VISUALIZAÇÃO DE DADOS

- Tarefa de importância fundamental no contexto da Análise de Dados, também conhecida como Plotagem. Provê recursos para facilitar a percepção e interpretação dos resultados.

- Por fazer parte do processo exploratório, auxilia a identificação de discrepâncias (*outliers*) e ainda, relações, transformações necessárias sobre os dados ou até mesmo sugerir modelos de representação.

- Existem várias bibliotecas associadas ao PYTHON para prover recursos gráficos voltados para Estatística Descritiva.

- MATPLOTLIB consiste numa biblioteca ampla com suporte para exportação de visualizações em vários formatos gráficos vetoriais, tais como: PDF, SVG, JPG, PNG, GIF etc)

- SEABORN: construída da parte superior da biblioteca MATPLOTLIB e provê integração com as estruturas de dados da biblioteca PANDAS. Com SEABORN é possível plotar:

1. Gráficos de linha
2. Gráficos de dispersão de pontos
3. Box plot
4. Gráfico de pontos
5. Gráfico de contagem
6. Trama de violino
7. Enredo de enxame
8. Gráfico de barra
9. Gráfico KDE

HISTOGRAMAS

É um gráfico para mostrar uma Distribuição de Frequência de dados numéricos ou categóricos.

Obs. Gráficos de Barras são similares, no entanto, inapropriados para dados contínuos.

Ilustração: Plotagem simples para *fi* de idade no arquivo PESSOAS.CSV

```
import matplotlib.pyplot as plt
df = pd.read_csv('pessoas.csv')
print(df)
plt.hist(df['idade'],bins=15)
plt.xlabel('Idade')
plt.ylabel('fi')
```

	nome	idade	sexo	peso	altura	salario	e_civil
0	Joao	20	M	50	1.70	1400	S
1	Maria	35	F	62	1.80	3000	C
2	Pedro	92	M	60	1.55	4800	S
3	Alice	20	F	50	1.49	1240	S
4	Amanda	38	F	65	1.70	2400	C
5	Sandro	27	M	57	1.63	2140	S
6	Clara	29	F	54	1.67	2000	C
7	Roberta	65	F	60	1.72	1500	C
8	Marcos	40	M	58	1.76	3500	C
9	Carol	45	F	70	1.85	1800	C
10	Cintia	20	F	64	1.58	1600	C
11	Jonas	19	M	70	1.95	1450	S
12	Silvia	40	F	57	1.68	2100	C
13	Ana	29	F	55	1.60	1300	C
14	Jonata	37	M	59	1.68	2300	C

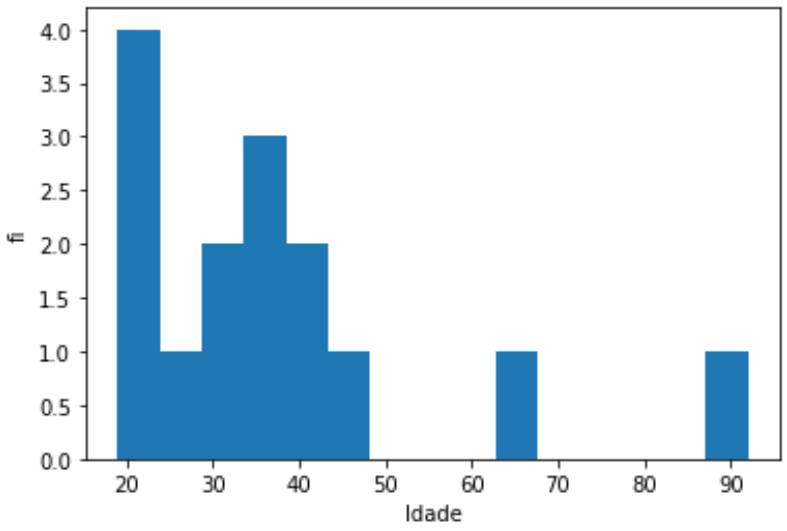


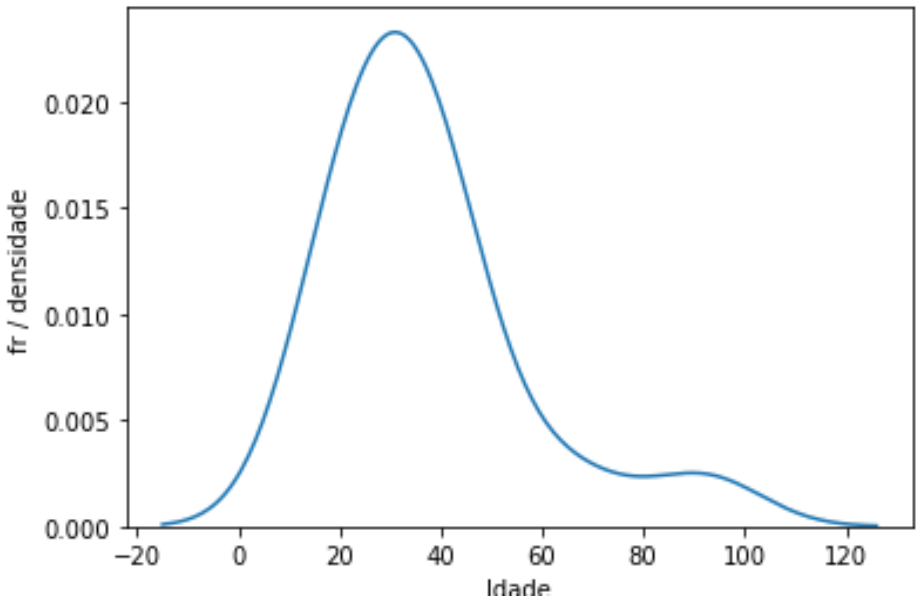
GRÁFICO DE DENSIDADE

Uma versão do Histograma que estima a Frequência Relativa a faz a plotagem do que sugere ser a ***Função de Densidade Probabilística***.

Ilustração: para o caso anterior, temos:

```
import matplotlib.pyplot as plt
import seaborn as sb
df = pd.read_csv('pessoas.csv')
print(df)
sb.kdeplot(df['idade'])
plt.xlabel('Idade')
plt.ylabel('fr / densidade')
```

```
      nome  idade sexo  peso  altura  salario e_civil
0     Joao     20    M    50    1.70     1400       S
1    Maria     35    F    62    1.80     3000       C
2    Pedro     92    M    60    1.55     4800       S
3    Alice     20    F    50    1.49     1240       S
4   Amanda     38    F    65    1.70     2400       C
5   Sandro     27    M    57    1.63     2140       S
6    Clara     29    F    54    1.67     2000       C
7  Roberta     65    F    60    1.72     1500       C
8   Marcos     40    M    58    1.76     3500       C
9    Carol     45    F    70    1.85     1800       C
10  Cintia     20    F    64    1.58     1600       C
11   Jonas     19    M    70    1.95     1450       S
12  Silvia     40    F    57    1.68     2100       C
13     Ana     29    F    55    1.60     1300       C
14  Jonata     37    M    59    1.68     2300       C
```



Obs. Estudaremos mais adiante!

Ilustração: Histograma para o atributo PESO

```
import matplotlib.pyplot as plt
import seaborn as sb
df = pd.read_csv('pessoas.csv')
print(df)
plt.figure(figsize=(8, 6))
plt.hist(df['peso'], bins=range(50,80,5))
plt.title('Distribuição de Pesos')
plt.xlabel('Peso')
plt.ylabel('Quantidade')
plt.savefig('peso-histograma.png')
plt.show()
plt.close()
```

	nome	idade	sexo	peso	altura	salario	e_civil
0	Joao	20	M	50	1.70	1400	S
1	Maria	35	F	62	1.80	3000	C
2	Pedro	92	M	60	1.55	4800	S
3	Alice	20	F	50	1.49	1240	S
4	Amanda	38	F	65	1.70	2400	C
5	Sandro	27	M	57	1.63	2140	S
6	Clara	29	F	54	1.67	2000	C
7	Roberta	65	F	60	1.72	1500	C
8	Marcos	40	M	58	1.76	3500	C
9	Carol	45	F	70	1.85	1800	C
10	Cintia	20	F	64	1.58	1600	C
11	Jonas	19	M	70	1.95	1450	S
12	Silvia	40	F	57	1.68	2100	C
13	Ana	29	F	55	1.60	1300	C
14	Jonata	37	M	59	1.68	2300	C

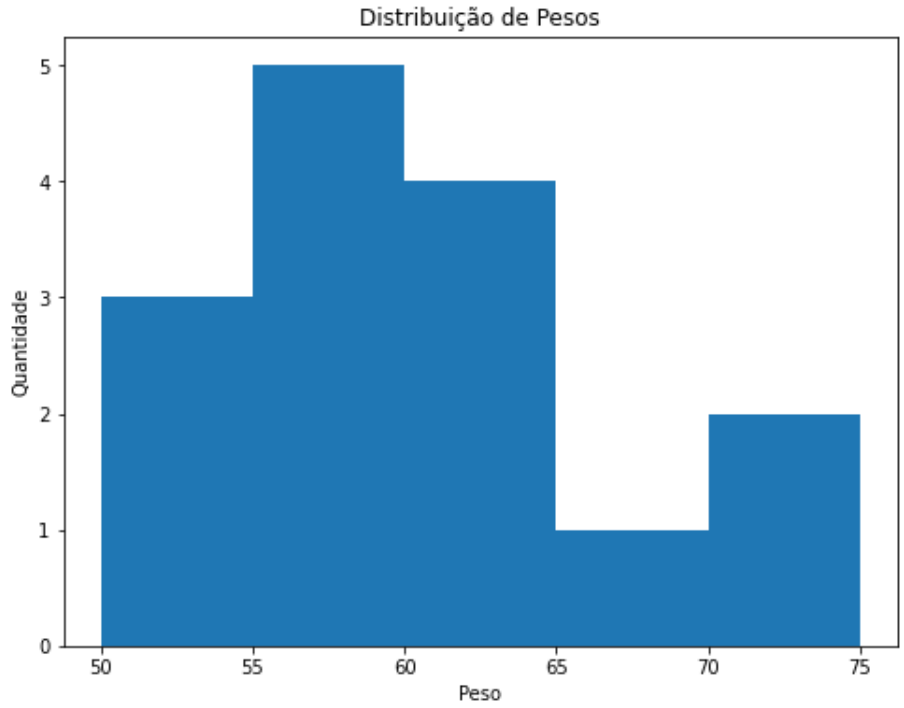
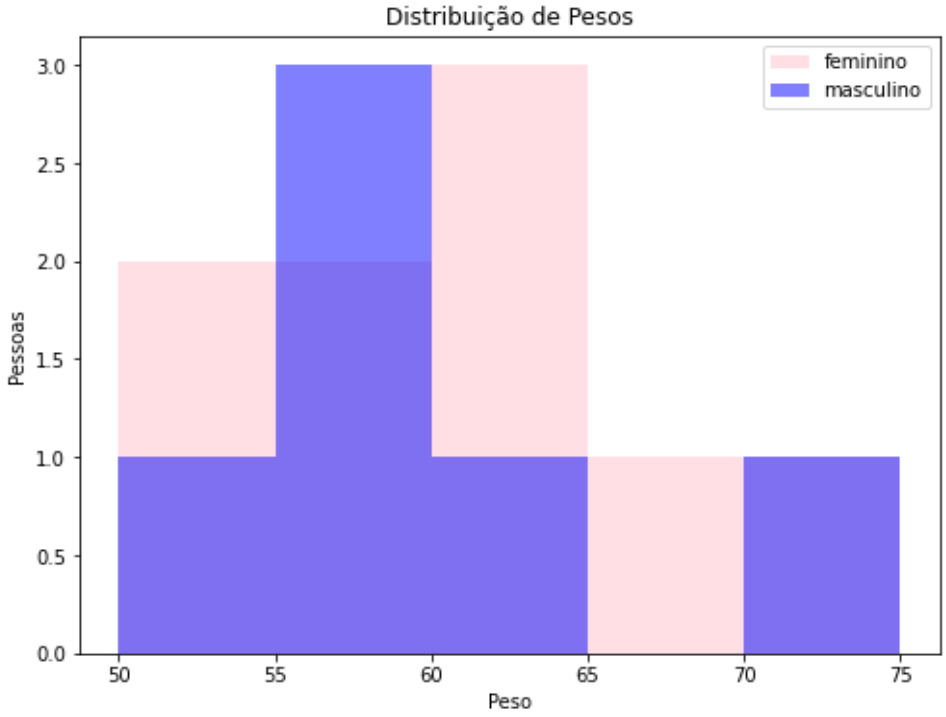


Ilustração: Histograma relacionando PESO → Homens x Mulheres

```python
# Cria um histograma comparando os pesos masculino x feminino em 'PESSOAS.CSV'
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('pessoas.csv')
print(df)
fem = df[df.sexo == 'F']
mas = df[df.sexo == 'M']
peso_fem = fem['peso']
peso_mas = mas['peso']

plt.figure(figsize=(8, 6))
plt.title('Distribuição de Pesos')
plt.xlabel('Peso')
plt.ylabel('Pessoas')
plt.hist(peso_fem,bins=range(50, 80,5),
         alpha=0.5, label='feminino', color='pink')
plt.hist(peso_mas,bins=range(50, 80,5),
         alpha=0.5, label='masculino', color='blue')
plt.legend(loc='upper right')
plt.savefig('peso-histograma-mas-x-fem.png')
plt.show()
plt.close()
```

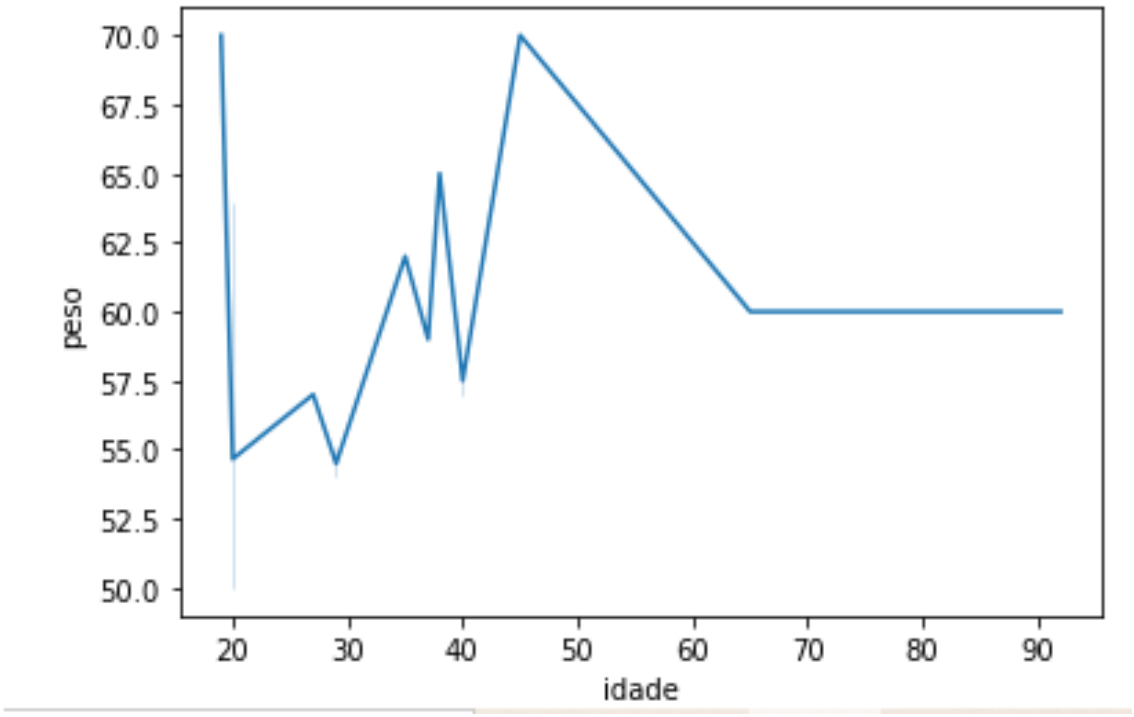


GRÁFICOS DE LINHA

- Também conhecidos por Gráficos de Representação Multivariável (*Line Plots*)
- Em muitos conjuntos de dados faz-se necessário uma análise do comportamento de determinadas variáveis em relação a outros em termos de uma evolução contínua (temporal, serial etc)

Ilustração: a relação IDADE x PESO em 'PESSOAS.CSV'

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
df = pd.read_csv('pessoas.csv')
#print(df)
sb.lineplot('idade','peso', data=df)
```



REVISANDO...

Conforme a necessidade de interpretação dos dados pode-se utilizar diferentes formas de visualização, fornecidas por um conjunto de bibliotecas que trabalham harmoniosamente com Python:

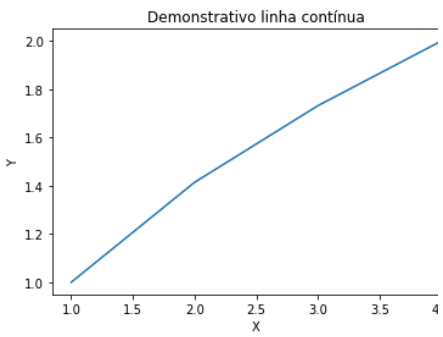
1. Gráficos Relacionais
    a. Dispersão (*scatterplot*)
    b. Linhas (*lineplot*)
2. Gráficos de Distribuição
    a. Histogramas (*hisplot*)
    b. Densidade (*kdeplot*)
3. Gráficos Categóricos
    a. Caixas (*boxplot*)
    b. Pontos (*pointplot*)
    c. Barras (*barplot*)

Continuação: **Gráficos de Linhas**

1. Função *plot()* - Biblioteca *matplotlib.pyplot*
2. Função *lineplot()* – Biblioteca *seaborn*

<u>Ilustração</u>:

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

x = np.array([1, 2, 3, 4])
y = x**0.5

plt.plot(x, y)
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Demonstrativo linha contínua")
plt.show()
A = [-1, 1, 3, 5]
B = [0, 2.5, 5, 7.5]
plt.plot(A, B,'^')
plt.xlabel("A")
plt.ylabel("B")
plt.title("Demonstrativo de pontos")
plt.show()
```



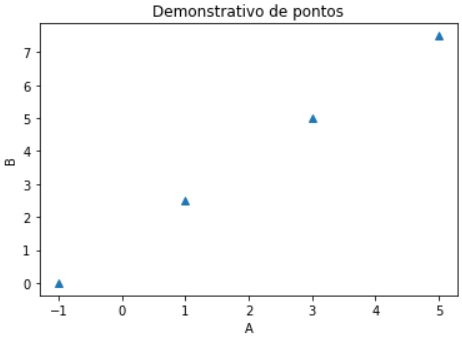


Ilustração:

Analisando o arquivo ‘vendas_filiais.csv’

```
import pandas as pd
import matplotlib.pyplot as plt
#import matplotlib as plt
import seaborn as sb
import numpy as np
vendas=pd.read_csv('vendas_filiais.csv')

meses=['jan','fev','mar','abr','mai','jun']

vendas = list(vendas['VENDAS'])
cont=jan=fev=mar=abr=mai=jun=0
for m in vendas:
    cont+=1
    if cont <=4:
        jan+=m
    elif cont <=8:
        fev+=m
    elif cont<=12:
        mar+=m
    elif cont<=16:
        abr+=m
    elif cont<=20:
        mai+=m
    else:
        jun+=m
y_vendas=[jan,fev,mar,abr,mai,jun]

plt.plot(meses, y_vendas)
plt.xlabel("MÊS")
plt.ylabel("VENDAS EM R$ mil")
plt.title("EVOLUÇÃO DAS VENDAS")
plt.show()
```

MES	FILIAL	VENDAS
jan	'Filial 1'	50
jan	'Filial 2'	45
jan	'Filial 3'	60
jan	'Filial 4'	38
fev	'Filial 1'	52
fev	'Filial 2'	51
fev	'Filial 3'	54
fev	'Filial 4'	50
mar	'Filial 1'	51
mar	'Filial 2'	48
mar	'Filial 3'	70
mar	'Filial 4'	35
abr	'Filial 1'	59
abr	'Filial 2'	42
abr	'Filial 3'	85
abr	'Filial 4'	31
mai	'Filial 1'	50
mai	'Filial 2'	40
mai	'Filial 3'	80
mai	'Filial 4'	32
jun	'Filial 1'	56
jun	'Filial 2'	45
jun	'Filial 3'	70
jun	'Filial 4'	37

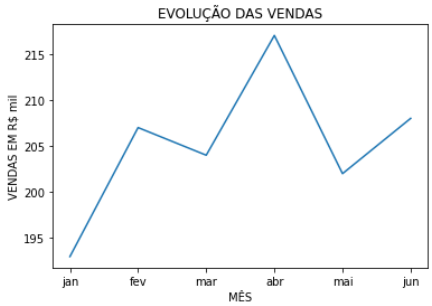
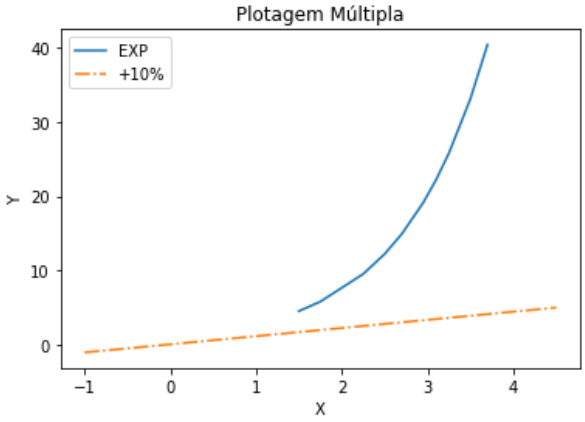


Ilustração: Plotagem Múltipla

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

# Dados para 1a. Plotagem
lista_A =[1.5,1.75,2.25,2.5,2.7,2.95,3.1,3.25,3.5,3.7]
lista_B =[]
for i in lista_A:
    lista_B.append(math.exp(i))

plt.plot(lista_A, lista_B,label='EXP')

# Dados para 2a. Plotagem
lista_X =[-1,0,1,1.5,2,2.5,3,3.5,4,4.5]
lista_Y =[]
for i in lista_X:
   lista_Y.append(i*1.1)

plt.plot(lista_X, lista_Y,ls='-.',label='+10%')

plt.xlabel("X")
plt.ylabel("Y")
plt.title('Plotagem Múltipla')
plt.legend()
plt.show()
```



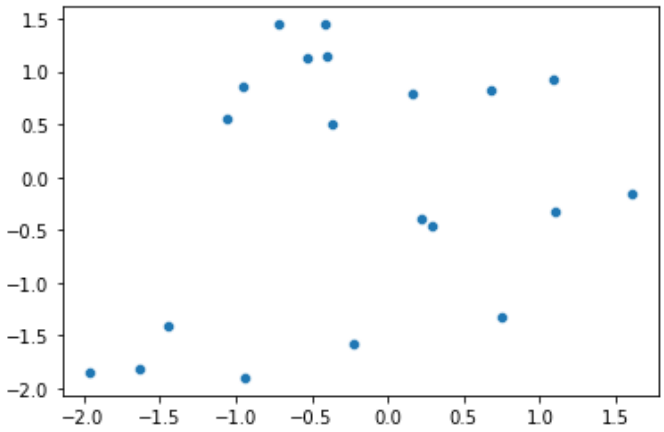
Gráficos de Dispersão

Função *scatterplot()* - Biblioteca *seaborn*

Um gráfico de dispersão mostra os pontos de uma relação (x_i,y_i). A plotagem desta relação pode contribuir com a visualização de um padrão (agrupamentos, afastamentos, tendências – no caso, sugerindo um tipo de regressão).

Ilustração: caso simples utilizando dados randômicos

```
import pandas as pd
import matplotlib as plt
import seaborn as sb
import numpy as np
tabela_x_y=np.random.randn(20,2)
xi=[]
yi=[]
print(tabela_x_y)
for i in range(0,20):
    nx = tabela_x_y[i,0]
    ny = tabela_x_y[i,1]
    xi.append(nx)
    yi.append(ny)

grafico=sb.scatterplot(data=tabela_x_y,x=xi,y=yi)
```



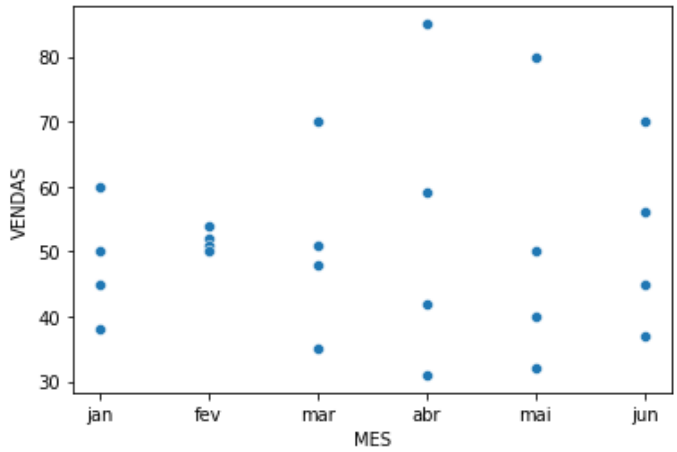

```
[[-0.40957143  1.44740705]
 [-0.40453402  1.14892501]
 [-0.93891049 -1.90146405]
 [ 1.09543335 -0.33354379]
 [-1.44184857 -1.41666675]
 [-1.62950152 -1.81525581]
 [-1.056169    0.55427493]
 [-0.95001801  0.85799935]
 [-0.53207925  1.12575886]
 [ 0.21865772 -0.39288196]
 [-0.22869727 -1.57625488]
 [ 0.16583301  0.78879681]
 [-1.96189224 -1.85203824]
 [ 1.09188661  0.93459086]
 [-0.71178556  1.44308301]
 [ 0.75231332 -1.33492077]
 [ 0.67663967  0.83110578]
 [-0.36505744  0.49677025]
 [ 0.28955915 -0.47020934]
 [ 1.60599264 -0.15366183]]
```

Ilustração:

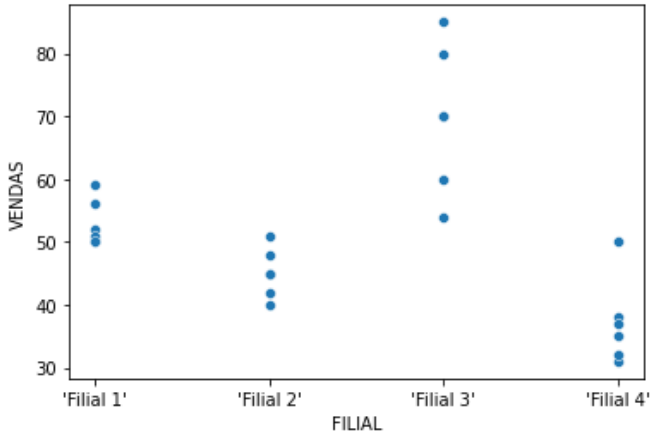
Analisando o arquivo ‘vendas_filiais.csv’

```
# DISPERSÃO DAS VENDAS POR MÊS
import pandas as pd
import matplotlib as plt
import seaborn as sb
import numpy as np
vendas=pd.read_csv('vendas_filiais.csv')
print(vendas)

grafico=sb.scatterplot(data=vendas,x=vendas['MES'],y=vendas['VENDAS'])
```

MES	FILIAL	VENDAS
jan	'Filial 1'	50
jan	'Filial 2'	45
jan	'Filial 3'	60
jan	'Filial 4'	38
fev	'Filial 1'	52
fev	'Filial 2'	51
fev	'Filial 3'	54
fev	'Filial 4'	50
mar	'Filial 1'	51
mar	'Filial 2'	48
mar	'Filial 3'	70
mar	'Filial 4'	35
abr	'Filial 1'	59
abr	'Filial 2'	42
abr	'Filial 3'	85
abr	'Filial 4'	31
mai	'Filial 1'	50
mai	'Filial 2'	40
mai	'Filial 3'	80
mai	'Filial 4'	32
jun	'Filial 1'	56
jun	'Filial 2'	45
jun	'Filial 3'	70
jun	'Filial 4'	37




```
# DISPERSÃO DAS VENDAS POR FILIAL
import pandas as pd
import matplotlib as plt
import seaborn as sb
import numpy as np
vendas=pd.read_csv('vendas_filiais.csv')

grafico=sb.scatterplot(data=vendas,x=vendas['FILIAL'],y=vendas['VENDAS'])
```



Dispersão com Regressão Linear

Função *regplot()* - Biblioteca *seaborn*

Neste caso, além da plotagem dos pontos também será apresentada uma Reta que representa a *Regressão Linear* tentando mostrar o padrão de correlação existente.

Nota: *ver Regressão Linear pelo Método dos Mínimos Quadrados*

$y = A + Bx$ *(Regressão Linear Simples)*

$y = A + B_0 x_0 + B_1 x_1 + B_2 x_2 + ... + B_n x_n + \varepsilon$ *(Regressão Linear Múltipla)*

Valores Residuais

Níveis de Confiança

Coeficiente de Determinação

Outros Indicadores Estatísticos a partir de DESCRIBE( ) e SUMARY( )

<u>Ilustração</u>:

```
import pandas as pd
import matplotlib as plt
import seaborn as sb
import numpy as np
tabela_x_y=np.random.randn(20,2)
xi=[]
yi=[]
print(tabela_x_y)
for i in range(0,20):
    nx = tabela_x_y[i,0]
    ny = tabela_x_y[i,1]
    xi.append(nx)
    yi.append(ny)
grafico1=sb.scatterplot(data=tabela_x_y,x=xi,y=yi)
grafico2=sb.regplot(data=tabela_x_y,x = xi,y = yi,ci=95, n_boot=1000)
```

```
[[ 1.0163212   0.15988328]
 [-0.02688839  0.76199226]
 [ 0.0802451  -0.59180897]
 [-0.47401727 -0.3223604 ]
 [ 0.54485209 -1.02765483]
 [ 0.99656301 -0.62774944]
 [-0.28037049 -0.61233171]
 [-0.09717936 -1.90130802]
 [ 0.94350782 -1.10579721]
 [ 0.51125228 -0.43000518]
 [-1.24746375 -0.96924212]
 [-0.78675831 -1.08516914]
 [-0.38341532 -0.64001635]
 [ 0.13043115  0.18062624]
 [-0.15229886  0.51698568]
 [-0.08347322  2.31990206]]
```

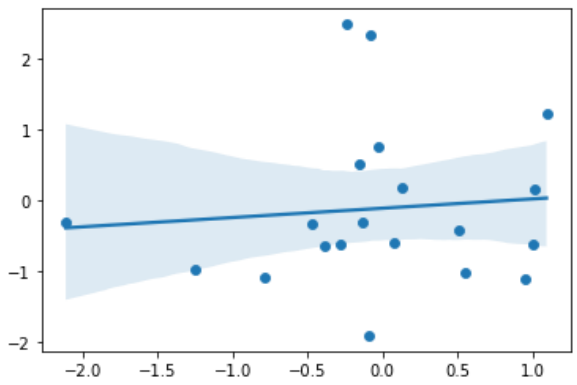


Ilustração:

```
import pandas as pd
import matplotlib as plt
import seaborn as sb
import numpy as np
requisicoes_acumuladas=[5,12,15,18,20,25,27,32]
tempo_resposta=[0.01,0.15,0.9,1.2,1.5,1.7,1.75,1.95]
grafico=sb.regplot(x = requisicoes_acumuladas, y = tempo_resposta, color='green')
grafico.set_title("Regressão Linear")
grafico.set_xlabel("Requisições acumuladas no sistema", fontsize = 9)
grafico.set_ylabel("Tempo de Resposta acumulado", fontsize = 9)
```

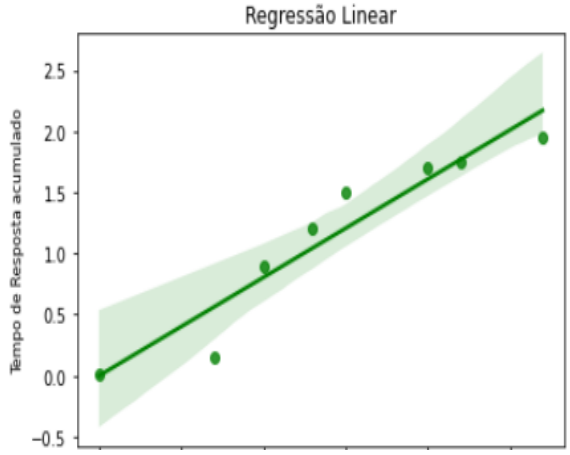
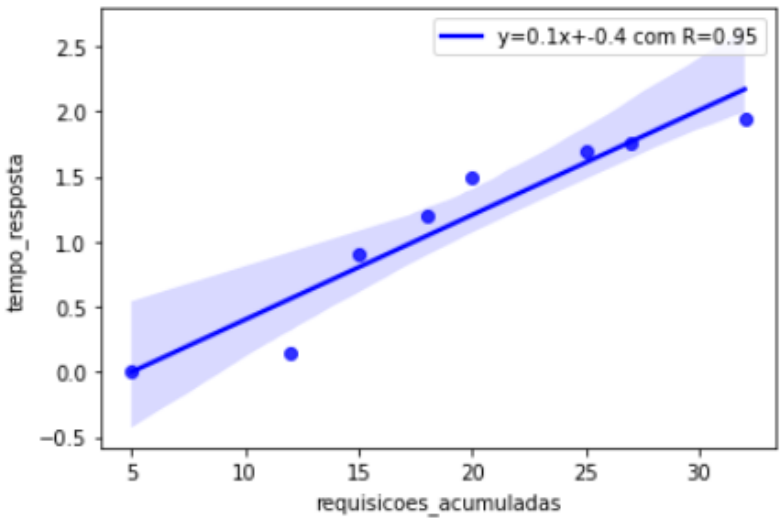


Ilustração: Exibindo o modelo de Regressão Linear (Equação da Reta)

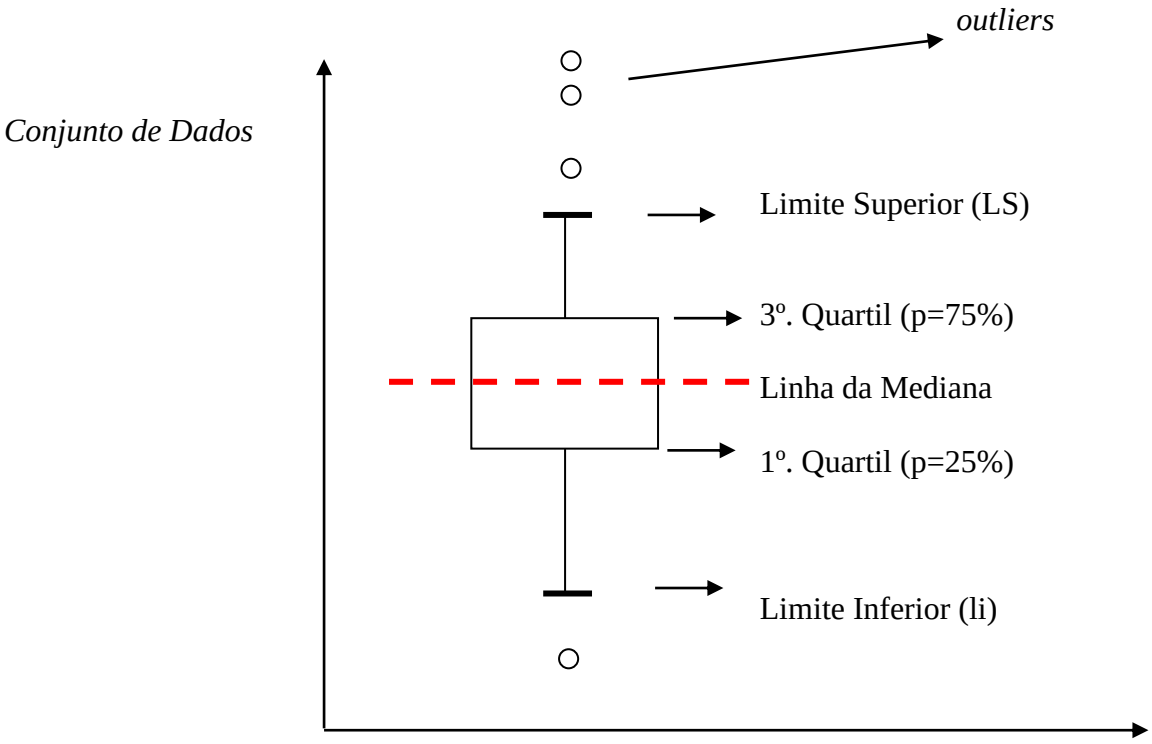
```
from scipy import stats
import pandas as pd
import matplotlib as plt
import seaborn as sb
import numpy as np

requisicoes_acumuladas=[5,12,15,18,20,25,27,32]
tempo_resposta=[0.01,0.15,0.9,1.2,1.5,1.7,1.75,1.95]
tabela=pd.DataFrame(list(zip(requisicoes_acumuladas,tempo_resposta)),columns=['requisicoes_acumuladas','tempo_resposta'])

# b = Coeficiente de Inclinação; a = Coeficiente de Intersecção
b, a, r, p, std = stats.linregress(tabela['requisicoes_acumuladas'],tabela['tempo_resposta'])

eixos = sb.regplot(x="requisicoes_acumuladas", y="tempo_resposta", data=tabela,color='b',
 line_kws={'label':"y={0:.1f}x+{1:.1f} com R={2:.2f}".format(b,a,r)})
```



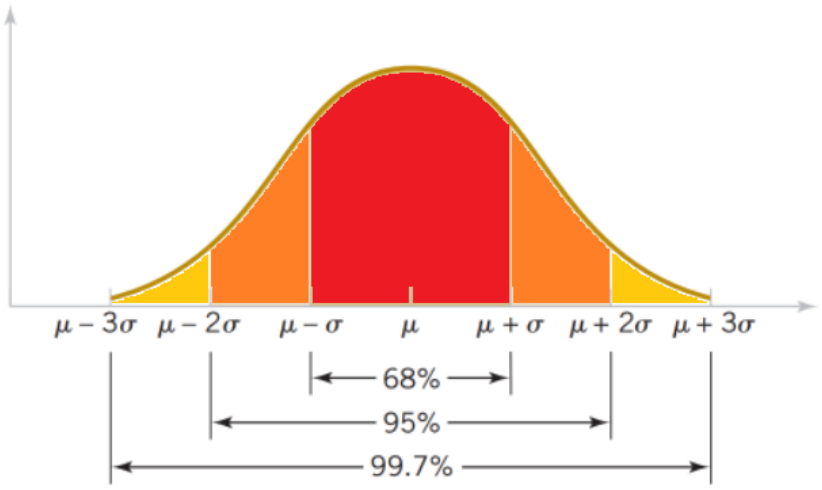
Diagramas BoxPlot

Função *boxplot()* - Biblioteca *seaborn*
Função *boxplot()* - Biblioteca *Matplotlib.pyplot*

Diagramas de caixas do tipo BoxPlot possuem a seguinte estrutura:



Obs. Visualização dos dados segundo a curva Normal de Distribuição:



- O quão deseja-se envolver ou não os dados *outliers* na análise?

Cálculo de *li* e LS:

*li = q1 - 1.5 * (q3 - q1)*

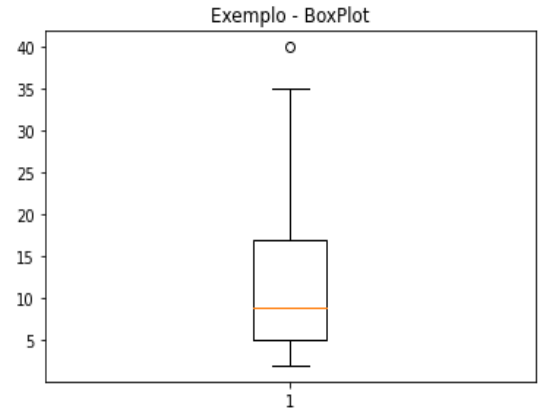
*LS = q3+1.5 * (q3 - q1)*

Exemplo (*passo a passo*...)

Suponha A = {-20,-18,-5,0,1,3,4,6,7,9,30}

Ilustração:

```
import matplotlib.pyplot as plt
import statistics
import seaborn as sb

A=[2,4,5,7,9,12,17,35,40]

plt.boxplot(A)
plt.title("Exemplo - BoxPlot")
plt.show()
q1=np.percentile(A,25)
print("1o. Quartil ------> ",q1)
print("Mediana ----------> ",statistics.median(A))
q3=np.percentile(A,75)
print("3o. Quartil ------> ",q3)
print("Mínimo-----------> ",q1-1.5*(q3-q1))
print("Máximo-----------> ",q3+1.5*(q3-q1))
```

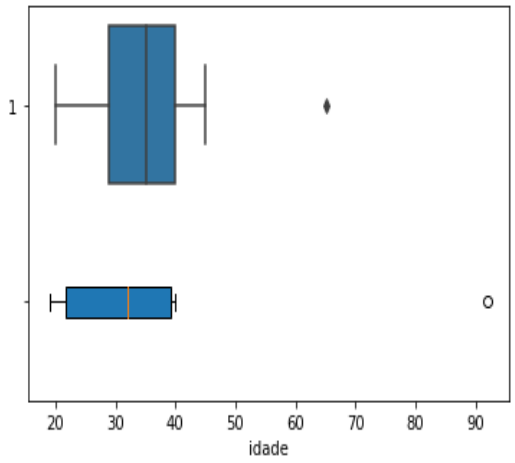



```
1o. Quartil ------>  5.0
Mediana ---------->  9
3o. Quartil ------>  17.0
Mínimo------------>  -13.0
Máximo------------>  35.0
```

Ilustração:

```
# Box Plot da Biblioteca seaborn
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
df=pd.read_csv('pessoas.csv')
print(df)
fem = df[df.sexo == 'F']
mas = df[df.sexo == 'M']
idade_fem = fem['idade']
iddae_mas = mas['idade']

sns.boxplot(x=idade_fem,data=df)
# OU...
grafico = plt.boxplot(idade_mas,  vert = 0, patch_artist = True)
```

```
      nome  idade sexo  peso  altura  salario e_civil
0     Joao     20    M    50    1.70     1400       S
1    Maria     35    F    62    1.80     3000       C
2    Pedro     92    M    60    1.55     4800       S
3    Alice     20    F    50    1.49     1240       S
4   Amanda     38    F    65    1.70     2400       C
5   Sandro     27    M    57    1.63     2140       S
6    Clara     29    F    54    1.67     2000       C
7  Roberta     65    F    60    1.72     1500       C
8   Marcos     40    M    58    1.76     3500       C
9    Carol     45    F    70    1.85     1800       C
10  Cintia     20    F    64    1.58     1600       C
11   Jonas     19    M    70    1.95     1450       S
12  Silvia     40    F    57    1.68     2100       C
13     Ana     29    F    55    1.60     1300       C
14  Jonata     37    M    59    1.68     2300       C
```



Exercícios: (aplicação em *Python*)

1. Considere a relação de dados:

Ano	Matrículas	Evasão Escolar
2010	3400	230
2011	3800	175
2012	3550	110
2013	3700	350
2014	3780	315
2015	3600	327
2016	3200	280
2017	3340	335
2018	3100	295
2019	2870	390
2020	2200	450
2021	1960	184

   a. Elabore um Histograma plotando a relação Ano X Matrículas e Ano X Evasão.
   b. Considerando o percentual de Evasão Escolar em relação às matriculas em cada ano, faça a plotagem de um gráfico de dispersão e discuta sobre a tendência dos dados.

2. A tabela registra as marcas das temperaturas, mínima e máxima de determinada região no período de uma semana:

Mínima	**Máxima**
12	26
15	28
11	30
13	25
14	27
10	24
12	29

Elabore a plotagem BoxPlot de ambas as colunas registradas e identifique os elementos que constituem os indicadores das caixas.

3. Considere o *dataset* "pessoas.csv" disponibilizado. Elabore dois diagramas com plotagens diferentes para discutir a relação *pessoas.peso* x *pessoas.altura*

4. Repita o ex. 3 apenas para mulheres.

5. O arquivo *"anotações.txt"* contém a coleta das medidas de circunferência de várias árvores destinadas a produção de madeira para indústria moveleira. Essas medidas são coletadas na altura de aproximadamente 1,5m (referência utilizada para cálculo do DAP = diâmetro na altura do peito). Você precisa fazer uma análise que melhor descreva estatisticamente o conjunto das anotações de medidas. Para tal é necessário implementar códigos que orientem:

    a. Qual a melhor medida padrão para uma decisão de venda das árvores?
    b. Identificar mudanças significativas se forem constituídos grupos distintos como medidas de circunferência padrão em relação ao tratamento como um único grupo?
    c. Cuidar com anotações supostamente erradas que podem deturpar a variabilidade das medidas

    Elabore:
    d. Diagrama de dispersão
    e. Histogramas que permitam a visualização das classes no entorno dos quartis dos conjuntos

6. Dado o arquivo *"health.csv"* elabore uma tabela de referência cruzada para mostrar:
    a. Min()
    b. Max()
    c. Mean()
    d. Count()
    e. Sum()

    na relação *year X Disease* com agregação pelo atributo *increase*. Também pode ser gerado o *Mapa de Calor* para visualização destes indicadores.