

INTRODUÇÃO A VISUALIZAÇÃO DE DADOS



Profa. Marcela Xavier Ribeiro

DC/UFSCar

O QUE É A VISUALIZAÇÃO DE DADOS?



- Representação gráfica de informações e dados;
- Usa elementos visuais, como diagramas, gráficos e mapas;
- Uma forma acessível de ver e entender exceções, tendências e padrões nos dados.



VISUALIZAÇÃO EFETIVA X INEFETIVA

Efetiva:

Rapidamente entendível;

Representativa;

Verdadeira.

Inefetiva:

Difícil de entender;

Distorce resultados;

Erros mais comuns da visualização (ver página web):

<https://medium.com/@eliezerfb/os-5-erros-comuns-que-levam-a-uma-visualiza%C3%A7%C3%A3o-de-dados-incorreta-8f1573e4d188>

REGRA DOS 5 SEGUNDOS

- A média de tempo de atenção para visualizar qualquer coisa on-line é de menos de 5 segundos;
- Portanto, se você não consegue chamar a atenção em 5 minutos, provavelmente perdeu o espectador;
- Inclua títulos e instruções claras e diga às pessoas sucintamente o que a visualização mostra e como interagir com ela.



SIMPLICIDADE

- Mantenha os gráficos simples e fáceis de interpretar;
- Em vez de sobrecarregar os cérebros dos espectadores com muitas informações, mantenha apenas os elementos necessários no gráfico;
- Ajude o público a entender rapidamente o que está acontecendo.

BONITO É DIFERENTE DE EFICAZ

- Há um equívoco de que a visualização esteticamente agradável é mais eficaz.
- Para chamar a atenção, às vezes queremos que eles sejam bonitos e atraentes.
- Mas, se não conseguir comunicar os dados corretamente, você perderá o interesse do público assim que conquistá-lo.

CORES APROPRIADAS

- use cor de forma propositada e eficaz
- a cor mais bonita e atraente pode distrair
- a cor deve ser usada apenas se ajudar a transmitir sua mensagem
- ser consistente com o esquema de cores ao qual a organização / consumidor está acostumado


MANIPULAÇÃO DE DADOS — PRÉ-PROCESSAMENTO

- **Por que é importante?**
- Sem qualidade nos dados ==> sem qualidade nos resultados da mineração.
- Decisões corretas precisam de dados corretos
- ex., dados duplicados distorcem resultados da mineração.
- Preparação de dados, limpeza, seleção e transformação compreende na maioria do trabalho da aplicação de mineração (90%).



PREPARAÇÃO DOS DADOS

A preparação dos dados é um grande problema para a mineração de dados, que inclui:

- limpeza de dados e **integração de dados;** 
- normalização;
- redução de dados;
- discretização.
- Muitos métodos têm sido propostos, mas ainda é uma área ativa de pesquisa.



NORMALIZAÇÃO

O que é?

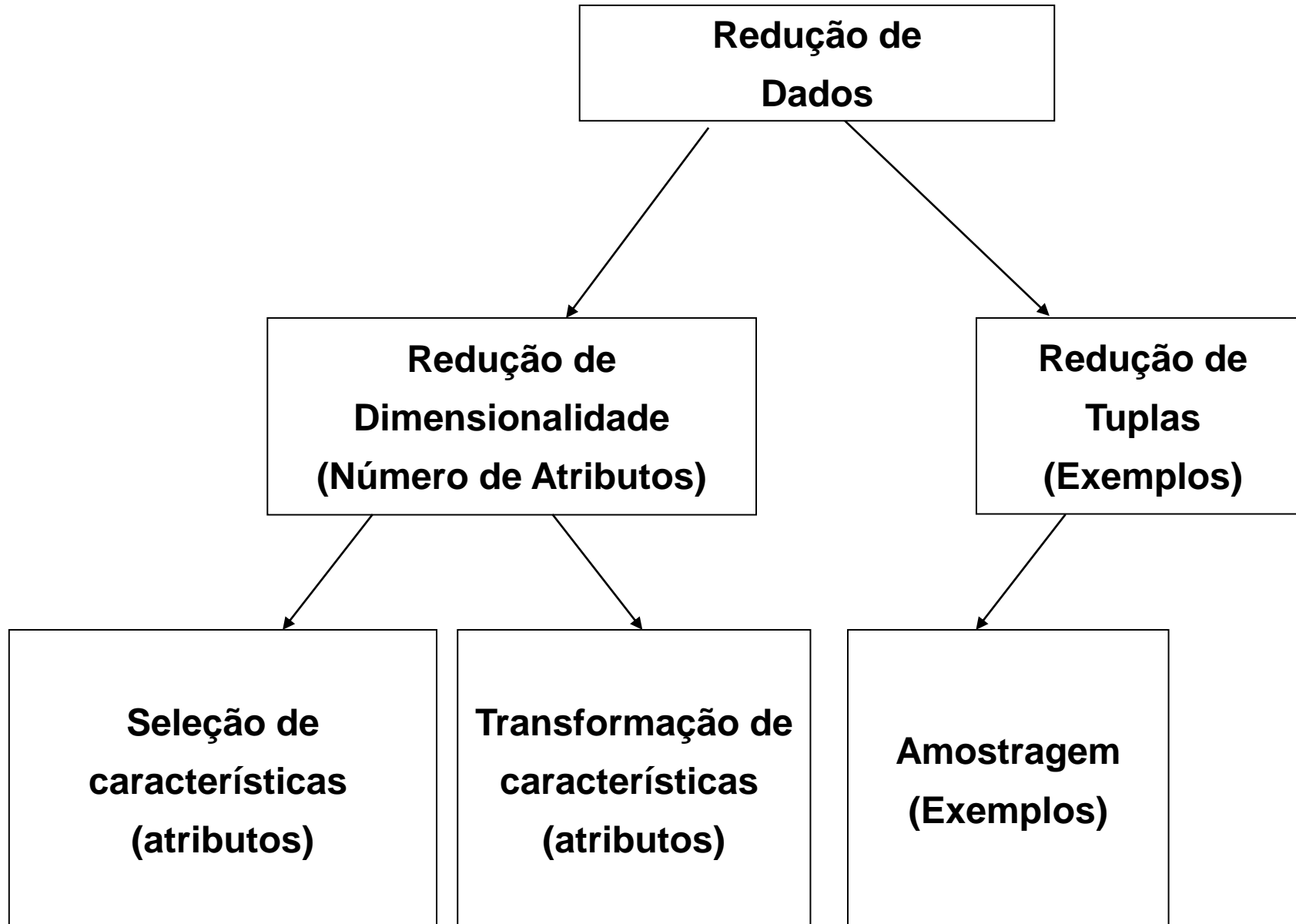
- Mapear os atributos contínuos para uma escala comum.

Para que normalizar?

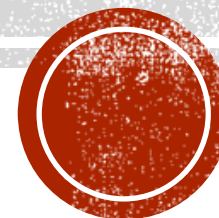
- Evitar distorções na mineração devido a diferentes escalas.



REDUÇÃO DE DADOS



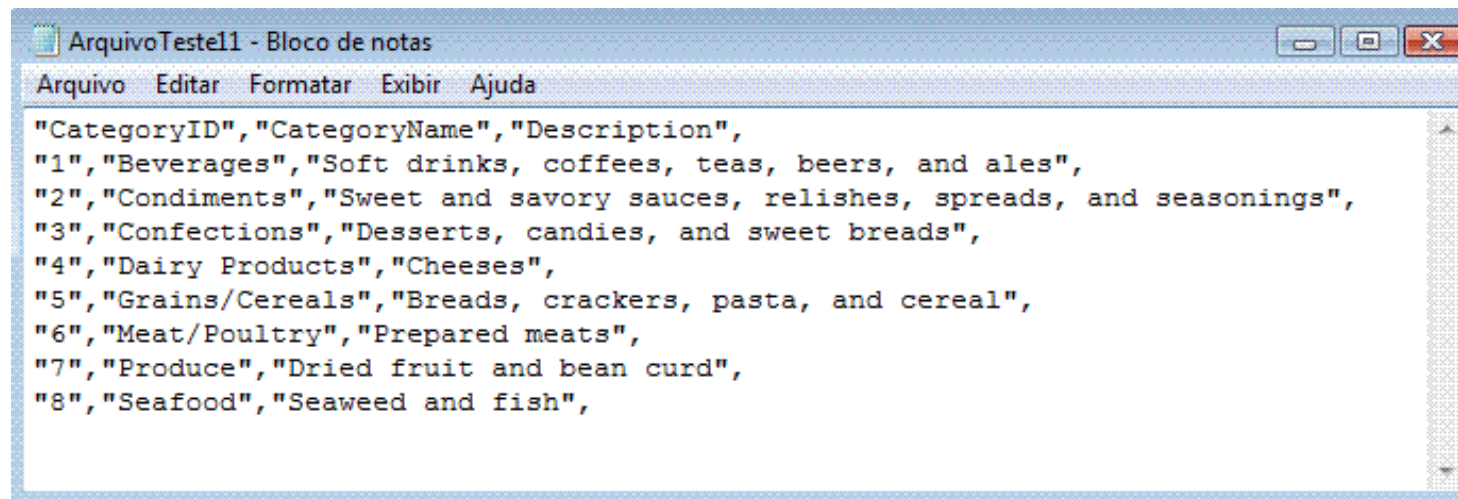
CARREGANDO OS DADOS



IMPORTAR ARQUIVOS .CSV NO PYTHON USANDO O COLAB

O QUE SÃO ARQUIVOS CSV?

- “CSV” significa *Comma Separated Values*
- é um arquivo de valores separados por vírgula.
- formato simples que agrupa informações de arquivos de texto em planilhas.



```
"CategoryID","CategoryName","Description",  
"1","Beverages","Soft drinks, coffees, teas, beers, and ales",  
"2","Condiments","Sweet and savory sauces, relishes, spreads, and seasonings",  
"3","Confections","Desserts, candies, and sweet breads",  
"4","Dairy Products","Cheeses",  
"5","Grains/Cereals","Breads, crackers, pasta, and cereal",  
"6","Meat/Poultry","Prepared meats",  
"7","Produce","Dried fruit and bean curd",  
"8","Seafood","Seaweed and fish",
```

POR QUE IMPORTAR ARQUIVOS CSV?

- Em geral os arquivos com dados que desejamos visualizar estão em planilhas.
- Essas planilhas são facilmente exportadas para .csv
- É muito útil carregar os dados diretamente desses arquivos para o Python, para posteriormente podermos gerar os gráficos.

ABRIR UM ARQUIVO .CSV NO COLAB

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

Passo 1: Importar o arquivo para a cloud (drive do colab)

```
from google.colab import files  
uploaded = files.upload()
```

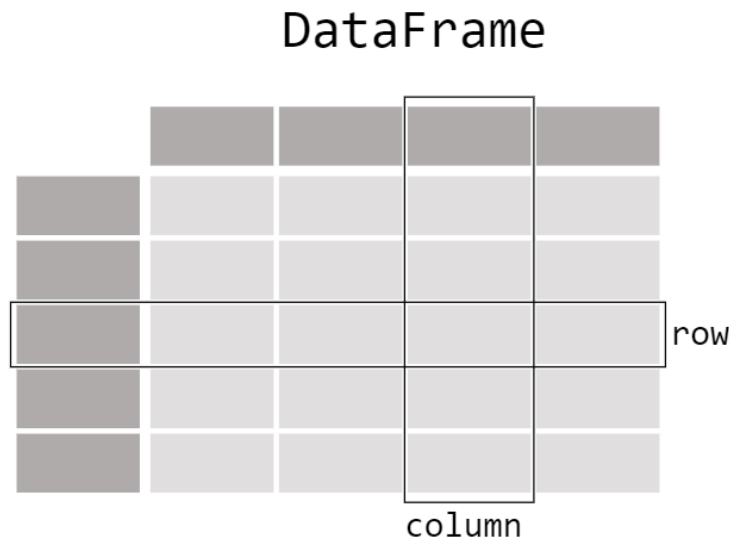


USANDO O PANDAS PARA CARREGAR O DATAFRAME

Passo 2: Ler o arquivo como .csv

```
import pandas as pd
import io
df = pd.read_csv(io.StringIO(uploaded['PoluicaoX.csv'].decode('utf-8')))
df
```

pandas data table representation



Fonte da Figura:

https://pandas.pydata.org/docs/getting_started/intro_tutorials/01_table_oriented.html



EXERCÍCIO: IMPORTAR O ARQUIVO POLUCAOX.CSV

```
[5] from google.colab import files
     uploaded = files.upload()
```

Escolher arquivos PoluicaoX.csv

- **PoluicaoX.csv**(application/vnd.ms-excel) - 2653 bytes, last modified: 13/04/2021 - 100% done
Saving PoluicaoX.csv to PoluicaoX (1).csv

```
import pandas as pd
import io
df = pd.read_csv(io.StringIO(uploaded['PoluicaoX.csv'].decode('utf-8')))
df
```



					pobr	mort	educ	precip	tempjan	umidd	HCpolui
13% a 18%	912	3	10	7	36	0	27	0	59	0	baixo
	823	8	12	1	28	0	32	0	54	0	baixo
	912	2	10	3	40	0	23	0	60	0	baixo

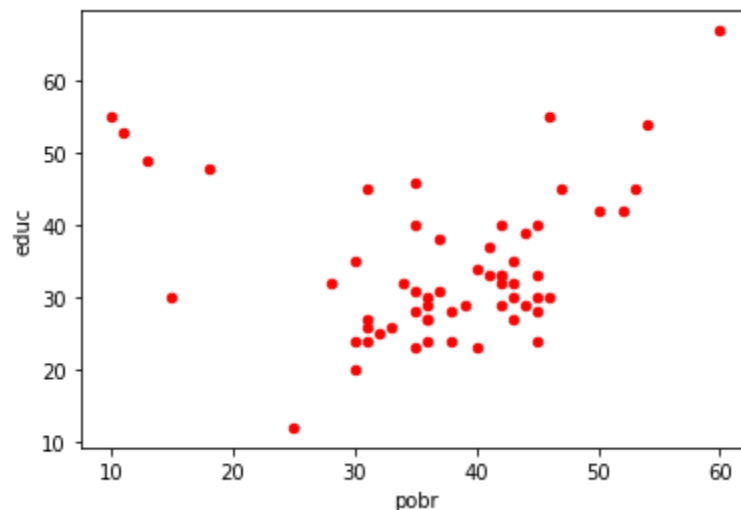
USANDO O DATAFRAME PARA UM GRÁFICO

#descreve as colunas e calcula medidas estatísticas de cada coluna

```
df.describe()
```

#desenha um grafico a partir das colunas do data frame

```
df.plot(kind='scatter',x='pobr',y='educ',color='red')
```



USANDO IMPORT CSV

```
import csv
with open('PoluicaoX.csv', 'r') as f:
    lines = list(csv.reader(f, delimiter=','))
print(lines[12])
```

MATPLOTLIB

MATPLOTLIB O QUE É?

- É uma biblioteca para a visualização de dados em Python.
- Apresenta uma API orientada a objetos que permite a criação de gráficos em 2D
- Disponibiliza diversos tipos de gráficos, como em barra, em linha, em pizza, histogramas...
- Foi projetada para ser compatível com o MATLAB.

VISUALIZAÇÃO DE DADOS COM PYPLOT

- O PyPlot é um módulo do matplotlib para criação de gráficos.
- Para utilizá-lo é necessário fazer a importação:

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

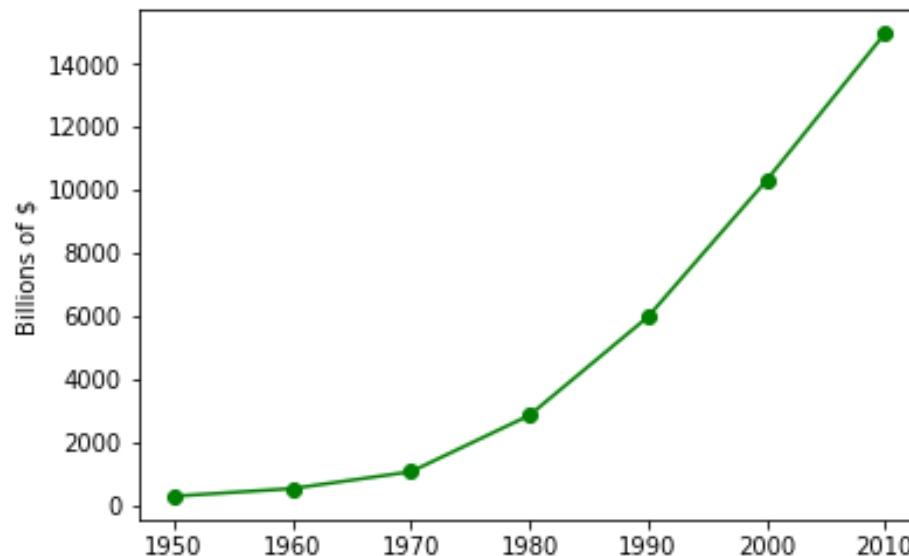
PYPLLOT - EXEMPLO

- Taxa de Crescimento do Gross Domestic Product (GDP) = Produto Interno Bruto (PIB)
- O produto interno bruto:
 - soma de todos os bens e serviços finais produzidos numa determinada região, durante um período determinado;
 - é um dos indicadores mais utilizados na macroeconomia;
 - tem o objetivo de quantificar a atividade econômica de uma região.

PYPLLOT - EXEMPLO

```
from matplotlib import pyplot as plt
years = [1950, 1960, 1970, 1980, 1990, 2000, 2010]
gdp = [300.2, 543.3, 1075.9, 2862.5, 5979.6, 10298.7, 14958.3]

#grafico de linha selecionando cor, marcador do ponto e estilo
#de linha
plt.plot(years, gdp, color="green", marker='o', linestyle='solid')
plt.ylabel('Billions of $')
plt.show()
```



Consultar todas as opções de parâmetros em:

https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.plot.html

EXERCÍCIO

- Faça um gráfico comparando Brasil e Argentina.
- São duas séries temporais artificiais disponibilizadas no `pibaumento.csv`



Fonte da Figura:

<https://pt.countryeconomy.com/paises/comparar/argentina/brasil>

Mais análises também disponíveis no site

VAMOS FAZER JUNTO PASSO A PASSO

- Vamos fazer uma vez da maneira mais difícil, sem usar um *dataframe* do pandas
- Passo 1: importar a base
- Passo 2: fazer o gráfico usando pyplot

PASSO 1:

```
from google.colab import files
uploaded = files.upload()
```

```
import csv
with open('pibaumento.csv', 'r') as f:
    lines = list(csv.reader(f, delimiter=';'))
print(lines[0])
```

```
['\uffffCountry Name', '1970', '1980', '1990', '2000', '2010', '2019', '2020']
```

```
anos = lines[0][1:]
print(anos)
```

```
['1970', '1980', '1990', '2000', '2010', '2019', '2020']
```

PASSO 1:

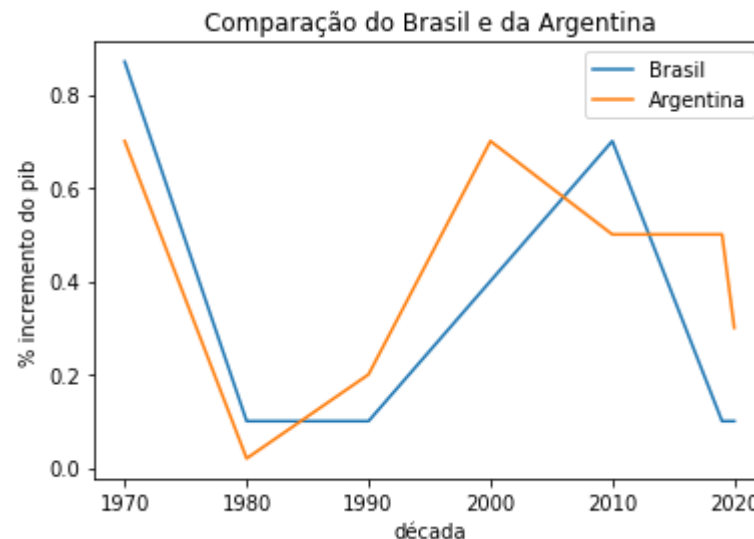
```
anosint= map(int, anos)
xanos = list(anosint)
print(xanos)           [1970, 1980, 1990, 2000, 2010, 2019, 2020]
```

```
brasil = lines[2][1:]
argentina = lines[1][1:]
print(brasil)
print(argentina)       ['0.87', '0.10', '0.10', '0.40', '0.70', '0.10', '0.10']
                        ['0.70', '0.02', '0.20', '0.70', '0.50', '0.50', '0.30']
```

```
brfloat = map(float,brasil)
arfloat = map(float,argentina)
ybr = list(brfloat)
yar = list(arfloat)
print(ybr)              ['0.87', '0.10', '0.10', '0.40', '0.70', '0.10', '0.10']
print(yar)              ['0.70', '0.02', '0.20', '0.70', '0.50', '0.50', '0.30']
```

PASSO 2:

```
import matplotlib.pyplot as plt
# série temporal do Brasil
plt.plot(xanos, ybr, label = "Brasil")
# série temporal da Argentina
plt.plot(xanos, yar, label = "Argentina")
plt.xlabel('década')
# coloca label do eixo y
plt.ylabel('% incremento do pib')
# coloca o título do gráfico
plt.title('Comparação do Brasil e da Argentina')
# adiciona a legenda do gráfico
plt.legend()
# mostra o gráfico
plt.show()
```



PLOTLY EXPRESS

Profa. Marcela Xavier Ribeiro
DC/UFSCar



PLOTLY EXPRESS

<https://plotly.com/python/plotly-express/>

- é biblioteca desenvolvida a partir da plotly para uma rápida exploração de dados e geração de figuras.
- já disponibiliza bases para analisar

https://www.plotly.express/plotly_express/data/index.html

- Depois de importar o Plotly Express, a maioria dos gráficos é feito com apenas uma chamada de função que aceita dados do Pandas

`px.scatter (data, x = "column_name", y = "column_name")`.

EXEMPLO — BASE POLUICAOX.CSV

```
from google.colab import files
uploaded = files.upload()
```



```
from google.colab import files
uploaded = files.upload()
```



Escolher arquivos PoluicaoX.csv

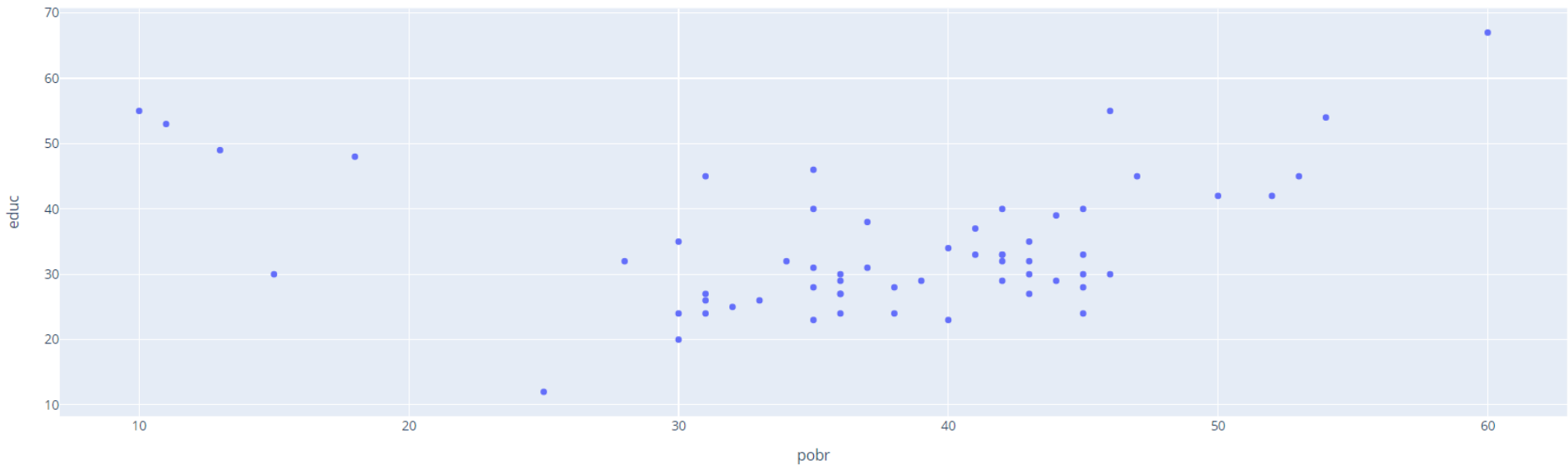
- **PoluicaoX.csv**(application/vnd.ms-excel) - 2653 bytes,
Saving PoluicaoX.csv to PoluicaoX.csv

```
import pandas as pd
import io
df = pd.read_csv(io.StringIO(uploaded['PoluicaoX.csv'].
decode('utf-8'))))
df
```

13% a 18%	954	4	10	7	38	0	28	0	58	0	mediano
	968	7	11	4	39	0	29	0	60	0	mediano
	1015	0	10	5	42	0	32	0	54	0	mediano
	958	8	11	9	37	0	31	0	58	0	mediano

EXEMPLO — BASE POLUICAOX.CSV

```
import plotly.express as px  
px.scatter (df, x = 'pobr', y = 'educ')
```



INTRODUÇÃO A VISUALIZAÇÃO DE DADOS



Profa. Marcela Xavier Ribeiro

DC/UFSCar