

Creado por:

Isabel Maniega

## -4- Matrices

### Transpuesta de una matriz

#### Intercambio de filas por columnas

```
In [2]: # pip install numpy
import numpy as np

In [3]: A = np.array([
[1, 2, 3],
[4, 5, 6],
[7, 8, 9],
])
A

Out[3]: array([[1, 2, 3],
[4, 5, 6],
[7, 8, 9]])

In [4]: # forma 1
A.T

Out[4]: array([[1, 4, 7],
[2, 5, 8],
[3, 6, 9]])

In [5]: # forma 2: np.transpose(array a modificar)
np.transpose(A)

Out[5]: array([[1, 4, 7],
[2, 5, 8],
[3, 6, 9]])
```

### Dimensiones de la matriz

```
In [6]: import numpy as np

In [7]: A = np.array([
[1, 2, 3],
[4, 5, 6]
])
A

Out[7]: array([[1, 2, 3],
[4, 5, 6]])

In [8]: len(A)

Out[8]: 2

In [9]: # Obtenemos la dimension filas x columnas
A.shape

Out[9]: (2, 3)

In [10]: # primer valor (0) --> filas
A.shape[0]

Out[10]: 2

In [11]: # segundo valor (1) --> columnas
A.shape[1]

Out[11]: 3

In [12]: len(A.T)

Out[12]: 3

In [13]: transpuesta_A = A.T
transpuesta_A

Out[13]: array([[1, 4],
[2, 5],
[3, 6]])

In [14]: transpuesta_A.shape

Out[14]: (3, 2)

In [15]: B = np.array([
[4, 2, 7],
[5, 6, 5]
])
B

Out[15]: array([[4, 2, 7],
[5, 6, 5]])

In [16]: B.shape

Out[16]: (2, 3)

In [17]: if A.shape == B.shape:
print("mismas dimensiones")
else:
print("dimensiones diferentes")

mismas dimensiones

In [18]: C = np.array([
[4, 8],
[5, 6]
])
C

Out[18]: array([[4, 8],
[5, 6]])

In [19]: C.shape

Out[19]: (2, 2)

In [34]: d = np.array([
1, 2,
3, 4,
6
])
d

Out[34]: array([1, 2, 3, 4, 6])

In [35]: # Agregar un valor como None
d = np.append(d, None)
d

Out[35]: array([1, 2, 3, 4, 6, None], dtype=object)

In [36]: # redimensionar la matriz
d.reshape(3, 2)

Out[36]: array([[1, 2],
[3, 4],
[6, None]], dtype=object)

In [37]: if A.shape == C.shape:
print("mismas dimensiones")
else:
print("dimensiones diferentes")

dimensiones diferentes

In [39]: # Transpuesta B
B_trans = B.T
B_trans

Out[39]: array([[4, 5],
[2, 6],
[7, 5]])

In [40]: B_trans.shape

Out[40]: (3, 2)
```

### Comparativa de varias matrices

```
In [41]: A = np.array([
[1, 2],
[3, 4]
])

B = np.array([
[5, 6],
[7, 8]
])

C = np.array([
[9, 10],
[11, 12]
])

In [42]: if A.shape == B.shape and A.shape == C.shape:
print('son iguales las 3 en dimensiones')
else:
print('son diferentes las 3 en dimensiones')

son iguales las 3 en dimensiones
```

## -4.1.- Dataframes

### Crear un dataframe paso a paso

- Creado a partir de listas

```
In [43]: # Nombres
P = ["Maria", "Luis", "Jose", "Ana"]
P

Out[43]: ['Maria', 'Luis', 'Jose', 'Ana']

In [44]: # Edades
Q = [10, 15, 20, 25]
Q

Out[44]: [10, 15, 20, 25]

In [45]: # pip install pandas
import pandas as pd

In [46]: df = pd.DataFrame(P, columns=["Estudiantes"])
df

Out[46]:
  Estudiantes
0      María
1       Luis
2       Jose
3        Ana

In [48]: df["Edad"] = Q
df

Out[48]:
  Estudiantes  Edad
0      María    10
1       Luis    15
2       Jose    20
3        Ana    25

• Crear un dataframe a partir de Tuplas

In [49]: # Nombres
P = ["Maria", "Luis", "Jose", "Ana"]
P

Out[49]: ['Maria', 'Luis', 'Jose', 'Ana']

In [50]: # Edades
Q = [10, 15, 20, 25]
Q

Out[50]: [10, 15, 20, 25]

In [51]: data = list(zip(P, Q))
data

Out[51]: [('Maria', 10), ('Luis', 15), ('Jose', 20), ('Ana', 25)]

In [52]: df = pd.DataFrame(data, columns=["Estudiantes", "Edad"])
df

Out[52]:
  Estudiantes  Edad
0      María    10
1       Luis    15
2       Jose    20
3        Ana    25

• Crear un dataframe a partir de un diccionario

In [53]: # Nombres
P = ["Maria", "Luis", "Jose", "Ana"]
P

Out[53]: ['Maria', 'Luis', 'Jose', 'Ana']

In [54]: # Edades
Q = [10, 15, 20, 25]
Q

Out[54]: [10, 15, 20, 25]

In [55]: df = pd.DataFrame({"Estudiantes": P, "Edad": Q})
df

Out[55]:
  Estudiantes  Edad
0      María    10
1       Luis    15
2       Jose    20
3        Ana    25

In [57]: # seleccionar columna
df["Estudiantes"]

Out[57]:
0      María
1       Luis
2       Jose
3        Ana
Name: Estudiantes, dtype: object

In [58]: # otra forma seleccionar la columna
df.Estudiantes

Out[58]:
0      María
1       Luis
2       Jose
3        Ana
Name: Estudiantes, dtype: object

In [60]: # Formato tabla, se asigna una nueva variable
df_edades = df[["Edad"]]
df_edades

Out[60]:
  Edad
0    10
1    15
2    20
3    25

In [61]: # seleccionar los alumnos con edades menores a 20 años
df_menores_iguales_20 = df[df["Edad"] <= 20]
df_menores_iguales_20

Out[61]:
  Estudiantes  Edad
0      María    10
1       Luis    15
2       Jose    20

In [62]: # seleccionar los alumnos con edades mayores a 20 años
df_mayores_iguales_20 = df[df["Edad"] >= 20]
df_mayores_iguales_20

Out[62]:
  Estudiantes  Edad
2       Jose    20
3        Ana    25

In [64]: # longitud del dataframe
len(df)

Out[64]: 4

In [66]: df

Out[66]:
  Estudiantes  Edad
0      María    10
1       Luis    15
2       Jose    20
3        Ana    25

In [65]: # filas y columnas que lo componen
df.shape

Out[65]: (4, 2)

In [67]: # resumen de los datos: numero de datos, media, desviación típica, min, max, cuartiles(25%, 50%, 75%):
df.describe()

Out[67]:
  Edad
count  4.000000
mean   17.500000
std     6.454972
min    10.000000
25%    13.750000
50%    17.500000
75%    21.250000
max    25.000000

In [68]: type(df)

Out[68]: pandas.core.frame.DataFrame

In [69]: df["Notas"] = [9, 8, 5, 6]
df

Out[69]:
  Estudiantes  Edad  Notas
0      María    10      9
1       Luis    15      8
2       Jose    20      5
3        Ana    25      6

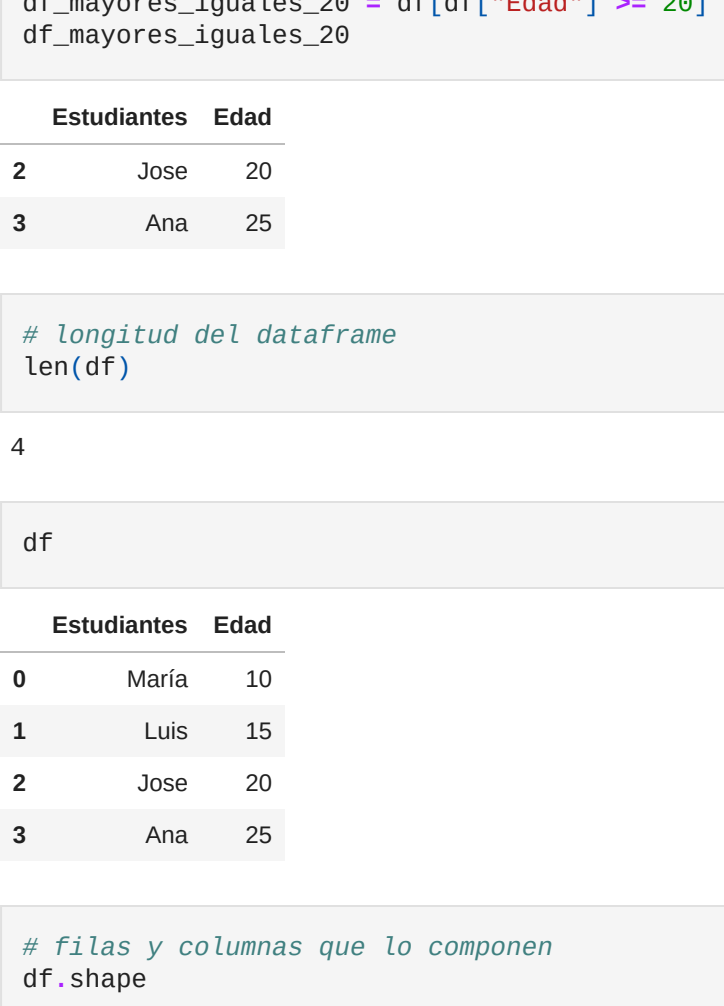
In [72]: df.Notas.plot(kind="bar")

Out[72]: <AxesSubplot:~>
```



```
In [73]: df.Edad.plot(kind="bar")

Out[73]: <AxesSubplot:~>
```



```
In [70]: import numpy as np

In [71]: np.mean([10, 30])

Out[71]: 20.0
```

Creado por:

Isabel Maniega