



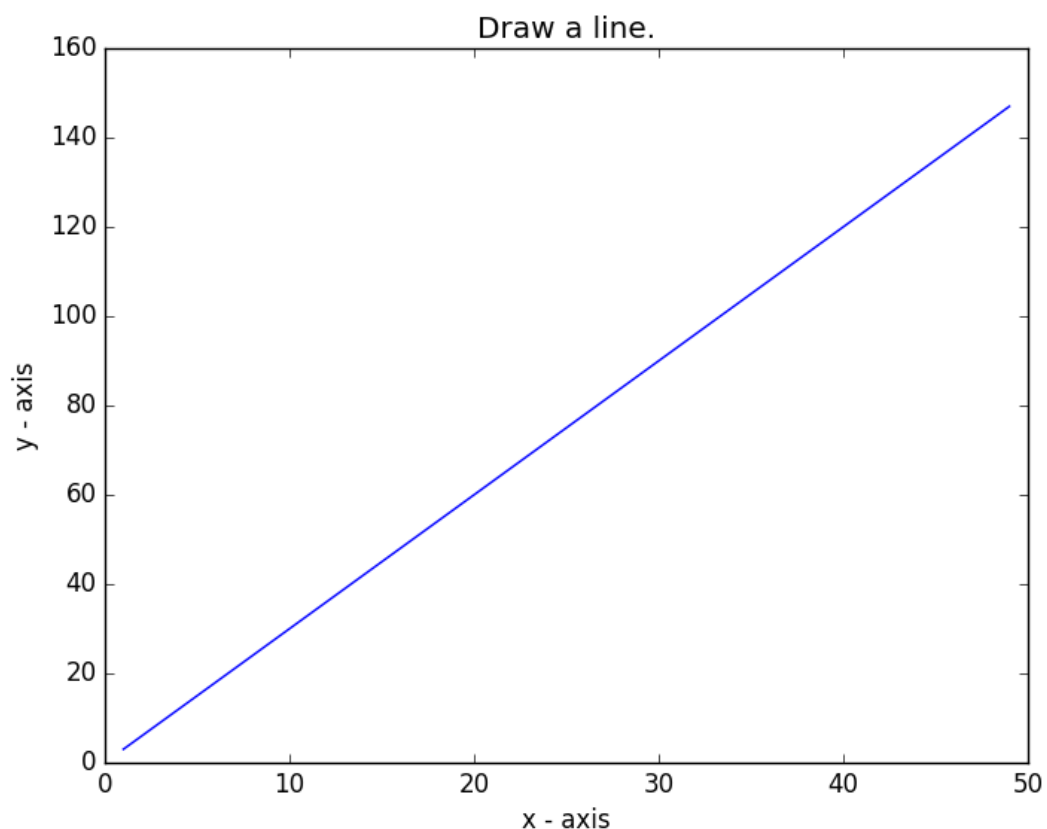
LISTA DE EXERCÍCIO 07 – Matplotlib e Seaborn

Exercício 01 - Escreva um programa Python para desenhar uma linha com rótulo adequado no eixo x, eixo y e um título.

Amostra de dados:

```
x = range(1, 50)
y = [value * 3 for value in x]
```

O código fornece a saída mostrada na captura de tela a seguir:



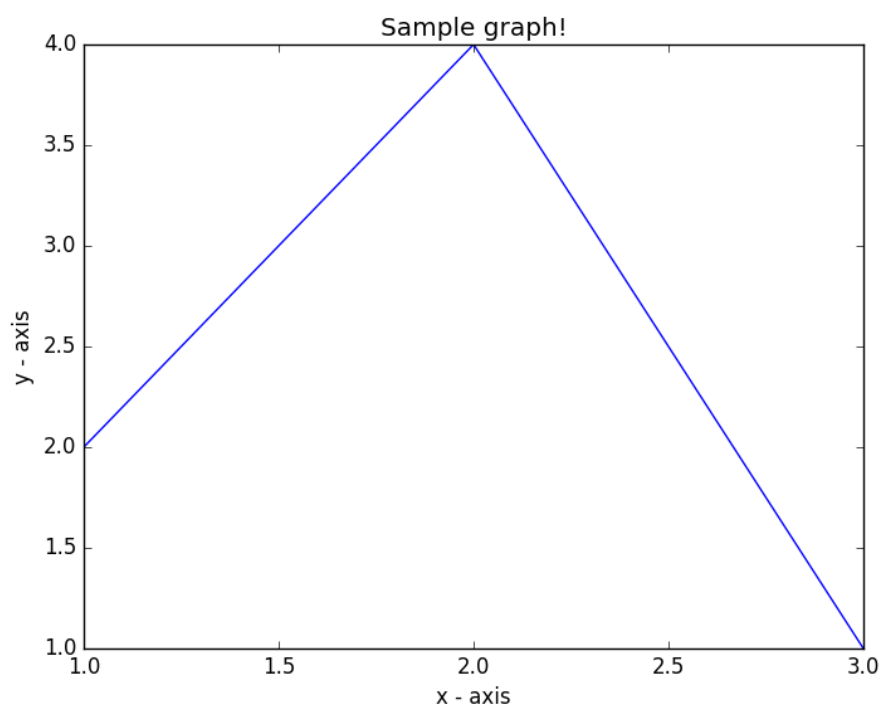


Exercício 02 - Escreva um programa Python para desenhar uma linha usando determinados valores de eixo com rótulo adequado no eixo x, eixo y e um título.

Amostra de dados:

```
# x axis values
x = [1,2,3]
# y axis values
y = [2,4,1]
```

O trecho de código fornece a saída mostrada na captura de tela a seguir





Exercício 03 - Escreva um programa Python para desenhar uma linha usando determinados valores de eixo retirados de um arquivo de texto, com rótulo adequado no eixo x, eixo y e um título.

Dados de teste:

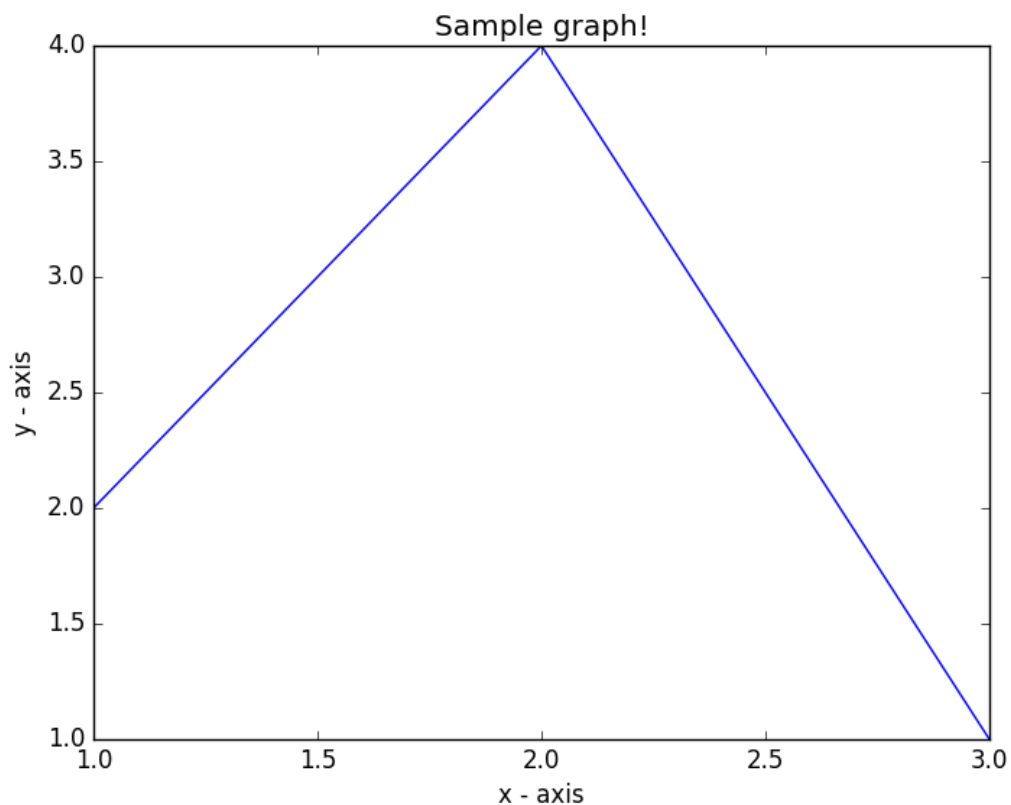
X,Y

1,2

3,4

3,1

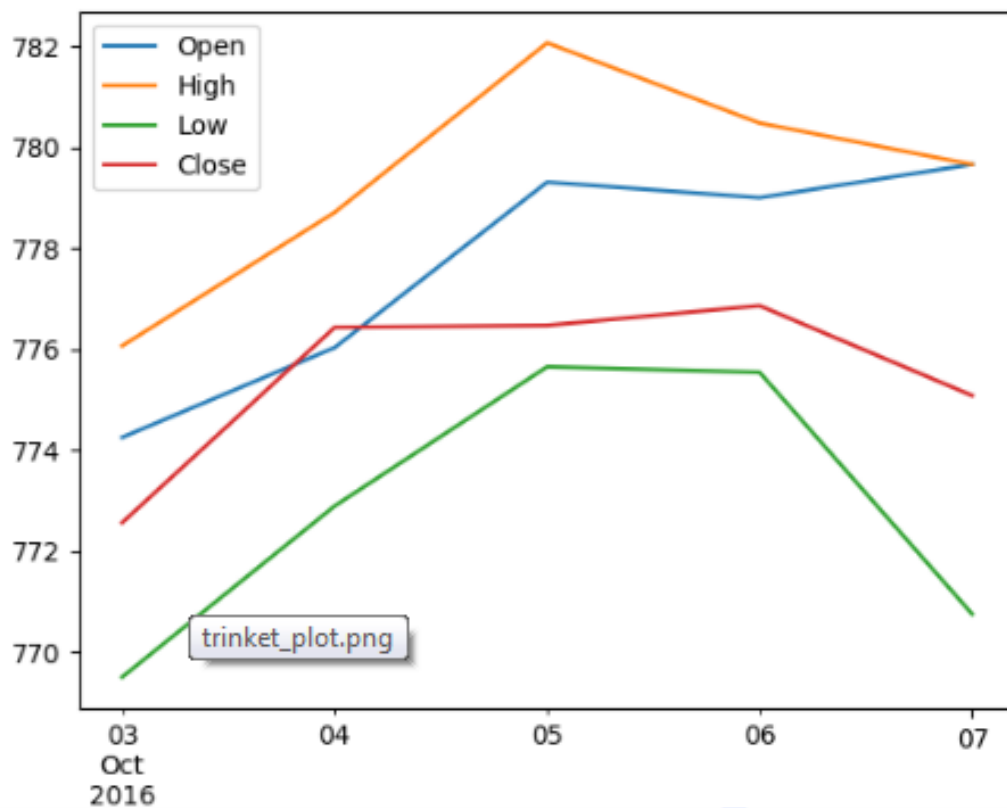
O trecho de código fornece a saída mostrada na captura de tela a seguir:



Exercício 04 - Escreva um programa Python para desenhar gráficos de linhas dos dados financeiros da Alphabet Inc. entre 3 de outubro de 2016 a 7 de outubro de 2016.

Exemplo de dados estão no arquivo `fdata.csv`.

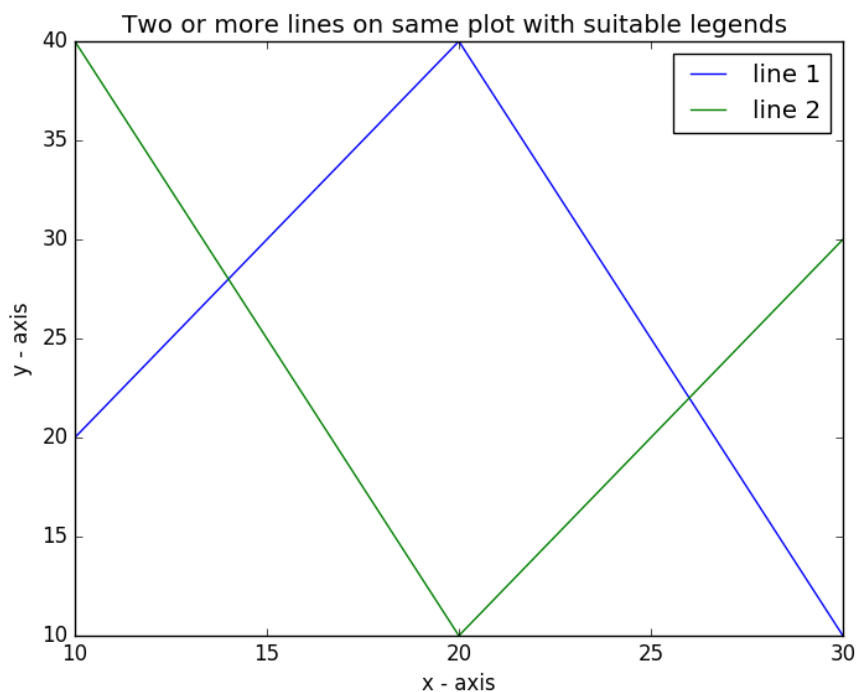
O trecho de código fornece a saída mostrada na captura de tela a seguir:



[trinket_plot.png](#)

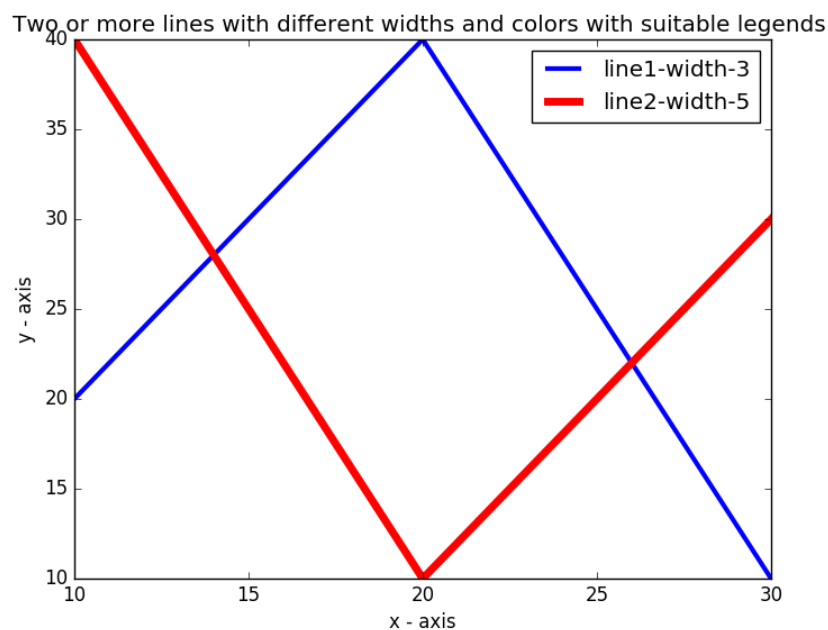
Exercício 05 - Escreva um programa Python para traçar duas ou mais linhas no mesmo gráfico com legendas adequadas para cada linha.

O trecho de código fornece a saída mostrada na captura de tela a seguir:



Exercício 06 - Escreva um programa Python para traçar duas ou mais linhas com legendas, larguras e cores diferentes.

O trecho de código fornece a saída mostrada na captura de tela a seguir:



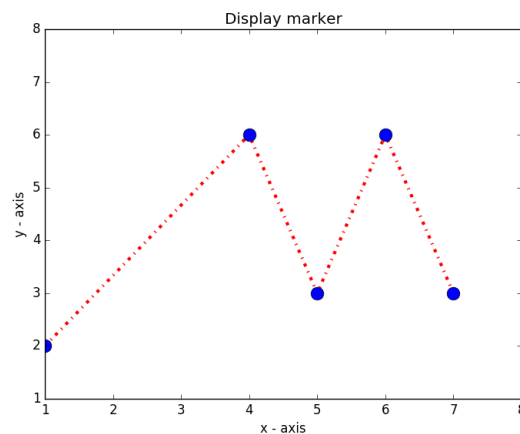


Exercício 07 - Escreva um programa Python para traçar duas ou mais linhas e definir os marcadores de linha.

Amostra de dados:

```
x = [1, 4, 5, 6, 7]
# y axis values
y = [2, 6, 3, 6, 3]
```

O trecho de código fornece a saída mostrada na captura de tela a seguir:

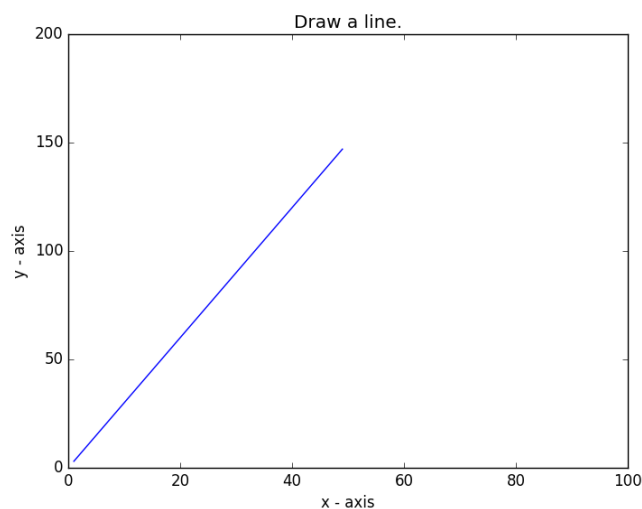


Exercício 08 - Escreva um programa Python para exibir os valores atuais dos limites do eixo e definir novos valores do eixo.

Amostra de dados:

```
X = range(1, 50)
Y = [value * 3 for value in X]
```

O trecho de código fornece a saída mostrada na captura de tela a seguir:

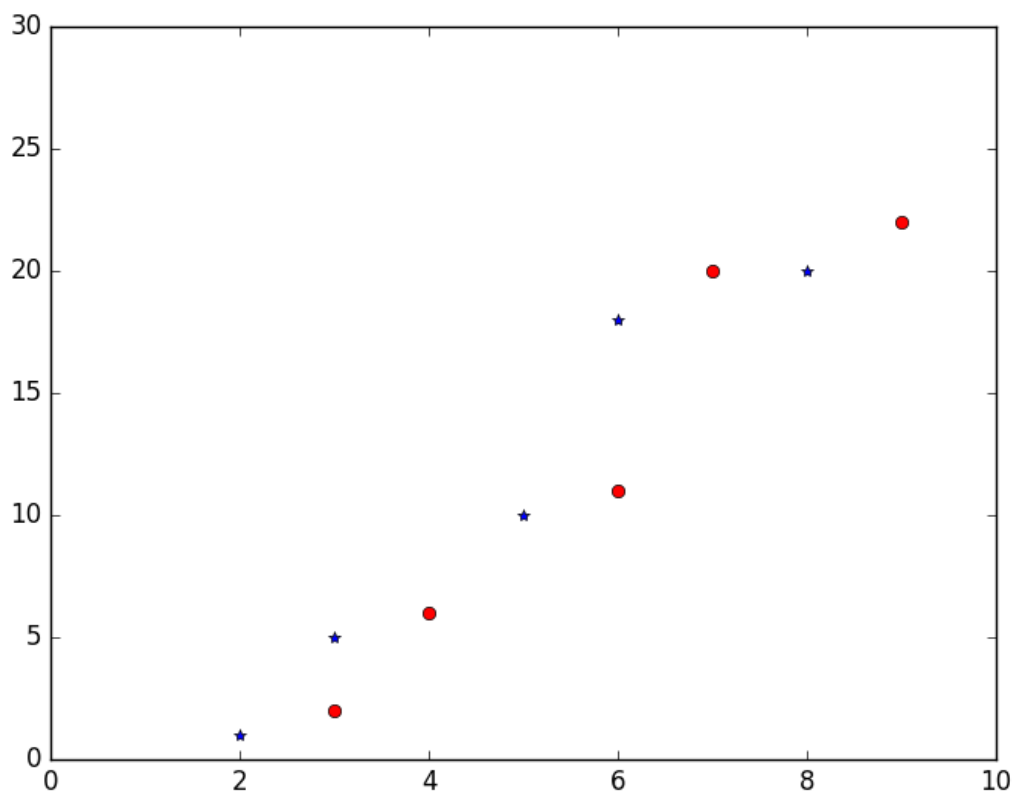


Exercício 09 - Escreva um programa Python para representar graficamente quantidades que possuem uma posição x e y.

Amostra de dados:

```
x1 = [2, 3, 5, 6, 8]
# Make an array of y values for each x value
y1 = [1, 5, 10, 18, 20]
# Make an array of x values
x2 = [3, 4, 6, 7, 9]
# Make an array of y values for each x value
y2 = [2, 6, 11, 20, 22]
```

O trecho de código fornece a saída mostrada na captura de tela a seguir:



Exercício 10 - Escreva um programa Python para plotar várias linhas com diferentes estilos de formato em um comando usando arrays.

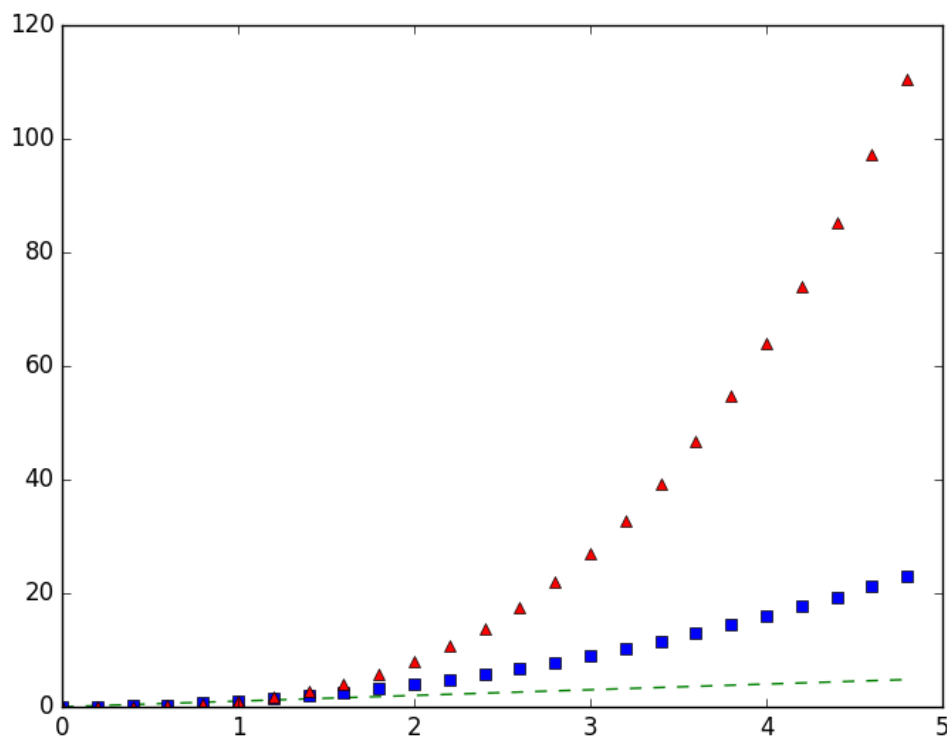
Código:

```
import numpy as np
import matplotlib.pyplot as plt

# Sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

# green dashes, blue squares and red triangles
plt.plot(t, t, 'g--', t, t**2, 'bs', t, t**3, 'r^')
plt.show()
```

O trecho de código fornece a saída mostrada na captura de tela a seguir:





Exercício 11 - Escreva um programa Python para criar vários tipos de gráficos (uma curva simples e traçar algumas quantidades) em um único conjunto de eixos.

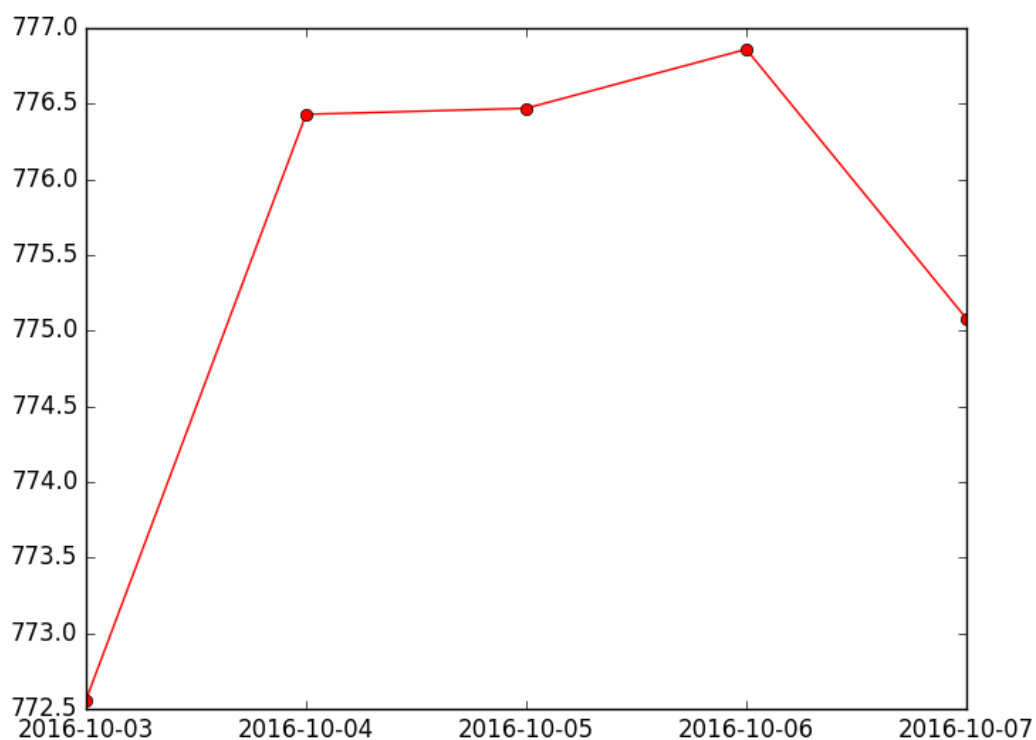
Dados de amostra:

```
import datetime as DT
from matplotlib import pyplot as plt
from matplotlib.dates import date2num

data = [(DT.datetime.strptime('2016-10-03', "%Y-%m-%d"), 772.559998),
        (DT.datetime.strptime('2016-10-04', "%Y-%m-%d"), 776.429993),
        (DT.datetime.strptime('2016-10-05', "%Y-%m-%d"), 776.469971),
        (DT.datetime.strptime('2016-10-06', "%Y-%m-%d"), 776.859985),
        (DT.datetime.strptime('2016-10-07', "%Y-%m-%d"), 775.080017 )]

x = [date2num(date) for (date, value) in data]
y = [value for (date, value) in data]
```

O trecho de código fornece a saída mostrada na captura de tela a seguir:





Gráficos de Dispersão

Da Wikipedia:

Um gráfico de dispersão (também chamado de gráfico de dispersão, gráfico de dispersão, gráfico de dispersão, diagrama de dispersão ou diagrama de dispersão) [3] é um tipo de gráfico ou diagrama matemático que usa coordenadas cartesianas para exibir valores para normalmente duas variáveis para um conjunto de dados. Se os pontos forem codificados por cores, uma variável adicional poderá ser exibida. Os dados são exibidos como uma coleção de pontos, cada um tendo o valor de uma variável que determina a posição no eixo horizontal e o valor da outra variável que determina a posição no eixo vertical. [4]

Exercício 12 – Criando um Gráfico de Dispersão com Matplotlib

Objetivo: Criar um gráfico de dispersão (scatter plot) personalizado utilizando a biblioteca Matplotlib em Python.

1. Importar as bibliotecas

Abra seu ambiente de desenvolvimento (como Jupyter Notebook ou um IDE Python) e inicie um novo projeto ou script. Importe a biblioteca Matplotlib como "plt" da seguinte forma:

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

2. Defina os dados

Defina os dados de exemplo para o eixo x e y. Você pode usar os dados fornecidos no exemplo ou criar seus próprios.

```
x = [43, 76, 34, 63, 56, 82, 87, 55, 64, 87, 95, 23, 14, 65, 67, 25, 23, 85]  
y = [34, 45, 34, 23, 43, 76, 26, 18, 24, 74, 23, 56, 23, 23, 34, 56, 32, 23]
```

3. Criar a figura e os eixos

Crie uma figura e um conjunto de eixos usando a função subplots. Personalize a figura conforme suas necessidades, como título, tamanho da figura etc.



```
fig, ax = plt.subplots(1, figsize=(10, 6))  
fig.suptitle('Exemplo de Gráfico de Dispersão')
```

4. Criar o gráfico de dispersão

Use a função `scatter` para criar o gráfico de dispersão. Personalize os parâmetros, como a cor dos pontos, o tamanho, a transparência e a largura das bordas, conforme indicado no exemplo.

```
ax.scatter(x, y,  
           color="blue", # Cor dos pontos  
           s=100,        # Tamanho dos pontos  
           alpha=0.5,    # Transparência dos pontos (0 é transparente, 1  
           linewidths=1) # Largura das bordas dos pontos
```

5. Mostrar o gráfico

Finalmente, exiba o gráfico chamando a função `show()`.

```
plt.show()
```

Exercício 13 - Criando um Gráfico de Dispersão com Rótulos usando Matplotlib

Objetivo: Criar um gráfico de dispersão com pontos rotulados usando a biblioteca Matplotlib em Python.

1. Importar as bibliotecas

Inicie um novo projeto ou script e importe a biblioteca Matplotlib como "plt" da seguinte forma:

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

2. Definir os dados

Defina os dados para o eixo x, y e os rótulos. Utilize os dados fornecidos no exemplo:

```
x = [21, 34, 44, 23]
y = [435, 334, 656, 1999]
labels = ["alice", "bob", "charlie", "diane"]
```

3. Criar a figura e os eixos

Crie uma figura e um conjunto de eixos usando a função `subplots`. Personalize a figura, como o título, o tamanho da figura, etc.

```
fig, ax = plt.subplots(1, figsize=(10, 6))
fig.suptitle('Exemplo de Gráfico de Dispersão com Rótulos')
```

4. Criar o gráfico de dispersão

Use a função `scatter` para criar o gráfico de dispersão. Personalize os parâmetros, como a cor dos pontos, o tamanho, a transparência e a largura das bordas, como mostrado no exemplo:

```
ax.scatter(x, y,
           color="blue", # Cor dos pontos
           s=100,        # Tamanho dos pontos
           alpha=0.5,    # Transparência dos pontos (0 é transparente, 1 é opaco)
           linewidths=1) # Largura das bordas dos pontos
```

5. Adicionar rótulos aos pontos

Utilize um loop para adicionar os rótulos aos pontos. A função `annotate` pode ser usada para adicionar rótulos personalizados a cada ponto. Certifique-se de ajustar os parâmetros da função `annotate` para que os rótulos estejam bem posicionados.

```
for x_pos, y_pos, label in zip(x, y, labels):
    ax.annotate(label, # O rótulo para este ponto
               xy=(x_pos, y_pos), # Posição do ponto correspondente
               xytext=(7, 0),     # Deslocamento do texto por 7 pontos para a direita
               textcoords='offset points', # Usar deslocamento em pontos
               ha='left',         # Alinhamento horizontal à esquerda
               va='center')       # Alinhamento vertical centralizado
```

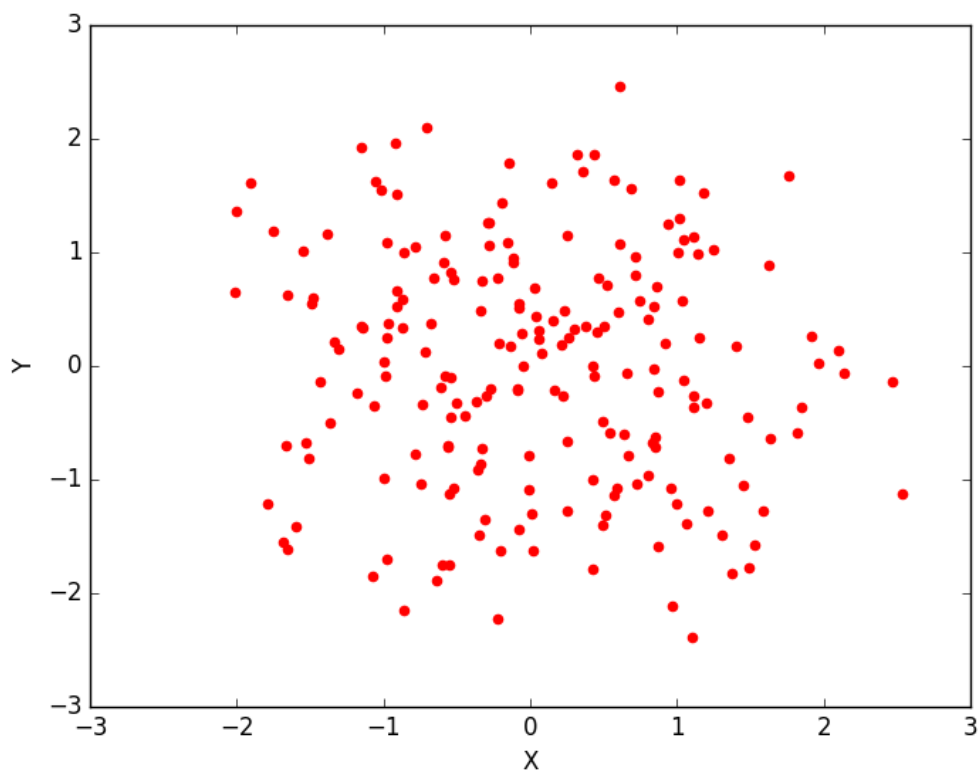
6. Mostrar o gráfico

Finalmente, exiba o gráfico chamando a função `show()`.

```
plt.show()
```

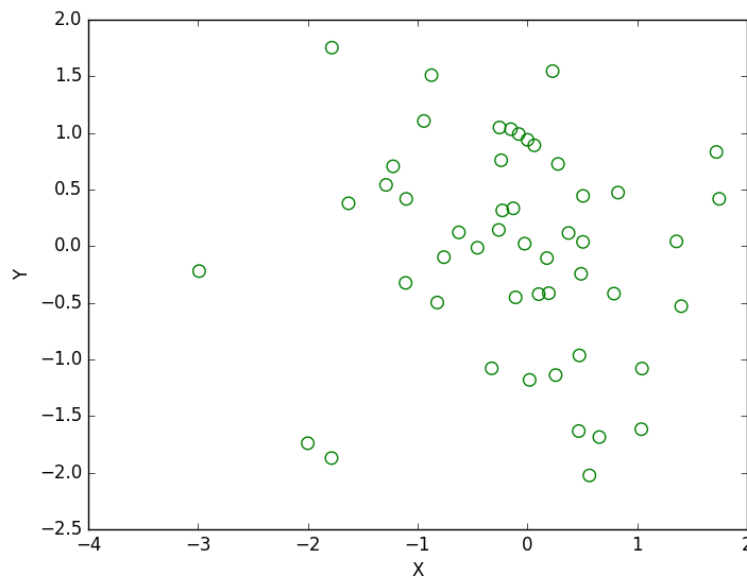
Exercício 14 - Escreva um programa Python para desenhar um gráfico de dispersão tomando uma distribuição aleatória em X e Y e plotando uma contra a outra.

O trecho de código fornece a saída mostrada na captura de tela a seguir:



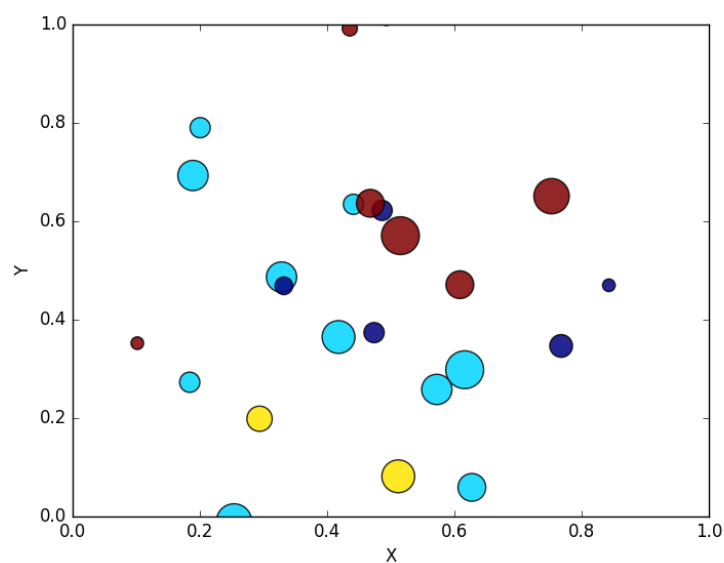
Exercício 15 - Escreva um programa Python para desenhar um gráfico de dispersão com círculos vazios tomando uma distribuição aleatória em X e Y e plotados uns contra os outros.

O trecho de código fornece a saída mostrada na captura de tela a seguir:



Exercício 16 - Escreva um programa Python para desenhar um gráfico de dispersão usando distribuições aleatórias para gerar bolas de tamanhos diferentes.

O trecho de código fornece a saída mostrada na captura de tela a seguir:





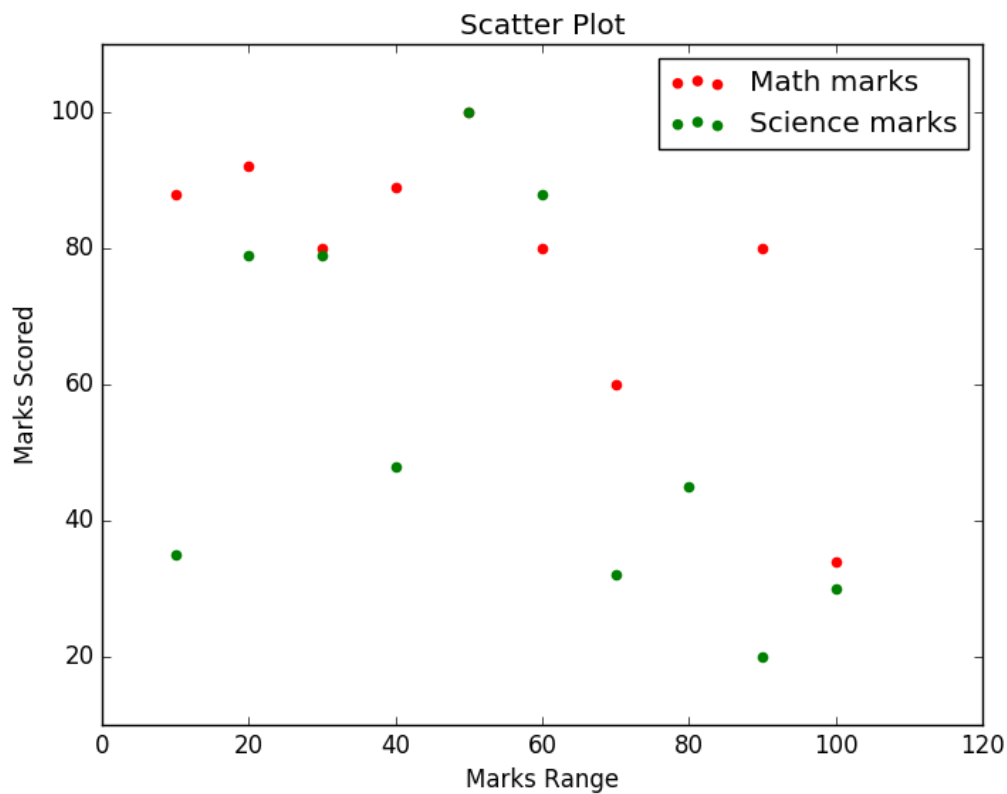
Exercício 17 - Escreva um programa Python para desenhar um gráfico de dispersão comparando duas notas de disciplinas de Matemática e Ciências. Use notas de 10 alunos.

Dados de teste:

```
notas_matemáticas = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
```

```
notas_científicas = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
```

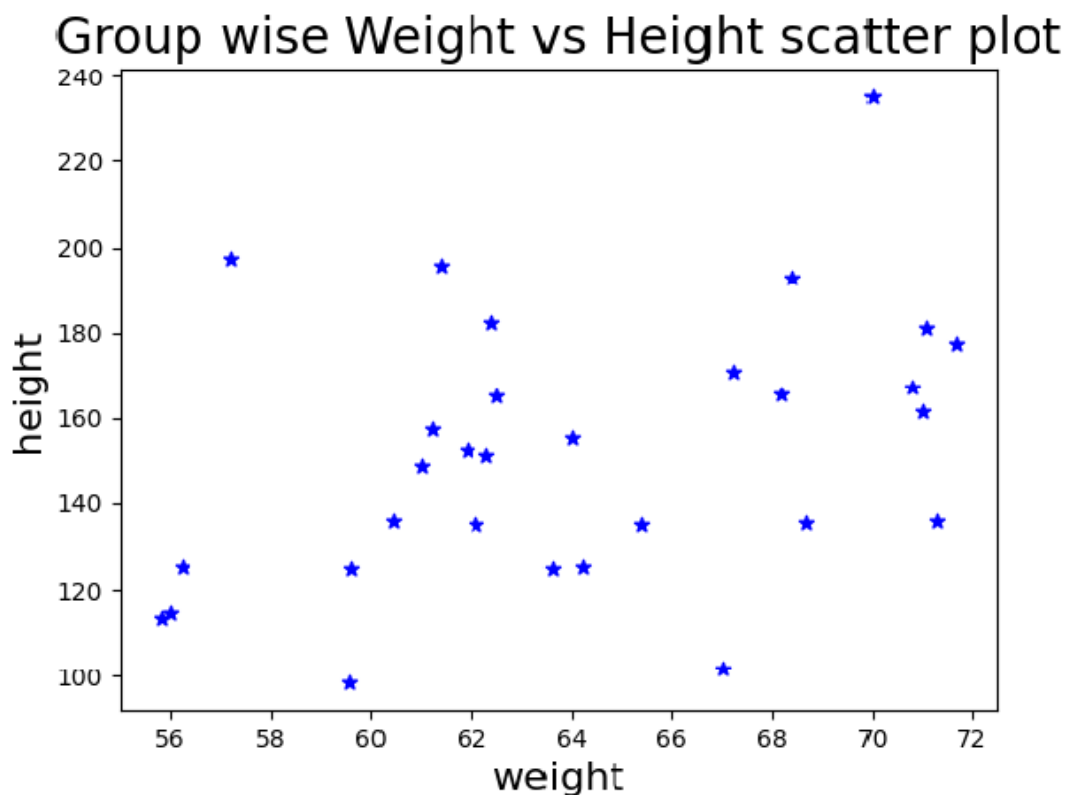
```
faixa_de_marcas = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```



Exercício 18 - Escreva um programa Python para desenhar um gráfico de dispersão para três grupos diferentes comparando pesos e alturas.

```
import matplotlib.pyplot as plt
import numpy as np
weight1=[67,57.2,59.6,59.64,55.8,61.2,60.45,61,56.23,56]
height1=[101.7,197.6,98.3,125.1,113.7,157.7,136,148.9,125.3,114.9]
weight2=[61.9,64,62.1,64.2,62.3,65.4,62.4,61.4,62.5,63.6]
height2=[152.8,155.3,135.1,125.2,151.3,135,182.2,195.9,165.1,125.1]
weight3=[68.2,67.2,68.4,68.7,71,71.3,70.8,70,71.1,71.7]
height3=[165.8,170.9,192.8,135.4,161.4,136.1,167.1,235.1,181.1,177.3]
weight=np.concatenate((weight1,weight2,weight3))
height=np.concatenate((height1,height2,height3))
plt.scatter(weight, height, marker='*', color=['blue'])
plt.xlabel('weight', fontsize=16)
plt.ylabel('height', fontsize=16)
plt.title('Group wise Weight vs Height scatter plot',fontsize=20)
plt.show()
```

O trecho de código fornece a saída mostrada na captura de tela a seguir:





Exercício 19 - Criando um Histograma com Pandas e Matplotlib

Neste exercício, você aprenderá como criar um histograma para visualizar a distribuição de alturas de árvores a partir de um conjunto de dados usando as bibliotecas Pandas, NumPy e Matplotlib.

Passo 1: Importação das Bibliotecas

Comece importando as bibliotecas necessárias para este exercício. Você precisará das bibliotecas Pandas, NumPy e Matplotlib. Use os seguintes comandos:

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

Passo 2: Carregamento da Base de Dados

Carregue a base de dados "trees.csv" usando o Pandas e armazene-a em um DataFrame chamado 'base'. Certifique-se de que o arquivo CSV esteja no caminho correto.

Passo 3: Verificando a Forma dos Dados

Use o método .shape para verificar o número de linhas e colunas na base de dados.

Passo 4: Visualizando os Primeiros Dados

Use o método .head() para visualizar as primeiras linhas da base de dados e entender sua estrutura.

Passo 5: Visualização do Histograma

Visualize o histograma com 6 divisões (bins) usando o Matplotlib. Configure o título, rótulos dos eixos e exiba o gráfico.

```
plt.hist(base.iloc[:,1], bins=6)
plt.title('Árvores')
plt.ylabel('Frequência')
plt.xlabel('Altura')
plt.show()
```



Exercício 20 – Criando Histogramas com Seaborn e Matplotlib

Neste exercício, você aprenderá como criar histogramas e distribuições de densidade a partir de um conjunto de dados usando as bibliotecas Pandas, Matplotlib e Seaborn.

Passo 1: Importação das Bibliotecas

Comece importando as bibliotecas necessárias para este exercício. Você precisará das bibliotecas Pandas, Matplotlib e Seaborn. Use os seguintes comandos:

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

Passo 2: Carregamento da Base de Dados

Carregue a base de dados "trees.csv" usando o Pandas e armazene-a em um DataFrame chamado 'base'. Certifique-se de que o arquivo CSV esteja no caminho correto.

Passo 3: Visualização do Primeiro Conjunto de Dados

Use o método .head() para visualizar as primeiras linhas da base de dados e entender sua estrutura.

Passo 4: Histograma com Matplotlib

Crie um histograma com 6 divisões (bins) usando o Matplotlib para representar a distribuição do primeiro atributo da base de dados (suponhamos que seja a altura das árvores).

```
plt.hist(base.iloc[:,1], bins=6)
```

Passo 5: Histograma e Linha de Distribuição de Frequência com Seaborn

Crie um gráfico que inclua o histograma e a linha de distribuição de frequência usando o Seaborn. Configure a cor das barras para azul e especifique que não há linha de densidade.

```
sns.distplot( base.iloc[:,1], hist. = True, kde = False, bins = 6,
              color='blue', hist_kws = {'edgecolor': 'black'})
```

Passo 6: Distribuição de Densidade com Seaborn

Crie um gráfico que mostre apenas a linha de distribuição de densidade usando o Seaborn. Configure a cor da linha para azul.

```
sns.distplot(base.iloc[:,1], hist=False, kde=True, bins=6,  
color='blue', hist_kws={'edgecolor': 'black'})
```

Passo 7: Histograma e Distribuição de Densidade com Seaborn

Crie um gráfico que inclua o histograma e a linha de distribuição de densidade usando o Seaborn. Configure a cor das barras para azul e a cor da linha para azul.

Exercício 21 - Uma grande empresa pretende adquirir uniformes para todos os seus 400 funcionários. Como as pessoas possuem diferentes estaturas, precisam de uniformes de tamanhos diferentes. Ao invés de medir a altura de todos os funcionários, o que levaria muito tempo e daria muito trabalho, a empresa decidiu utilizar um método estatístico, neste caso, utilize o arquivo *alturas.xlsx* e gere um histograma para analisar a distribuição das alturas.

Exercício 25 – Histogramas para dados de produção

A empresa TODOSUNIDOS recebeu um lote inicial de eixos para montagem de motores destinados a servomecanismos de aeronaves. Foi necessária uma verificação para identificar a capacidade de cada um de três fornecedores, visto que pretendia-se firmar um futuro contrato de parceria. Optou-se por utilizar histograma da distribuição dos diâmetros externos de uma amostra aleatória de 240 eixos, proveniente dos três fornecedores. O limite de especificação dos diâmetros é 5 ± 2 (cm).

Arquivo: eixos.xls

Planilha: eixos

- Que conclusões você obtém por meio de uma análise preliminar do histograma com todos os eixos?
- Que conclusões você obtém por meio de uma análise preliminar dos histogramas por fornecedor?
- Que outro gráfico auxiliaria você na análise solicitada no item anterior?



- d) Se você fosse o gestor desse processo, baseando-se somente nessa ferramenta, qual(is) o(s) fornecedor(es) que você selecionaria para o contrato de parceria?

Exercício 22– Criando gráficos de dispersão com Seaborn e Matplotlib

Crie um programa para gerar um gráfico de dispersão, utilizando a fonte de dados `trees.csv`. Siga os passos a seguir para criar as análises:

1. Importe as bibliotecas Pandas, Matplotlib e Seaborn;
2. Mapeie o caminho do documento e adicione em um dataframe;
3. Visualize o cabeçalho do dataframe;
4. Crie um gráfico de dispersão utilizando a biblioteca matplotlib, para analisar a relação entre as variáveis Girth e Volume. Modifique as cores dos pontos para azul.
5. Crie um gráfico de dispersão utilizando a biblioteca seaborn, para analisar a relação entre as variáveis Girth e Volume.

Exercício 23 – Criando Boxplot com Seaborn

1. Importe as bibliotecas Pandas e Seaborn;
2. Mapeie o caminho do documento e adicione em um dataframe;
3. Abra o arquivo `trees.csv`;
4. Visualize o cabeçalho dos dados;
5. Crie um box plot para visualizar a variável volume;
6. Crie um box plot para visualizar todas as variáveis dos dados.

Exercício 24 – Uma empresa tem um conjunto de dados sobre a dimensão de um determinado produto feito com matéria-prima de três fornecedores. Você foi designado a desenvolver uma análise para conseguir analisar quais desses fornecedores tem a matéria prima que não garante o dimensional final deste produto, a media máxima é de 120,15 mm e a dimensão mínima é de 119,9 mm. Crie um histograma e um boxplot para analisar as medidas de cada fornecedor e um boxplot para comparar os três fornecedores. Utilize o arquivo `fornecedores.xlsx`



Exercício 26 – Gráfico de dispersão

Uma indústria alimentícia, produtora de peças de presunto e de outros embutidos, iniciou o giro do ciclo PDCA de melhorias com a meta de reduzir para 5% a porcentagem de quebra durante o fatiamento do produto, até o final do ano. Esta decisão foi tomada após a confirmação de que a atual porcentagem de 12% não atendia às expectativas do cliente.

Na etapa observação do problema, levantou-se a hipótese de que a quebra no fatiamento após o resfriamento do presunto poderia ser dependente da temperatura interna do produto (decorrente do processo de fabricação). Com base nesta suposição, a equipe responsável pela solução do problema providenciou um levantamento de dados a fim de conhecer melhor a relação entre a temperatura interna do produto, a quebra no fatiamento e o tipo de produto fabricado.

- a) Analisar a relação entre temperatura interna do produto e % de quebra durante fatiamento do produto considerando todas as observações.
- b) Analisar a relação entre temperatura interna do produto e % de quebra durante fatiamento do produto estratificando por tipo de produto.