Notebook_Numpy_Pandas

November 4, 2025

0.1 INTRODUCCIÓN A NUMPY

¿QUE ES NUMPY? Es una biblioteca de Python de código abierto, que proporciona soporte para vectores y matrices multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas. Es muy usada para análisis de datos y prerpocesamiento para algoritmos de IA.

0.1.1 INSTALACIÓN

[ ]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in
c:\users\homer\appdata\local\programs\python\python313\lib\site-packages (2.3.4)
Note: you may need to restart the kernel to use updated packages.


[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[35]:
```
import numpy as np
print (np.__version__)
```

```
2.3.4
```

0.1.2 CREACIÓN DE ARRAYS

Una dimensión

[3]:
```
a=np.array([1,2,3,4,5])
print(a)
```

```
[1 2 3 4 5]
```

Dos o más dimensiones

[5]:
```
b=np.array([[1,2,3],[4,5,6],[7,8,9]])
print(b)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

0.1.3 FUNCIONES CON ARRAYS

ZEROS() genera un array de ceros del tamaño que le indiquemos:

[15]:
```
c=np.zeros([5,5])
print(c)
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

ONES() genera un array de unos del tamaño que le indiquemos:

[30]:
```
d=np.ones(5)
print(d)
```

```
[1. 1. 1. 1. 1.]
```

ARANGE() genera un array entre dos valores indicados y a intervalos específicos, similar a la función range() de Python:

[22]:
```
e=np.arange(0, 10)
e
```

[22]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

LINSPACE() genera un array entre dos valores y con un tamaño indicado, compuesto por valores equidistantes entre cada elemento:

[42]:
```
f=np.linspace(0, 2, 5)
f
```

[42]:
```
array([0. , 0.5, 1. , 1.5, 2. ])
```

EYE() genera una matriz identidad del tamaño indicado:

[27]:
```
g=np.eye(5)
g
```

[27]:
```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

RANDOM.RAND() genera un array del tamaño indicado, con valores aleatorios entre 0 y 1:

[28]:
```
h=np.random.rand(5)
h
```

[28]:
```
array([0.23967125, 0.94585285, 0.95548386, 0.54172266, 0.89953928])
```

También podemos usar **OPERACIONES MATEMÁTICAS** entre arrays o entre un array y una constante, como sumar, restar, multiplicar y dividir:

[31]:
```
a+d
```

[31]:
```
array([2., 3., 4., 5., 6.])
```

[32]:
```
g*7
```

[32]:
```
array([[7., 0., 0., 0., 0.],
       [0., 7., 0., 0., 0.],
       [0., 0., 7., 0., 0.],
       [0., 0., 0., 7., 0.],
       [0., 0., 0., 0., 7.]])
```

0.2 INTRODUCCIÓN A PANDAS

¿QUÉ ES PANDAS? Es una biblioteca de código abierto de Python, muy utilizada en el análisis y manipulación de datos. Proporciona las estructuras de datos **SERIES** y **DATAFRAME**, que permiten trabajar con datos tabulares (similares a una hoja de cálculo) de una forma sencilla e intuitiva. Las series se asemejarían a una fila o columna; mientras que un dataframe será como una hoja de cálculo. Es muy utilizada en la ciencia de datos para tareas de limpieza, transformación y análisis de datos para aprendizaje automático.

INSTALACIÓN

[ ]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in
c:\users\homer\appdata\local\programs\python\python313\lib\site-packages (2.3.3)
Requirement already satisfied: numpy>=1.26.0 in
c:\users\homer\appdata\local\programs\python\python313\lib\site-packages (from
pandas) (2.3.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\users\homer\appdata\roaming\python\python313\site-packages (from pandas)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\homer\appdata\local\programs\python\python313\lib\site-packages (from
pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\homer\appdata\local\programs\python\python313\lib\site-packages (from
pandas) (2025.2)
Requirement already satisfied: six>=1.5 in
c:\users\homer\appdata\roaming\python\python313\site-packages (from python-
```

```
dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

[3]:
```
import pandas as pd
```

CREACIÓN DE SERIES

[4]:
```
serie1 = pd.Series([1,2,3])
serie1.name="primeros"
print(serie1)
```

```
0    1
1    2
2    3
Name: primeros, dtype: int64
```

CREACIÓN DE DATAFRAMES

[5]:
```
df1 = pd.DataFrame({"columna1":[3,2,5,4,1], "columna2":["a","b","c","d","e"],␣
↪"columna3":["?","!","#","@","&"]})
df1
```

[5]:

	columna1	columna2	columna3
0	3	a	?
1	2	b	!
2	5	c	#
3	4	d	@
4	1	e	&

CREAR UN DATAFRAME A PARTIR DE UN FICHERO CSV Debemos especificar la ruta o URL de nuestro fichero:

[13]:
```
df2 = pd.read_csv("https://gist.githubusercontent.com/curran/
↪a08a1080b88344b0c8a7/raw/0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv")
print(df2.head(3))
```

```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
```

PRINCIPALES FUNCIONES CON DATAFRAMES Ver cuantos elementos tiene nuestro Dataframe.

[9]:
```
print(df2.size)
```

```
750
```

Ver cuantas filas y columnas tiene nuestro Dataframe.

[10]:
```
print(df2.shape)
```

```
(150, 5)
```

Muestra las X primeras filas de nuestro dataframe (por defecto 5):

[ ]:
```
df1.head(2)
```

Muestra las X últimas filas de nuestro dataframe (por defecto 5):

[16]:
```
print(df2.tail())
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica
```

Podemos personalizar los nombres de las columnas:

[19]:
```
df2.
  ↪columns=["largo_sepalo","ancho_sepalo","largo_petalo","ancho_petalo","especie"]
print(df2.head())
```

```
   largo_sepalo  ancho_sepalo  largo_petalo  ancho_petalo especie
0           5.1           3.5           1.4           0.2  setosa
1           4.9           3.0           1.4           0.2  setosa
2           4.7           3.2           1.3           0.2  setosa
3           4.6           3.1           1.5           0.2  setosa
4           5.0           3.6           1.4           0.2  setosa
```

Devuelve los identificadores de columna de nuestro dataframe:

[23]:
```
df2.columns
```

[23]:
```
Index(['largo_sepalo', 'ancho_sepalo', 'largo_petalo', 'ancho_petalo',
       'especie'],
      dtype='object')
```

Devuelve los identificadores de fila de nuestro dataframe:

[36]:
```
df1.index
```

[36]:
```
RangeIndex(start=0, stop=5, step=1)
```

Muestra estadísticas generícas del dataframe:

[25]:
```
df2.describe()
```

[25]:
```
       largo_sepalo  ancho_sepalo  largo_petalo  ancho_petalo
count    150.000000    150.000000    150.000000    150.000000
mean       5.843333      3.054000      3.758667      1.198667
std        0.828066      0.433594      1.764420      0.763161
min        4.300000      2.000000      1.000000      0.100000
25%        5.100000      2.800000      1.600000      0.300000
50%        5.800000      3.000000      4.350000      1.300000
75%        6.400000      3.300000      5.100000      1.800000
max        7.900000      4.400000      6.900000      2.500000
```

Muestra un recuento de valores de la columna o columnas indicadas:

[29]:
```
df2["especie"].value_counts()
```

[29]:
```
especie
setosa        50
versicolor    50
virginica     50
Name: count, dtype: int64
```

Muestra los tipos de datos por columnas:

[30]:
```
df2.dtypes
```

[30]:
```
largo_sepalo    float64
ancho_sepalo    float64
largo_petalo    float64
ancho_petalo    float64
especie          object
dtype: object
```

Muestra el espacio de memoria que ocupa el Dataframe:

[34]:
```
df2.memory_usage().sum()
```

[34]:
```
np.int64(6132)
```

Podemos transponer las filas y columnas del dataframe:

[38]:
```
print(df2.T.head())
```

```
                0       1       2       3       4       5       6       7    \
largo_sepalo    5.1     4.9     4.7     4.6     5.0     5.4     4.6     5.0
ancho_sepalo    3.5     3.0     3.2     3.1     3.6     3.9     3.4     3.4
largo_petalo    1.4     1.4     1.3     1.5     1.4     1.7     1.4     1.5
ancho_petalo    0.2     0.2     0.2     0.2     0.2     0.4     0.3     0.2
especie      setosa  setosa  setosa  setosa  setosa  setosa  setosa  setosa

                8       9    …       140       141       142       143  \
```

```
largo_sepalo     4.4     4.9  …        6.7        6.9        5.8        6.8
ancho_sepalo     2.9     3.1  …        3.1        3.1        2.7        3.2
largo_petalo     1.4     1.5  …        5.6        5.1        5.1        5.9
ancho_petalo     0.2     0.1  …        2.4        2.3        1.9        2.3
especie       setosa  setosa  …  virginica  virginica  virginica  virginica

                  144        145        146        147        148        149
largo_sepalo      6.7        6.7        6.3        6.5        6.2        5.9
ancho_sepalo      3.3        3.0        2.5        3.0        3.4        3.0
largo_petalo      5.7        5.2        5.0        5.2        5.4        5.1
ancho_petalo      2.5        2.3        1.9        2.0        2.3        1.8
especie       virginica  virginica  virginica  virginica  virginica  virginica

[5 rows x 150 columns]
```

Ordenar valores

[ ]:
```
print(df1.sort_values("columna1", ascending=False))
```

[ ]:
```
   columna1 columna2 columna3
0         3        a        ?
1         2        b        !
2         5        c        #
3         4        d        @
4         1        e        &
```

Podemos especificar una columna concreta del dataframe:

[47]:
```
print(df2.especie)
```

```
0         setosa
1         setosa
2         setosa
3         setosa
4         setosa
         …
145    virginica
146    virginica
147    virginica
148    virginica
149    virginica
Name: especie, Length: 150, dtype: object
```

Podemos especificar una serie de columnas y filas concretas a mostrar:

[46]:
```
df1.iloc[[1,4],[1,2]]
```

[46]:
```
  columna2 columna3
1        b        !
```

```
4        e        &
```

Tambien especificar varias columnas por sus identificadores:

[50]:
```
print(df1[["columna1", "columna3"]])
```

```
   columna1 columna3
0         3        ?
1         2        !
2         5        #
3         4        @
4         1        &
```

Ver valores de cierta columna que cumplan un criterio:

[51]:
```
df1[df1["columna1"]>2]
```

[51]:
```
   columna1 columna2 columna3
0         3        a        ?
2         5        c        #
3         4        d        @
```

-

[64]:
```
df1.columna1[3]=np.nan
df1.isna().sum()
```

[64]:
```
columna1    1
columna2    0
columna3    0
dtype: int64
```

-

[ ]:
```
df1["letras"][1:3]=np.nan
df1.head()
```

-

[ ]:
```
df1["letras"] = df1["letras"].fillna(8)
df1.head()
```

-

[ ]:
```
df1["numeros"].mean()
```

-

[ ]:
```
df1["numeros"].median()
```

Podemos organizar valores en base a una columna, por ejemplo, obtener la media de alturas por clase:

[ ]:
```
media_alturas = df2.groupby("clase")["altura"].mean()
media_alturas.name="media altura por clase"
```

Podemos unir una serie de datos a nuestro dataframe

[ ]:
```
df3 = df2.join(media_alturas, on="clase", how="inner")
df3.head()
```

0.3 INTRODUCCIÓN A MATPLOTLIB

[ ]:
```
pip install matplotlib
```

```
Collecting matplotlib
  Downloading matplotlib-3.10.7-cp313-cp313-win_amd64.whl.metadata (11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.3-cp313-cp313-win_amd64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.60.1-cp313-cp313-win_amd64.whl.metadata (114 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Downloading kiwisolver-1.4.9-cp313-cp313-win_amd64.whl.metadata (6.4 kB)
Requirement already satisfied: numpy>=1.23 in
c:\users\homer\appdata\local\programs\python\python313\lib\site-packages (from
matplotlib) (2.3.4)
Requirement already satisfied: packaging>=20.0 in
c:\users\homer\appdata\roaming\python\python313\site-packages (from matplotlib)
(25.0)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-12.0.0-cp313-cp313-win_amd64.whl.metadata (9.0 kB)
Collecting pyparsing>=3 (from matplotlib)
  Downloading pyparsing-3.2.5-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\homer\appdata\roaming\python\python313\site-packages (from matplotlib)
(2.9.0.post0)
Requirement already satisfied: six>=1.5 in
c:\users\homer\appdata\roaming\python\python313\site-packages (from python-
dateutil>=2.7->matplotlib) (1.17.0)
Downloading matplotlib-3.10.7-cp313-cp313-win_amd64.whl (8.1 MB)
   ---------------------------------------- 0.0/8.1 MB ? eta -:--:--
   ----------- ---------------------------- 2.4/8.1 MB 11.6 MB/s eta 0:00:01
   ---------------------------- ------------ 5.5/8.1 MB 12.8 MB/s eta 0:00:01
   ---------------------------------------- 8.1/8.1 MB 12.6 MB/s  0:00:00
Downloading contourpy-1.3.3-cp313-cp313-win_amd64.whl (226 kB)
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
```

```
Downloading fonttools-4.60.1-cp313-cp313-win_amd64.whl (2.3 MB)
   ---------------------------------------- 0.0/2.3 MB ? eta -:--:--
   ---------------------------------- -- 2.1/2.3 MB 17.4 MB/s eta 0:00:01
   ---------------------------------------- 2.3/2.3 MB 6.0 MB/s  0:00:00
Downloading kiwisolver-1.4.9-cp313-cp313-win_amd64.whl (73 kB)
Downloading pillow-12.0.0-cp313-cp313-win_amd64.whl (7.0 MB)
   ---------------------------------------- 0.0/7.0 MB ? eta -:--:--
   ------------------ --------------------- 3.4/7.0 MB 16.6 MB/s eta 0:00:01
   ------------------------------------ - 6.8/7.0 MB 17.1 MB/s eta 0:00:01
   ---------------------------------------- 7.0/7.0 MB 14.6 MB/s  0:00:00
Downloading pyparsing-3.2.5-py3-none-any.whl (113 kB)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler,
contourpy, matplotlib

   ---------------------------------------- 0/7 [pyparsing]
   ---------------------------------------- 0/7 [pyparsing]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----- ---------------------------------- 1/7 [pillow]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
   ----------------- ---------------------- 3/7 [fonttools]
```

```
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
```

```
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
----------------- ---------------------- 3/7 [fonttools]
---------------------------- ----------- 5/7 [contourpy]
---------------------------- ----------- 5/7 [contourpy]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
---------------------------------- ----- 6/7 [matplotlib]
```

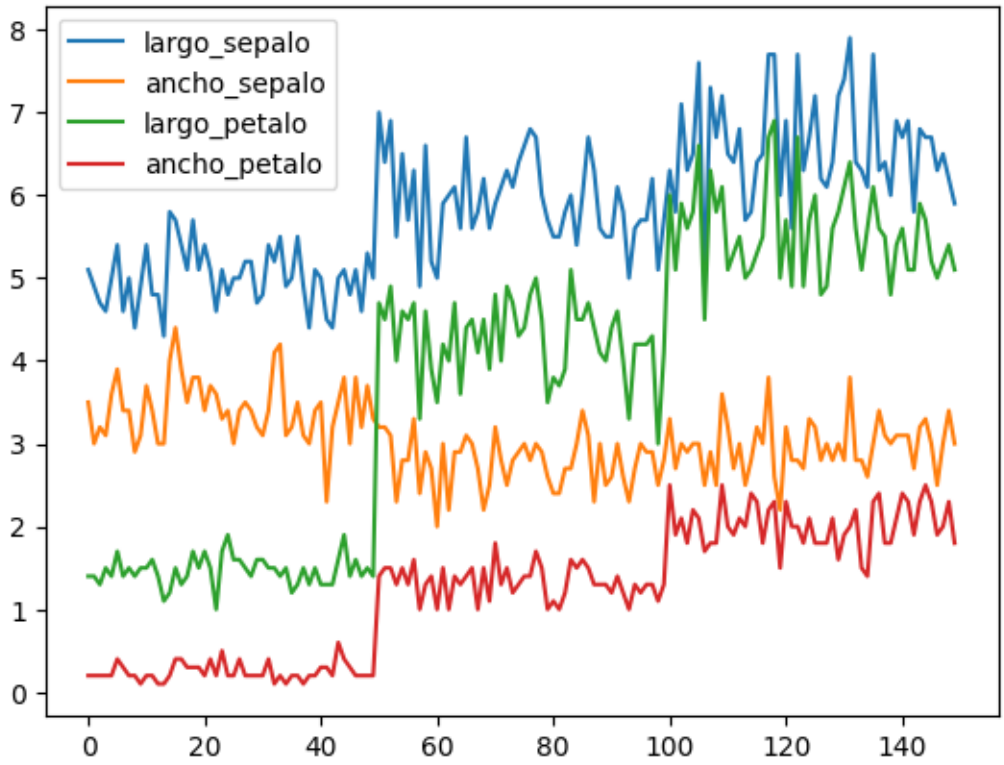
```
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------- ----- 6/7 [matplotlib]
   ---------------------------------------- 7/7 [matplotlib]

Successfully installed contourpy-1.3.3 cycler-0.12.1 fonttools-4.60.1
kiwisolver-1.4.9 matplotlib-3.10.7 pillow-12.0.0 pyparsing-3.2.5
Note: you may need to restart the kernel to use updated packages.
```

•

[67]:
```python
import matplotlib.pyplot as plt
df2.plot()
```

```
Matplotlib is building the font cache; this may take a moment.
```

[67]: <Axes: >



-

```
[ ]: df3.plot()
```

-

```
[ ]: df3.plot(kind='bar')
```