

Aula 05 | PosTech | Análise Exploratória de Dados

Anotações sobre a quarta aula da PosTech FIAP ✨✨

<https://on.fiap.com.br/mod/conteudoshtml/view.php?id=307787&c=8729&sesskey=AlUOg2UtXh>

Temas abordados:

- Qual é o melhor tipo de gráfico para representar o que eu quero?

Dependências:

- Baixar o arquivo zip:

<https://github.com/alura-tech/pos-datascience-analise-e-exploracao-de-dados/tree/aula1>

- Documentação Pandas:

<https://pandas.pydata.org/>

- Documentação Matplotlib:

<https://matplotlib.org/>

- Google Colab:

<https://colab.research.google.com/>

- TabNet:

<https://datasus.saude.gov.br/informacoes-de-saude-tabnet/>


- Jupiter Notebook completo:

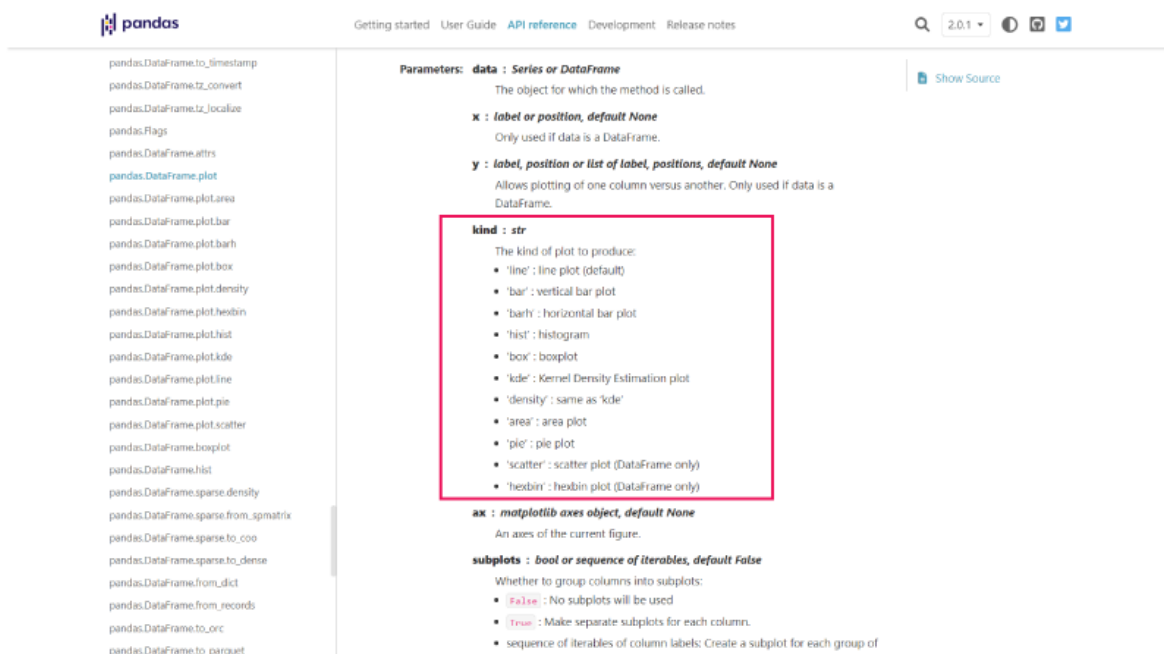
https://github.com/alura-tech/pos-datascience-analise-e-exploracao-de-dados/blob/aula5/Produção_Hospitalar-51.ipynb

Aula 5 - Manipulação e interpretação de gráficos

Para entendermos qual gráfico utilizar, podemos ir nesse endereço:

`pandas.DataFrame.plot` — pandas 2.0.2 documentation

 <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html>



Parameters:

- data** : *Series or DataFrame*
The object for which the method is called.
- x** : *label or position, default None*
Only used if data is a DataFrame.
- y** : *label, position or list of label, positions, default None*
Allows plotting of one column versus another. Only used if data is a DataFrame.
- kind** : *str*
The kind of plot to produce:
 - 'line': line plot (default)
 - 'bar': vertical bar plot
 - 'barh': horizontal bar plot
 - 'hist': histogram
 - 'box': boxplot
 - 'kde': Kernel Density Estimation plot
 - 'density': same as 'kde'
 - 'area': area plot
 - 'pie': pie plot
 - 'scatter': scatter plot (DataFrame only)
 - 'hexbin': hexbin plot (DataFrame only)
- ax** : *matplotlib axes object, default None*
An axes of the current figure.
- subplots** : *bool or sequence of iterables, default False*
Whether to group columns into subplots:
 - **False**: No subplots will be used
 - **True**: Make separate subplots for each column.
 - sequence of iterables of column labels: Create a subplot for each group of

Documentação `.plot()` do pandas 😊

No parâmetro `kind`, temos alguns tipos de gráficos e, para cada um deles, existe uma melhor escolha na hora de decidir qual utilizar. Exemplo, se tiver poucas categorias onde quer representar a **porcentagem em relação ao todo**, o ideal é o **“pie”**, mais conhecido como “gráfico de setor”. Se quer **analisar uma série temporal**, o ideal é que utilize o **“line”**, que já é um parâmetro default, ou seja, não precisa passar esse parâmetro.

Além do tipo de gráfico, temos uma série de **parâmetros que ajudam a modificar o gráfico**. Só tome cuidado, porque certas modificações precisarão de bibliotecas gráficas, como o Matplotlib.

Gráficos mais comuns com a função `.plot()` do Pandas:

1. **Line Plot (Gráfico de Linha)**: exibir tendências em dados de séries temporais;
2. **Bar Plot (Gráfico de Barras)**: comparar valores categóricos;
3. **Histogram (Histograma)**: visualizar a distribuição de frequência de dados numéricos;
4. **Pie Chart (Gráfico de Pizza)**: representar a proporção de cada categoria em relação ao todo;
5. **Scatter Plot (Gráfico de Dispersão)**: visualizar relações entre dois conjuntos de dados numéricos;

Exemplos de código em Python que gerem esses gráficos:

```
import pandas as pd
import matplotlib.pyplot as plt

data = {'Year': [2010, 2011, 2012, 2013, 2014],
        'Sales': [1000, 1100, 1050, 1075, 1150]}
```

```
df = pd.DataFrame(data)

df.plot(x='Year', y='Sales', kind='line')

plt.show()
```



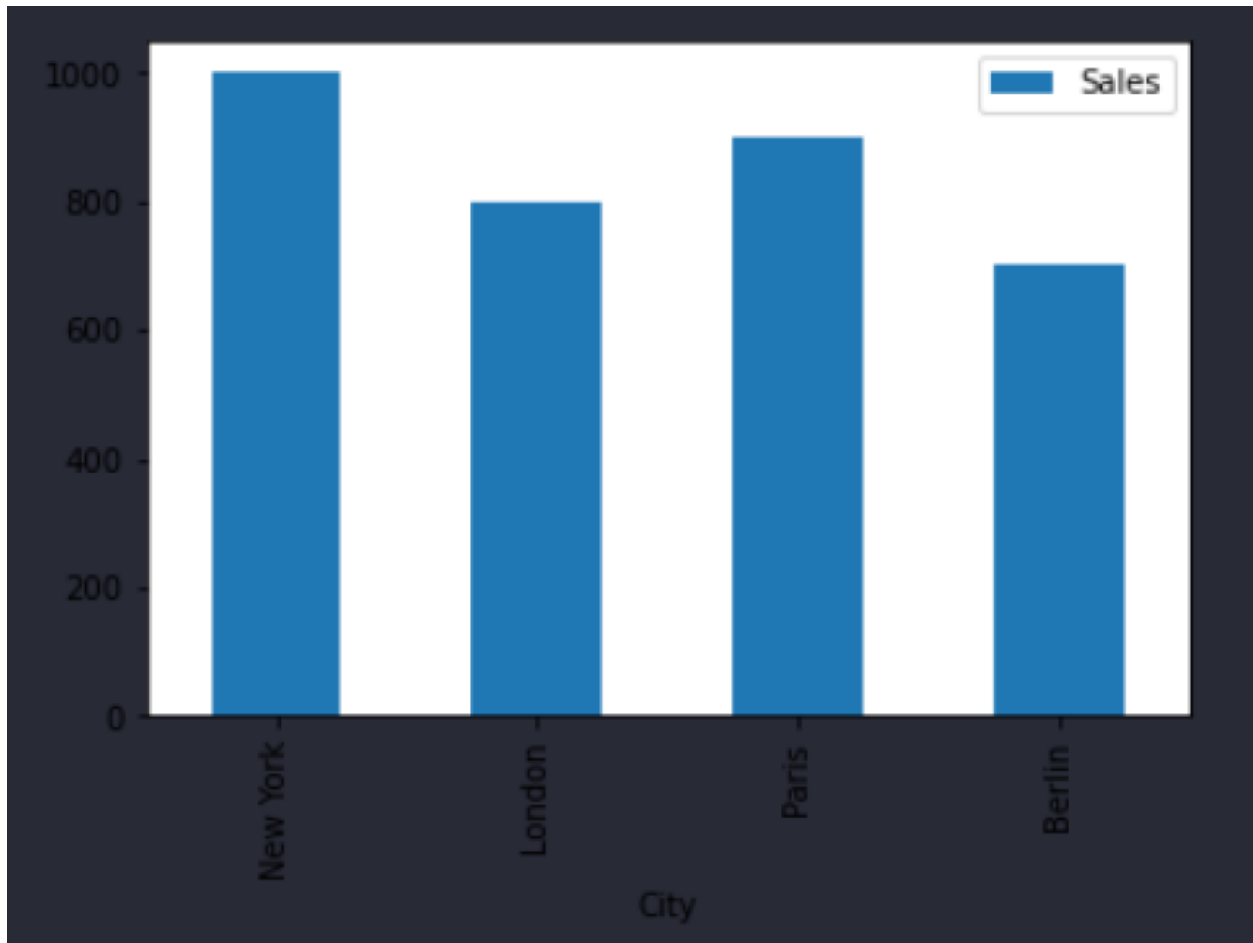
Resultado: Gráfico de Linha.

```
import pandas as pd
import matplotlib.pyplot as plt

data = {'City': ['New York', 'London', 'Paris', 'Berlin'],
        'Sales': [1000, 800, 900, 700]}

df = pd.DataFrame(data)
```

```
df.plot(x='City', y='Sales', kind='bar')  
  
plt.show()
```

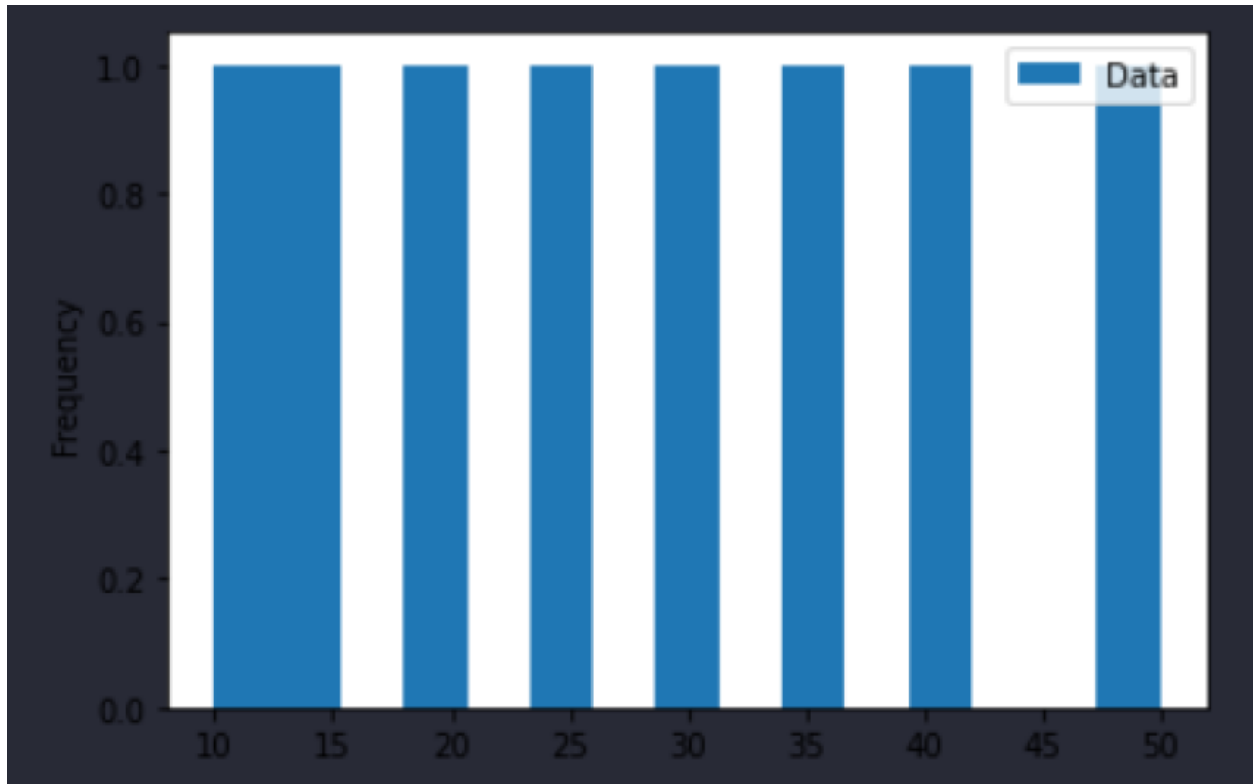


Resultado: Gráfico de Barras.

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
data = [10, 20, 15, 30, 25, 40, 35, 50]  
  
df = pd.DataFrame(data, columns=['Data'])
```

```
df.plot(kind='hist', bins=15)

plt.show()
```



Resultado: Histograma.

```
import pandas as pd
import matplotlib.pyplot as plt

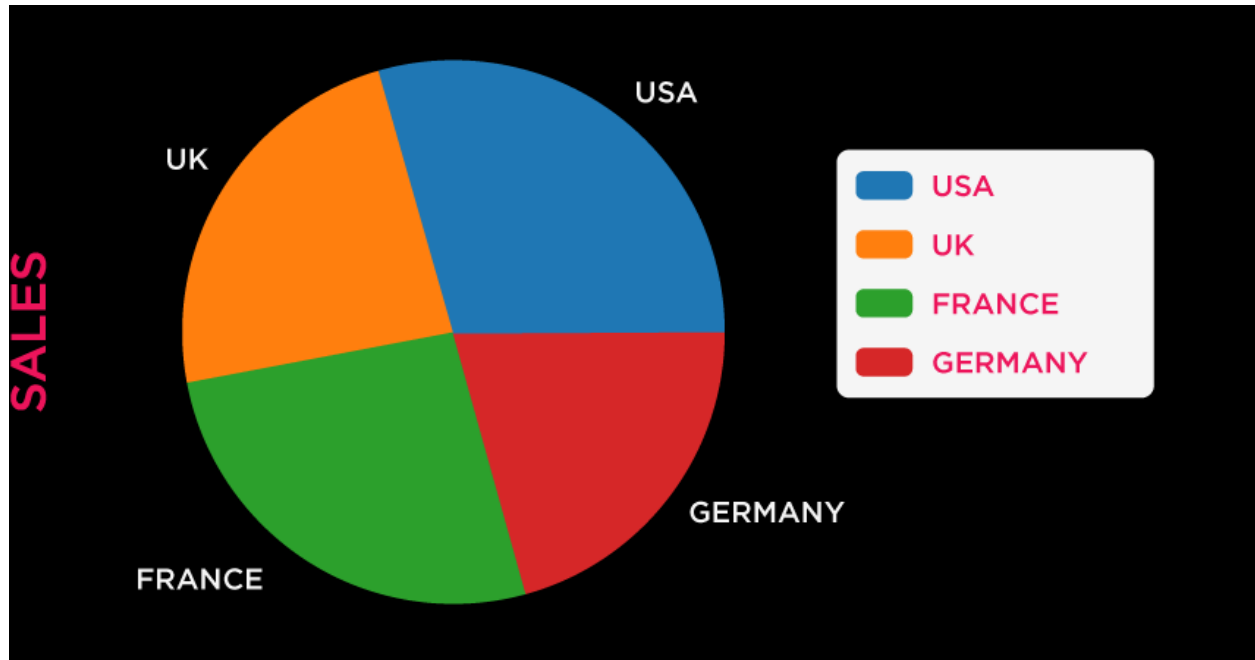
data = {'Country': ['USA', 'UK', 'France', 'Germany'],
        'Sales': [1000, 800, 900, 700]}

df = pd.DataFrame(data)

df.plot(kind='pie', y='Sales', labels=df['Country'])

plt.axis('equal')
```

```
plt.show()
```



Resultado: Gráfico de Pizza.

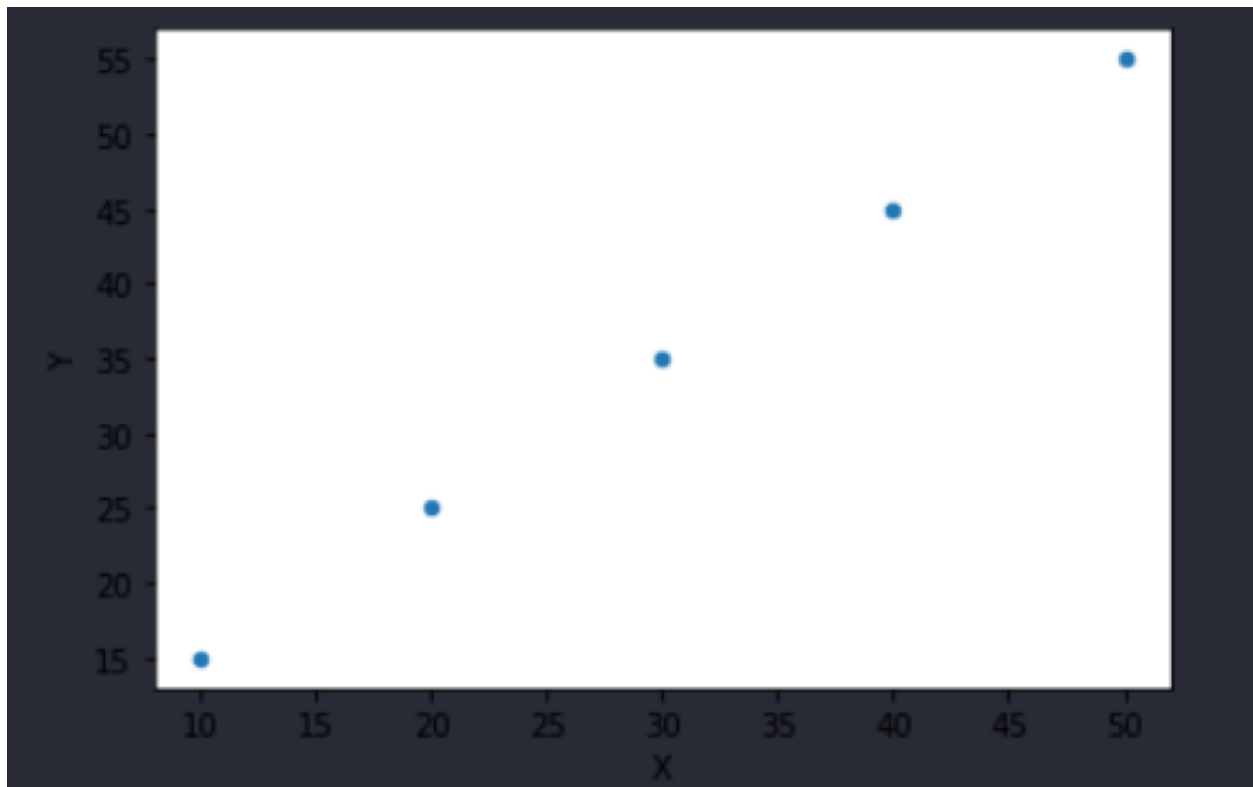
```
import pandas as pd
import matplotlib.pyplot as plt

data = {'X': [10, 20, 30, 40, 50],
        'Y': [15, 25, 35, 45, 55]}

df = pd.DataFrame(data)

df.plot(x='X', y='Y', kind='scatter')

plt.show()
```



Resultado: Scatter Plot.

1 | MANIPULAÇÃO E INTERPRETAÇÃO DE GRÁFICOS

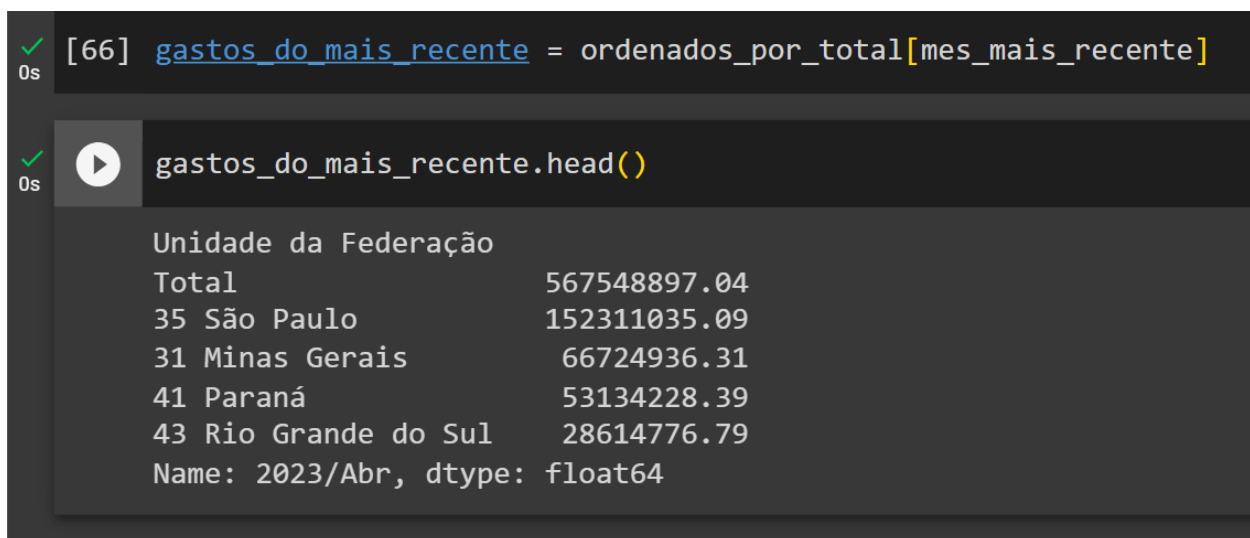
☐ `mes_mais_recente = ordenados_por_total.columns[-1]`

☐ `mes_mais_recente`

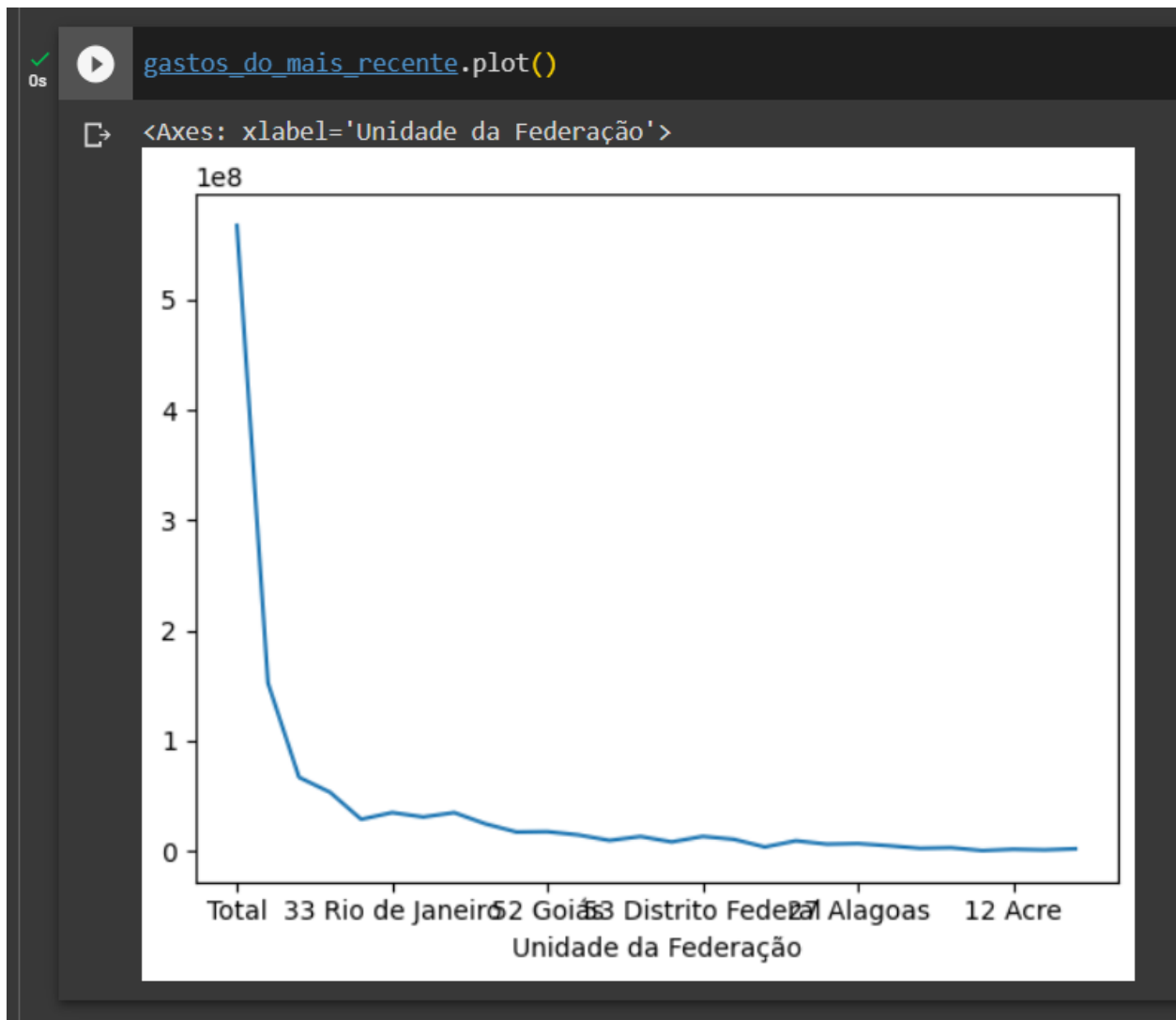


☐ `gastos_do_mais_recente = ordenados_por_total[mes_mais_recente]`

☐ `gastos_do_mais_recente.head()`




☐ `gastos_do_mais_recente.plot()`



Esse gráfico não faz sentido, então procuramos por “pandas plot kind”:

pandas.DataFrame.plot — pandas 2.0.2 documentation

 <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html>

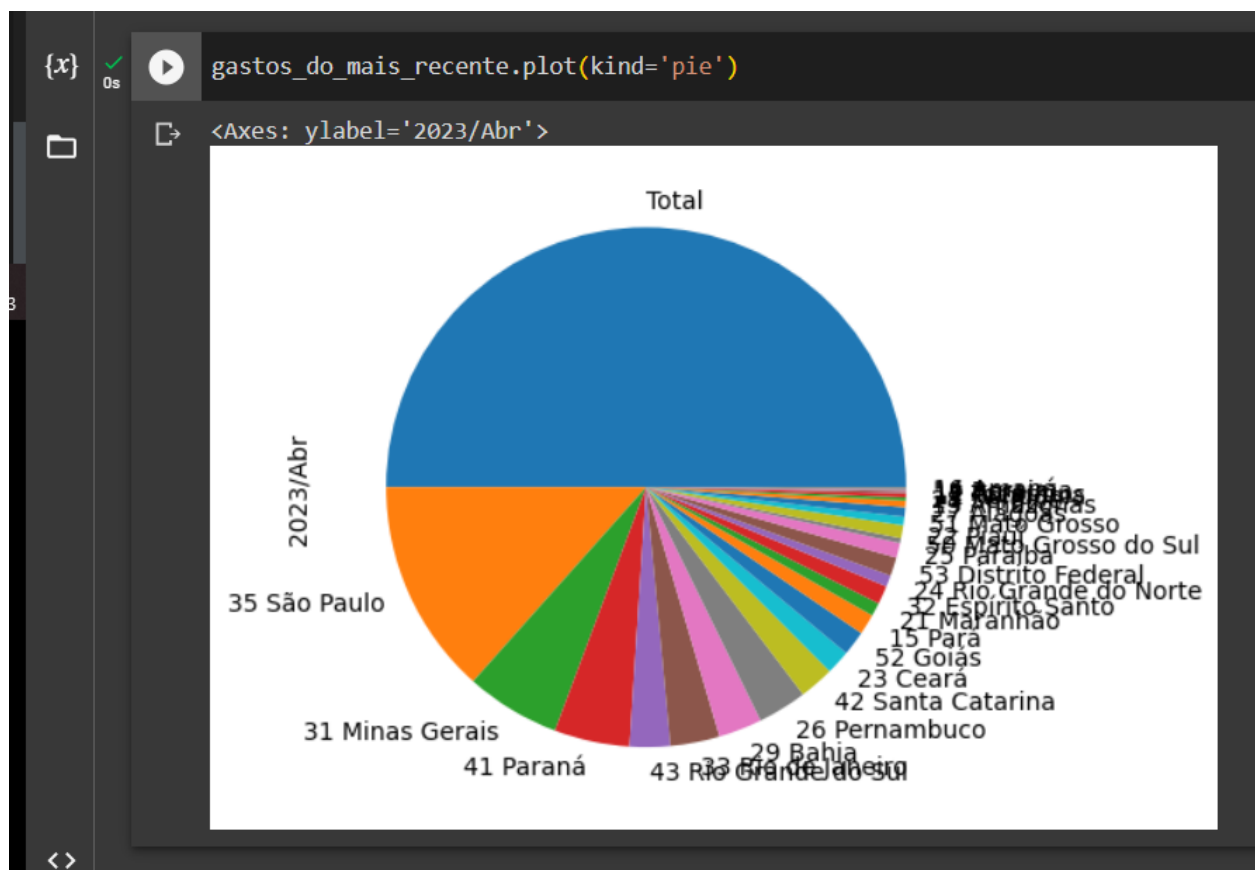
kindstr

The kind of plot to produce:

- ‘line’ : line plot (default)

- 'bar' : vertical bar plot
- 'barh' : horizontal bar plot
- 'hist' : histogram
- 'box' : boxplot
- 'kde' : Kernel Density Estimation plot
- 'density' : same as 'kde'
- 'area' : area plot
- 'pie' : pie plot
- 'scatter' : scatter plot (DataFrame only)
- 'hexbin' : hexbin plot (DataFrame only)

☐ `gastos_do_mais_recente.plot(kind='pie')`



Esse gráfico está horrível hahahah nenhuma conclusão. “Se o gráfico tiver nome de comida, não é para usar esse gráfico”. Tem muito preconceito sobre gráfico de pizza hahahahaha

- 1 Difícil de identificar elementos;
- 2 Número de elementos é enorme
- 3 Cores chamam mais atenção e parece ser uma fatia maior (dimensionalidade de um gráfico);
- 4 Qual o maior e o menor? Qual é maior que qual?
- 5 Qual a diferença de proporcionalidade?

→ Não é uma questão apenas de arrumar legenda, não é comparável.

Desafio 01: esse gráfico está ordenado ou não? Qual o motivo disso ter acontecido?

Não dá para saber a ordem de forma alguma apenas olhando o gráfico. Estamos sendo iludidos em relação às proporções.

☐ gastos_do_mais_recente



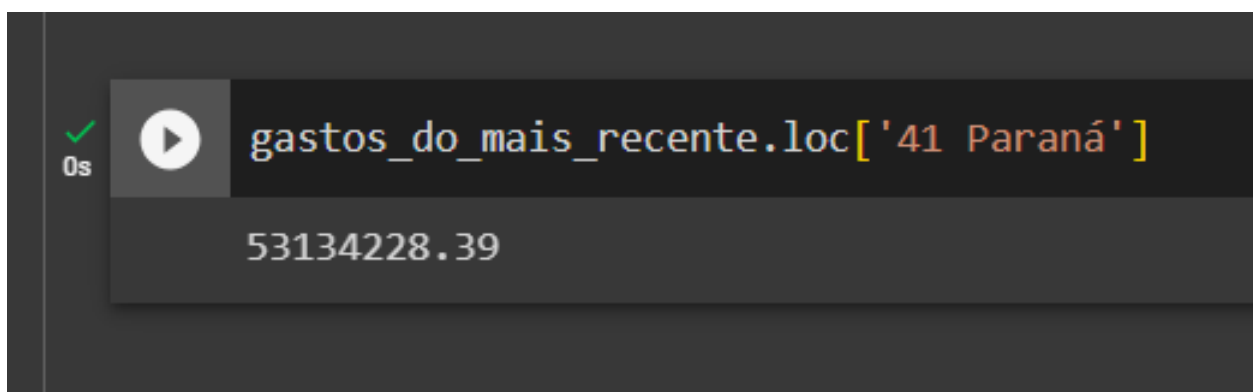
Unidade da Federação	
Total	567548897.04
35 São Paulo	152311035.09
31 Minas Gerais	66724936.31
41 Paraná	53134228.39
43 Rio Grande do Sul	28614776.79
33 Rio de Janeiro	34715797.10
29 Bahia	30716940.30
26 Pernambuco	34676781.08
42 Santa Catarina	24507992.14
23 Ceará	17094223.01
52 Goiás	17394131.69
15 Pará	14416398.87
21 Maranhão	9437097.82
32 Espírito Santo	13025442.16
24 Rio Grande do Norte	8058974.26
53 Distrito Federal	13099103.44
25 Paraíba	10457707.79
50 Mato Grosso do Sul	3464911.18
22 Piauí	9105939.36
51 Mato Grosso	6042972.96
27 Alagoas	6570640.67
13 Amazonas	4607011.87
28 Sergipe	2274803.87
11 Rorônia	2809110.10
17 Tocantins	226111.49
12 Acre	1401760.07
14 Roraima	836884.29
16 Amapá	1823184.04
Fonte: Ministério da Saúde - Sistema de Informações Hospitalares do SUS (SIH/SUS)	NaN
Notas:	NaN
Dados referentes aos últimos seis meses, sujeitos a atualização.	NaN
A partir do processamento de junho de 2012, houve mudança na classificação da natureza e esfera dos estabelecimentos. Com isso, temos que:	NaN

Agora, conseguimos responder à pergunta acima. Não está ordenado, está QUASE ordenado.

O motivo: lá atrás, fizemos a filtragem e depois ordenamos as informações.

São Paulo gastou 3x mais que o Paraná nesse mês mais recente.

☐ `gastos_do_mais_recente.loc["41 Parana"]`



☐ `gastos_do_mais_recente / gastos_do_mais_recente.loc["41 Paraná"]`

```
1 gastos_do_mais_recente / gastos_do_mais_recente.loc["41 Paraná"]
```

```
Unidade da Federação
35 São Paulo          3.31
31 Minas Gerais       1.53
41 Paraná             1.00
43 Rio Grande do Sul  0.82
33 Rio de Janeiro     1.03
29 Bahia              0.68
26 Pernambuco         0.79
42 Santa Catarina     0.54
23 Ceará              0.39
52 Goiás              0.49
15 Pará               0.23
21 Maranhão           0.28
32 Espírito Santo     0.30
24 Rio Grande do Norte 0.18
25 Paraíba            0.18
53 Distrito Federal   0.24
50 Mato Grosso do Sul 0.12
```

✓ 0s completed at 3:45 PM

Essa tabela aqui é muito mais valiosa para poder fazer uma comparação de gastos em estado (absoluto, não per capita).

Tendo essa tabela em mãos, vou pegar os 5 primeiros estados:

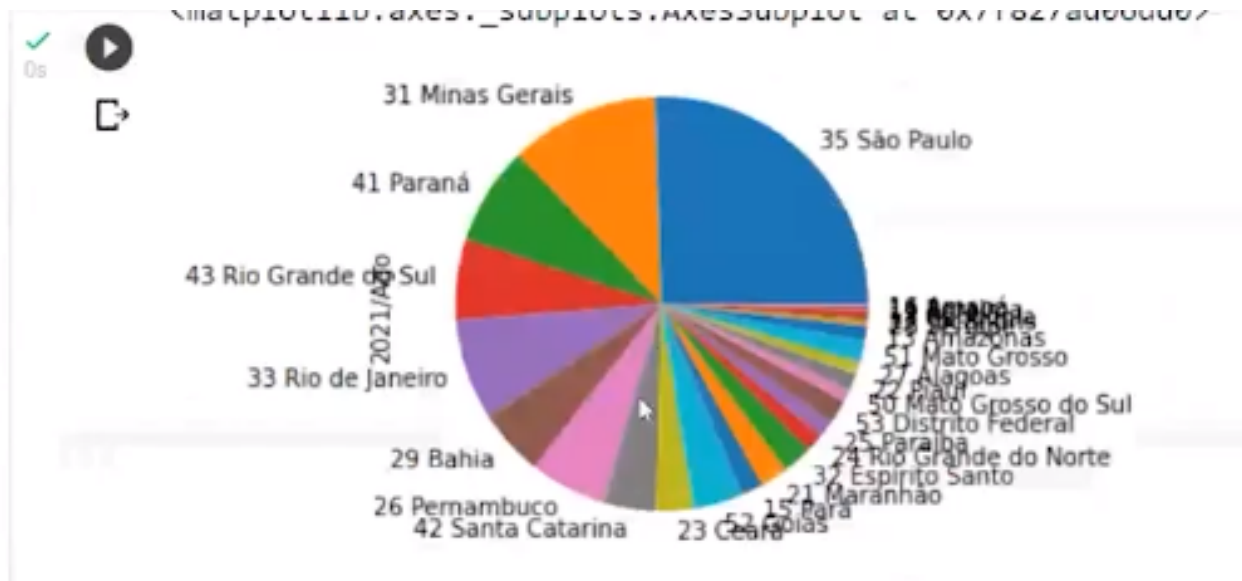
```
tabela_de_comparacao = gastos_do_mais_recente / gastos_do_mais_recente.loc["41 Paraná"]
```

```
tabela_de_comparacao.head(5)
```

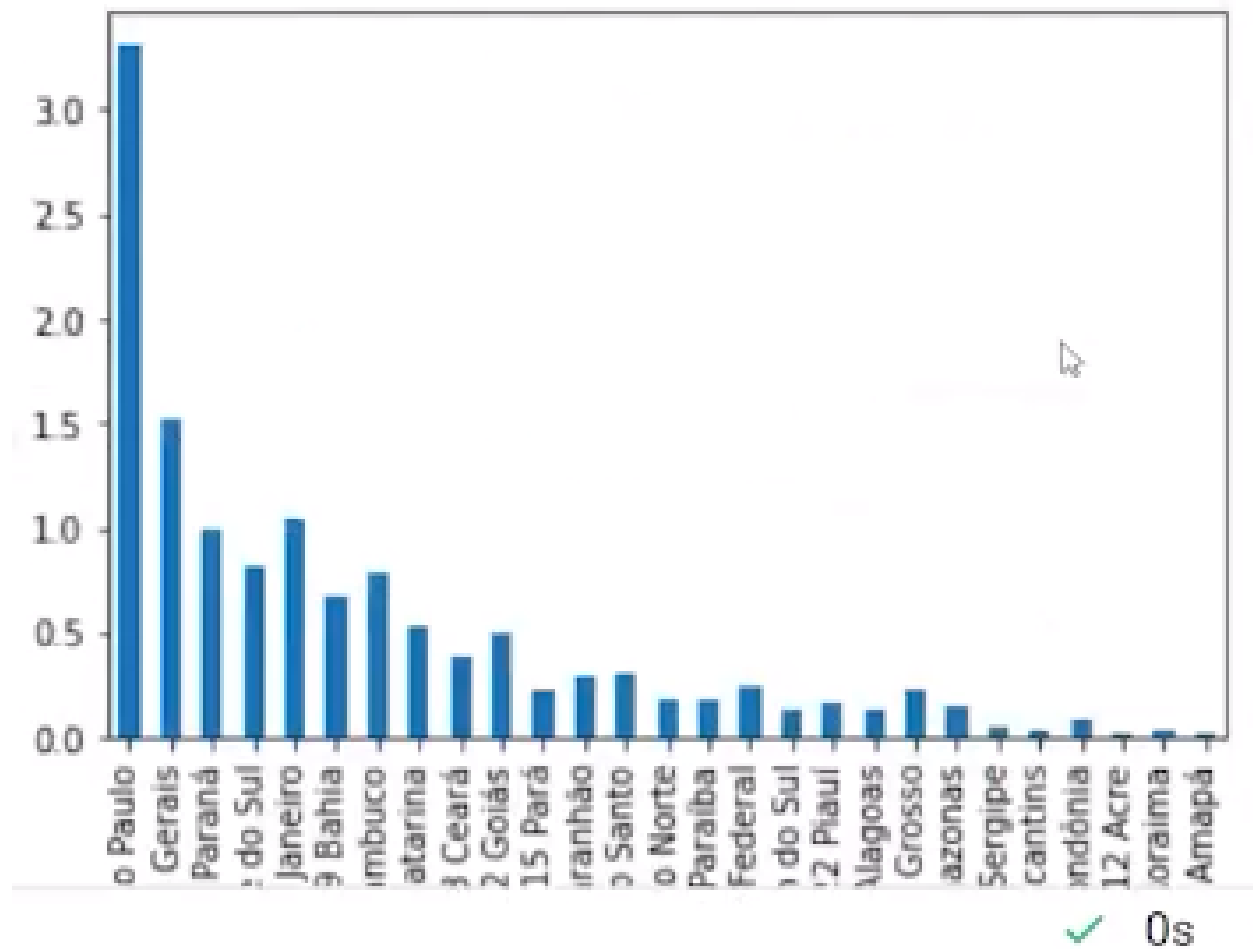
```
[235] 1 tabela_de_comparacao = gastos_do_mais_recente / gastos_do_mais_recente.loc["41 Paraná"]
      2 tabela_de_comparacao.head(5)
```

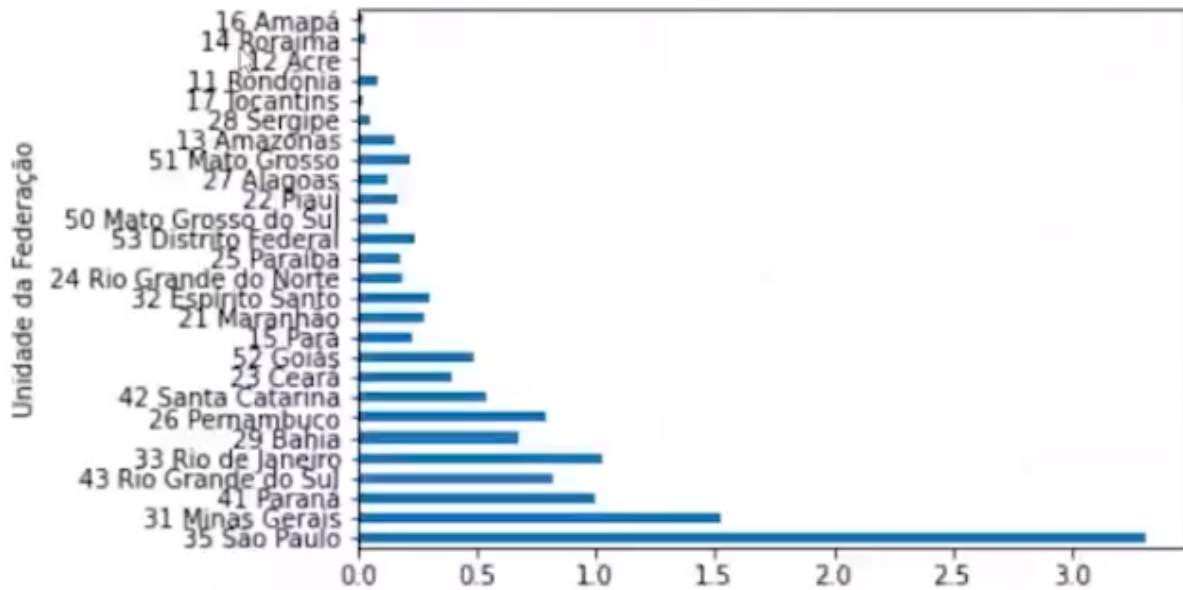
```
Unidade da Federação
35 São Paulo          3.31
31 Minas Gerais       1.53
41 Paraná             1.00
43 Rio Grande do Sul  0.82
33 Rio de Janeiro     1.03
Name: 2021/Ago, dtype: float64
```

☐ `tabela_de_comparacao.plot(kind="pie")`



☐ `tabela_de_comparacao.plot(kind="bar")`

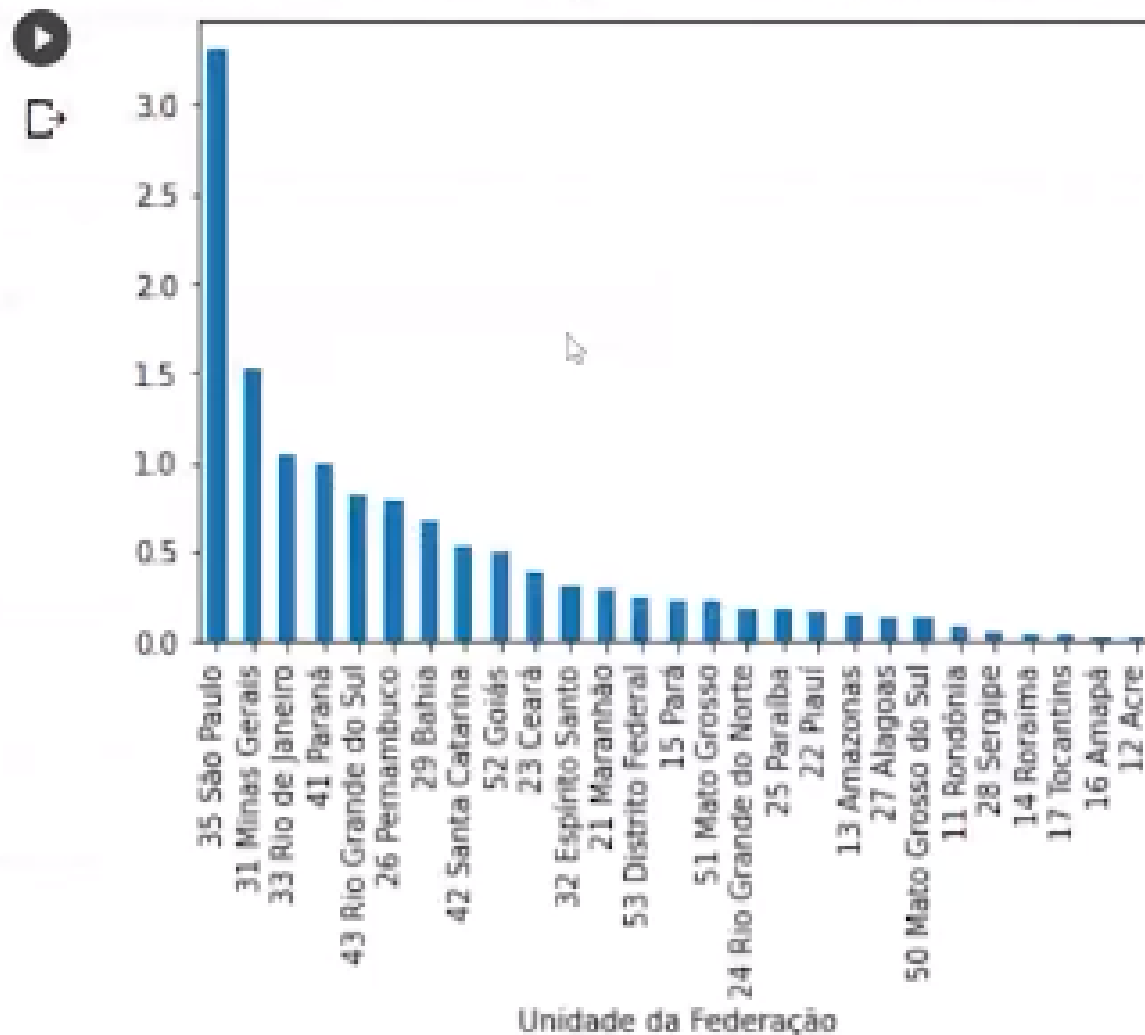




Nesse caso, podemos preferir as barras verticais, porque quero comparar grandeza, quantas vezes a mais. A barra horizontal é interessante em algumas situações, mas nessa não muito. ;)

☐ `tabela_de_comparacao = tabela_de_comparacao.sort_values(ascending=False)`

☐ `tabela_de_comparacao.plot(kind="bar")`



Agora temos um gráfico do **menor** para o **maior**, **tudo ordenadinho**.

Desafio 02: passar uma linha horinzotal no seu estado.
Anotando o gráfico com uma linha.

Desafio 03: atualizar o último gráfico para refletir seu estado, incluindo grid, eixos, etc.

Desafio 04: Colorir seu estado com um tom diferente. Colorir os outros estados de acordo com gasto maior ou menor.

Desafio 05: Gasto por população de dois estados. Escolher dois estados, plotar a comparação desses gastos de acordo com a população deles (usar base IBGE, por exemplo).

Desafio 06: Explore gráficos e as tabelas, encontre o que você acha de interessante, levante perguntas e hipóteses.

Desafio 07: Escolha outro valor além de “Valor Aprovado” no Tabnet.

2 | CONCLUSÃO

- Carregar dados;
- Explorar dados em formato de tabela;
- Visualizar isso de diversas formas, não só por gráficos, mas como através de tabelas também é muito importante, até mesmo através de texto;
- Se preocupar com eixo, cores, formato etc;
- Quanto mais, não necessariamente é melhor. Aprender a filtrar informações;
- Explorar wikipedia: análise exploratória de dados (exploratory data analysis), visualização de dados (data visualization);
- Ler documentação do pandas aos poucos;

