



Curso de Python

29-01-2021

<https://github.com/tsbressan/CursoPython>

Bibliotecas para processamento

- **Pandas: dataframe**
- Combinar dados (concatenar)
- `pd.concat([dt1,dt2])`
- `pd.concat([dt1,dt2], axis=1)`

----Exemplo com importação de dados (.xlsx e .csv)

Bibliotecas para processamento

- **Pandas: dataframe**
- Combinar dados (merge - juntar)
- `pd.merge(dt1,dt2, how='left', on='Nome_Rocha')`

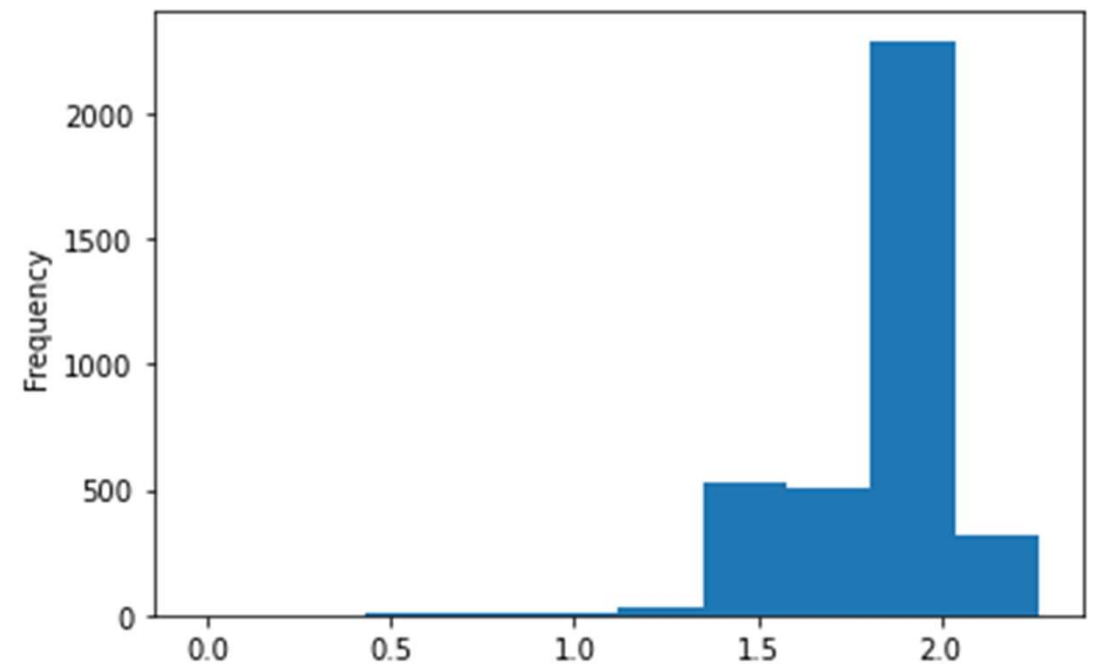
----Exemplo com importação de dados (.xlsx e .csv)

Bibliotecas para procesamiento

- **Pandas: dataframe**
- Plotting (gráficos simples)
- https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

Bibliotecas para processamento

- **Pandas: dataframe**
- Plotting (gráficos simples)
- Histograma:
`dataset["..."].plot.hist()`



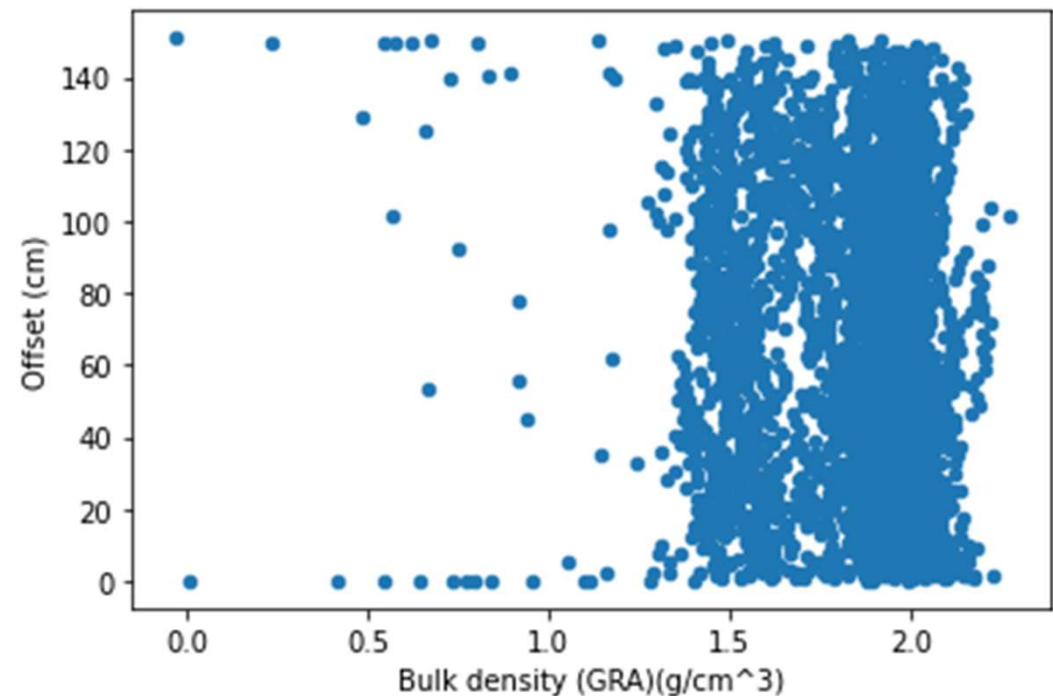
Bibliotecas para processamento

- **Pandas: dataframe**

- Plotting (gráficos simples)

- Scatter plot

```
dataset.plot.scatter(y='Offset(cm)',  
x="Bulkdensity(GRA)")
```



Bibliotecas para processamento

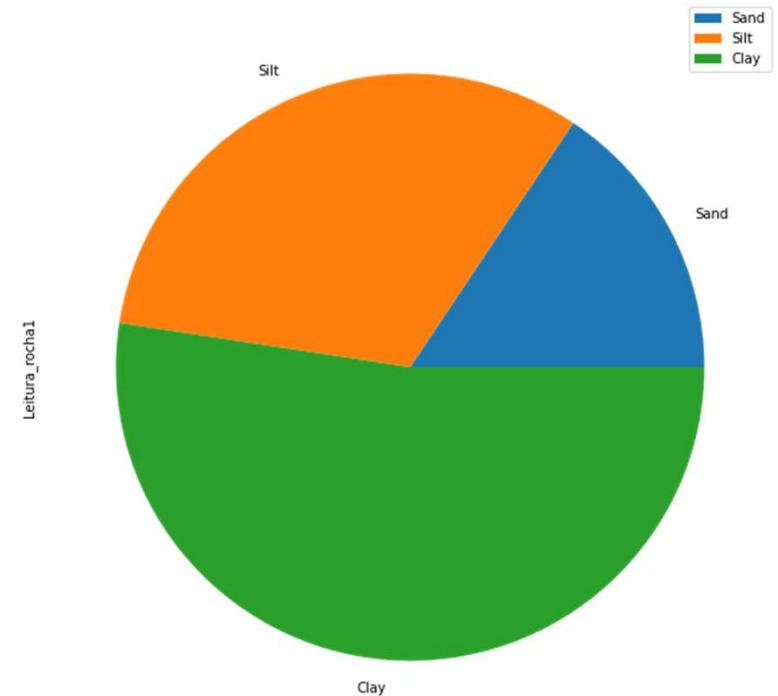
- **Pandas: dataframe**

- Plotting (gráficos simples)

- Pie

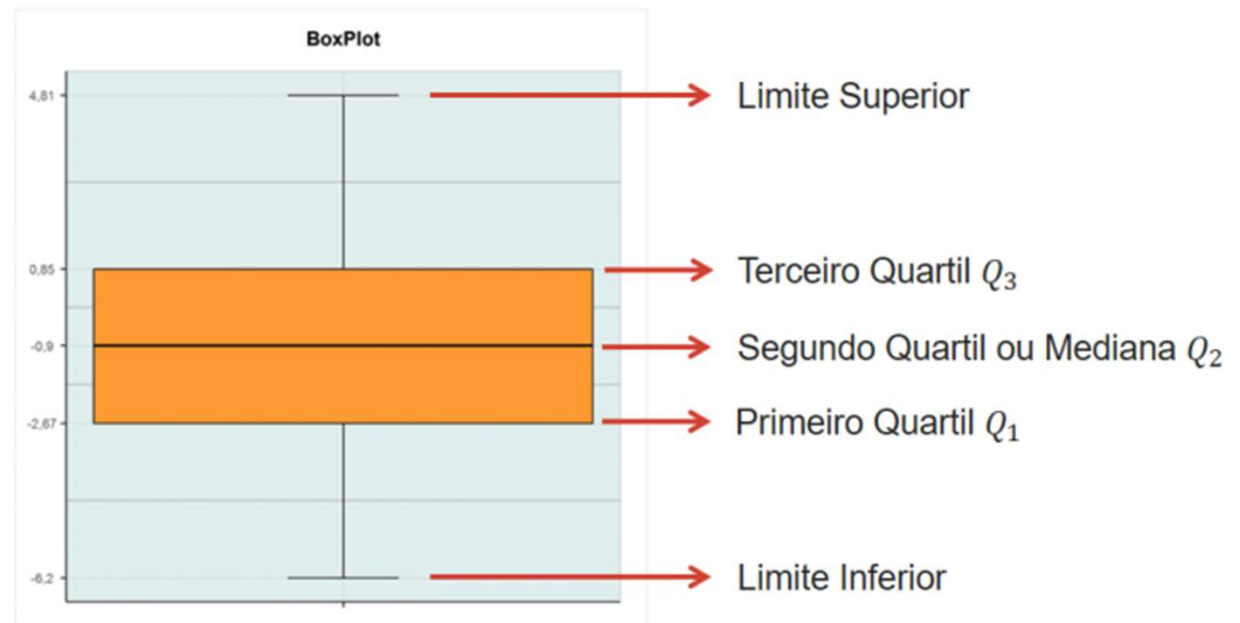
```
df = pd.DataFrame({'Leitura_rocha1': [1.2, 2.45 , 4.02],  
                  'Leitura_rocha2': [550.7, 480.8, 369.5]},  
                  index=['Sand', 'Silt', 'Clay'])
```

```
plot = df.plot.pie(y='Leitura_rocha1', figsize=(10, 10))
```



Bibliotecas para processamento

- **Pandas: dataframe**
- Plotting (gráficos simples)
- Boxplot



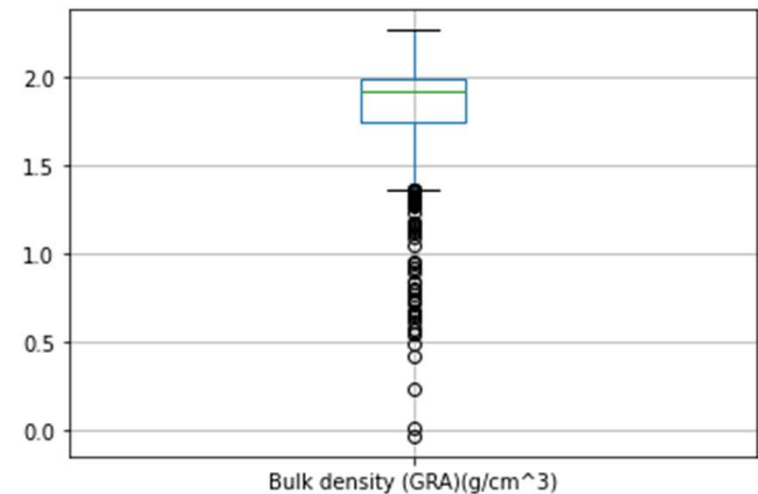
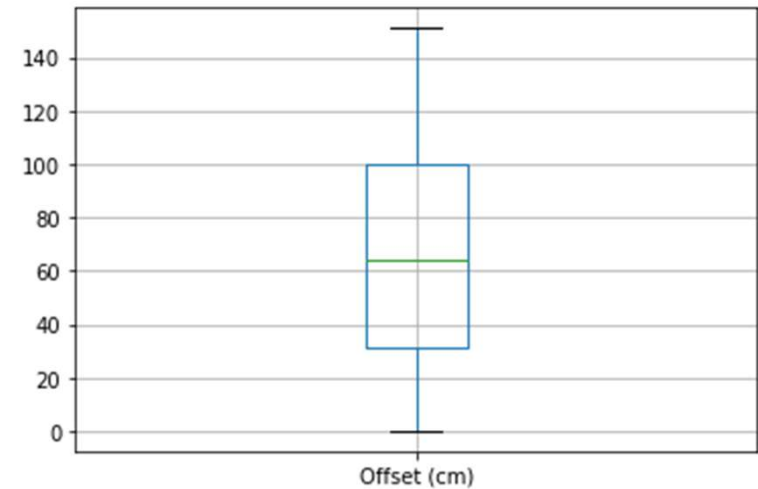
<http://www.portalection.com.br/sites/default/files/resize/EstatisticaBasica/figuras/boxplot1-700x354.png>

Bibliotecas para processamento

- **Pandas: dataframe**
- Plotting (gráficos simples)
- Boxplot

```
dataset.boxplot(column=['Offset (cm)'])
```

```
dataset.boxplot(column=['Bulk density (GRA)(g/cm^3)'])
```



Bibliotecas para processamento

- **Numpy:** <https://numpy.org/>
- Principal biblioteca para computação científica em Python
- Biblioteca complementar do Panda
- Necessita ser importada antes da utilização:
- `import numpy as np`
- Suporte a array unidirecionais (vetor) ou multidimensionais(matriz)

Bibliotecas para processamento

- **numpy: consultar instalação**

```
import numpy as np
```

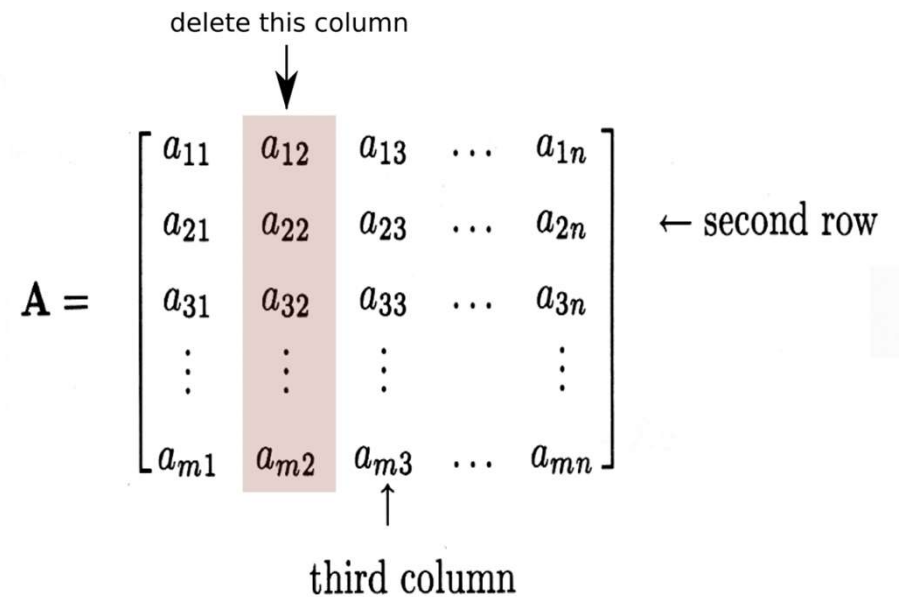
```
np.__version__
```

Bibliotecas para processamento

- **numpy: array**

Arrays NumPy

- tipo de dado chamado ndarray



Bibliotecas para processamento

- **numpy: array**

Exemplo:

```
import numpy as np
```

```
ar_1 = np.arange(10)
```

```
print(ar_1)
```

```
#unidimensional de 0 a 9
```

Bibliotecas para processamento

- **numpy: array**

Exemplo:

```
import numpy as np
```

```
ar_1 = np.array([1, 2, 3, 4, 5])
```

```
print(ar_1)
```

```
print (ar_1.shape)  #dimensões da array
```

```
#array unidimensional de inteiros(igual a uma lista)
```

Bibliotecas para processamento

- **numpy: array**

Exemplo:

```
import numpy as np
```

```
ar_1 = np.array([[1, 2, 3, 4, 5], [10, 21, 35, 46, 500]])
```

```
print(ar_1)
```

```
print (ar_1.shape)
```

```
#matriz de duas linhas e 5 colunas
```

Bibliotecas para processamento

- **numpy: array**

Exemplo:

Acessar elemento por elemento:

```
print(ar_1[0])
```

```
print(ar_1[1])
```

```
print(ar_1[0][0])
```

```
>>> a[0,3:5]  
array([3,4])
```

```
>>> a[4:,4:]  
array([[44, 45],  
       [54, 55]])
```

```
>>> a[:,2]  
array([2,12,22,32,42,52])
```

```
>>> a[2::2,::2]  
array([[20,22,24],  
       [40,42,44]])
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

http://www.estruturas.ufpr.br/wp-content/uploads/2015/01/numpy_indexing-300x171.png

Bibliotecas para processamento

- **numpy: array**

Exemplo:

Criar uma array vazia:

```
ar_2 = np.empty([3,2], dtype = int)
```

#3 linhas e 2 colunas – atribui valores inteiros aleatórios na matriz

```
ar_2 = np.zeros((5,2))
```

matriz com valores igual a zero

#Ou com valores aleatórios

```
ar_3 = np.random.random((5,2))
```

Bibliotecas para processamento

- **numpy: array**

Exemplo:

Operações - cálculos matemáticos - entre arrays: +, -, /, *

```
import numpy as np
```

```
a = np.array([[10.0, 2.0], [30.0, 4.0]])
```

```
b = np.array([[50.0, 6.0], [7.0, 80.0]])
```

```
soma = a+b
```

#possui também o operador `sum()`, `prod()`,

Bibliotecas para processamento

- **numpy: array**

Exemplo:

Coeficiente de Correlação Pearson (corrcoef) – R^2

```
import numpy as np  
a = np.array([10.0, 2.0])  
b = np.array([50.0, 6.0])  
np.corrcoef([a,b])
```

Bibliotecas para processamento

- **numpy: array**

Exemplo:

Normalização (linalg.norm)

```
import numpy as np
a = np.array([ 1.797, 2.7607, 0.4383, 0.7866, 1.8091, 0.1954,
               3.6307, 4.6599, 0.1065, 2.0508])
norm = np.linalg.norm(a)
a' = a/norm
```

<https://numpy.org/doc/stable/reference/generated/numpy.linalg.norm.html#numpy.linalg.norm>