

# PERSPECTIVAS

Programando o Futuro

Podemos “ver” informação?  
Sim... podemos!

Desafio de programação - Python



A Cristina continua dedicada ao seu estudo sobre robôs. Como uma boa pesquisadora, ela coletou e organizou uma série de dados a respeito de robôs sociais, robôs industriais e drones. Ela organizou os dados em variáveis do tipo lista. Veja na figura 1 todos os dados que ela conseguiu:

```
[ ] nome_sociais = ["Pepper", "Paro", "Nao", "Photon", "QTrobot", "Asimo", "Fable", "EMYS", "Misty II", "Sophia"]
altura_sociais = [28, 16, 58, 19, 64, 130, 32, 38, 35.5, 167]
peso_sociais = [120, 2.7, 5.5, 0.69, 5, 48, 2.1, 4, 2.7, 20]
ano_sociais = [2014, 2004, 2008, 2017, 2016, 2017, 2000, 2017, 2018, 2016]

nome_industriais = ["Harvey", "Baxter", "OTTO", "LBR iiwa", "Meca500", "LD", "Nextage", "Expliner", "Handle", "Freight"]
altura_industriais = [53, 94, 30.1, 126.6, 33, 38.3, 71.4, 150, 200, 35.6]
peso_industriais = [38.5, 75, 127, 22, 4.5, 62, 28, 95, 150, 68]
ano_industriais = [2012, 2012, 2015, 2013, 2015, 2013, 2009, 2010, 2019, 2014]

nome_drones = ["Anafi", "EASE", "eBee", "Elios", "Lucie", "Mambo FPV", "Mavic 2", "Nano Hummingbird", "Raven",
               "Skydio R1"]
comprimento_drones = [24, 30, 96, 40, 15.5, 24.2, 16, 140, 40]
peso_drones = [0.32, 1.3, 0.67, 0.05, 18, 0.9, 0.019, 1.9, 1]
ano_drones = [2018, 2012, 2012, 2016, 2017, 2018, 2008, 2008, 2018]
```

Figura 1: Dados obtidos por Cristina registrados em uma célula de código do Google Colab

Ter organizado todos esses dados já foi um trabalho incrível. Mas a Cristina quer mais do que dados, ela quer **INFORMAÇÃO**. Com informação, ela poderá avançar no seu entendimento sobre as diferenças e semelhanças entre diferentes tipos de robôs e também entender um pouco mais sobre drones.

**LEMBRETE:**

**Dados e informações** - Dados são observações documentadas ou resultados de algum tipo de medição. Os dados não necessariamente possuem um significado relevante ou permitem algum tipo de compreensão. Já a informação é a ordenação e organização dos dados de forma a obter algum conhecimento em um determinado assunto, tópico ou contexto.

Por exemplo: Se fizermos uma comparação entre os dados referentes à medição de alturas de robôs feita pela Cristina, nós obteremos a informação de qual dos robôs é o mais alto dentre aqueles que estão sendo estudados.



*Drones são veículos aéreo não tripulados. Esse tipo de veículo, também chamado de aeronave remotamente pilotada, não necessita de um piloto embarcado para ser guiada, ela é guiada remotamente e pode ser controlada diante de três eixos (para frente e para trás, para cima e para baixo, para a direita e para a esquerda).*

*Drone é uma palavra da língua inglesa e ela possui vários significados. Um deles é, segundo o dicionário Merriam-Webster [1], “uma abelha macho sem ferrão que tem o papel de acasalar com a rainha e não coleta néctar ou pólen”, ou seja, ZANGÃO. Interessante, não?*

[1] <https://www.merriam-webster.com/>

Por enquanto ela está interessada em responder às seguintes perguntas:

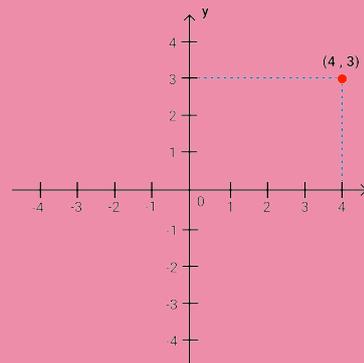
1. Entre os três tipos de equipamentos (robôs sociais, robôs industriais e drones), qual apresenta maior variação de tamanho e a maior variação de peso?
2. Qual tipo de equipamento apresenta menor variação de tamanho e e a menor variação de peso?
3. Ao analisarmos os dados de tamanho e peso dos equipamentos, quais tipos são mais similares?

Depois de pensar sobre como conseguir a informação que responde a essas perguntas, Cristina concluiu que uma boa maneira de realizar esta tarefa seria usando uma forma de **visualização da informação**, e pensou em criar um gráfico.

Diferentes tipos de gráficos podem ser criados usando programação em Python, mas Cristina acha que seria ideal usar um gráfico que permitisse analisar as duas medidas (altura e peso) ao mesmo tempo. Então, ela resolveu criar um **gráfico de dispersão**.



O **gráfico de dispersão**, também conhecido como **diagrama de dispersão**, é uma ferramenta para análise visual de dados na qual é possível representar os valores de duas variáveis quantitativas simultaneamente. Na construção deste gráfico, os valores das variáveis são exibidos como uma coleção de pontos. Os pontos são formados a partir da combinação dos valores das duas variáveis - uma variável terá seus valores posicionados no eixo horizontal do gráfico e a outra no eixo vertical; o ponto será formado na interseção de de linhas retas desenhadas a partir dessas posições.



**Variáveis quantitativas:** são variáveis que assumem valores numéricos.

Ambos os conceitos são muito importantes e úteis quando queremos conhecer mais sobre o que os dados podem nos dizer sobre um fenômeno. No nosso caso, o que os dados podem nos dizer sobre os tipos de equipamentos.

A decisão da Cristina por usar um gráfico de dispersão é motivada pela possibilidade de representar os dados de altura e peso de cada equipamento como pontos que estarão distribuídos no gráfico. A partir dessa distribuição, a análise visual dos dados poderá ajudar a encontrar a resposta para as perguntas que ela quer responder.



Porém, desenvolver um gráfico de dispersão manualmente para todos os dados que estão na base de dados da Cristina pode ser uma tarefa um tanto quanto trabalhosa, pois temos muitos dados para colocar no gráfico. Que tal você ajudar a Cristina criando um programa em Python que crie o gráfico de dispersão para ela?

Vamos estudar alguns comandos novos para aprender a fazer gráficos em Python e então ajudar a Cristina!!!!

A linguagem Python possui ferramentas específicas para criar gráficos. A mais famosa dessas ferramentas é a biblioteca matplotlib. Com o auxílio desta biblioteca, vamos criar um programa que desenha um gráfico de dispersão. Depois de observar como fazer o gráfico, tente desenhar um gráfico que a Cristina precisa para responder às questões.

*Uma biblioteca de uma linguagem de programação é um conjunto de programas que possuem funções específicas e que podem ser usados no desenvolvimento de outros programas. Quando você está desenvolvendo um programa e precisa resolver um problema que já está resolvido em um dos programas de uma biblioteca, você pode usar esse programa já existente dentro do programa que você está construindo. Esses programas existentes dentro das bibliotecas já estão testados e possuem sua qualidade verificada. Usá-los lhe dará garantias de que está resolvendo um problema corretamente e tornará o seu trabalho mais simples de ser feito.*

*O nome da biblioteca usada nesse desafio é matplotlib. Esse nome vem das palavras em inglês: math, plot, library. Que tal pesquisa sobre o significado dessas palavras?*



## Desenhando um gráfico de dispersão com Python.

A biblioteca *matplotlib* possui uma grande quantidade de funcionalidades (programas pré construídos) cujo objetivo é construir visualizações de dados. Nós podemos usar essas funcionalidades a partir de funções especiais.

*Em programação uma função é programa que resolve um problema específico. Esse programa é organizado de tal forma que podemos executá-la por meio de uma “chamada” - uma “chamada de função”. Ou seja, o programa referente à função recebe um nome e nós podemos usar esse nome dentro de nosso programa para solicitar que a função seja executada.*

Para desenhar um gráfico de dispersão nós usamos a função **scatter()**. Esse comando exige que informemos dois parâmetros:

- o primeiro são os valores da variável que serão posicionados no eixo horizontal (eixo x) do gráfico;
- o segundo são os valores da variável que serão posicionados no eixo vertical (eixo y) do gráfico.

Esses valores serão usados pelo programa que implementam a função **scatter()** para posicionar pontos no gráfico. Vamos ver isso em mais detalhes programando e executando o programa mostrado nas figuras 2 e 3.

```
[ ] # Para usar os recursos da uma biblioteca é necessário importar a biblioteca  
# e atribuir um "apelido" a ela para ser usado no nosso programa.  
# Esse "apelido" deverá aparecer como um indicador do uso de uma função da biblioteca associada a ele.  
# Vamos usar o apelido 'plt'.  
import matplotlib.pyplot as plt
```

Figura 2: Importando a biblioteca e definindo um “apelido”



```
[ ] # Inicializando duas listas com dados referentes ao peso e tamanho (altura) de cinco robôs sociais
    peso_robos_sociais = [2.7, 0.69, 2.1, 4, 2.7]
    altura_robos_sociais = [16, 19, 32, 38, 35.5]

# Utilizando a função 'scatter' para desenhar os pontos no gráfico
# Observe que usamos o apelido 'plt' para indicar que a função 'scatter' é da biblioteca matplotlib
# (apelidada por nós de 'plt')
plt.scatter(x=peso_robos_sociais, y=altura_robos_sociais, color = "green", label = "Sociais")
```

Figura 3: Inicializando um conjunto de dados e utilizando a função `scatter()` para gerar um gráfico

*Você percebeu que no código que escrevemos estamos usando variáveis do tipo lista??*

*Vimos esse conceito na aula sobre tipos e variáveis. Para lembrar: em Python, uma lista é representada como uma sequência de itens separados por vírgula e dentro de colchetes [ ].*

Ao executar o código acima você verá o gráfico da figura 4:

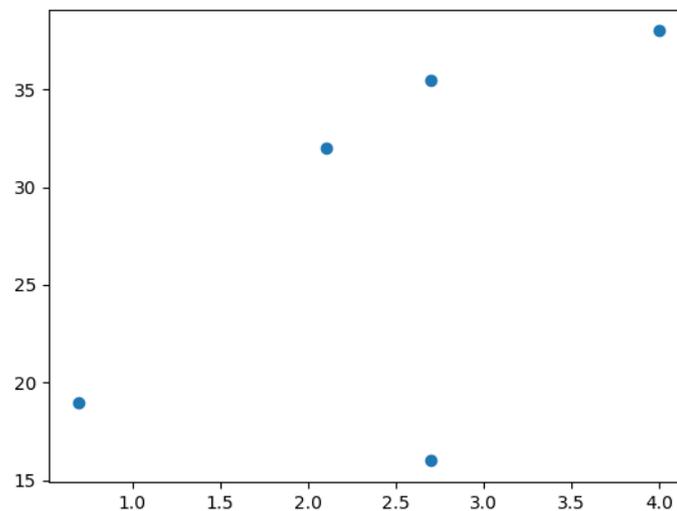
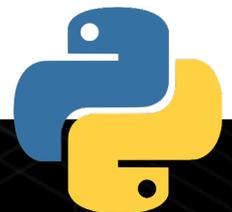
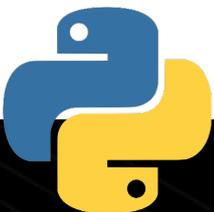


Figura 4: Gráfico de dispersão obtido ao executar o código

Cada ponto no gráfico representa os dados sobre peso e altura dos robôs. Veja em mais detalhes na figura 5:



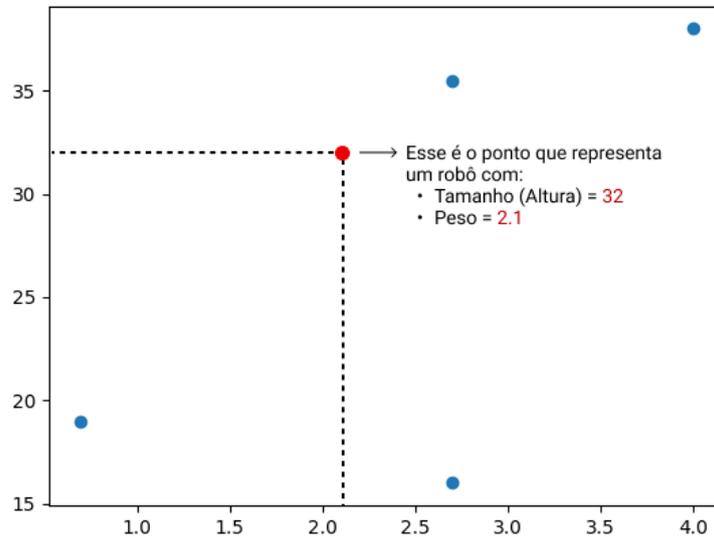


Figura 5: Analisando um ponto do gráfico

Um detalhe importante é que nós precisamos tomar muito cuidado com a forma como organizamos os dados nas variáveis lista. A função **scatter()** entenderá que o primeiro elemento da variável do tipo lista `peso_robos_sociais` e o primeiro elemento da variável do tipo lista `tamanho_robos_sociais` formam um ponto para ser desenhado no gráfico. Logo, precisamos informar os dados de um mesmo robô no primeiro elemento das duas variáveis. Isso vale também para o segundo elemento das listas, para o terceiro ... e assim por diante.

Agora vamos adicionar mais dados no nosso gráfico. O código das figuras 6, 7 e 9:

- Coloca dados referentes a diferentes variáveis no mesmo gráfico: vamos colocar dados de robôs sociais e drones. Podemos exibir mais de um conjunto de dados no mesmo gráfico, só precisamos repetir o mesmo processo.
- Define cores e rótulos para os conjuntos de pontos conforme nós achamos mais adequado: vamos usar mais alguns parâmetros na função **scatter()**, para definir cor (parâmetro 'color') e rótulos (parâmetro 'label').
- Cria uma legenda para o gráfico: usamos a função **legend()**, para exibir uma legenda de acordo com as cores e rótulos já definidos.



```
[ ] # Inicializando duas listas com dados referentes ao peso e tamanho (altura) de cinco robôs sociais
    peso_robos_sociais = [2.7, 0.69, 2.1, 4, 2.7]
    altura_robos_sociais = [16, 19, 32, 38, 35.5]

# Utilizando a função 'scatter' para desenhar os pontos no gráfico
# Observe que usamos o apelido 'plt' para indicar que a função 'scatter' é da biblioteca matplotlib
# (apelidada por nós de 'plt')
plt.scatter(x=peso_robos_sociais, y=altura_robos_sociais, color = "green", label = "Sociais")

# Inicializando duas listas com dados referentes ao peso e tamanho (comprimento) de cinco drones
peso_robos_drones = [0.32, 1.3, 0.05, 0.9, 1]
comprimento_robos_drones = [24, 30, 40, 24.2, 40]

# Utilizando a função scatter para desenhar os outros pontos no gráfico
plt.scatter(x=peso_robos_drones, y=comprimento_robos_drones, color = "blue", label = "Drones")
```

Figura 6: Adicionando novos dados e modificando nosso gráfico

Observe atentamente as alterações realizadas nas seguintes linhas de código:

```
[ ] # Inicializando duas listas com dados referentes ao peso e tamanho (altura) de cinco robôs sociais
    peso_robos_sociais = [2.7, 0.69, 2.1, 4, 2.7]
    altura_robos_sociais = [16, 19, 32, 38, 35.5]

# Utilizando a função 'scatter' para desenhar os pontos no gráfico
# Observe que usamos o apelido 'plt' para indicar que a função 'scatter' é da biblioteca matplotlib
# (apelidada por nós de 'plt')
plt.scatter(x=peso_robos_sociais, y=altura_robos_sociais, color = "green", label = "Sociais")

# Inicializando duas listas com dados referentes ao peso e tamanho (comprimento) de cinco drones
peso_robos_drones = [0.32, 1.3, 0.05, 0.9, 1]
comprimento_robos_drones = [24, 30, 40, 24.2, 40]

# Utilizando a função scatter para desenhar os outros pontos no gráfico
plt.scatter(x=peso_robos_drones, y=comprimento_robos_drones, color = "blue", label = "Drones")

# Utilizando a função 'legend' para exibir a legenda de acordo com as cores
# e os rótulos já definidos
plt.legend()
```

Figura 7: Utilizando os parâmetros *color*, *label* e a função *legend()* para modificar o gráfico



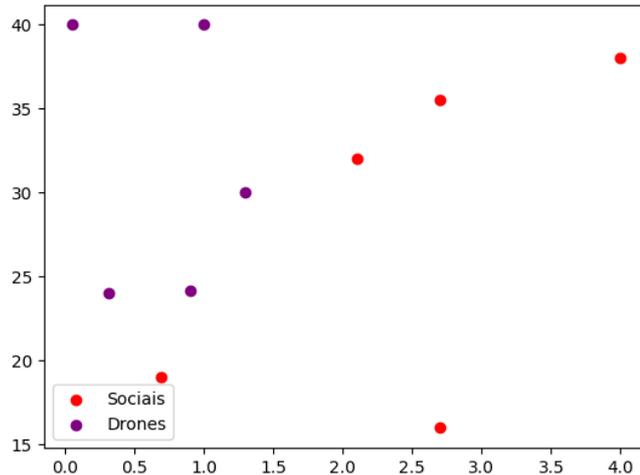


Figura 8: Adicionando novos dados e modificando

- Podemos adicionar rótulos nos eixos do gráfico: todo gráfico precisa ter rótulos nos eixos para que possamos saber qual é o dado que está sendo posicionado de acordo com cada eixo; faremos isso usando as funções `xlabel()` (para colocar o rótulo no eixo horizontal) e `ylabel()` (para colocar o rótulo no eixo vertical).

```
[ ] # Inicializando duas listas com dados referentes ao peso e tamanho (altura) de cinco robôs sociais
peso_robos_sociais = [2.7, 0.69, 2.1, 4, 2.7]
altura_robos_sociais = [16, 19, 32, 38, 35.5]

# Utilizando a função 'scatter' para desenhar os pontos no gráfico
# Observe que usamos o apelido 'plt' para indicar que a função 'scatter' é da biblioteca matplotlib
# (apelidada por nós de 'plt')
plt.scatter(x=peso_robos_sociais, y=altura_robos_sociais, color = "green", label = "Sociais")

# Inicializando duas listas com dados referentes ao peso e tamanho (comprimento) de cinco drones
peso_robos_drones = [0.32, 1.3, 0.05, 0.9, 1]
comprimento_robos_drones = [24, 30, 40, 24.2, 40]

# Utilizando a função scatter para desenhar os outros pontos no gráfico
plt.scatter(x=peso_robos_drones, y=comprimento_robos_drones, color = "blue", label = "Drones")

# Utilizando a função 'legend' para exibir a legenda de acordo com as cores
# e os rótulos já definidos
plt.legend()

plt.xlabel("Peso (Quilogramas)")
plt.ylabel("Tamanho - altura ou comprimento (Centímetros)")
```

Figura 9: Utilizando as funções `xlabel()` e `ylabel()` para adicionar rótulos aos eixos



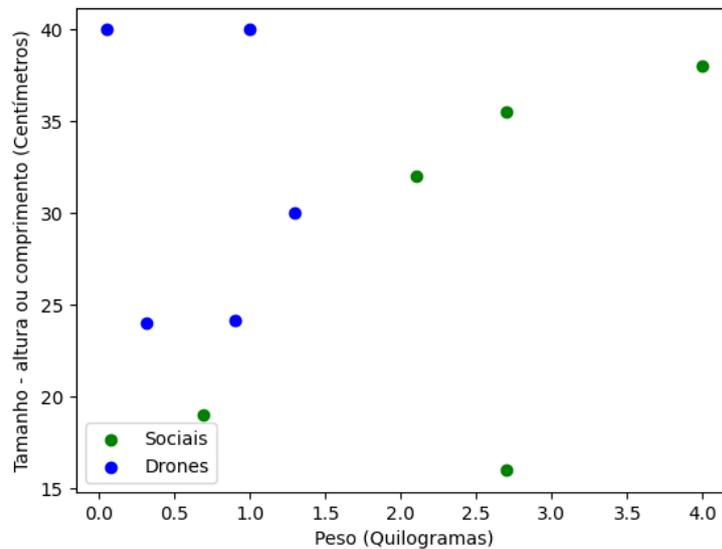


Figura 10: Gráfico finalizado, com a adição de rótulos aos eixos

Pronto, agora você pode fazer o gráfico que a Cristina precisa. Lembre-se que ela precisa de um gráfico do tipo dispersão, contendo dados de robôs sociais, robôs industriais e drones.

*Depois que você criar o gráfico, será que você consegue responder às três perguntas que Cristina formulou?*

