

Apresentação

Thiago Kuma

- Cientista de Dados na Loggi
- Hipnólogo de entretenimento nas horas vagas

Visualização de Dados

Leitura Recomendada:

Alberto Cairo

- Professor na Universidade de Miami
 - Visualização de Dados
- Livro [Art Introduction Information Visuzalization](https://www.amazon.com.br/Functional-Art-introduction-information-visualization-ebook/dp/B0091SXD0M?__mk_pt_BR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=alberto+cairo&qid=1537789336&sr=8-2&ref=sr_1_2) (https://www.amazon.com.br/Functional-Art-introduction-information-visualization-ebook/dp/B0091SXD0M?__mk_pt_BR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=alberto+cairo&qid=1537789336&sr=8-2&ref=sr_1_2)

Base dos Gráficos

Framework

- Base
- Escalas
- Legendas

Visual encodings

- O mais importante
- Como dar significado aos números, ex. gráfico de barras quando colocamos proporção

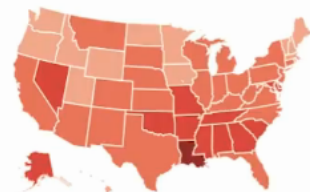
Annotations

- Texto que dá contexto
- Pode servir de introdução

COMPLETE CHART

MURDERS

Rate per 100,000 people (2015)

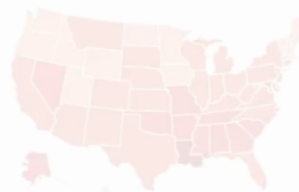


0 3 6 9 >9
Source: FBI Uniform Crime Reports

FRAMEWORK

MURDERS

Rate per 100,000 people (2015)

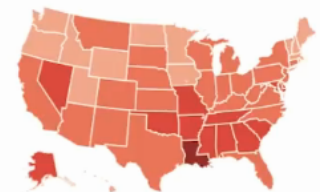


0 3 6 9 >9
Source: FBI Uniform Crime Reports

CONTENT (ENCODING)

MURDERS

Rate per 100,000 people (2015)

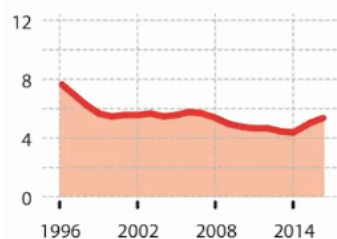


0 3 6 9 >9
Source: FBI Uniform Crime Reports

COMPLETE CHART

MURDERS

Rate per 100,000 people

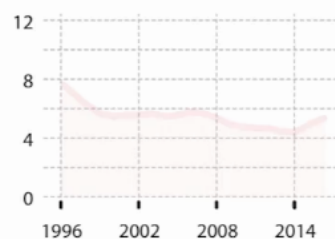


Source: FBI Uniform Crime Reports

FRAMEWORK

MURDERS

Rate per 100,000 people

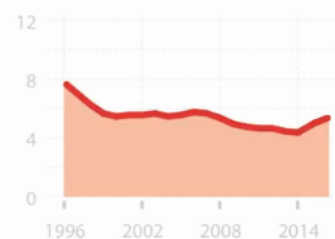


Source: FBI Uniform Crime Reports

CONTENT (ENCODING)

MURDERS

Rate per 100,000 people



Source: FBI Uniform Crime Reports

Falando em visualização de Dados e Mapa...

Mapa VAGAS de Carreira

[Mapa de Carreira \(https://www.vagas.com.br/mapa-de-carreiras/\)](https://www.vagas.com.br/mapa-de-carreiras/)

- *Processamento de 10 milhões de currículos*

- *Representação baseada em grafo*

- *Representação Gênero*

- *Representação Salarial*

- *Representação das Ocupações*

[Professora Pedagoga \(https://www.vagas.com.br/mapa-de-carreiras/cargos/professora-pedagoga/0\)](https://www.vagas.com.br/mapa-de-carreiras/cargos/professora-pedagoga/0)

- *Distribuição para demonstrar proximidade*

[Médico \(https://www.vagas.com.br/mapa-de-carreiras/cargos/medico/0\)](https://www.vagas.com.br/mapa-de-carreiras/cargos/medico/0)

Vamos olhar mais tecnicamente

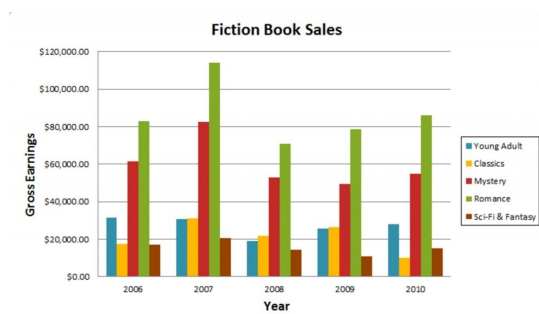
Gráficos de barras

Situação de uso: quando queremos ter uma ideia da evolução temporal das nossas variáveis

Título: Vendas de livros de ficção

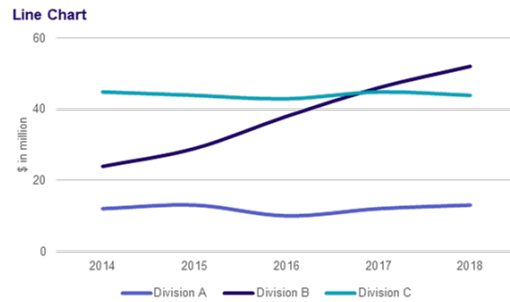
Eixo Y: Ganhos brutos

Eixo X: Ano



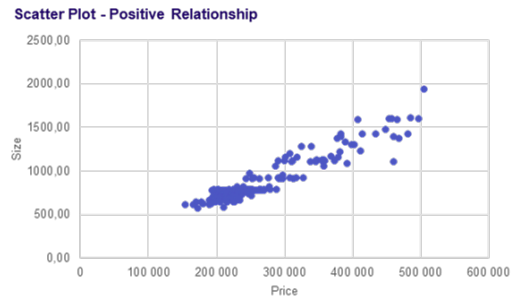
Gráficos de linhas

Situação de uso: quando queremos ter uma ideia da evolução temporal contínua das nossas variáveis. Por causa disso é o plot mais comum em time series



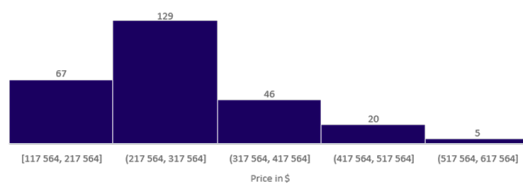
Scatter Plot

Situação de uso: quando queremos estudar a relação entre duas variáveis quantitativas



Histogram

Situação de uso: analisar a distribuição de uma variável quantitativa

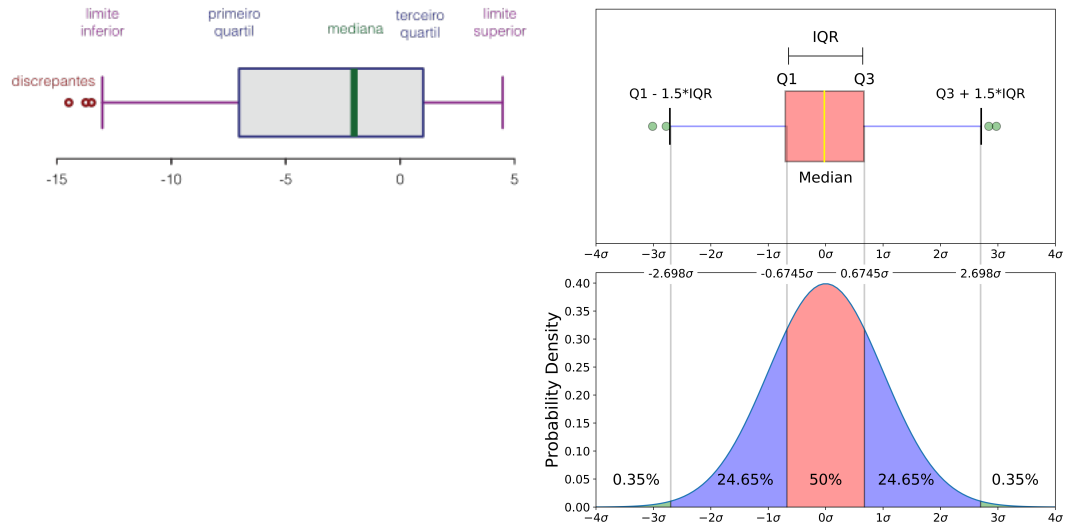


Box plot

Situação de uso: quando queremos ter um entendimento dos momentos estatísticos básicos dos nossos dados

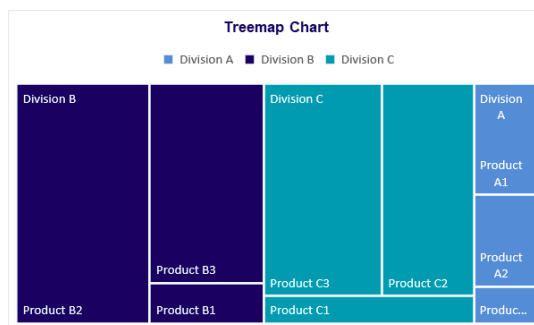
Quando falha: quando você tem muitas classes e o plot fica confuso

Como ler um boxplot:



Treemap Chart

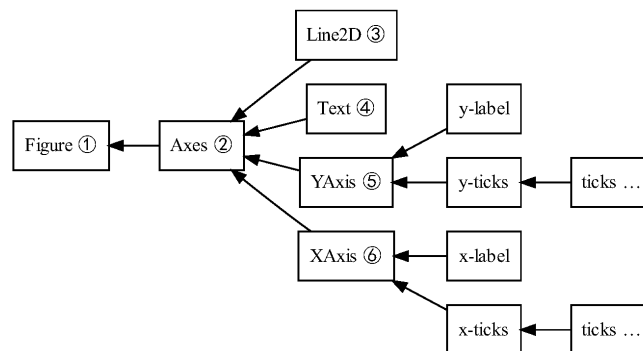
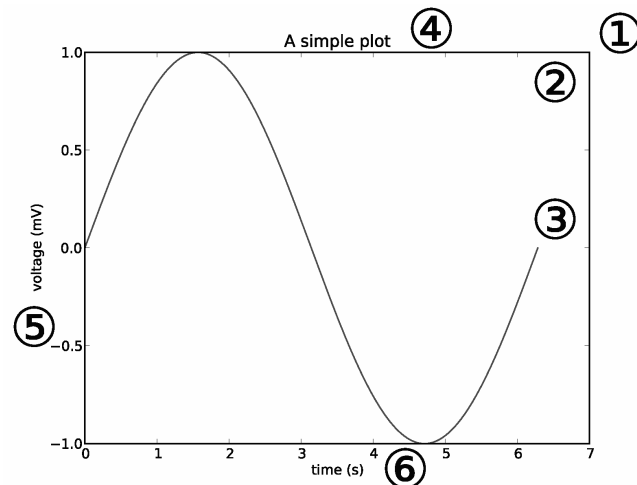
Situação de uso: quando você possui classes e clusters relativos a elas e precisa de uma foto de informação proporcional entre elas



Primeiros Gráficos

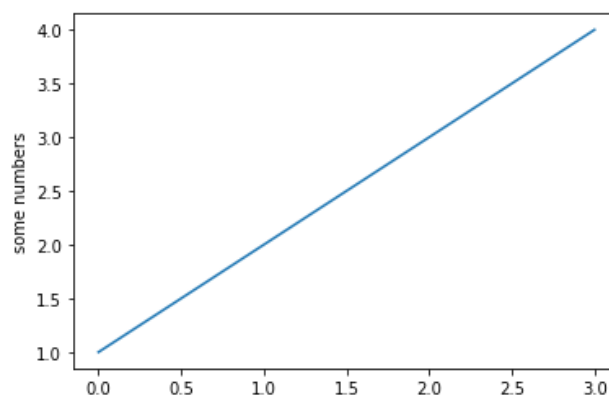
```
In [1]: >> import matplotlib.pyplot as plt
```

API do Pyplot: [matplotlib \(https://matplotlib.org/api/pyplot_api.html\)](https://matplotlib.org/api/pyplot_api.html)

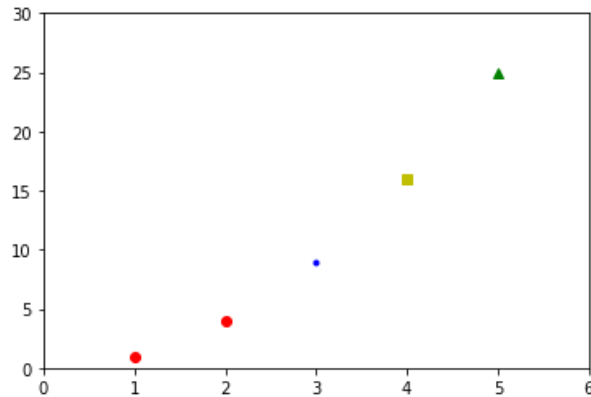


Tutorial: [matplotlib \(https://matplotlib.org/tutorials/introductory/pyplot.html\)](https://matplotlib.org/tutorials/introductory/pyplot.html)

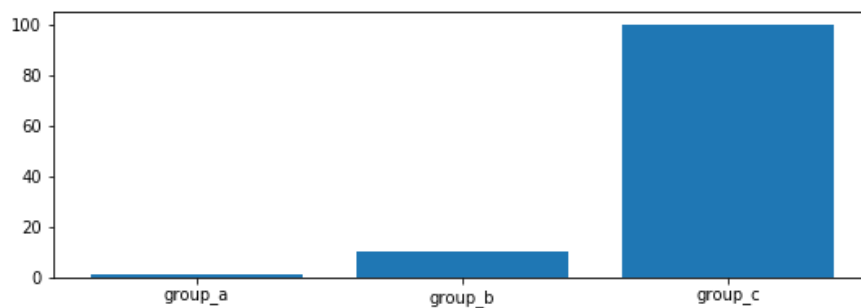
```
In [2]: ▶ plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```



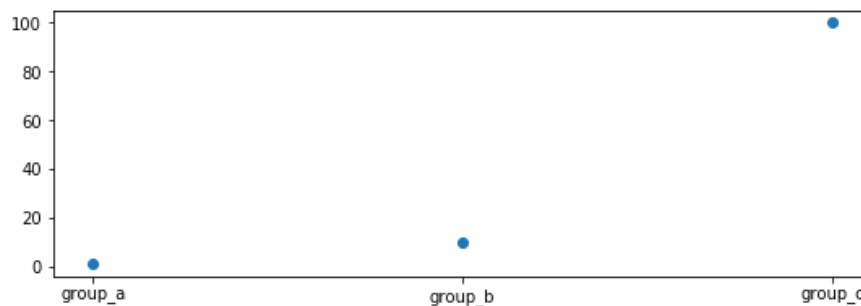
```
In [3]: ▶ plt.plot([1, 2], [1, 4], 'ro') #red circle  
plt.plot([3], [9], 'b.') #blue point  
plt.plot([4], [16], 'ys') #yellow square  
plt.plot([5], [25], 'g^') #green triangles  
plt.axis([0, 6, 0, 30])  
plt.show()
```



```
In [4]: ▶ names = ['group_a', 'group_b', 'group_c']  
values = [1, 10, 100]  
  
plt.figure(figsize=(9, 3))  
  
plt.bar(names, values)  
plt.show()
```



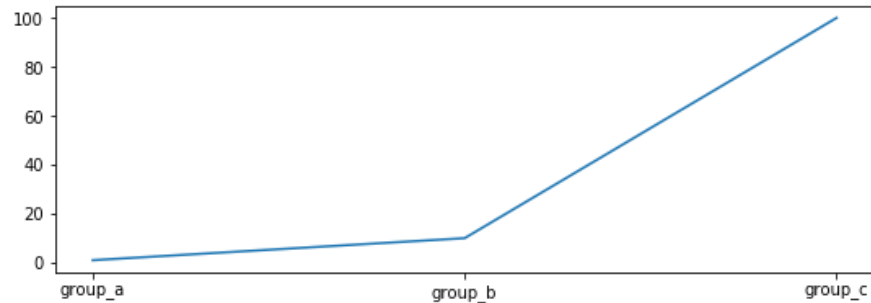
```
In [5]: ▶ names = ['group_a', 'group_b', 'group_c']  
values = [1, 10, 100]  
  
plt.figure(figsize=(9, 3))  
  
plt.scatter(names, values)  
plt.show()
```



```
In [6]: ▶ names = ['group_a', 'group_b', 'group_c']
values = [1, 10, 100]

plt.figure(figsize=(9, 3))

plt.plot(names, values)
plt.show()
```



Gráficos Mais Complexos

```
In [7]: ▶ import pandas as pd
kc = pd.read_csv(r'KingCounty_house_data.csv')
print(kc.size)
df = kc.sample(5000)
df.head()
```

453873

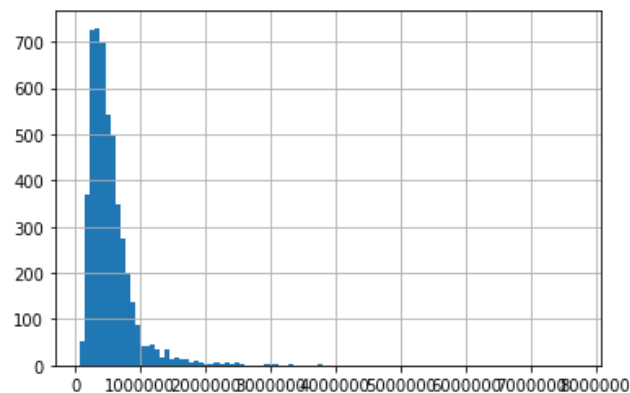
Out[7]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	wa
9355	3172600031	20150327T000000	325000.0	3	1.5	1590	7936	1.0	
2772	2325069117	20140805T000000	960000.0	5	3.5	4510	16305	2.0	
16949	418000415	20150319T000000	191000.0	2	1.0	700	5000	1.0	
4897	5648600190	20150429T000000	310000.0	3	2.5	1670	5791	2.0	
1965	8952900260	20140919T000000	375000.0	3	1.0	1130	12500	1.5	

5 rows × 21 columns

```
In [8]: ▶ df.price.hist(bins=100).plot()
# https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.hist.html
```

Out[8]: []




```
In [10]: ▶ import seaborn as sns
sns.jointplot(x="sqft_living", y="price", data=df, kind = 'reg', fit_reg= Tr
plt.show()
# https://seaborn.pydata.org/generated/seaborn.jointplot.html
```

