



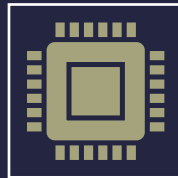
Introdução ao Pandas

Prof. Eduardo Gomes Carvalho

Introdução



Contém estruturas de dados e ferramentas para manipulação de dados.



Com frequência usado em conjunto com:

ferramentas de
processamento numérico
com NumPy e Scipy,
bibliotecas de análise como
statsmodel e scikit-learn
bibliotecas de visualização
de dados como a matplotlib.



Adota partes significativas do estilo idiomático do NumPy para processamento baseado em arrays.



Apesar de adotar muitos idioms de programação do NumPy foi projetado para trabalhar como dados tabulares e heterogêneos.

Importação do pandas

```
import pandas as pd
```

```
from pandas import Series,  
DataFrame
```

Estruturas de
dados do
pandas

Series

DataFrame



Series

- Uma Series é um objeto do tipo array unidimensional contendo uma sequência de valores (de tipos semelhantes aos tipos do NumPy) e um array associado de rótulos (labels) de dados, chamado de índice. A Series mais simples é composta de apenas um array de dados.

Series

✓
0s



```
import pandas as pd
```

✓
0s

```
[3] obj = pd.Series([4, 7, -5, 3])  
obj
```

```
0    4  
1    7  
2   -5  
3    3  
dtype: int64
```

✓
0s

```
[4] obj.values
```

```
array([ 4,  7, -5,  3])
```

✓
0s



```
obj.index
```

```
RangeIndex(start=0, stop=4, step=1)
```

Series

```
✓ [7] obj2 = pd.Series([4, 7, -5, 3], index = ['d', 'b', 'a', 'c'])  
0s obj2
```

```
d    4  
b    7  
a   -5  
c    3  
dtype: int64
```

```
✓ [8] obj2.index  
0s
```

```
Index(['d', 'b', 'a', 'c'], dtype='object')
```

```
✓ [13] obj2['a'] = -6  
0s
```

```
✓ [ ] obj2[obj2>0]  
0s
```

```
➞ d    4  
   b    7  
   c    3  
   dtype: int64
```

Series

```
✓ [15] import pandas as pd  
0s      import numpy as np
```

```
✓ [16] obj2 = pd.Series([4, 7, -5, 3], index = ['d', 'b', 'a', 'c'])  
0s      obj2
```

```
d      4  
b      7  
a     -5  
c      3  
dtype: int64
```

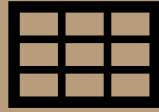
```
✓ [17] obj2 * 2  
0s
```

```
d      8  
b     14  
a    -10  
c      6  
dtype: int64
```

```
✓ [18] np.exp(obj2)  
0s
```

```
➞ d      54.598150  
   b    1096.633158  
   a      0.006738  
   c     20.085537  
   dtype: float64
```

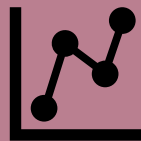

DataFrame



Um DataFrame representa uma tabela de dados retangular e contém uma coleção ordenada de colunas, em que cada uma pode ter um tipo de valor diferente (numérico, string, booleano etc.).



O DataFrame tem índice tanto para linha quanto para coluna.



Embora o DataFrame seja fisicamente bidimensional, podemos usá-lo para representar dados de dimensões maiores em um formato tabular usando indexação hierárquica.

DataFrame

✓
0s



```
data = {'estado': ['Minas Gerais', 'Minas Gerais', 'Minas Gerais', 'São Paulo', 'São Paulo', 'São Paulo'],  
        'ano': [2000, 2001, 2002, 2001, 2002, 2003],  
        'pop': [1.5, 1.7, 3.6, 2.4, 2.9, 3.2]}  
frame = pd.DataFrame(data)  
frame
```



	estado	ano	pop
0	Minas Gerais	2000	1.5
1	Minas Gerais	2001	1.7
2	Minas Gerais	2002	3.6
3	São Paulo	2001	2.4
4	São Paulo	2002	2.9
5	São Paulo	2003	3.2



✓
0s



```
frame.head() # pega as cinco primeiras linhas
```

	estado	ano	pop
0	Minas Gerais	2000	1.5
1	Minas Gerais	2001	1.7
2	Minas Gerais	2002	3.6
3	São Paulo	2001	2.4
4	São Paulo	2002	2.9





```
data = {'estado': ['Minas Gerais', 'Minas Gerais', 'Minas Gerais', 'São Paulo', 'São Paulo', 'São Paulo'],  
        'ano': [2000, 2001, 2002, 2001, 2002, 2003],  
        'pop': [1.5, 1.7, 3.6, 2.4, 2.9, 3.2]}  
pd.DataFrame(data, columns=['ano', 'estado', 'pop']) #altera a ordem das colunas
```



	ano	estado	pop
0	2000	Minas Gerais	1.5
1	2001	Minas Gerais	1.7
2	2002	Minas Gerais	3.6
3	2001	São Paulo	2.4
4	2002	São Paulo	2.9
5	2003	São Paulo	3.2



```
frame2=pd.DataFrame(data, columns=['ano', 'estado', 'pop', 'dívida'],  
                     index = ['um', 'dois', 'três', 'quatro', 'cinco', 'seis']) #insere uma coluna não contida no dicionário, que fica  
frame2
```

	ano	estado	pop	dívida
um	2000	Minas Gerais	1.5	NaN
dois	2001	Minas Gerais	1.7	NaN
três	2002	Minas Gerais	3.6	NaN
quatro	2001	São Paulo	2.4	NaN
cinco	2002	São Paulo	2.9	NaN



DataFrame

✓
0s



frame2.ano



```
um      2000
dois    2001
três    2002
quatro  2001
cinco   2002
seis    2003
Name: ano, dtype: int64
```

✓
0s

[9] frame2.loc['três']

```
ano      2002
estado  Minas Gerais
pop      3.6
dívida   NaN
Name: três, dtype: object
```

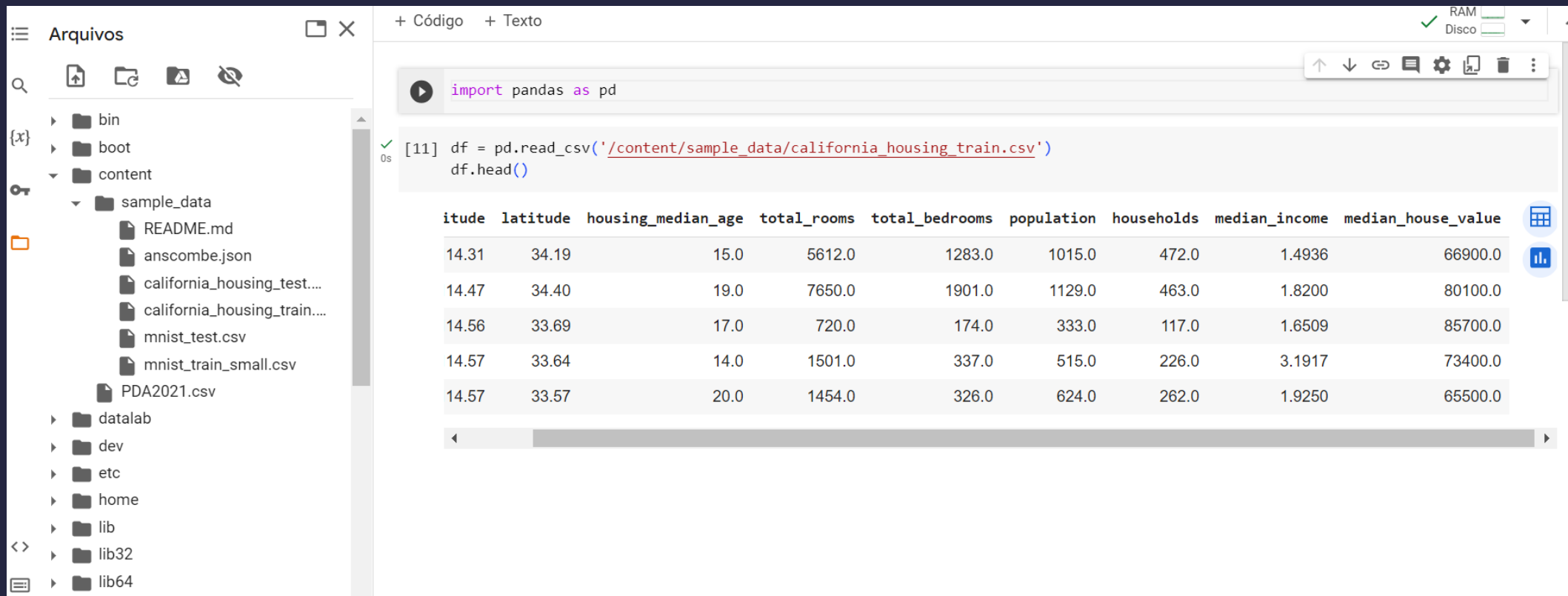
✓
0s



```
frame2['dívida']=16.5
frame2
```

	ano	estado	pop	dívida
um	2000	Minas Gerais	1.5	16.5
dois	2001	Minas Gerais	1.7	16.5
três	2002	Minas Gerais	3.6	16.5
quatro	2001	São Paulo	2.4	16.5

Transformando arquivos CSV em DataFrame



The screenshot displays a JupyterLab environment. On the left, the 'Arquivos' (Files) panel shows a directory tree with folders like 'bin', 'boot', and 'content'. Under 'content', there is a 'sample_data' folder containing several files, including 'california_housing_train.csv'. On the right, the '+ Código' (Code) panel shows a Jupyter notebook with two cells. The first cell contains the code `import pandas as pd`. The second cell contains the code `df = pd.read_csv('/content/sample_data/california_housing_train.csv')` followed by `df.head()`. Below the code, the output shows the first six rows of the DataFrame, with columns: `itude`, `latitude`, `housing_median_age`, `total_rooms`, `total_bedrooms`, `population`, `households`, `median_income`, and `median_house_value`.

itude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
14.31	34.19	15.0	5612.0	1283.0	1015.0	472.0	1.4936	66900.0
14.47	34.40	19.0	7650.0	1901.0	1129.0	463.0	1.8200	80100.0
14.56	33.69	17.0	720.0	174.0	333.0	117.0	1.6509	85700.0
14.57	33.64	14.0	1501.0	337.0	515.0	226.0	3.1917	73400.0
14.57	33.57	20.0	1454.0	326.0	624.0	262.0	1.9250	65500.0

Estatísticas Descritivas

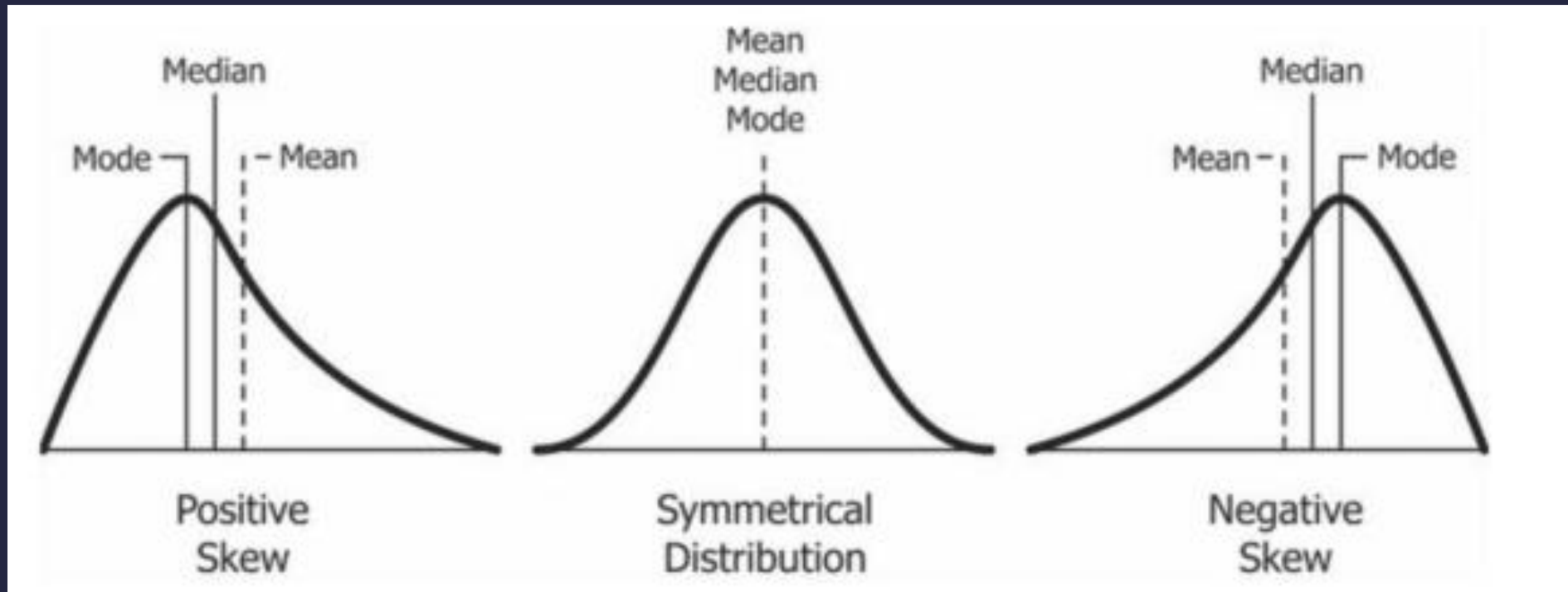
✓ [18] df.describe()
0s

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income
count	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000
mean	-119.562108	35.625225	28.589353	2643.664412	539.410824	1429.573941	501.221941	3.883578
std	2.005166	2.137340	12.586937	2179.947071	421.499452	1147.852959	384.520841	1.908157
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900
25%	-121.790000	33.930000	18.000000	1462.000000	297.000000	790.000000	282.000000	2.566375
50%	-118.490000	34.250000	29.000000	2127.000000	434.000000	1167.000000	409.000000	3.544600
75%	-118.000000	37.720000	37.000000	3151.250000	648.250000	1721.000000	605.250000	4.767000
max	-114.310000	41.950000	52.000000	37937.000000	6445.000000	35682.000000	6082.000000	15.000100

✓ df.sum()
0s

```
longitude      -2.032556e+06
latitude        6.056288e+05
housing_median_age  4.860190e+05
total_rooms      4.494230e+07
total_bedrooms    9.169984e+06
population      2.430276e+07
```

Skewness (assimetria)



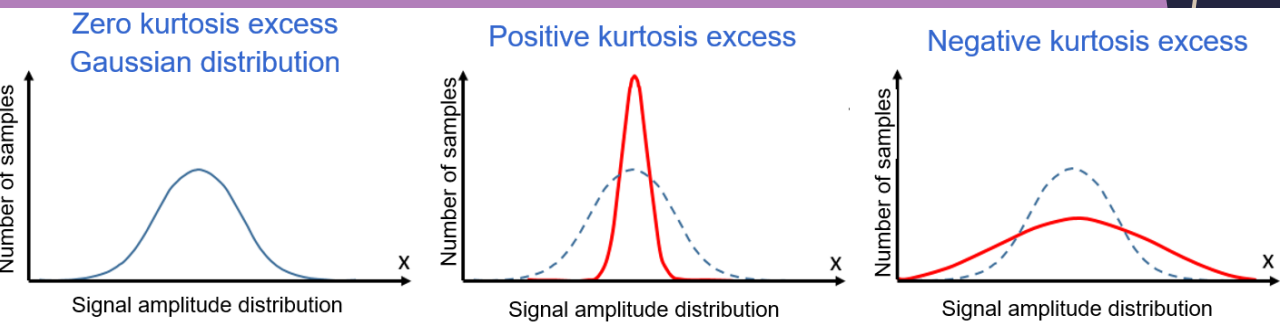
Skewness (assimetria)



```
df.skew()
```



longitude	-0.304003
latitude	0.471801
housing_median_age	0.064894
total_rooms	4.002730
total_bedrooms	3.322637
population	5.187212
households	3.342668
median_income	1.626693
median_house_value	0.973037
dtype:	float64



Kurtosis (Curtose)

Medida de dispersão que caracteriza o "achatamento" da curva da função de distribuição.

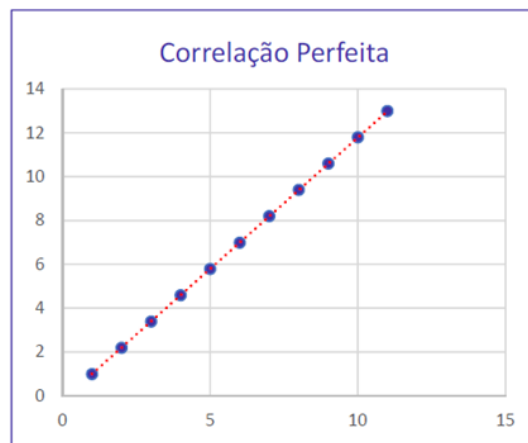
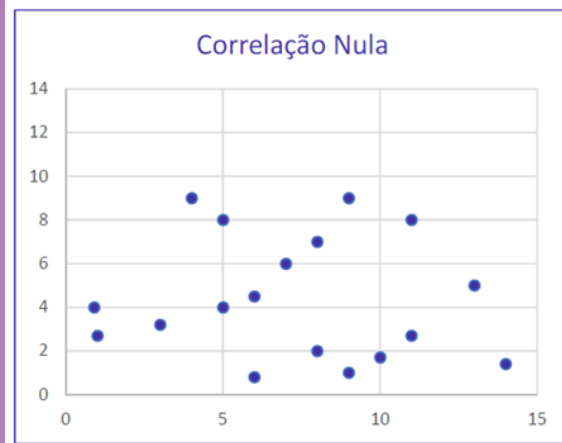
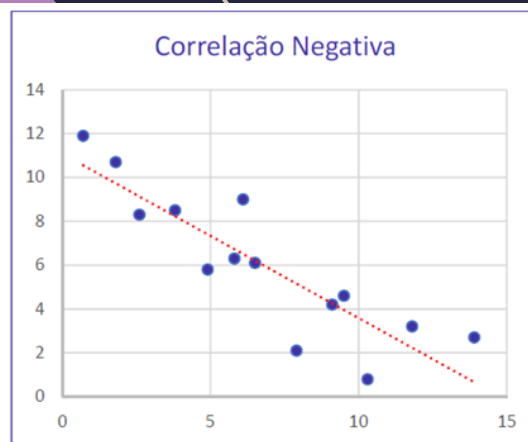
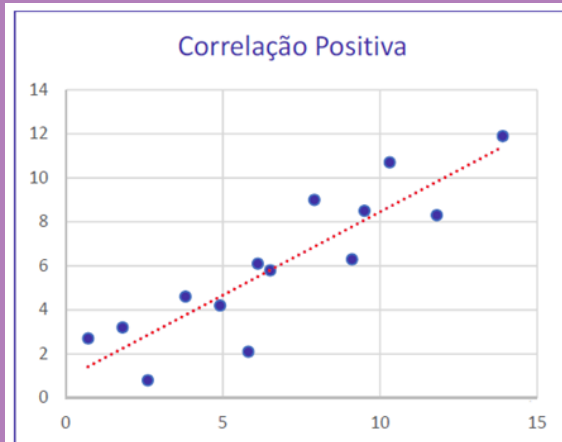
✓
0s



```
df.kurtosis()
```


```
longitude      -1.322330  
latitude        -1.112226  
housing_median_age -0.800826  
total_rooms     29.515885  
total_bedrooms  19.692750  
population      80.861997  
households      20.692645  
median_income    4.764145  
median_house_value 0.303998  
dtype: float64
```

Kurtosis (Curtose)



Correlação

Correlação

0s  df.corr()

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income
longitude	1.000000	-0.925208	-0.114250	0.047010	0.071802	0.101674	0.059628	-0.015485
latitude	-0.925208	1.000000	0.016454	-0.038773	-0.069373	-0.111261	-0.074902	-0.080303
housing_median_age	-0.114250	0.016454	1.000000	-0.360984	-0.320434	-0.295890	-0.302754	-0.115932
total_rooms	0.047010	-0.038773	-0.360984	1.000000	0.928403	0.860170	0.919018	0.195383
total_bedrooms	0.071802	-0.069373	-0.320434	0.928403	1.000000	0.881169	0.980920	-0.013495
population	0.101674	-0.111261	-0.295890	0.860170	0.881169	1.000000	0.909247	-0.000638
households	0.059628	-0.074902	-0.302754	0.919018	0.980920	0.909247	1.000000	0.007644
median_income	-0.015485	-0.080303	-0.115932	0.195383	-0.013495	-0.000638	0.007644	1.000000
median_house_value	-0.044982	-0.144917	0.106758	0.130991	0.045783	-0.027850	0.061031	0.691871

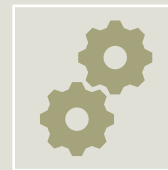
Estatísticas com pandas

- `groupby()`: Permite agrupar dados com base em valores de uma ou mais colunas e, em seguida, aplicar funções de agregação, como soma, média, contagem, etc.
- `crosstab()`: Cria uma tabela de frequência entre duas ou mais variáveis categóricas.
- `isnull()`, `notnull()`: Identifica valores nulos em um DataFrame ou Series.
- `dropna()`: Remove linhas ou colunas com valores nulos.
- `fillna()`: Preenche valores nulos com um valor específico.

Considerações Finais



Importância do
Pandas na Análise de
Dados;



Flexibilidade e
Eficiência das
Estruturas de Dados;



Manipulação de
Dados e Análise
Exploratória;



Integração com
Outras Bibliotecas e
Ferramentas.

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = [10, 20, 30, 40, 50]
```

```
series = pd.Series(data, index=['a', 'b', 'c', 'd', 'e'])
```

```
print(series['c'])
```

Resposta

- A série tem valores indexados de 'a' a 'e'. O valor correspondente ao índice 'c' é 30.

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = [15, 25, 35, 45]
```

```
series = pd.Series(data)
```

```
print(series[series > 30])
```

Resposta

2 35

3 45

dtype: int64

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = {'a': 5, 'b': 10, 'c': 15}
```

```
series = pd.Series(data)
```

```
print(series + 5)
```

Resposta

a 10

b 15

c 20

dtype: int64

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = [2, 4, 6, 8, 10]
```

```
series = pd.Series(data)
```

```
print(series.mean())
```

Resposta

- A média dos valores 2, 4, 6, 8 e 10 é 6.0.

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = [1, 2, 3, 4, 5]
```

```
series = pd.Series(data)
```

```
series[1:4] = 0
```

```
print(series)
```

Resposta

0 1

1 0

2 0

3 0

4 5

dtype: int64

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = {'A': [1, 2, 3], 'B': [4, 5, 6], 'C': [7, 8, 9]}
```

```
df = pd.DataFrame(data)
```

```
print(df.iloc[1])
```

Resposta

A 2

B 5

C 8

Name: 1, dtype: int64

- A segunda linha (índice 1) contém os valores 2, 5 e 8 nas colunas A, B e C, respectivamente.

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = {'A': [1, 2, 3], 'B': [4, 5, 6], 'C': [7, 8, 9]}
```

```
df = pd.DataFrame(data)
```

```
print(df['B'].sum())
```

Resposta

- A soma dos valores na coluna 'B' ($4 + 5 + 6$) é 15.

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = {'A': [10, 20, 30], 'B': [40, 50, 60], 'C': [70, 80, 90]}
```

```
df = pd.DataFrame(data)
```

```
df['D'] = df['A'] + df['B']
```

```
print(df)
```

Resposta

	A	B	C	D
0	10	40	70	50
1	20	50	80	70
2	30	60	90	90

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = {'A': [1, 2], 'B': [3, 4]}
```

```
df = pd.DataFrame(data, index=['row1', 'row2'])
```

```
print(df.loc['row1', 'B'])
```

Resposta

- O valor na linha 'row1' e na coluna 'B' é 3

Qual é o valor exibido no console?

```
import pandas as pd
```

```
data = {'A': [10, 20, 30], 'B': [40, 50, 60]}
```

```
df = pd.DataFrame(data)
```

```
df = df.drop('B', axis=1)
```

```
print(df)
```

Resposta

A

0 10

1 20

2 30

A close-up photograph of a laboratory experiment. A glass pipette is shown dispensing a small, clear drop of red liquid into a test tube. The background is a soft-focus blue, with other laboratory glassware visible in the foreground. The text "Nos vemos no laboratório!" is overlaid on the left side of the image.

Nos vemos no
laboratório!