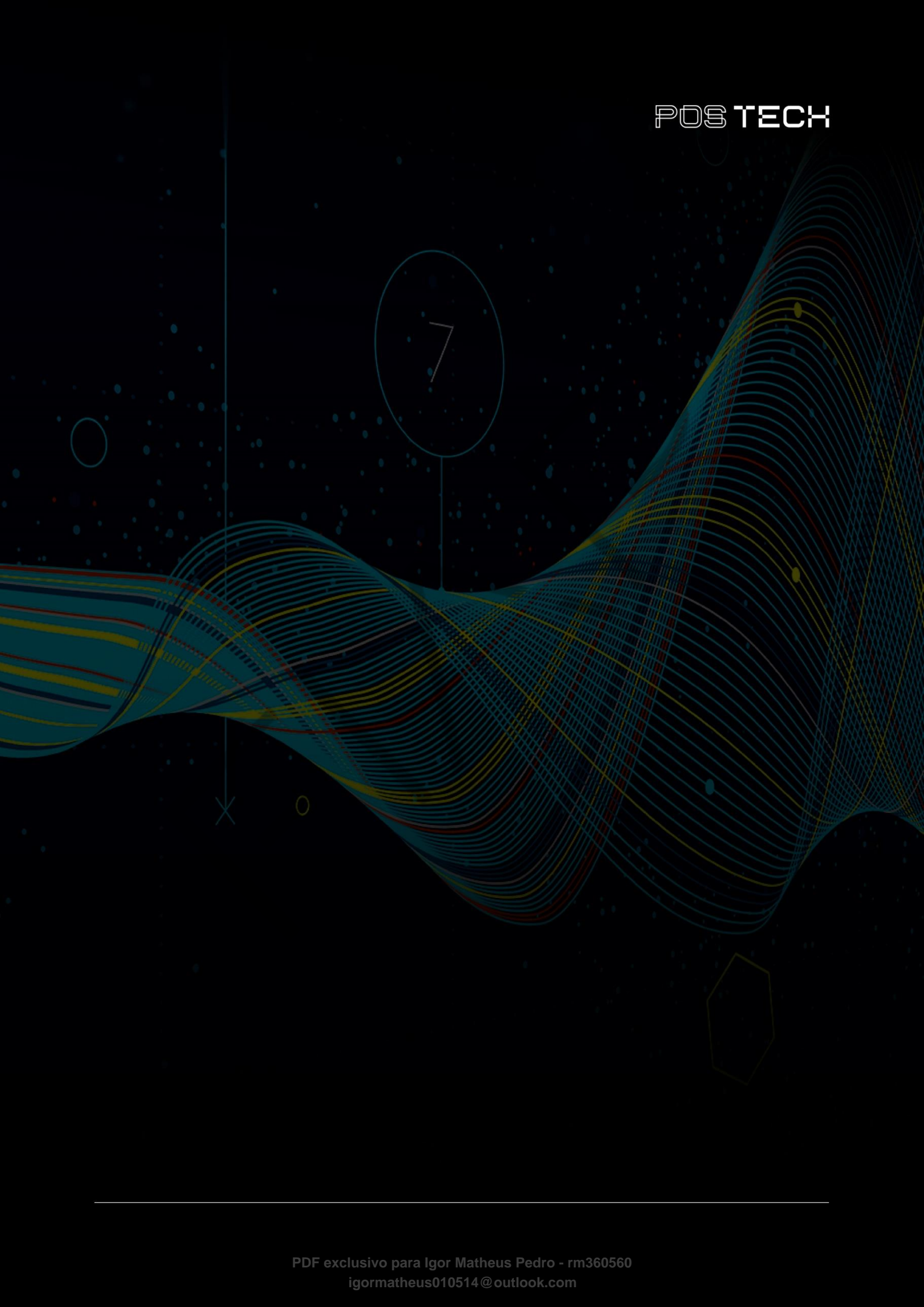


POSTECH



**SUMÁRIO**

O QUE VEM POR AÍ? .....	3
HANDS ON .....	4
SAIBA MAIS.....	5
O QUE VOCÊ VIU NESTA AULA? .....	10
REFERÊNCIAS.....	11

EMSE

## O QUE VEM POR AÍ?

Fala analytics expert, tudo certo?!

Chegamos em mais um ponto interessante em nossa jornada, onde continuaremos o aprofundamento nas manipulações de dados.

Até o momento, você partiu do ambiente do Google Colab, onde construímos nossas primeiras visualizações acerca dos dados disponíveis do DATASUS e, além disso, aprendemos conceitos base de análise como a aleatoriedade, a reprodutibilidade e a ordenação.

E, se por acaso, você não lembra direito o código exato, acesse-o no [nosso GitHub!](#)

Nesta aula, você entenderá a importância de uma boa apresentação gráfica, para que não passemos uma imagem errada sobre os dados apresentados, e muito menos geremos uma interpretação dicotômica. É muito importante que você sempre revise as aulas anteriores, faça os exercícios e bote em prática cada detalhe. Lembre-se que o hands on é fundamental para o aprendizado!

Agora, vamos aprender mais sobre esse maravilhoso mundo dos dados!

## HANDS ON

Agora, chegou o momento de ver, na prática, como começar a importar nossos dados e trabalhar com eles via programação. A ideia é não se limitar apenas ao código explícito no hands on, então recomendamos que procure a documentação das bibliotecas, explore novas funcionalidades e muito mais! Lembrando que os nossos docentes e pessoa de Community Management estão disponíveis no Discord para te ajudar com eventuais dúvidas.

Confira a videoaula e faça as sugestões de desafios! O material completo da aula você encontra aqui, no [nosso GitHub](#).

Essa aula envolve processos de exploração gráfica, e o ideal é que você não se limite ao código da aula. Procure explorar e vivenciar os processos de construção, combinado?

## SAIBA MAIS

Jovem padawan, muito provavelmente você já esbarrou em situações em que fosse difícil olhar para um gráfico e interpretar os valores por conta de sua legenda, ou até mesmo por ter muita informação, o que faz com que você se depare com uma enorme poluição visual.

E, para te explicarmos mais sobre este ponto, é muito importante que você confira a super aula que preparamos para você!

Parece bobagem, mas devido ao nosso padrão evolutivo, nosso cérebro tende a aceitar melhor informações otimizadas. Para um melhor entendimento, imagine este primeiro gráfico (que aparece em nossa videoaula):

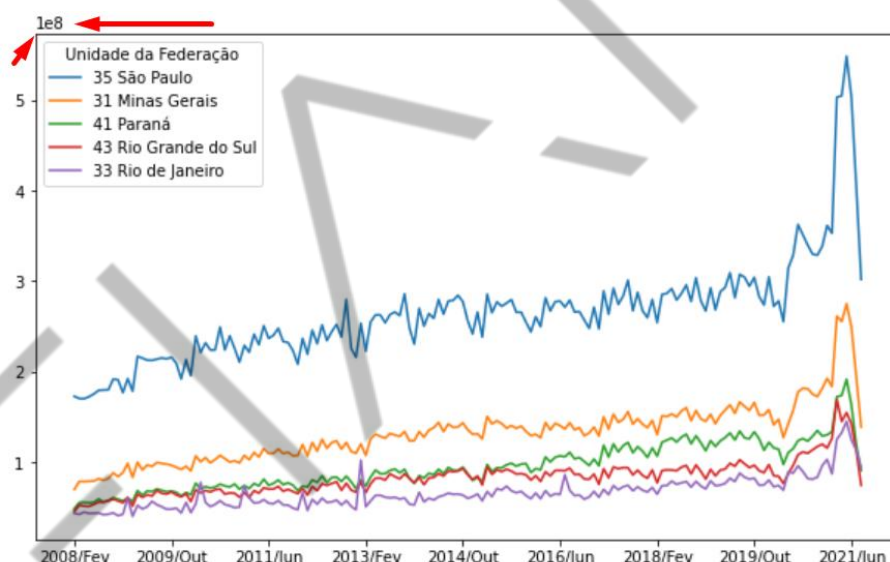


Figura 1 – Gráfico não ajustado em y  
Fonte: Elaborado pelo autor (2023)

As setas vermelhas mostram a escala em que o eixo y está, então, se você não domina matemática elementar e suas notações, ficará difícil interpretar a situação. Essa escala “1e8” significa: “ $10^8$ ” ou 100 milhões.

Perceba que, de uma maneira visual, isso não está bacana. Em m contrapartida, existe uma solução (que é apresentada em nossa videoaula), onde temos uma adição importante no gráfico:

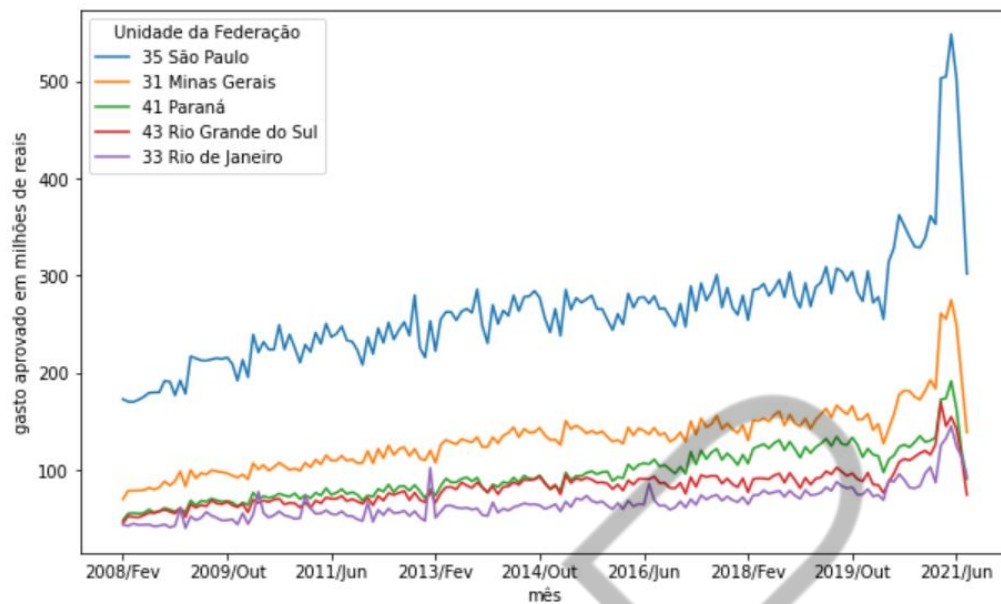


Figura 2 – Gráfico ajustado em y  
 Fonte: Elaborado pelo autor (2023)

Veja que, adicionando uma simples legenda e ajustando a escala, a visualização se tornou mais fácil de compreender, deixando a mensagem de que o eixo vertical quer passar simplesmente ajustando os labels e “normalizando” os valores, e para isso, apenas duas células foram necessárias.

```
ordenados_por_total = ordenados_por_total / 1_000_000
```

```
axis = ordenados_por_total.head(5).T.plot(figsize=(10,6))
axis.set_ylabel("gasto aprovado em milhões de reais")
axis.set_xlabel("mês")
```

Notebook Aula 4 – Comandos de ajuste do eixo y do gráfico  
 Fonte: Elaborado pelo autor (2023)

Então, veja a importância de uma simples mudança em uma visualização, mas que faz toda a diferença em uma análise. Principalmente se esse tipo de gráfico for apresentado para alguém que não seja técnico, mas que precisa entender a mensagem descrita ali.

Um outro ponto é o eixo “x”, que ao ser alongado ou encurtado pode passar mensagens visuais diferentes, mas que pode variar o tipo de problema em que você vai atuar, principalmente quando o assunto é a tal “série temporal”. Uma maneira



interessante de ver o mesmo gráfico em tamanhos diferentes é fazer o mesmo que é apresentado nas figuras 3 – “Gráfico ajustado em x (1°)”, figura 4 – “Gráfico ajustado em x (2°)” e figura 5 – “Gráfico ajustado em x (3°)”:

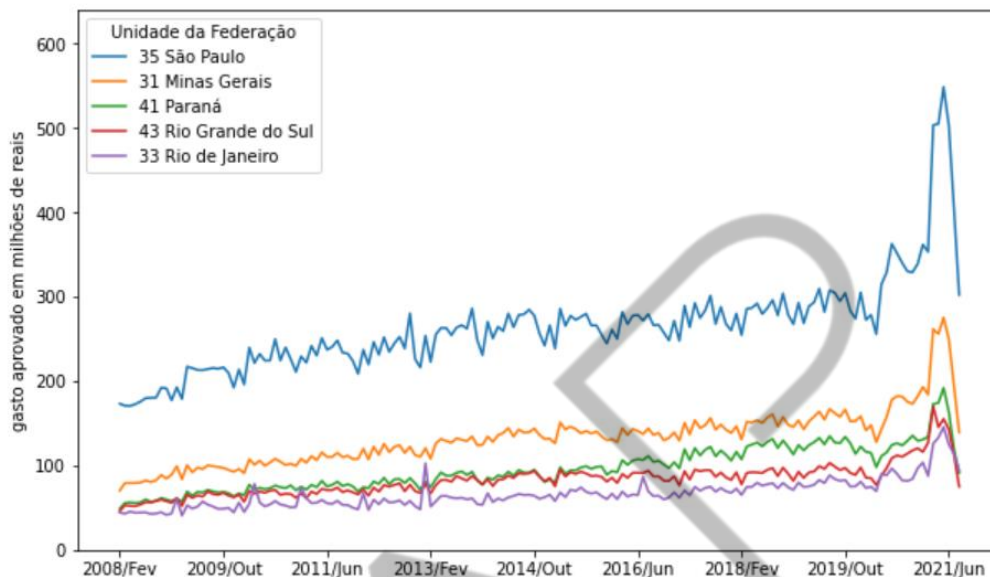


Figura 3 – Gráfico ajustado em x (1°)

Fonte: Elaborado pelo autor (2023)

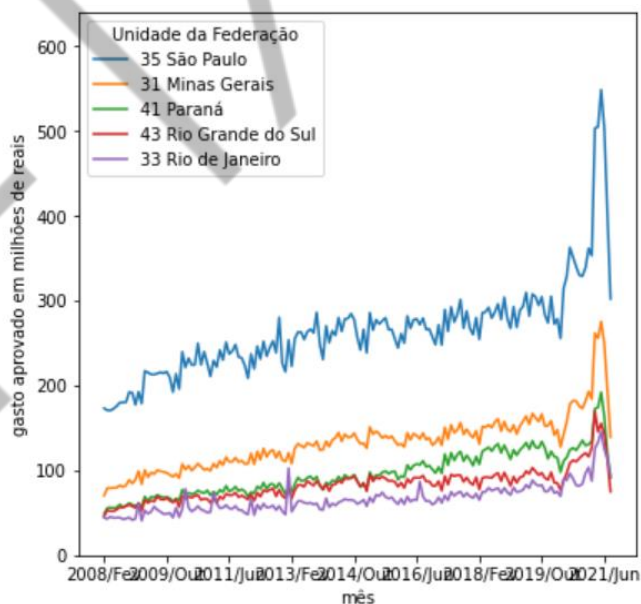


Figura 4 – Gráfico ajustado em x (2°)

Fonte: Elaborado pelo autor (2023)

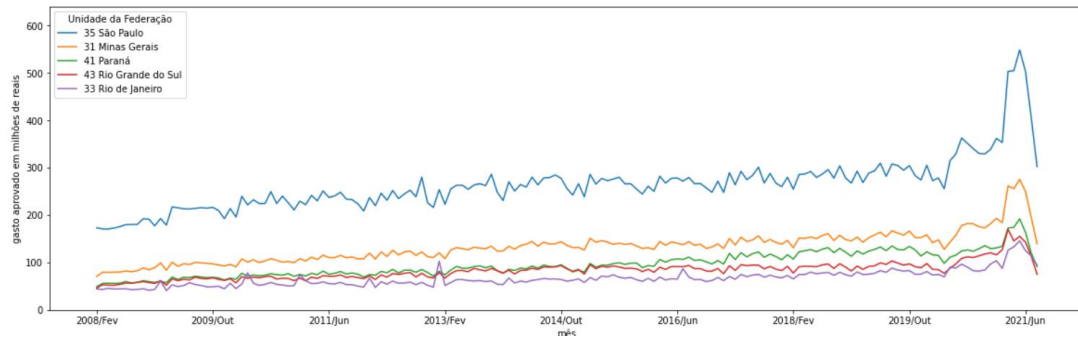


Figura 5 – Gráfico ajustado em x (3º)  
Fonte: Elaborado pelo autor (2023)

### Dica de leitura:

Uma ótima maneira de entender o que está ocorrendo nas funções de determinadas bibliotecas, é acessar a sua documentação. Veja a documentação do Pandas [Plot](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html) [<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html>](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html) para entender melhor e também o **plot()** do Matplotlib [<https://matplotlib.org/>](https://matplotlib.org/).

Ajustar as visualizações no Matplotlib é importante, porque permite que você controle a aparência e a estética dos gráficos produzidos. Isso inclui aspectos como títulos, legendas, rótulos de eixos, escala, cores e muito mais. Ajustar as visualizações ajuda a tornar os gráficos mais claros e fáceis de entender para o público-alvo, o que pode ser importante em contextos científicos ou de negócios. Além disso, o ajuste adequado das visualizações pode destacar informações relevantes e tornar os dados mais atraentes e impactantes.

Alguns dos recursos mais comumente usados no Matplotlib para ajustar as visualizações incluem:

- Títulos e legendas: para fornecer contexto e explicação para o gráfico.
- Rótulos de eixos: para identificar as escalas de valores no gráfico.
- Escala: para controlar a escala dos valores exibidos nos eixos.
- Cores: para destacar informações específicas ou para tornar o gráfico mais atraente.
- Marcadores: para identificar pontos específicos no gráfico.



- **Grade:** para tornar o gráfico mais claro e fácil de ler.
- **Estilos de linha:** para controlar o tipo de linha usado no gráfico, por exemplo, contínua, tracejada, pontilhada etc.

O Matplotlib é uma biblioteca poderosa e versátil para visualização de dados, mas pode ter limitações em algumas situações. Algumas dessas limitações incluem:

**Complexidade:** O Matplotlib pode ser complexo de ser usado para criar gráficos mais avançados ou interativos.

**Visualizações limitadas:** O Matplotlib tem suporte para uma ampla gama de tipos de gráficos, mas pode não ter suporte para algum tipo de visualização específica que você precise.

**Gráficos interativos:** O Matplotlib tem recursos limitados para criar gráficos interativos, que permitem ao usuário manipular o gráfico com o mouse ou outros dispositivos de entrada.

Nessas situações, pode ser necessário usar outras bibliotecas gráficas adicionais, como o Bokeh ou o Plotly, que oferecem recursos adicionais para criar gráficos interativos e mais avançados. No entanto, o Matplotlib é uma escolha sólida para a maioria das necessidades de visualização de dados e é frequentemente usado como uma base para outras bibliotecas gráficas.

## O QUE VOCÊ VIU NESTA AULA?

Você aprendeu como olhar e interpretar um gráfico de linhas realizando comparações, e como a manipulação dos gráficos pode transmitir uma informação falsa.

**IMPORTANTE:** não esqueça de praticar com o desafio da disciplina, para que assim você possa aprimorar os seus conhecimentos!

Você não está sozinho ou sozinha nesta jornada! Te esperamos no Discord e nas lives com os professores especialistas, onde você poderá tirar dúvidas, compartilhar conhecimentos e estabelecer conexões!

## REFERÊNCIAS

DATASUS. <<https://datasus.saude.gov.br/>>. Acesso em: 07 fev 2023.

DOCUMENTAÇÃO PANDAS. <<https://pandas.pydata.org/>>. Acesso em: 07 fev 2023.

GOOGLE COLAB. <<https://colab.research.google.com/>>. Acesso em: 07 fev 2023.

TABNET. <<https://datasus.saude.gov.br/informacoes-de-saude-tabnet/>>. Acesso em: 07 fev 2023.

## **PALAVRAS-CHAVE**

Python. Pandas. Dataframe.

EMAP

The background is a dark blue field filled with numerous small, light blue dots. Overlaid on this are several large, wavy, translucent lines in shades of blue, yellow, and red. These lines flow from the left side towards the right, creating a sense of motion. Scattered throughout the composition are various geometric shapes: a circle with the number '7' inside, a circle with an 'X' inside, a circle with an 'O' inside, and a hexagon in the bottom right corner. The overall aesthetic is futuristic and technological.

# POSTECH