



**VEJA OS PLANOS**

PROGRAMAÇÃO \_

DATA SCIENCE \_

DEVOPS \_

MOBILE \_

FRONT-END \_

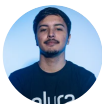
INTELIGÊNCIA ARTIFICIAL \_

UX & DESIGN \_

INOVAÇÃO & GESTÃO \_

Artigos > **Data Science**

# Data Visualization: conhecendo as bibliotecas do Python



**Marcus Almeida**

04/05/2023

COMPARTILHE



Esse artigo faz parte da  
**Formação Python para Data Science**

A Ciência de Dados é o conjunto de técnicas que trabalha em analisar e interpretar informações úteis e valiosas por trás de grandes volumes de dados que são gerados todos os dias, nos mais diferentes setores. Nesse contexto, surge também a **visualização de dados** que desempenha um papel importante quando precisamos identificar problemas, anomalias e tendências no processo de análise de dados. Logo, é uma ferramenta essencial para profissionais de dados e para o público em geral que consome essas informações.

Para resolver essas tarefas, surgem bibliotecas com o papel de unir a visualização de dados ao poder do Python e suas bibliotecas para Data Science, como [NumPy](#) e [Pandas](#), que já atuam em toda a cadeia do trabalho de extração, limpeza e transformação de dados.

### Confira neste artigo:

- [O que é Visualização de Dados \(DataViz\)?](#)
- [Matplotlib](#)
- [Seaborn](#)
- [Plotly](#)
- [O que vem depois?](#)
- [Conclusão](#)

Nesse artigo, vamos conhecer um pouco mais sobre a visualização de dados e algumas bibliotecas muito utilizadas.

# O que é Visualização de Dados (DataViz)?

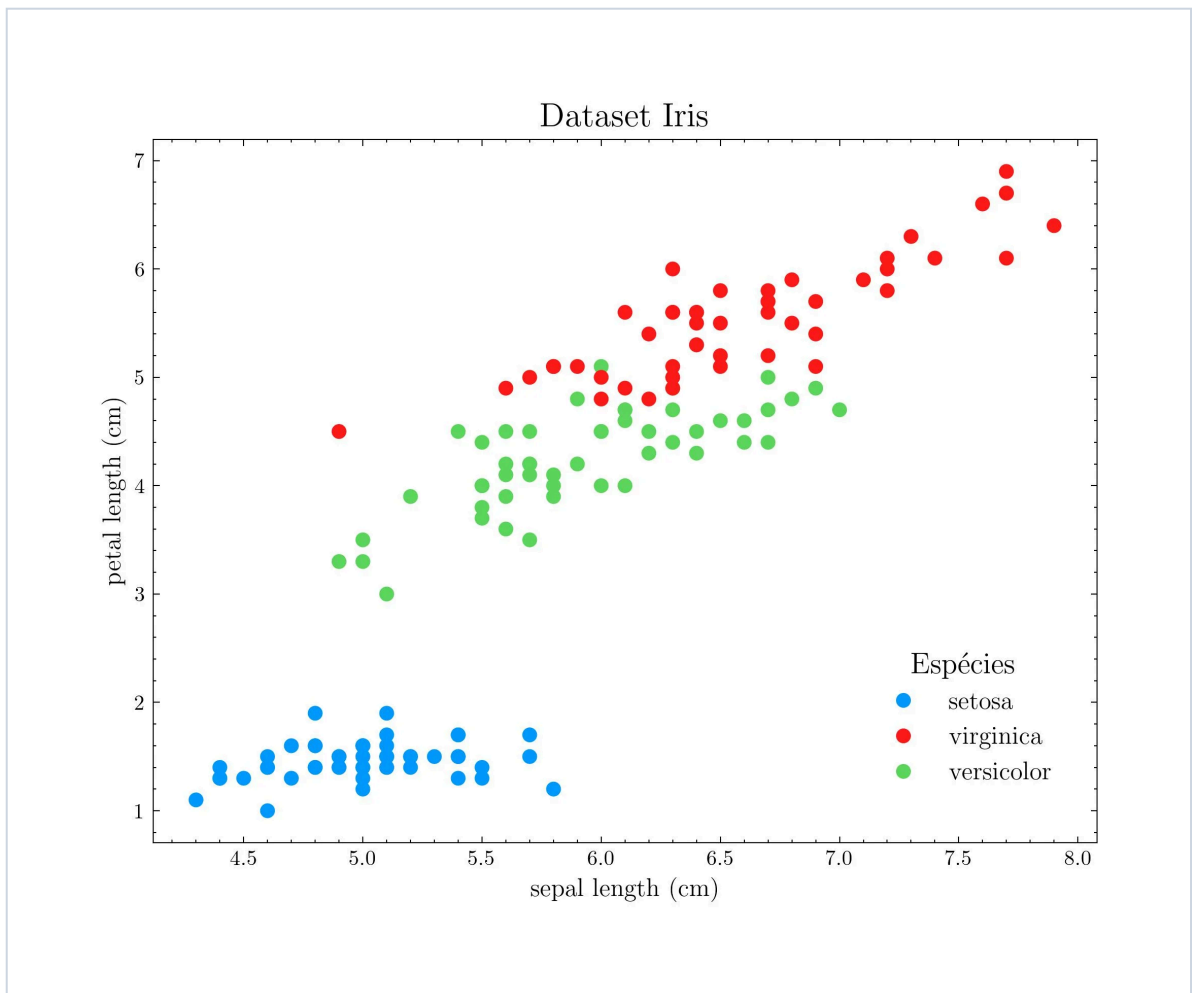
A **visualização de dados**, também conhecida por termos como *data visualization* ou *DataViz*, é o conjunto de técnicas para representar informações e dados de forma visual de maneira compreensível a fim transmitir resultados de análise de dados. Essas técnicas são utilizadas para transformar grandes conjuntos de dados e registros em gráficos, tabelas e diagramas que podem ser processados mais facilmente e por qualquer pessoa.

Por exemplo, utilizando um conjunto de dados (*dataset*) como o *Iris*, que possui 150 registros com o comprimento (length) de sépalas e pétalas de algumas espécies de flores Iris (a saber, *virginica*, *versicolor* e *setosa*), nós podemos apresentar as informações contidas nesse conjunto de algumas maneiras. Neste caso, podemos expor a tabela de dados diretamente:

	sepal length (cm)	petal length (cm)	class
0	5.1	1.4	setosa
1	4.9	1.4	setosa
2	4.7	1.3	setosa

	sepal length (cm)	petal length (cm)	class
3	4.6	1.5	setosa
4	5.0	1.4	setosa
..	...	...	...
145	6.7	5.2	virginica
146	6.3	5.0	virginica
147	6.5	5.2	virginica
148	6.2	5.4	virginica
149	5.9	5.1	virginica

Ou através do uso de gráficos, resumindo as informações desse conjunto de dados de maneira visual. O exemplo abaixo é um gráfico de dispersão (*scatter*) que traz os mesmos registros da tabela completa anterior:



Dessa maneira, nós conseguimos gerar *insights* de forma mais rápida e com uma linguagem simples. Para analisar o gráfico acima, não é preciso muito conhecimento técnico para compreender as características individuais de cada espécie de Iris, os tamanhos mínimos e máximos ou qual espécie tem pétala ou sépala maior, por exemplo. Isso facilita o processo de observar e comunicar resultados em análise de dados.

Na Ciência de Dados, a visualização de dados tem o papel fundamental de **conectar** os setores de cada área do negócio. Os projetos são executados por pessoas analistas e cientistas de dados e passam por uma fase na qual a apresentação dos resultados é estruturada.

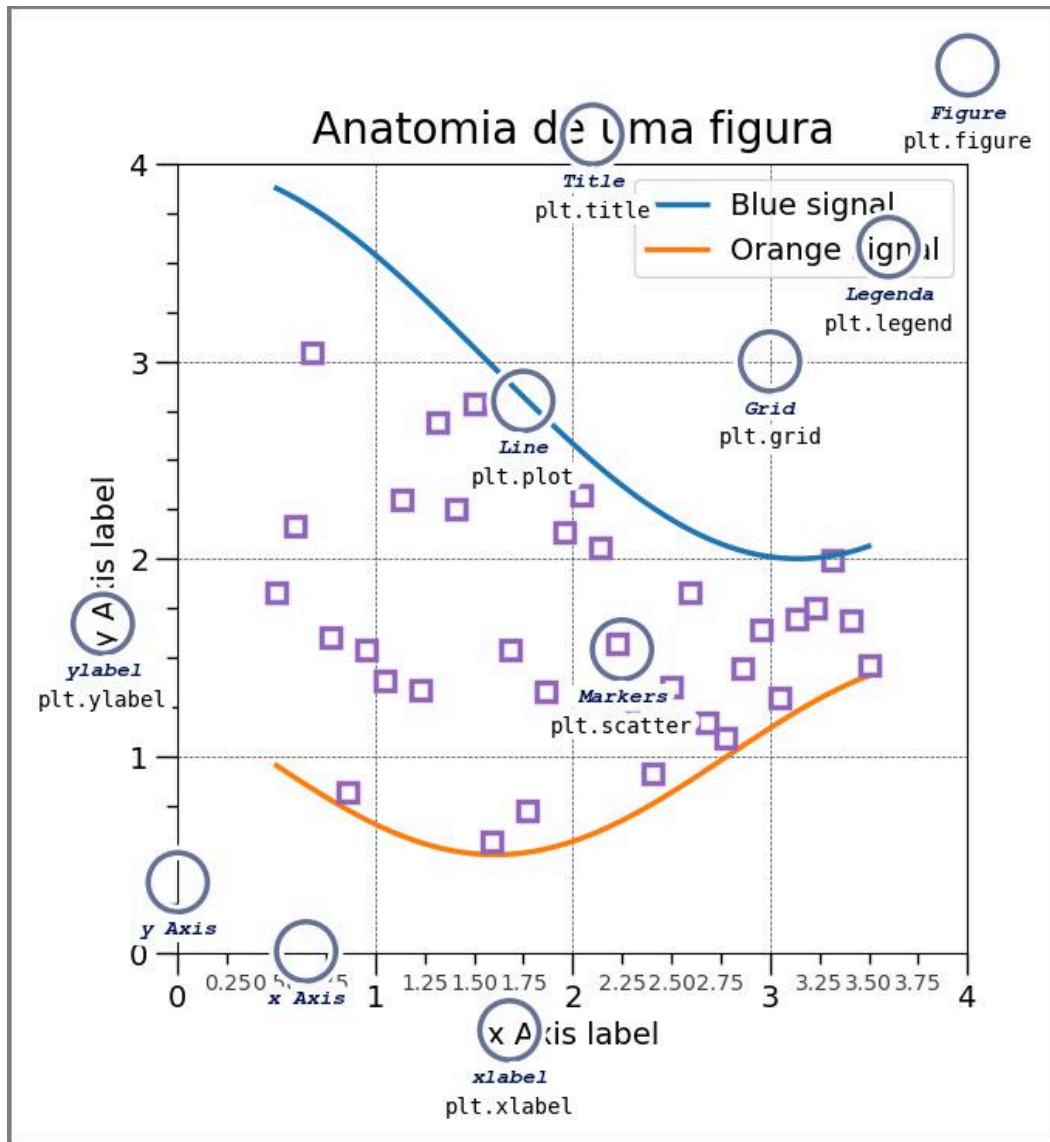
Nesse ponto, cria-se um **storytelling** (uma maneira de apresentar informação ao público alvo) e para dar suporte a esse processo criativo e analítico, os dados agora transformam-se em resultados por meio de indicadores ou [KPIs](#), insights, números, gráficos, diagramas, entre outros.

## Terminologias

O inglês é a língua utilizada para estruturar os códigos em Python e de suas bibliotecas. Então, é importante conhecer alguns dos termos utilizados em tarefas de visualização de dados, tais como:

<b>Termo [em inglês]</b>	<b>Tradução</b>	<b>Observações</b>
Chart	Gráfico	Geralmente é acompanhado do tipo do gráfico. Ex: Pie Chart (Pizza), Bar Chart (Barras), Line Chart (Linhas), Bubble Chart (Bolhas), etc.
Plot	Plotagem	Produção de uma imagem por meio de desenhos e linhas. Ação de construir o gráfico.
Axis / Axes	Eixo	Eixo dos gráficos. Ideia análoga à Matemática, na qual temos X e Y para o caso 2D e X, Y, Z para o caso 3D.
Label	Rótulo	São os rótulos que dão o nome e acompanham cada eixo. Geralmente aparecem acompanhados do respectivo eixo. Ex: x_label, y_label, etc.
Grid	Grade/Malha	É a estrutura de fundo de um gráfico. Linhas horizontais e verticais espaçadas que servem de referência para as unidades dos eixos.
Legend	Legenda	Caixa de texto descritivo, contendo informações sobre os elementos do gráfico. Geralmente traz informação sobre o que significa determinada cor ou traçado e/ou quais as variáveis sendo plotadas.

Esses termos também podem ser vistos por meio da imagem abaixo que mostra a estrutura “anatômica” de um gráfico, ou seja, como os elementos são chamados. Mesmo em bibliotecas diferentes, ainda conseguimos observar várias semelhanças.



Agora que já conhecemos os principais termos e componentes dos gráficos e figuras, vamos partir para as bibliotecas.

imersão **IA\_**

Mergulhe em **Inteligência Artificial** com a **Alura** e o **Google**

Cinco aulas **gratuitas** para você aprender a usar IA na prática e desenvolver habilidades essenciais para o mercado de trabalho

**Inscreva-se Agora**

**alura** + **Google**

100% gratuito

De 06 a 11 de maio

Certificado Alura

# Matplotlib



A **Matplotlib** é uma das bibliotecas mais populares para visualização de dados em Python. Criada em 2003, pelo cientista da computação John D. Hunter, o projeto Matplotlib foi desenvolvido com o objetivo de promover um ambiente de criação de gráficos (plots) semelhantes ao do software MATLAB.

O projeto é de código aberto (open source) e possui uma variedade de gráficos, como: o de linhas, dispersão, histogramas, barras, e muito mais. Ele também permite um alto nível de personalização, desde o trabalho com as cores de cada elemento, fontes, escalas, entre outros. Para os gráficos em imagens, é possível trabalhar com vários tipos de saída, desde os mais comuns PNG, PDF, JPEG, SVG e EPS, sendo este último um formato bastante utilizado para artigos acadêmicos e técnicos.

## Galeria de gráficos da Matplotlib

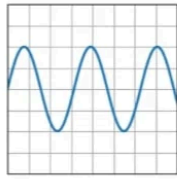
A documentação do Matplotlib disponibiliza dois links interessantes: o **Plot Types** e o **Examples**.

- [Plot types](#): página dedicada a mostrar os tipos de visualizações possíveis com Matplotlib. Alguns deles são: *plot*, *scatter*, *bar*, *stem*, *step* e *fill\_between*.

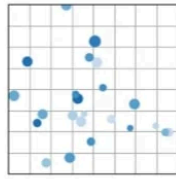


## Basic

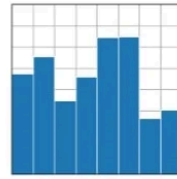
Basic plot types, usually y versus x.



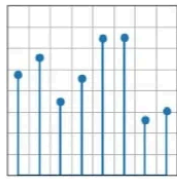
`plot(x, y)`



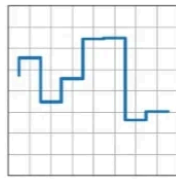
`scatter(x, y)`



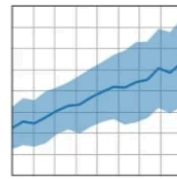
`bar(x, height)`



`stem(x, y)`



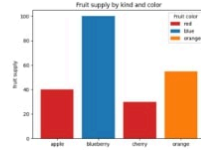
`step(x, y)`



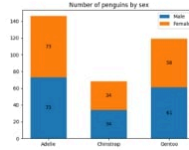
`fill_between(x, y1, y2)`

- [Examples](#): página com exemplos diversos e junção de funções de criação e customização diferentes do matplotlib. Alguns dos exemplos mostrados são: *Bar color demo*, *Bar Label Demo*, *Stacked bar chart*, *Grouped bar chart with labels*, *Horizontal bar chart* e *Broken Barh*.

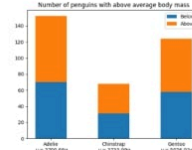
## Lines, bars and markers



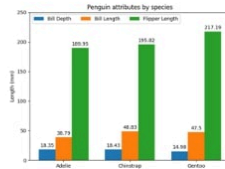
Bar color demo



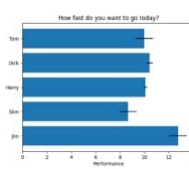
Bar Label Demo



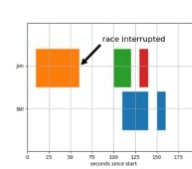
Stacked bar chart



Grouped bar chart with labels



Horizontal bar chart



Broken Barh

Além da grande variedade de gráficos 2D estáticos, a Matplotlib também permite a criação de animações no formato GIF, gráficos em 3D e a junção de vários gráficos em um só (subplots). Tudo isso com o uso de módulos específicos dentro da biblioteca.

## Seaborn

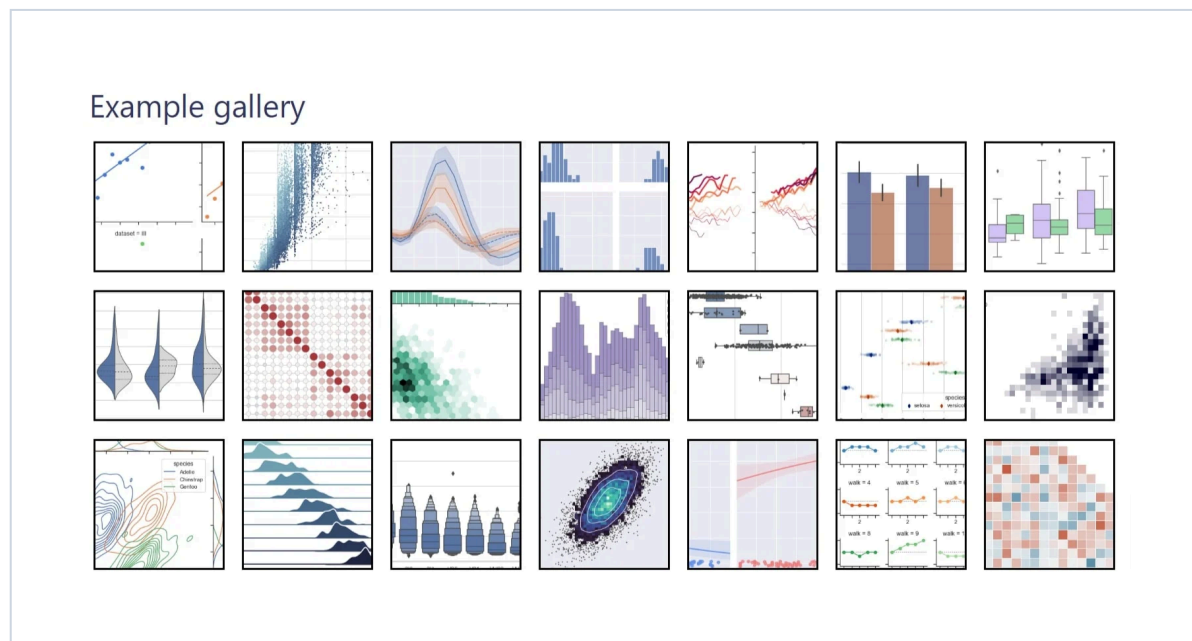


Em 2012, sob orientação do cientista de dados americano Michael Waskom, surge a **Seaborn**, uma biblioteca de código aberto baseada no Matplotlib. Ela propõe uma interface de alto nível para trabalhar com gráficos mais atrativos e com informações estatísticas, com a ideia principal que, segundo Waskom, “a Seaborn torna coisas difíceis em coisas muito fáceis de serem feitas”.

A biblioteca é geralmente utilizada em conjunto com outras bibliotecas de análise de dados, das quais tem uma ótima aderência entre os seus objetos, como NumPy e Pandas, nos quais é possível indicar de maneira simples as variáveis que estão sendo utilizadas e gerar rapidamente resultados.

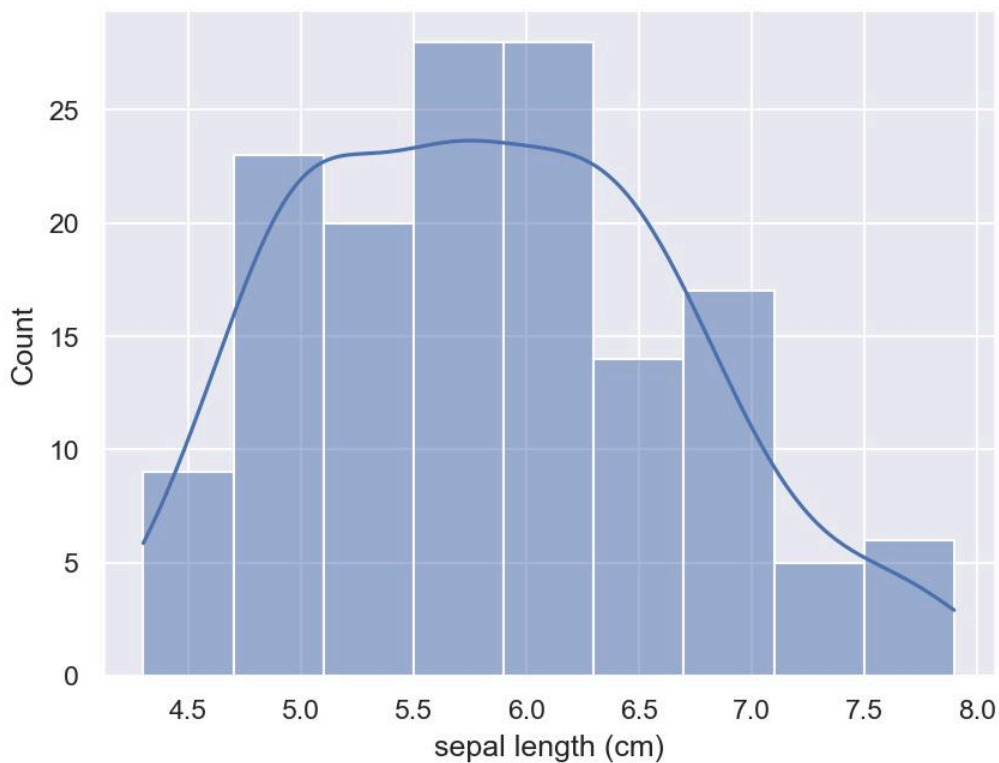
## Galeria de gráficos da Seaborn

A Seaborn também possui uma [galeria de imagens](#) dos gráficos feitos utilizando a biblioteca.



E uma das características mais importantes da Seaborn é a **facilidade em tratar informações estatísticas** do conjunto de dados durante o processo de *plot* dos gráficos. No exemplo abaixo construímos um [gráfico de histograma](#) e, apenas adicionando o parâmetro `kde=True`, conseguimos criar uma nova linha no gráfico que faz uma estimativa de uma função estatística para esse mesmo conjunto de dados.

```
import seaborn as sns
sns.histplot(data=df, x='sepal length (cm)', kde=True)
plt.show()
```



## Plotly



A Plotly é uma biblioteca de código aberto para visualização de dados **de forma interativa** em Python e que possui suporte em outras linguagens de programação. Criada em 2012 pelos cientistas de dados Alex Johnson, Jack Parmer e Chris Parmer, a Plotly chegou ao cenário com o objetivo de preencher a lacuna na visualização de dados interativa.

A principal característica da Plotly é a capacidade de criar gráficos que podem ser manipulados em tempo real pelas pessoas usuárias, permitindo que a exploração dos dados seja dinâmica e imersiva. Na interface do gráfico, é possível usar ações como diminuir e aumentar o zoom, mudar a região de observação (pan), utilizar filtros de dados e também animações.

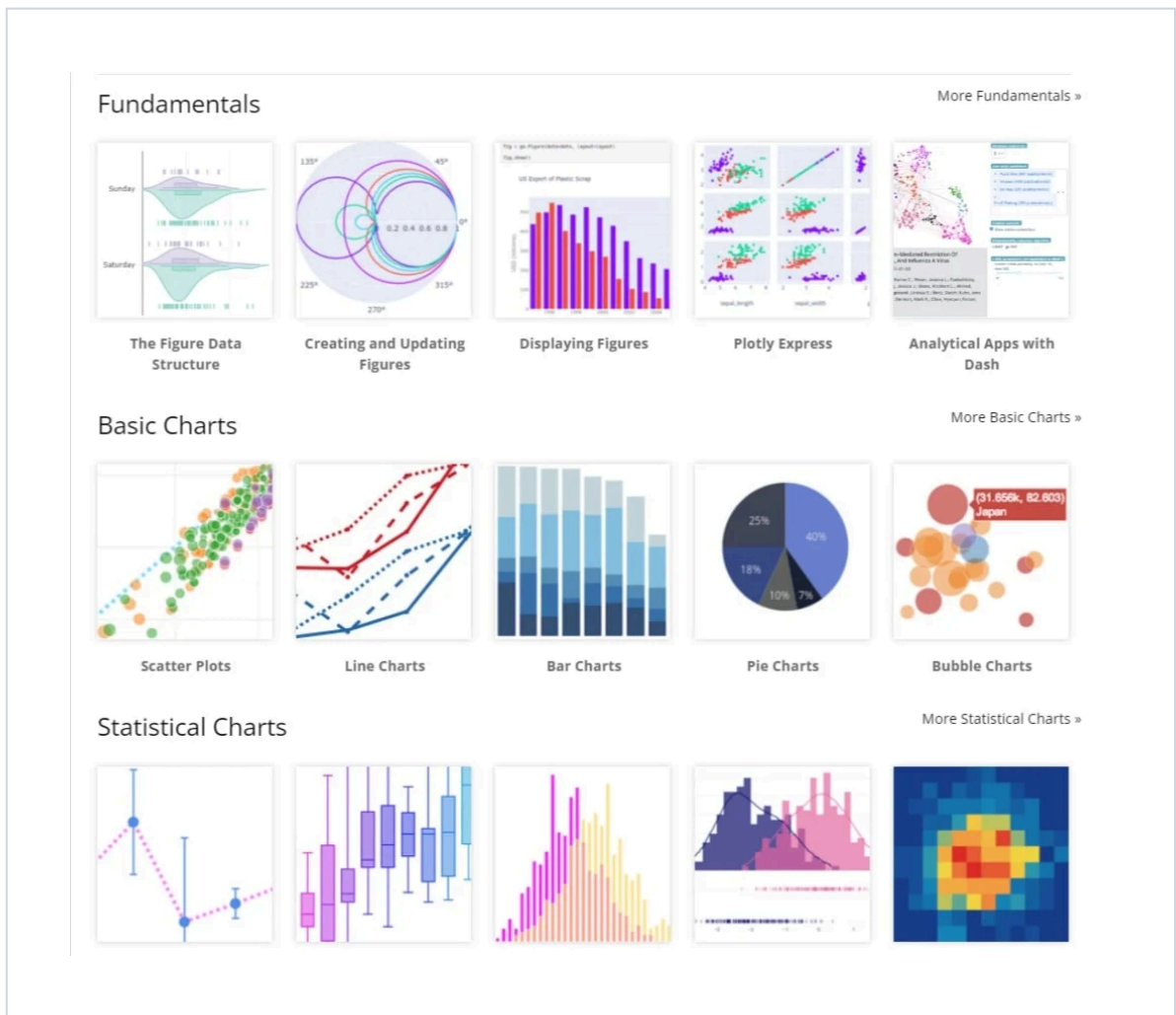
A Plotly funciona muito bem com várias plataformas usadas em Ciência de Dados e desenvolvimento de softwares, tais como: [Jupyter Notebook](#), Dash (framework de dashboards web), aplicativos webs, [Streamlit](#) e possui a possibilidade de exportação de HTML para páginas web.

Na Plotly, o gráfico de dispersão, mostrado no início do artigo, torna-se interativo e permite explorar várias ações.



## Galeria de gráficos da Plotly

A galeria de gráficos da Plotly está disponível na página [Plotly Open Source Graphing Library for Python](#) e conta com diversos exemplos de gráficos básicos, estatísticos, mapas e vários exemplos aplicados.



## O que vem depois?

O trabalho com visualização de dados e criação de gráficos e figuras não para por aqui. Também podemos incorporar esses itens em relatórios estáticos (documentos, PDFs, etc.) ou em relatórios dinâmicos, como os relatórios e dashboards criados em [Power BI](#). Além disso, podemos utilizar esses itens em páginas web, complementando tecnologias como o Flask, Django e as tecnologias para *webapps* voltadas a dados já mencionadas, como o Dash e Streamlit.

## Conclusão

E aí, curtiu? Aqui na Alura nós temos vários conteúdos voltados à Ciência de Dados e Visualização de Dados. Convidamos você a conhecer a [Formação Python para Data Science](#), uma trilha de estudos completa que inicia em Python, passa pelas principais bibliotecas de Ciência de Dados (como NumPy e Pandas) e trabalha as bibliotecas de visualização de dados. Lembrando que não precisa de pré-requisitos em programação para começar!

Se você já deu seus primeiros passos em Ciência de Dados, também te convidamos a conhecer o [Challenge de Data Science 1ª Edição](#), um desafio pensado para ser trabalhado em 4 semanas, no qual você vai desenvolver um conjunto de análises e modelos de machine learning para auxiliar na resolução de um case. Lembrando que somente estudantes da Alura têm acesso ao Challenge.

Mergulhe em Tecnologia! 🗨️🌐

## Créditos

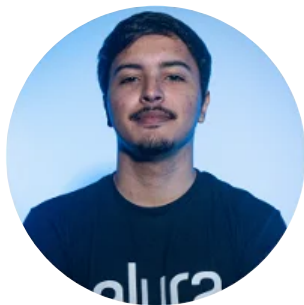
Conteúdo: [Marcus Almeida](#)

Produção técnica: [Rodrigo Dias](#)

Produção didática: [Morgana Gomes](#)

Designer gráfico: [Alysson Manso](#)





### Marcus Almeida

Bacharelando em Engenharia Elétrica pelo Instituto Federal do Maranhão. Atuou como parte do Scuba Team da Escola de Dados na Alura, trabalhando com conteúdos voltados a Data Science, Machine Learning, Python e SQL. Adora conversar tecnologia, universo geek, games e também aprender coisas novas.

[Artigo Anterior](#)

**SQL e NoSQL: trabalhando com bancos relacionais e não relacionais**

[Próximo Artigo](#)

**Business Intelligence: o que é?**

Veja outros artigos sobre  
[Data Science](#)

## Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do mercado de trabalho, ciência e desenvolvimento de software



Escreva seu email

**ME INSCREVA**

## Nossas redes e apps



### Institucional

[Sobre nós](#)

[Trabalhe conosco](#)

[Para Empresas](#)

[Para Sua Escola](#)

[Política de Privacidade](#)

[Compromisso de Integridade](#)

[Termos de Uso](#)

[Status](#)

### A Alura

[Formações](#)

[Como Funciona](#)

[Todos os cursos](#)

[Depoimentos](#)

[Instrutores\(as\)](#)

[Dev em <T>](#)

[Luri, a inteligência artificial da Alura](#)

## Conteúdos

Alura Cases

Imersões

Artigos

Podcasts

Artigos de educação  
corporativa

## Fale Conosco

Email e telefone

Perguntas frequentes

## Novidades e Lançamentos

Email\*

ENVIAR

## CURSOS

### Cursos de Programação

Lógica | Python | PHP | Java | .NET | Node JS | C | Computação | Jogos | IoT

### Cursos de Front-end

HTML, CSS | React | Angular | JavaScript | jQuery

### Cursos de Data Science

Ciência de dados | BI | SQL e Banco de Dados | Excel | Machine Learning | NoSQL | Estatística

### Cursos de Inteligência Artificial

IA para Programação | IA para Dados

### Cursos de DevOps

AWS | Azure | Docker | Segurança | IaC | Linux

### Cursos de UX & Design

Usabilidade e UX | Vídeo e Motion | 3D

### Cursos de Mobile

React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

## Cursos de Inovação & Gestão

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas