

# Aula 03 | PosTech | Ticks, escalas e formação de imagem

Anotações sobre a segunda aula da PosTech FIAP ✨✨

<https://on.fiap.com.br/mod/conteudoshtml/view.php?id=307796&c=8729&sesskey=ZuKoJQwSR0>

## Temas abordados:

- Como criar um Scatterplot e Lineplot com o Seaborn;
- Como formatar ticks ajustando sua localização da forma mais adequada possível;
- Como criar um grid em suas visualizações para facilitar a análise.

## Pré-requisitos:

- Base de dados:

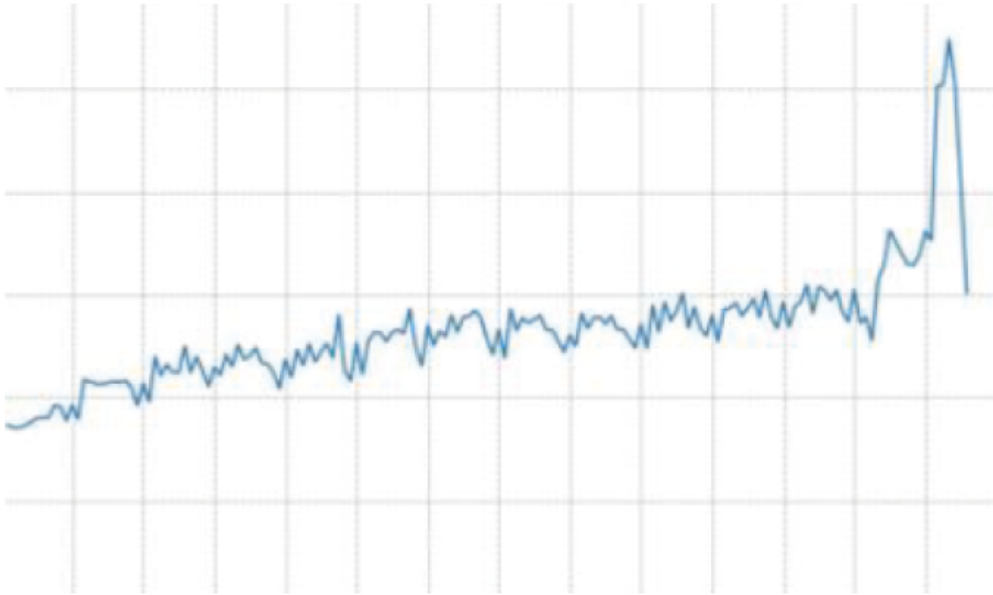
<https://github.com/alura-tech/pos-datascience-introducao-a-visualizacao/archive/refs/heads/dados.zip>

- Importar essa base de dados no Colaboratory

---

E se quisermos colocar um fundo de papel quadriculado em vez de branco? Com o Python podemos fazer isso! 😊

□ `plt.grid()`

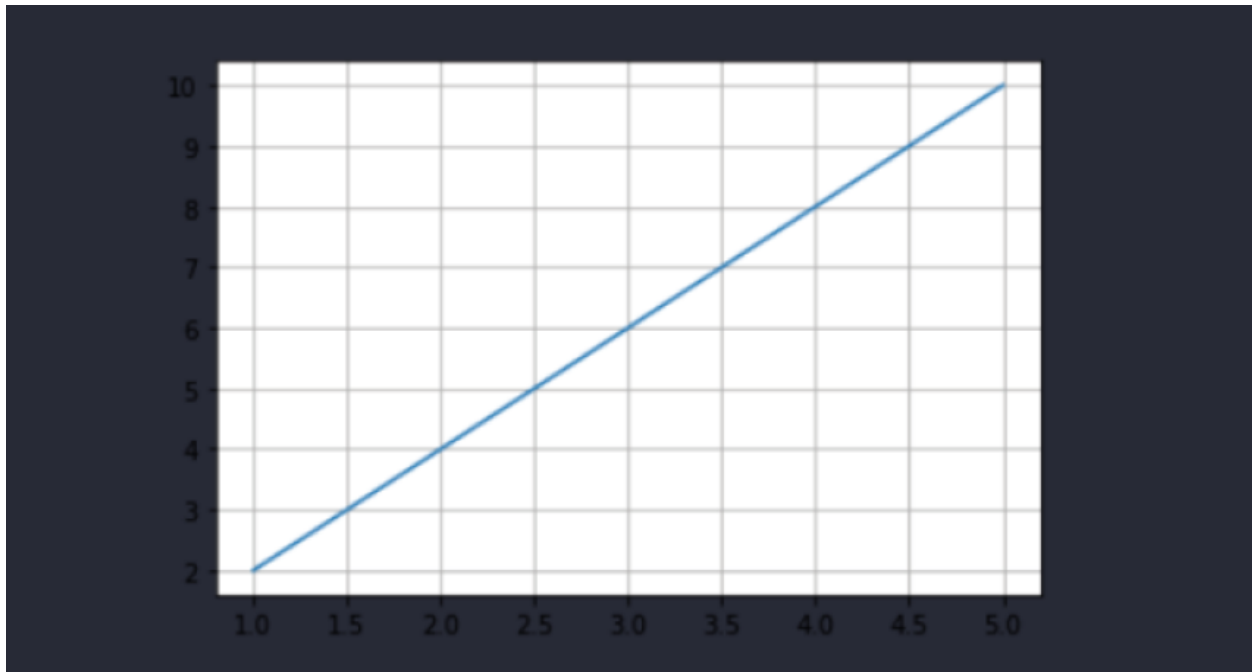


O `grid()` é uma função do Matplotlib que serve para desenhar uma grade no gráfico. A grade pode ser usada para ajudar na leitura dele, tornando mais fácil para o usuário identificar valores específicos em um eixo.

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

plt.plot(x, y)
plt.grid(True)
plt.show()
```



*Grid com código.*

Esse exemplo desenha uma grade sobre o gráfico de linha plotado. A opção `True` passada como argumento para `grid()` significa que a grade deve ser exibida no gráfico. Se você passar `False` como argumento, a grade não será exibida.

Além disso, podemos alterar rótulos utilizando `set_major_locator()` do Matplotlib, que é usada para definir a localização dos marcadores principais em um eixo. Os marcadores principais são valores numerais exibidos ao longo do eixo, que ajudam a identificar as posições dos dados no gráfico.

Por exemplo, você pode usar a função `set_major_locator` para exibir marcadores principais em intervalos regulares de tempo, como dias, meses ou anos, ou em valores numéricos regulares, como 1, 2, 3 etc.

Veja um caso utilizando Python:

```

import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import numpy as np
import datetime

dates = [datetime.datetime(2021, 1, 1),

datetime.datetime(2021, 2, 1),

datetime.datetime(2021, 3, 1),

datetime.datetime(2021, 4, 1),

datetime.datetime(2021, 5, 1),

datetime.datetime(2021, 6, 1),

datetime.datetime(2021, 7, 1)]

y = np.array([1, 4, 9, 16, 25, 36, 49])

plt.plot(dates, y)

# Define a localização dos marcadores principais como intervalos de 1 mês

ax = plt.gca()

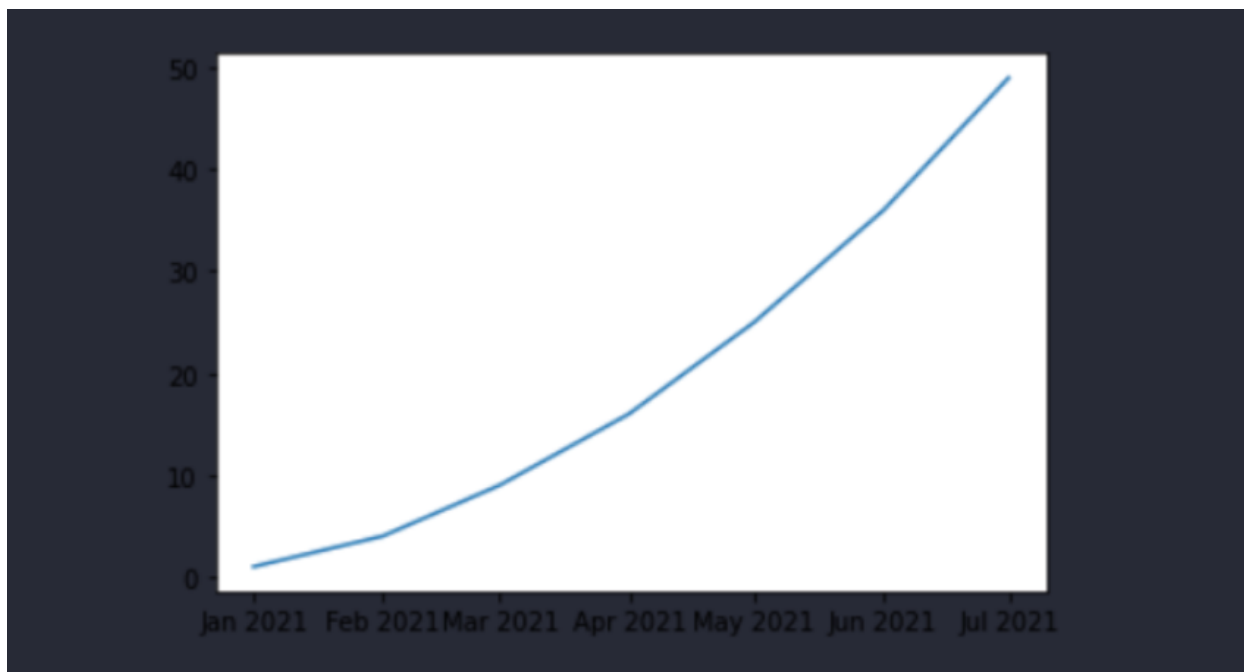
ax.xaxis.set_major_locator(mdates.MonthLocator())

# Formata os marcadores principais como meses e anos

ax.xaxis.set_major_formatter(mdates.DateFormatter('%b %Y'))

plt.show()

```



*Gráfico com ajuste de datas.*

Neste exemplo, a função `set_major_locator` é usada para **definir a localização dos marcadores principais**, como intervalos de 1 mês, usando o `MonthLocator` do Matplotlib. Além disso, a função `set_major_formatter` é usada para formatar os marcadores principais como meses e anos.

```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.ticker as ticker

x = np.arange(0, 10, 0.1)

y = x ** 2

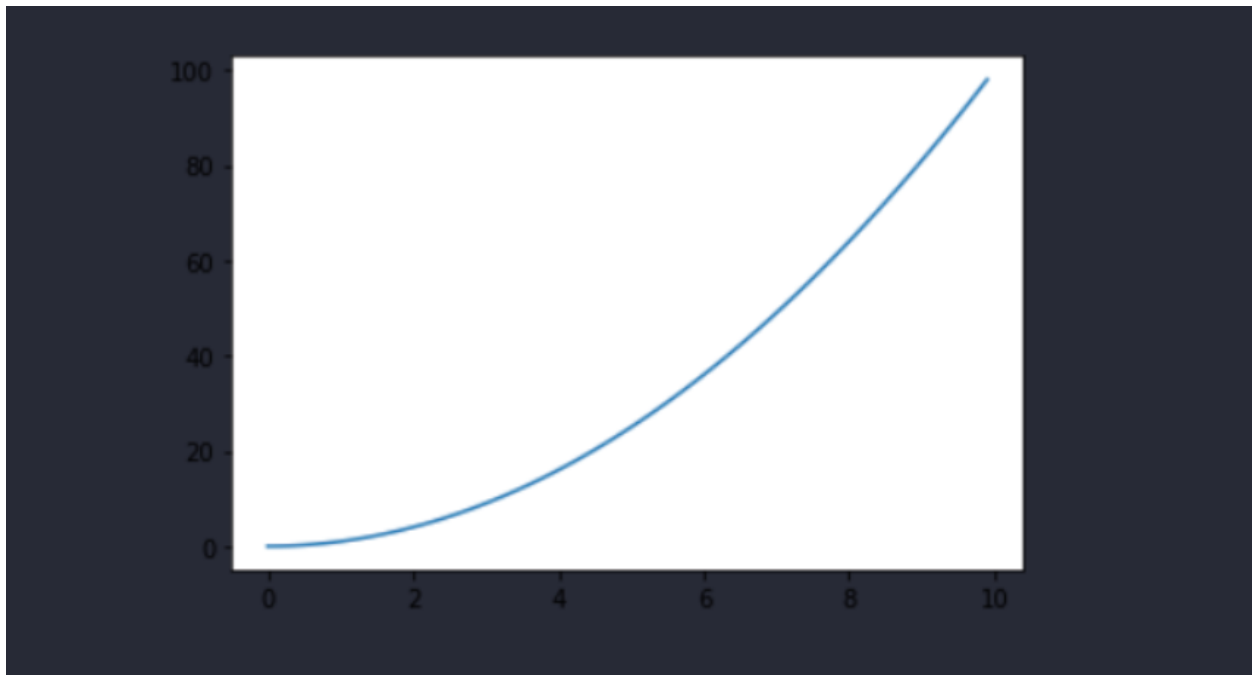
plt.plot(x, y)

# Define a localização dos marcadores principais como intervalos de 2

ax = plt.gca()
```

```
ax.xaxis.set_major_locator(ticker.MultipleLocator(2))

plt.show()
```



*Outro exemplo de gráfico.*

Neste caso, a função `set_major_locator` é usada para **definir a localização dos marcadores principais como intervalos de 2**, usando o `MultipleLocator` do Matplotlib. Isso significa que os **marcadores principais serão exibidos a cada 2 unidades no eixo x**.

## Parte 1 | Ticks, escalas e formatação de imagem

Agora vamos pegar nossos dados “ordenados\_por\_total”

☐ `ordenados_por_total.head()`



The screenshot shows a Jupyter Notebook titled "Produção Hospitalar". The table displays hospital production data for five states (São Paulo, Minas Gerais, Paraná, Rio Grande do Sul, and Rio de Janeiro) across 12 months (February 2008 to January 2009). The values represent production levels, with São Paulo having the highest values and Rio de Janeiro the lowest.

	2008/Fev	2008/Mar	2008/Abr	2008/Mai	2008/Jun	2008/Jul	2008/Ago	2008/Set	2008/Out	2008/Nov	2008/Dez	2009/Ja
Unidade da Federação												
São Paulo	173.06	170.62	170.39	172.51	175.56	179.61	180.10	180.21	192.00	191.26	177.10	192.4
Minas Gerais	70.30	79.12	79.05	79.27	79.63	82.22	80.37	82.42	88.61	84.49	88.93	99.0
Paraná	48.86	55.61	56.12	55.57	56.09	59.80	56.70	58.09	61.37	59.67	57.82	60.3
Rio Grande do Sul	45.90	52.93	51.97	51.68	53.76	56.39	56.23	57.91	59.59	57.57	55.70	61.5
Rio de Janeiro	44.13	42.91	45.32	43.95	44.12	44.36	42.50	43.07	44.55	41.35	43.04	60.8

5 rows x 163 columns

Agora, eu quero um gráfico de um estado específico. Vou precisar buscar esses dados, portanto.

☐ `ordenados_por_total["Sao Paulo"]`

Mas eu quero buscar pelo índice São Paulo:

☐ `ordenados_por_total.loc["Sao Paulo"]`



```
1 ordenados_por_total.loc["São Paulo"]
```



```
2008/Fev    173.06
2008/Mar    170.62
2008/Abr    170.39
2008/Mai    172.51
2008/Jun    175.56
...
2021/Abr    504.43
2021/Mai    548.11
2021/Jun    503.16
2021/Jul    404.37
2021/Ago    301.99
Name: São Paulo, Length: 163, dtype: float64
```

Aqui ele me trouxe uma série (uma sequência), não uma tabela. Eu quero plotar isso, então:



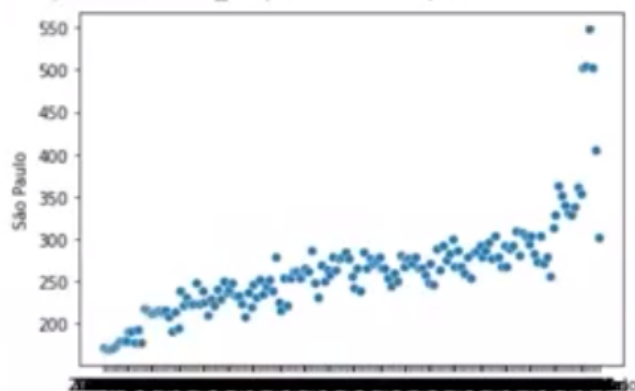
```
sns.scatterplot(x=ordenados_por_total.columns, y=ordenados_por_total.loc["Sao Paulo"])
```



```
1 sns.scatterplot(x=ordenados_por_total.columns, y=ordenados_por_total.loc["São Paulo"])
```



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3b25ee5cd0>
```





Como de costume, gráfico péssimo, não dá pra ver nada! Vamos agora tentar simplificar.

Podemos TRANSPOR as informações da seguinte forma:

☐ `mensal = ordenados_por_total.T`

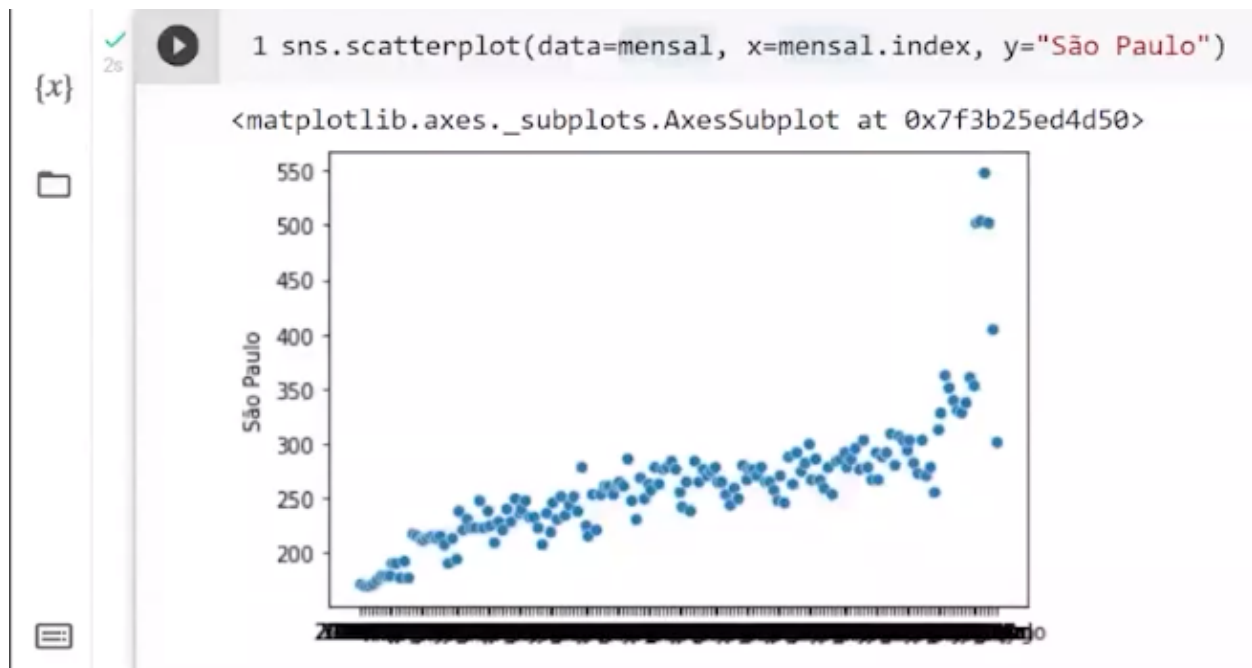
☐ `mensal.head()`

```
[104] 1 mensal = ordenados_por_total.T
      2 mensal.head()
```

Unidade da Federação	São Paulo	Minas Gerais	Paraná	Rio Grande do Sul	Rio de Janeiro	Bahia	Pernambuco	Santa Catarina	Ceará	Goiás	Pará	Maranhão	Espírito Santo	Rio Grande do Norte
2008/Fev	173.06	70.30	48.86	45.90	44.13	33.41	26.22	24.41	27.64	18.15	20.13	13.45	10.03	9.11
2008/Mar	170.62	79.12	55.61	52.93	42.91	41.33	30.72	27.48	30.09	20.99	22.15	14.75	12.26	10.13
2008/Abr	170.39	79.05	56.12	51.97	45.32	42.83	30.59	27.81	31.38	21.05	23.44	15.15	12.45	10.96
2008/Mai	172.51	79.27	55.57	51.68	43.95	42.10	31.42	28.35	31.19	19.16	23.86	14.54	11.79	11.03
2008/Jun	175.56	79.63	56.09	53.76	44.12	39.91	28.95	28.99	29.46	20.39	23.50	14.55	11.89	10.45

Agora, temos a coluna “São Paulo”. Além de mais legível fica também mais natural.

☐ `sns.scatterplot(data=mensal, x=mensal.index, y="São Paulo")`



A partir de agora, se quiser imprimir de outro estado é só imprimir. 😊

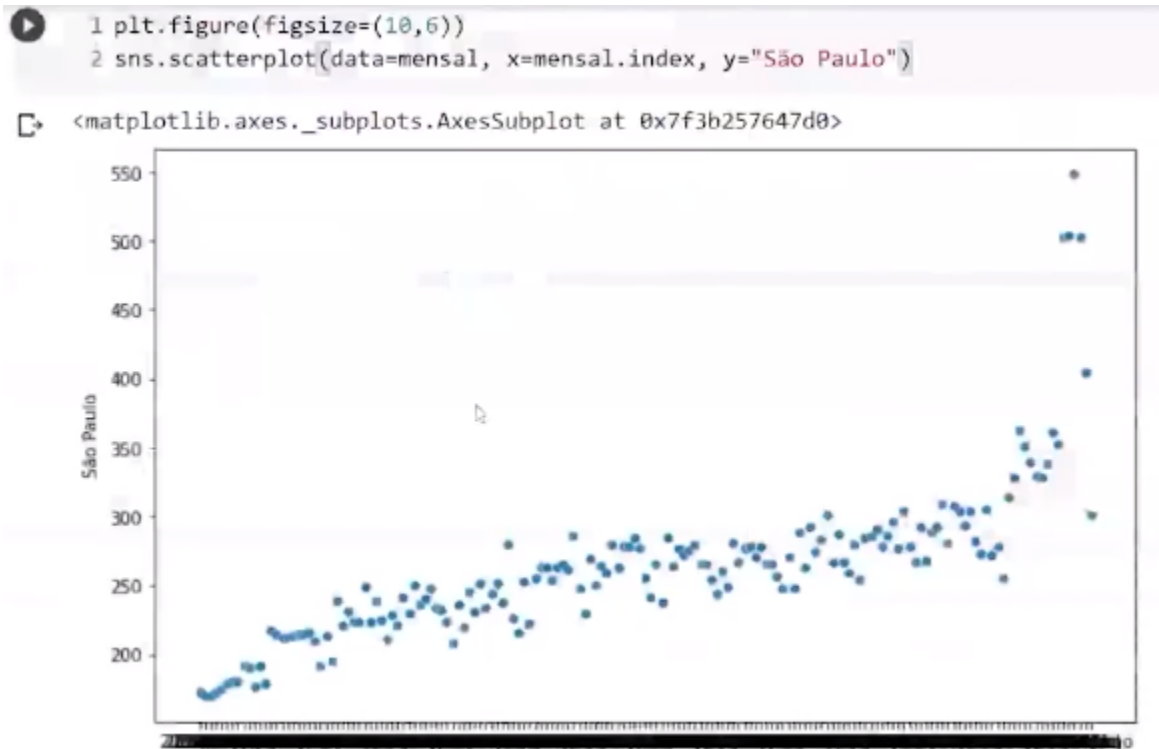
## DESAFIO 01: fazer gráfico com dois estados, uma cor diferente para cada estado.

Agora vamos melhorar a legibilidade desse gráfico...

Primeiro, vamos mudar o tamanho dessa figura:

☐ `plt.figure(figsize=(10,6))` # não há motivo para esse valor

☐ `sns.scatterplot(data=mensal, x=mensal.index, y="São Paulo")`

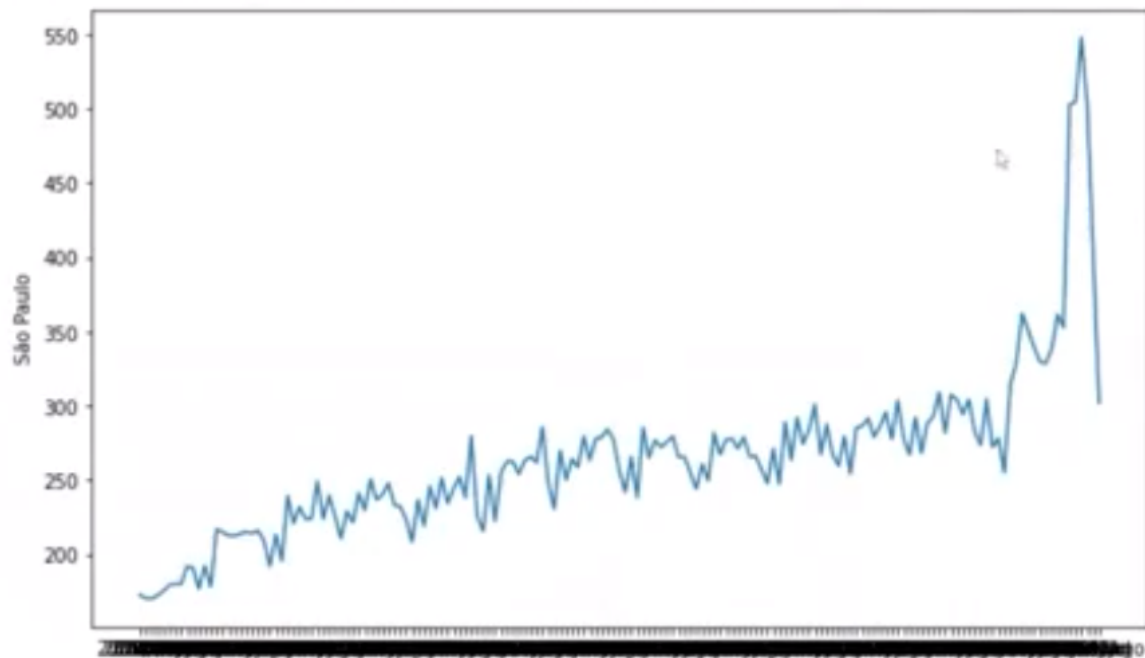


Há um problema no tamanho dos pontos e no uso de pontos, porque entre 1º de Janeiro e 30 de janeiro, há 30 dias de diferença. Há algo acontecendo nesse período, portanto. É muito comum quando temos o **tipo de gráfico em que o eixo x é uma série temporal**.

Em vez de scatterplot, vamos usar **lineplot**:

- ☐ `plt.figure(figsize=(10,6))` # não há motivo para esse valor
- ☐ `sns.lineplot(data=mensal, x=mensal.index, y="São Paulo")`

☞ <matplotlib.axes.\_subplots.AxesSubplot at 0x7f3b254d4bd0>



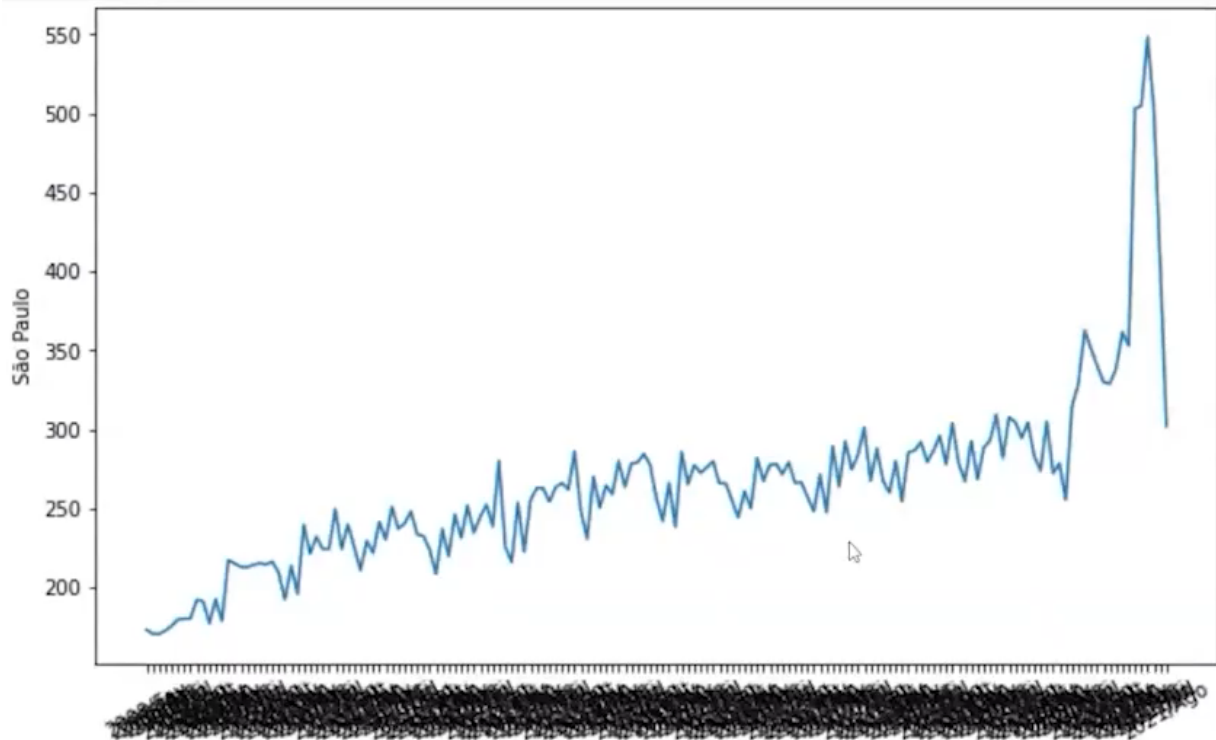
Vamos ajustar algumas coisas aqui. Por exemplo, os ticks do eixo x:

☐ `plt.figure(figsize=(10,6))` # não há motivo para esse valor

☐ `sns.lineplot(data=mensal, x=mensal.index, y="São Paulo")`

☐ `plt.xticks(rotation=30)`

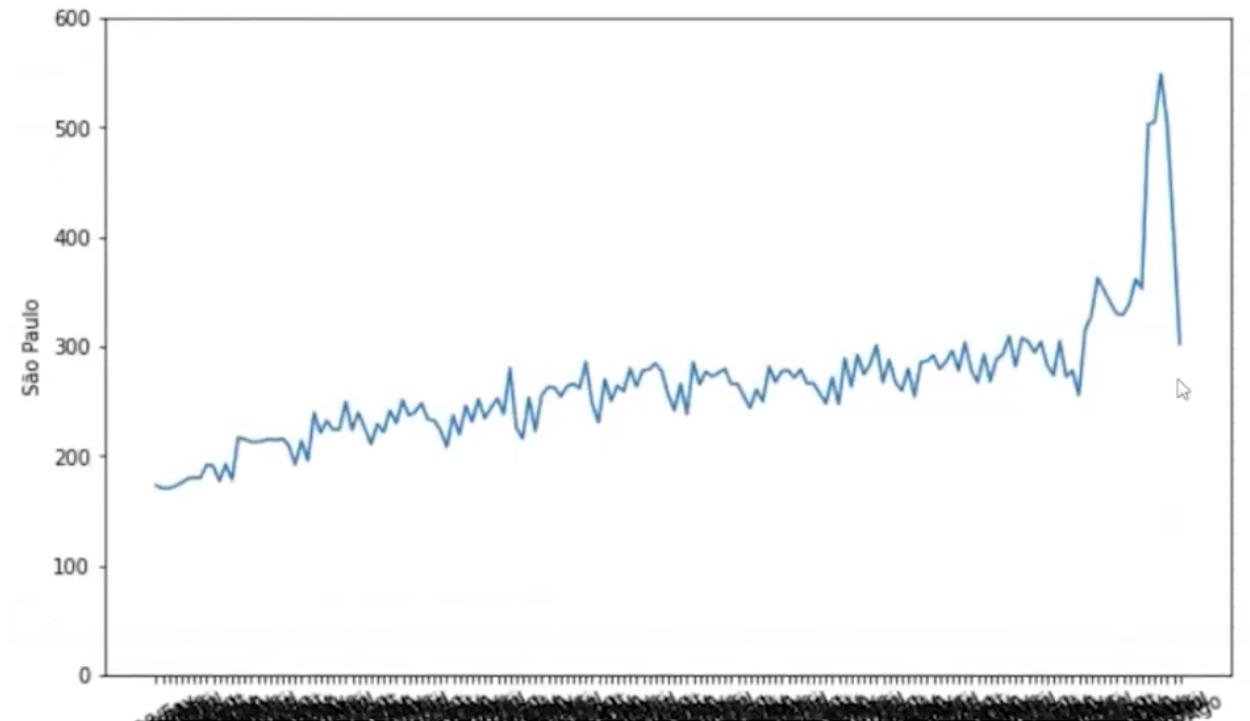
☐ `plt.show`



Ainda está péssimo, mas está mais legível e vamos arrumar. kkkkkk

O eixo y não começa do zero!

- ☐ `plt.figure(figsize=(10,6))` # não há motivo para esse valor
- ☐ `sns.lineplot(data=mensal, x=mensal.index, y="São Paulo")`
- ☐ `plt.xticks(rotation=30)`
- ☐ `plt.ylim(0, 600)`
- ☐ `plt.show`



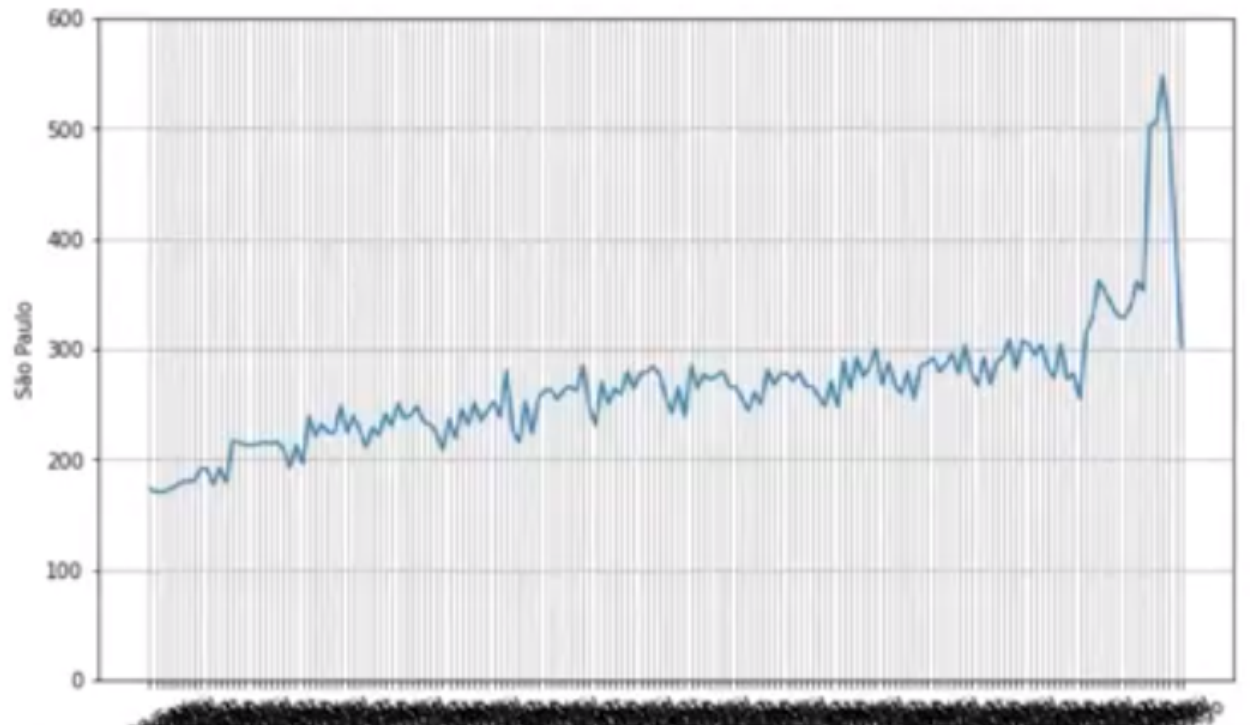
Certo, mas e se esse valor passar dos 600, o que vai acontecer?

**DESAFIO 02:** no gráfico acima, utilizar um valor máximo do eixo y que caiba todos os valores da nossa série. Além disso, precisa de um “respiro” (white space).

Vamos colocar agora o grid...

- ☐ `plt.figure(figsize=(10,6))` # não há motivo para esse valor
- ☐ `sns.lineplot(data=mensal, x=mensal.index, y="São Paulo")`
- ☐ `plt.xticks(rotation=30)`
- ☐ `plt.ylim(0, 600)`

☐ `plt.show()`



☐ `plt.figure(figsize=(10,6))` # não há motivo para esse valor

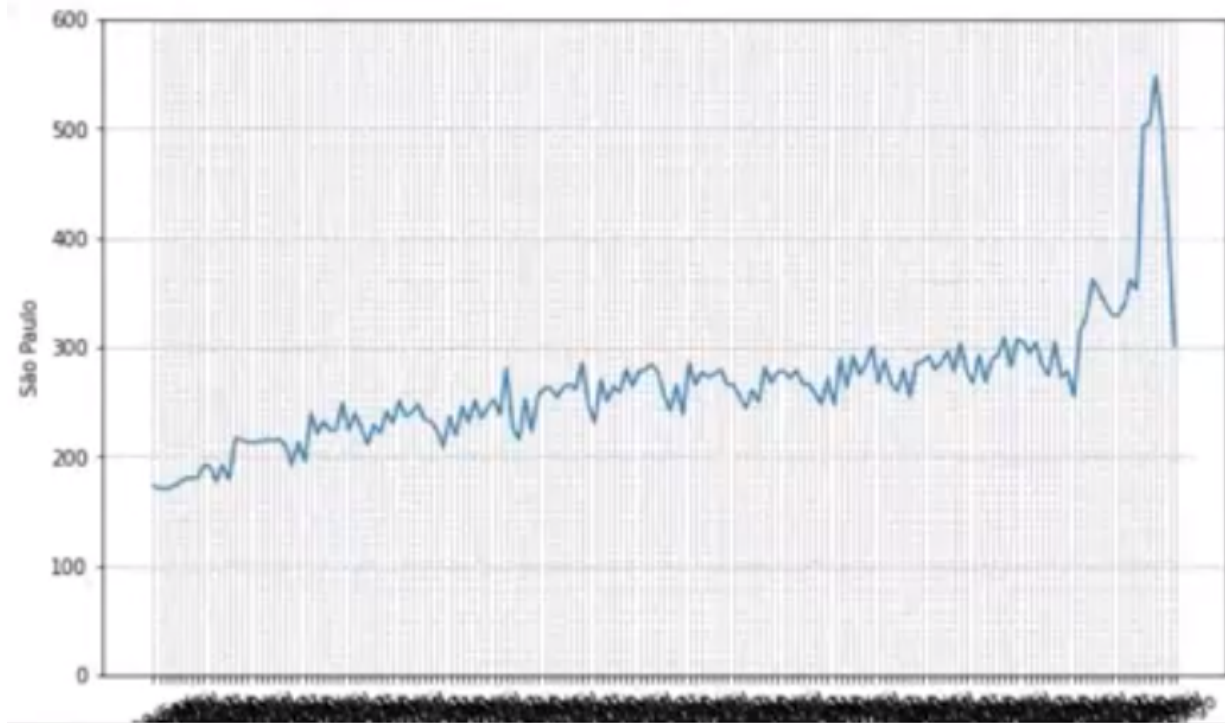
☐ `sns.lineplot(data=mensal, x=mensal.index, y="São Paulo")`

☐ `plt.xticks(rotation=30)`

☐ `plt.ylim(0, 600)`

☐ `plt.grid(linestyle="--")` # para ficar um tracejado mais levinho

☐ `plt.show()`



Mas o ticks não deveria estar a cada mês, não fica legível. Não vou colocar uma linha e escrever o valor mensal. Agora, quando eu chamar a lineplot, vou atribuir um valor da variável à x (x vem de axis, lembra?).

- ☐ `plt.figure(figsize=(10,6))` # não há motivo para esse valor
- ☐ `axis = sns.lineplot(data=mensal, x=mensal.index, y="São Paulo")`
- ☐ `plt.xticks(rotation=30)`
- ☐ `plt.ylim(0, 600)`
- ☐ `axis.xaxis.set_major_locator(ticker.IndexLocator(base=12, offset=0))`
- ☐ `plt.grid(linestyle="--")` # para ficar um tracejado mais levinho
- ☐ `plt.show()`

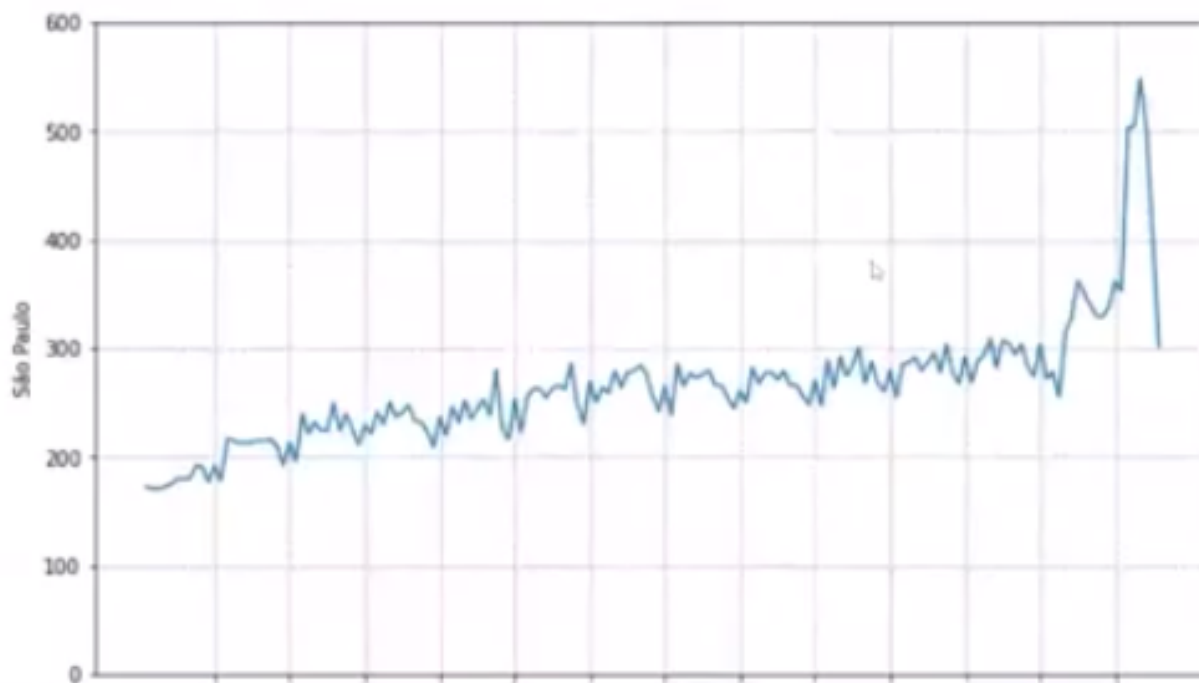
Ticker locator Matplotlib → queremos algo com **intervalos fixos**, a cada 12 meses (uma vez por ano),



[https://matplotlib.org/3.1.1/gallery/ticks\\_and\\_spines/tick-locators.html](https://matplotlib.org/3.1.1/gallery/ticks_and_spines/tick-locators.html)



- ☐ `plt.figure(figsize=(10,6))` # não há motivo para esse valor
- ☐ `axis = sns.lineplot(data=mensal, x=mensal.index, y="São Paulo")`
- ☐ `plt.xticks(rotation=30)`
- ☐ `plt.ylim(0, 600)`
- ☐ `axis.xaxis.set_major_locator(ticker.IndexLocator(base=12, offset=10))` # aqui posso chutar 10
- ☐ `plt.grid(linestyle="--")` # para ficar um tracejado mais levinho
- ☐ `plt.show()`



O legal é que a cada espacinho do grid, temos um ano inteiro. Podemos fazer comparações a partir disso.

**DESAFIO 03:** parece que em Fevereiro o valor sempre cai. Qual seria o possível motivo? Qual a causa da queda, talvez?

**DESAFIO 04:** colocar um minor locator no mês de Julho.