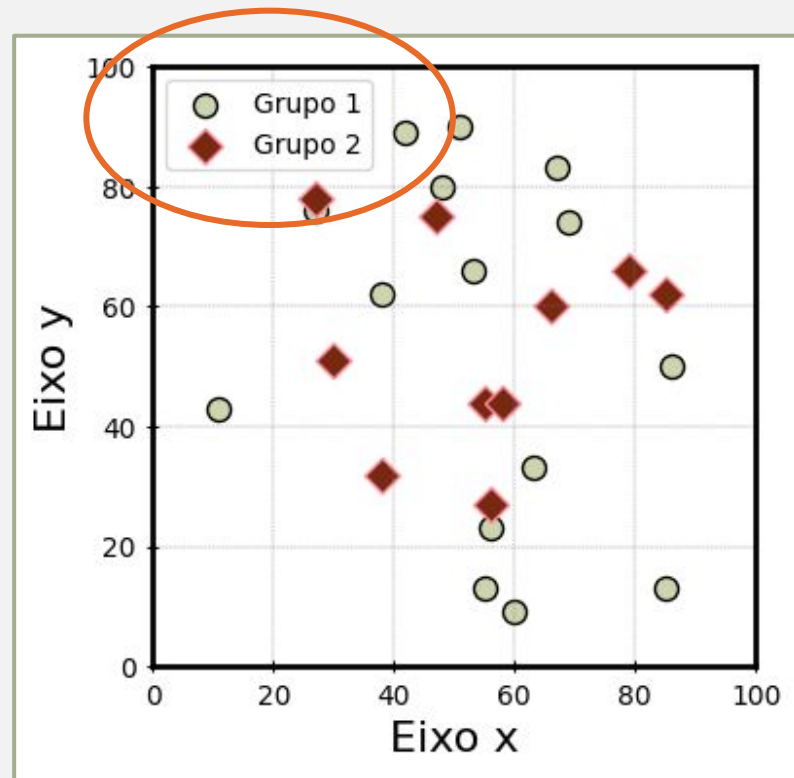


Dica!

É importante que as legendas NÃO ocultem seus dados. Elas devem ficar em uma região vazia do gráfico ou do lado externo.

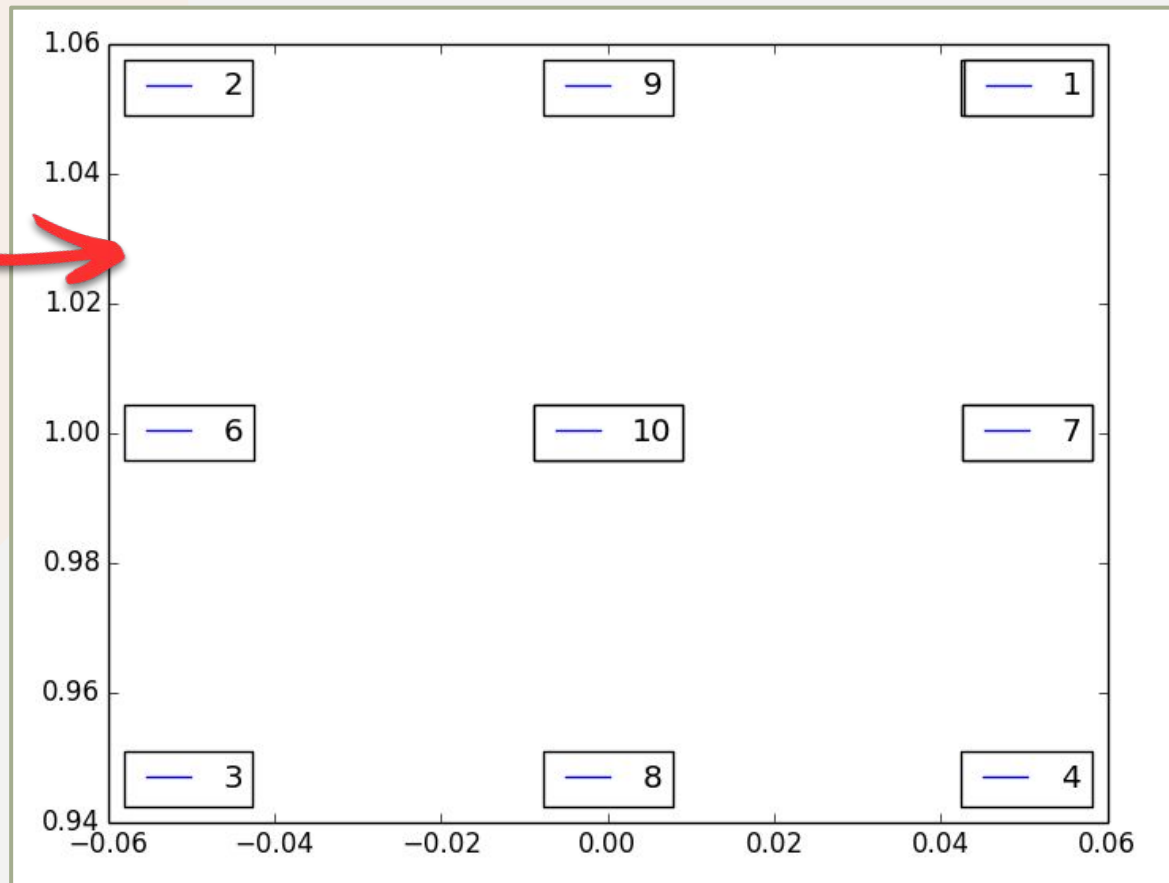
Para isso, você pode alterar a posição da legenda com o parâmetro `loc`

```
plt.legend(loc = 1)
```



```
plt.legend(loc = 1)
```

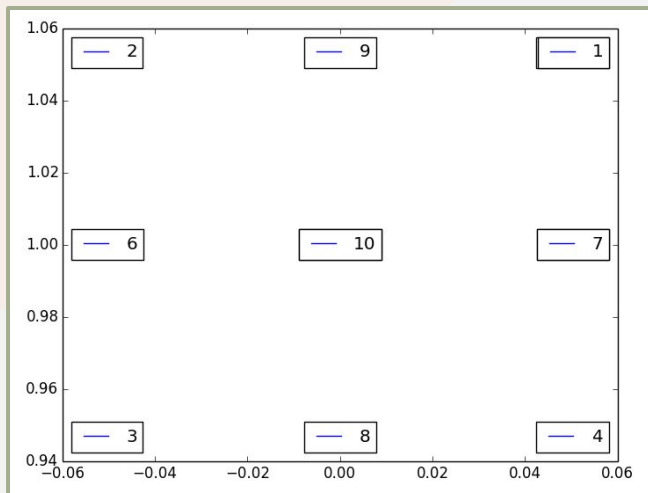
Cada número  
representa uma  
posição para a  
legenda dentro do  
quadro



```
plt.legend(loc = 1)
```

```
plt.legend(loc = 'upper right')
```

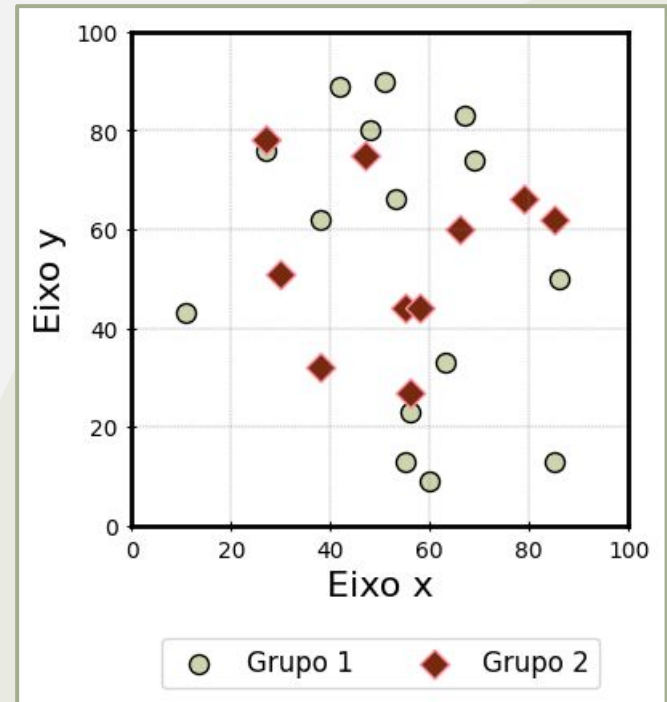
Pode escrever o nome da  
posição ao invés de colocar o  
número



Location String	Location Code
'best' (Axes only)	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

Dica!

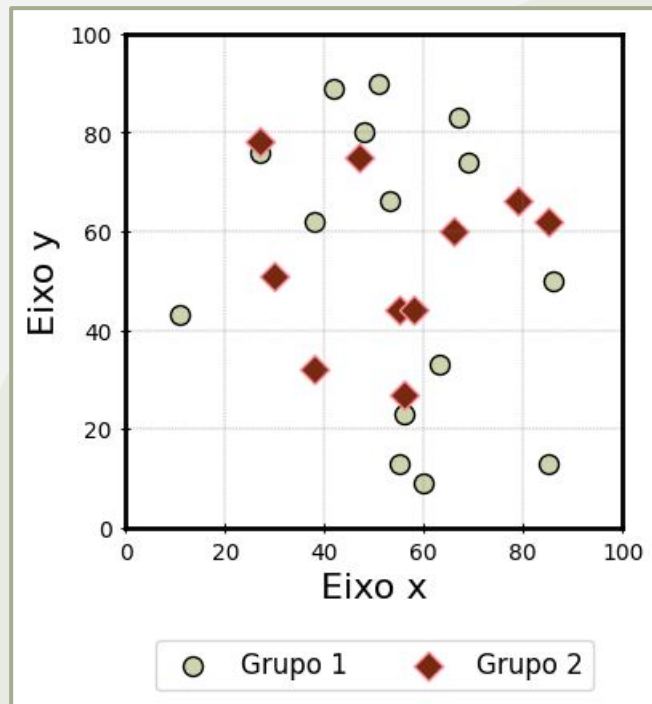
Se o gráfico já estiver com muita informação ou sem espaço vazio para colocar a legenda, opte por colocá-la do lado externo





Você pode usar o parâmetro ***bbox\_to\_anchor*** para ajustar a legenda e deixá-la na parte externa.

```
plt.legend(loc = 'lower center', bbox_to_anchor=(0.5, -0.35))
```



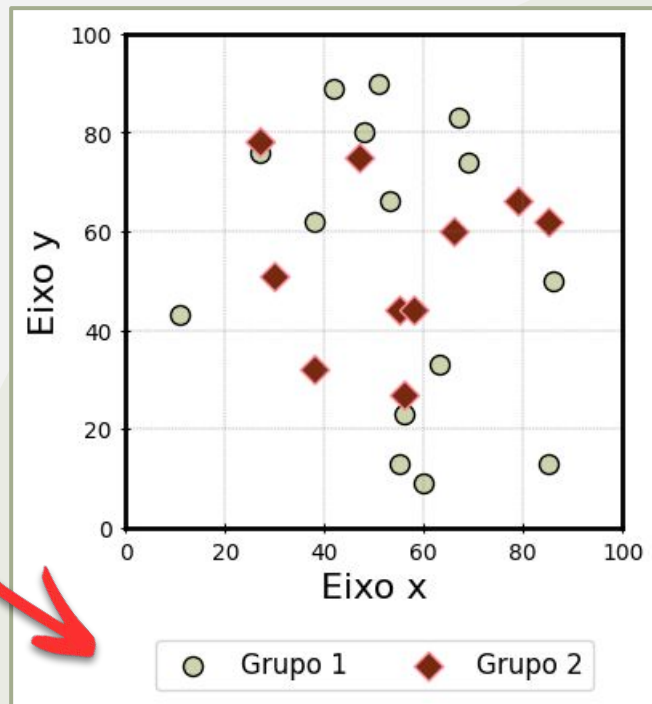
Você pode usar o parâmetro **bbox\_to\_anchor** para ajustar a legenda e deixá-la na parte externa.

```
plt.legend(loc = 'lower center', bbox_to_anchor=(0.5, -0.35))
```

Os parâmetros **fontsize** e **ncols** são úteis para deixar a legenda mais apresentável

- **fontsize** define o tamanho da fonte;
- **ncols** define em quantas colunas os dados da legenda serão separados

ncols = 2



# Como deixar o gráfico visualmente melhor?



Borda, eixos, título,  
grade



Tamanho e formato  
dos pontos



Espessura e estilo  
das linhas



Legendas



Cores



Outros

05

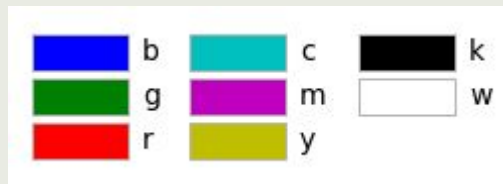
Cores



No site oficial do matplotlib, é possível encontrar as seguintes cores para serem usadas para pontos e linhas:



E as cores básicas podem ser identificadas pelas letras:



[https://matplotlib.org/stable/gallery/color/named\\_colors.html](https://matplotlib.org/stable/gallery/color/named_colors.html)

<span style="color: black;">■</span> black	<span style="color: #f5deb3;">■</span> bisque	<span style="color: #008000;">■</span> forestgreen	<span style="color: #4682b4;">■</span> slategray
<span style="color: #696969;">■</span> dimgray	<span style="color: #ffa500;">■</span> darkorange	<span style="color: #32cd32;">■</span> limegreen	<span style="color: #add8e6;">■</span> lightsteelblue
<span style="color: #808080;">■</span> dimgrey	<span style="color: #d2b48c;">■</span> burlywood	<span style="color: #006400;">■</span> darkgreen	<span style="color: #6495ed;">■</span> cornflowerblue
<span style="color: #a9a9a9;">■</span> gray	<span style="color: #fafade;">■</span> antiquewhite	<span style="color: #00ff00;">■</span> green	<span style="color: #4169e1;">■</span> royalblue
<span style="color: #bdbdbd;">■</span> grey	<span style="color: #f5deb3;">■</span> tan	<span style="color: #90ee90;">■</span> lime	<span style="color: #f0f0f0;">■</span> ghostwhite
<span style="color: #c0c0c0;">■</span> darkgray	<span style="color: #fffaf0;">■</span> navajowhite	<span style="color: #2e8b57;">■</span> seagreen	<span style="color: #d8bfd8;">■</span> lavender
<span style="color: #808080;">■</span> darkgrey	<span style="color: #fff5ee;">■</span> blanchedalmond	<span style="color: #3cb371;">■</span> mediumseagreen	<span style="color: #191970;">■</span> midnightblue
<span style="color: #c0c0c0;">■</span> silver	<span style="color: #fff5ee;">■</span> papayawhip	<span style="color: #00ff00;">■</span> springgreen	<span style="color: #00008b;">■</span> navy
<span style="color: #d3d3d3;">■</span> lightgray	<span style="color: #f5deb3;">■</span> moccasin	<span style="color: #90ee90;">■</span> mintcream	<span style="color: #0000cd;">■</span> darkblue
<span style="color: #e0e0e0;">■</span> lightgrey	<span style="color: #ffa500;">■</span> orange	<span style="color: #00ff00;">■</span> mediumspringgreen	<span style="color: #0000ff;">■</span> mediumblue
<span style="color: #d3d3d3;">■</span> gainsboro	<span style="color: #f5deb3;">■</span> wheat	<span style="color: #4682b4;">■</span> mediaquamarine	<span style="color: #0000ff;">■</span> blue
<span style="color: #f5f5dc;">■</span> whitesmoke	<span style="color: #f5deb3;">■</span> oldlace	<span style="color: #7fffd4;">■</span> aquamarine	<span style="color: #6a5acd;">■</span> slateblue
<span style="color: #ffffff;">■</span> white	<span style="color: #fffaf0;">■</span> floralwhite	<span style="color: #40e0d0;">■</span> turquoise	<span style="color: #2f4f4f;">■</span> darkslateblue
<span style="color: #fff2cc;">■</span> snow	<span style="color: #8b4513;">■</span> darkgoldenrod	<span style="color: #7fffd4;">■</span> lightseagreen	<span style="color: #6a5acd;">■</span> mediumslateblue
<span style="color: #d2b48c;">■</span> rosybrown	<span style="color: #d4af37;">■</span> goldenrod	<span style="color: #4682b4;">■</span> mediumturquoise	<span style="color: #8a2be2;">■</span> mediumpurple
<span style="color: #ff7f50;">■</span> lightcoral	<span style="color: #fff5ee;">■</span> cornsilk	<span style="color: #add8e6;">■</span> azure	<span style="color: #800080;">■</span> rebeccapurple
<span style="color: #cd5c5c;">■</span> indianred	<span style="color: #ffd700;">■</span> gold	<span style="color: #87ceeb;">■</span> lightcyan	<span style="color: #8a2be2;">■</span> blueviolet
<span style="color: #8b4513;">■</span> brown	<span style="color: #fffacd;">■</span> lemonchiffon	<span style="color: #afEEEE;">■</span> paleturquoise	<span style="color: #800080;">■</span> indigo
<span style="color: #dc143c;">■</span> firebrick	<span style="color: #f0e68c;">■</span> khaki	<span style="color: #2f4f4f;">■</span> darkslategray	<span style="color: #800080;">■</span> darkorchid
<span style="color: #8b0000;">■</span> maroon	<span style="color: #f0e68c;">■</span> palegoldenrod	<span style="color: #2f4f4f;">■</span> darkslategrey	<span style="color: #800080;">■</span> darkviolet
<span style="color: #8b0000;">■</span> darkred	<span style="color: #a0a080;">■</span> darkkhaki	<span style="color: #008080;">■</span> teal	<span style="color: #800080;">■</span> mediumorchid
<span style="color: #ff0000;">■</span> red	<span style="color: #ffff00;">■</span> ivory	<span style="color: #008080;">■</span> darkcyan	<span style="color: #d8bfd8;">■</span> thistle
<span style="color: #ffb6c1;">■</span> mistyrose	<span style="color: #f5deb3;">■</span> beige	<span style="color: #00bfff;">■</span> aqua	<span style="color: #dda0dd;">■</span> plum
<span style="color: #ffa07a;">■</span> salmon	<span style="color: #ffff00;">■</span> lightyellow	<span style="color: #00bfff;">■</span> cyan	<span style="color: #dda0dd;">■</span> violet
<span style="color: #ff4500;">■</span> tomato	<span style="color: #fffacd;">■</span> lightgoldenrodyellow	<span style="color: #00bfff;">■</span> darkturquoise	<span style="color: #800080;">■</span> purple
<span style="color: #ffa07a;">■</span> darksalmon	<span style="color: #808000;">■</span> olive	<span style="color: #4682b4;">■</span> cadetblue	<span style="color: #800080;">■</span> darkmagenta
<span style="color: #ff4500;">■</span> coral	<span style="color: #ffff00;">■</span> yellow	<span style="color: #add8e6;">■</span> powderblue	<span style="color: #ff00ff;">■</span> fuchsia
<span style="color: #ffa07a;">■</span> orangered	<span style="color: #808000;">■</span> olivedrab	<span style="color: #add8e6;">■</span> lightblue	<span style="color: #ff00ff;">■</span> magenta
<span style="color: #ffa07a;">■</span> lightsalmon	<span style="color: #9acd32;">■</span> yellowgreen	<span style="color: #00bfff;">■</span> deepskyblue	<span style="color: #ff69b4;">■</span> orchid
<span style="color: #d2691e;">■</span> sienna	<span style="color: #6b8e23;">■</span> darkolivegreen	<span style="color: #6495ed;">■</span> skyblue	<span style="color: #800080;">■</span> mediumvioletred
<span style="color: #f5deb3;">■</span> seashell	<span style="color: #9acd32;">■</span> greenyellow	<span style="color: #6495ed;">■</span> lightskyblue	<span style="color: #ff69b4;">■</span> deeppink
<span style="color: #8b4513;">■</span> chocolate	<span style="color: #32cd32;">■</span> chartreuse	<span style="color: #4682b4;">■</span> steelblue	<span style="color: #ff69b4;">■</span> hotpink
<span style="color: #8b4513;">■</span> saddlebrown	<span style="color: #32cd32;">■</span> lawngreen	<span style="color: #add8e6;">■</span> aliceblue	<span style="color: #f0f0f0;">■</span> lavenderblush
<span style="color: #d2b48c;">■</span> sandybrown	<span style="color: #90ee90;">■</span> honeydew	<span style="color: #4169e1;">■</span> dodgerblue	<span style="color: #d8bfd8;">■</span> palevioletred
<span style="color: #ffdeb3;">■</span> peachpuff	<span style="color: #6b8e23;">■</span> darkseagreen	<span style="color: #4682b4;">■</span> lightslategray	<span style="color: #dc143c;">■</span> crimson
<span style="color: #8b4513;">■</span> peru	<span style="color: #90ee90;">■</span> palegreen	<span style="color: #4682b4;">■</span> lightslategrey	<span style="color: #ffb6c1;">■</span> pink
<span style="color: #f5deb3;">■</span> linen	<span style="color: #32cd32;">■</span> lightgreen	<span style="color: #4682b4;">■</span> slategray	<span style="color: #ffb6c1;">■</span> lightpink

Porém...

Existe um parque de diversões chamado **COOLORS**

1°

PASSO:

Digite “colors” no google



colors



Todas Shopping Imagens Vídeos Vídeos curtos Notícias Web : Mais



Coolors

<https://coolors.co> · Traduzir esta página

**Coolors - The super fast color palettes generator!**

The super fast color palettes generator! Create the perfect palette or get inspired by thousands of beautiful color schemes. ... Create, browse and save palettes ...

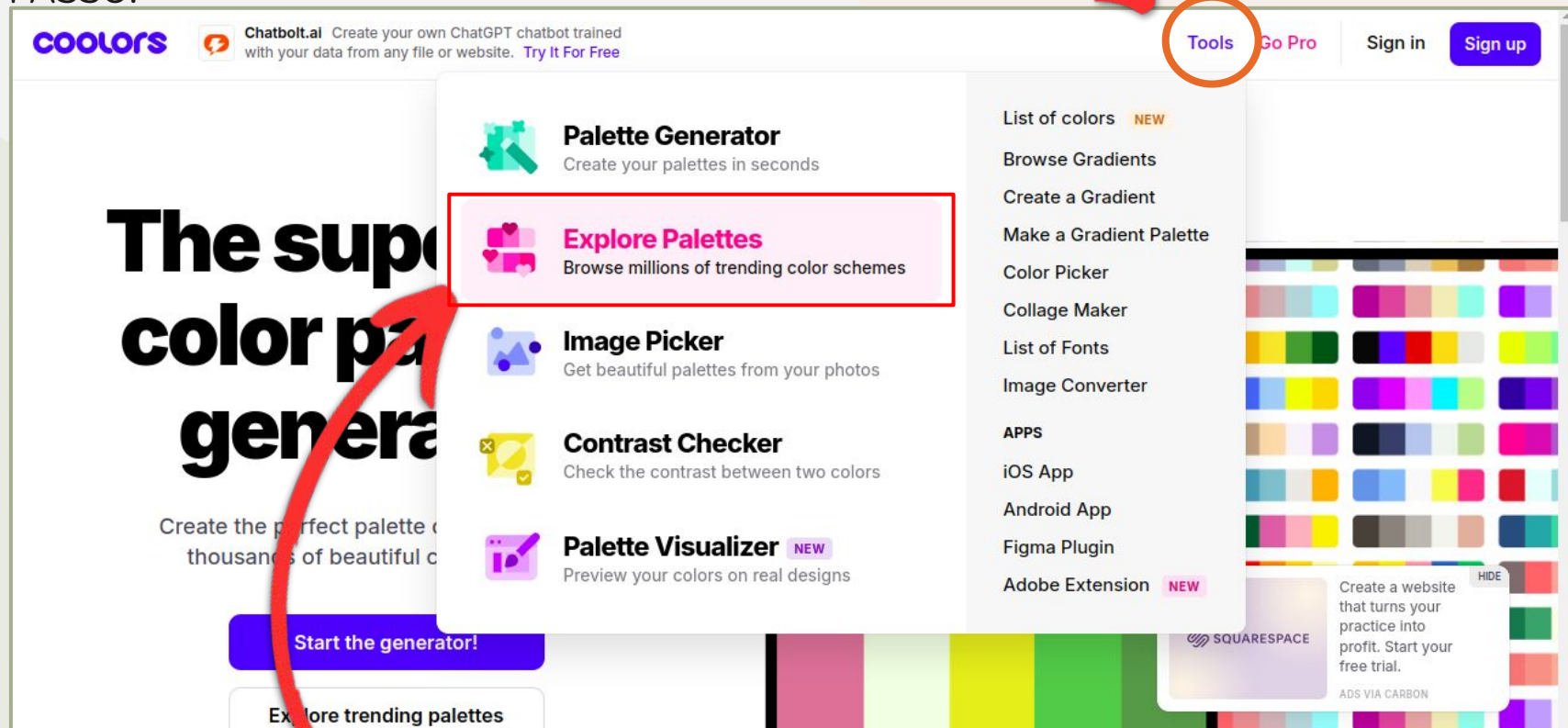
Selecione esse link



2°

PASSO:

Venha em “Tools” no menu superior direito



3°

PASSO:

Selecione “Explore Palettes”

Search with colors, topics, styles or hex values...



# Trending Color Palettes

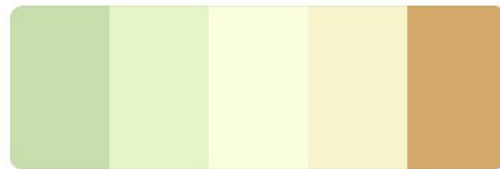
Get inspired by thousands of beautiful color schemes and make something cool!



♡ 90K ...



♡ 63.3K ...



♡ 80.1K ...



Design your in



♡ 90K ...

4°

PASSO:

Coloque o cursor do mouse sobre a cor desejada e clique, isso irá copiar o código da cor



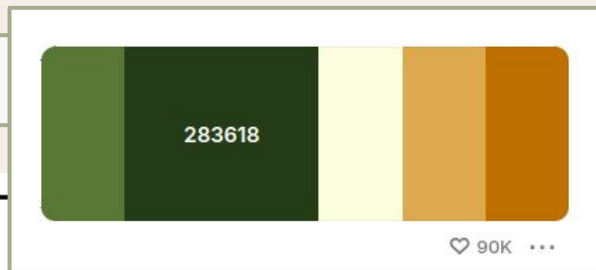
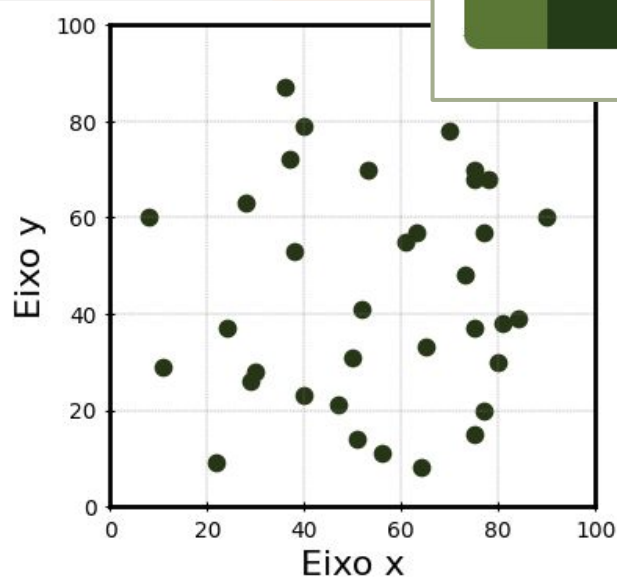
5°

PASSO:

No parâmetro *color* da sua linha de código no python, coloque “#código da cor”  
Obs: não irá funcionar sem o #

Exemplo:

```
plt.scatter(x, y, marker = 'o', color = '#283618')
```



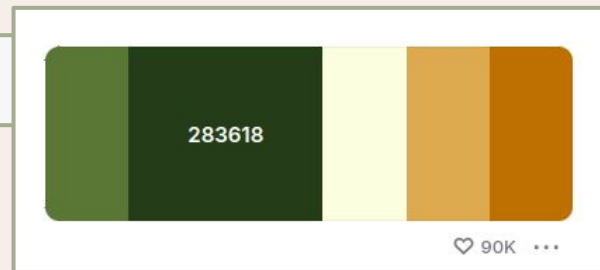
5°

PASSO:

No parâmetro *color* da sua linha de código no python, coloque “#(código da cor)”  
Obs: não irá funcionar sem o #

Exemplo:

```
plt.scatter(x, y, marker = 'o', color = '#283618')
```



Divirta-se

E use com moderação,  
porque há sérios riscos de  
você perder horas tentando  
escolher cores!

# Como deixar o gráfico visualmente melhor?



Borda, eixos, título,  
grade



Tamanho e formato  
dos pontos



Espessura e estilo  
das linhas



Legendas



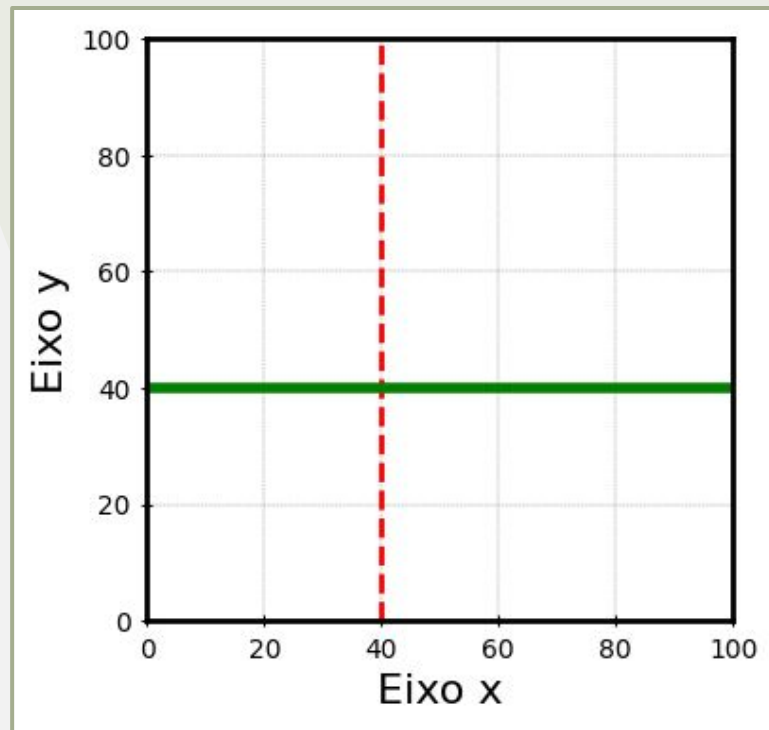
Cores



Outros

06

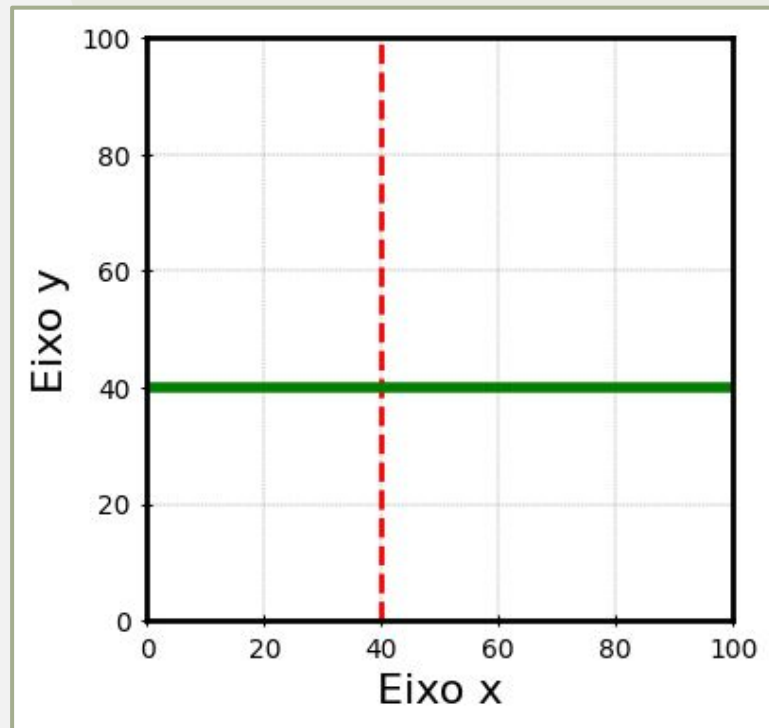
Outros



# Funções *axhline* e *axvline*

Se você deseja apenas fazer uma linha vertical ou horizontal no gráfico, pode usar as funções ***axhline*** (linha horizontal) e ***axvline*** (vertical), como no exemplo:

```
plt.axvline(40, color = 'red', ls = '--', lw = 2)  
plt.axhline(40, color = 'green', ls = '-', lw = 4)
```



[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.axvline.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.axvline.html)

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.axhline.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.axhline.html)

# Outras funções úteis

→ **`fill_between(x, y1, y2, color)`**: Traça uma região colorida entre  $y_1$  e  $y_2$

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.fill\\_between.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.fill_between.html)

→ **`axvspan ou axhspan(ymin, ymax, xmin, xmax, color)`**: segue o mesmo princípio de `axhline` e `ahvline` do slide anterior. Porém, ao invés de traçar uma linha, irá traçar uma faixa colorida

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.axes.Axes.axhspan.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.axhspan.html)

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.axvspan.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.axvspan.html)

→ **`plt.text(x,y,'texto', color)`**: insere um texto na posição  $x,y$  que foi indicada

[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.text.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.text.html)