```python
# Pandas es una biblioteca para acceso y manipulación de datos
import pandas as pd
```

```python
datos = pd.read_csv('https://bit.ly/31B56KB')
type(datos)
```

```
pandas.core.frame.DataFrame
```

```python
datos.head(3)
```

	Contaminacion_SO2	Temperatura	Fabricas	Habitantes	Velocidad_viento	Lluvia
0	10	70.3	213	582	6.0	7.05
1	13	61.0	91	132	8.2	48.52
2	12	56.7	453	716	8.7	20.66

```python
datos.tail(3)
```

	Contaminacion_SO2	Temperatura	Fabricas	Habitantes	Velocidad_viento	Lluvi
38	29	51.1	379	531	9.4	38.7
39	31	55.2	35	71	6.5	40.7
40	16	45.7	569	717	11.8	29.0

```python
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41 entries, 0 to 40
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Contaminacion_SO2  41 non-null     int64  
 1   Temperatura        41 non-null     float64
 2   Fabricas           41 non-null     int64  
 3   Habitantes         41 non-null     int64  
 4   Velocidad_viento   41 non-null     float64
 5   Lluvia             41 non-null     float64
 6   Dias_Lluvia        41 non-null     int64  
dtypes: float64(3), int64(4)
memory usage: 2.4 KB
```

```python
datos.describe()
```

	Contaminacion_SO2	Temperatura	Fabricas	Habitantes	Velocidad_viento	
count	41.000000	41.000000	41.000000	41.000000	41.000000	4
mean	30.048780	55.763415	463.097561	608.609756	9.443902	3
std	23.472272	7.227716	563.473948	579.113023	1.428644	1
min	8.000000	43.500000	35.000000	71.000000	6.000000	
25%	13.000000	50.600000	181.000000	299.000000	8.700000	3
50%	26.000000	54.600000	347.000000	515.000000	9.300000	3
75%	35.000000	59.300000	462.000000	717.000000	10.600000	4

```
len(datos)
```

```
41
```

```
datos.columns
```

```
Index(['Contaminacion_SO2', 'Temperatura', 'Fabricas', 'Habitantes',
       'Velocidad_viento', 'Lluvia', 'Dias_Lluvia'],
      dtype='object')
```

```
datos.dtypes
```

```
Contaminacion_SO2      int64
Temperatura          float64
Fabricas               int64
Habitantes             int64
Velocidad_viento     float64
Lluvia               float64
Dias_Lluvia            int64
dtype: object
```

```
datos.shape
```

```
(41, 7)
```

```
datos.shape[0]
```

```
41
```

```
datos.values
```

```
array([[  10.  ,   70.3 ,  213.  ,  582.  ,    6.  ,    7.05,   36.  ],
       [  13.  ,   61.  ,   91.  ,  132.  ,    8.2 ,   48.52,  100.  ],
       [  12.  ,   56.7 ,  453.  ,  716.  ,    8.7 ,   20.66,   67.  ],
       [  17.  ,   51.9 ,  454.  ,  515.  ,    9.  ,   12.95,   86.  ],
       [  56.  ,   49.1 ,  412.  ,  158.  ,    9.  ,   43.37,  127.  ],
       [  36.  ,   54.  ,   80.  ,   80.  ,    9.  ,   40.25,  114.  ],
       [  29.  ,   57.3 ,  434.  ,  757.  ,    9.3 ,   38.89,  111.  ],
       [  14.  ,   68.4 ,  136.  ,  529.  ,    8.8 ,   54.47,  116.  ],
       [  10.  ,   75.5 ,  207.  ,  335.  ,    9.  ,   59.8 ,  128.  ],
       [  24.  ,   61.5 ,  368.  ,  497.  ,    9.1 ,   48.34,  115.  ],
       [ 110.  ,   50.6 , 3344.  , 3369.  ,   10.4 ,   34.44,  122.  ],
       [  28.  ,   52.3 ,  361.  ,  746.  ,    9.7 ,   38.74,  121.  ],
       [  17.  ,   49.  ,  104.  ,  201.  ,   11.2 ,   30.85,  103.  ],
       [   8.  ,   56.6 ,  125.  ,  277.  ,   12.7 ,   30.58,   82.  ],
       [  30.  ,   55.6 ,  291.  ,  593.  ,    8.3 ,   43.11,  123.  ],
       [   9.  ,   68.3 ,  204.  ,  361.  ,    8.4 ,   56.77,  113.  ],
       [  47.  ,   55.  ,  625.  ,  905.  ,    9.6 ,   41.31,  111.  ],
       [  35.  ,   49.9 , 1064.  , 1513.  ,   10.1 ,   30.96,  129.  ],
       [  29.  ,   43.5 ,  699.  ,  744.  ,   10.6 ,   25.94,  137.  ],
       [  14.  ,   54.5 ,  381.  ,  507.  ,   10.  ,   37.  ,   99.  ],
       [  56.  ,   55.9 ,  775.  ,  622.  ,    9.5 ,   35.89,  105.  ],
       [  14.  ,   51.5 ,  181.  ,  347.  ,   10.9 ,   30.18,   98.  ],
       [  11.  ,   56.8 ,   46.  ,  244.  ,    8.9 ,    7.77,   58.  ],
       [  46.  ,   47.6 ,   44.  ,  116.  ,    8.8 ,   33.36,  135.  ],
       [  11.  ,   47.1 ,  391.  ,  463.  ,   12.4 ,   36.11,  166.  ],
       [  23.  ,   54.  ,  462.  ,  453.  ,    7.1 ,   39.04,  132.  ],
       [  65.  ,   49.7 , 1007.  ,  751.  ,   10.9 ,   34.99,  155.  ],
       [  26.  ,   51.5 ,  266.  ,  540.  ,    8.6 ,   37.01,  134.  ],
       [  69.  ,   54.6 , 1692.  , 1950.  ,    9.6 ,   39.93,  115.  ],
       [  61.  ,   50.4 ,  347.  ,  520.  ,    9.4 ,   36.22,  147.  ],
       [  94.  ,   50.  ,  343.  ,  179.  ,   10.6 ,   42.75,  125.  ],
       [  10.  ,   61.6 ,  337.  ,  624.  ,    9.2 ,   49.1 ,  105.  ],
       [  18.  ,   59.4 ,  275.  ,  448.  ,    7.9 ,   46.  ,  119.  ],
       [   9.  ,   66.2 ,  641.  ,  844.  ,   10.9 ,   35.94,   78.  ],
       [  10.  ,   68.9 ,  721.  , 1233.  ,   10.8 ,   48.19,  103.  ],
       [  28.  ,   51.  ,  137.  ,  176.  ,    8.7 ,   15.17,   89.  ],
       [  31.  ,   59.3 ,   96.  ,  308.  ,   10.6 ,   44.68,  116.  ],
       [  26.  ,   57.8 ,  197.  ,  299.  ,    7.6 ,   42.59,  115.  ],
       [  29.  ,   51.1 ,  379.  ,  531.  ,    9.4 ,   38.79,  164.  ],
       [  31.  ,   55.2 ,   35.  ,   71.  ,    6.5 ,   40.75,  148.  ],
       [  16.  ,   45.7 ,  569.  ,  717.  ,   11.8 ,   29.07,  123.  ]])
```

```
datos['Temperatura'].head(3)
```

```
0    70.3
1    61.0
2    56.7
Name: Temperatura, dtype: float64
```

```
datos.Temperatura.head(3)
```

```
0    70.3
1    61.0
2    56.7
Name: Temperatura, dtype: float64
```

```
# Series: almacena una columna
s = datos.Contaminacion_SO2
type(s)
```

```
pandas.core.series.Series
```

```
s.dtype
```

```
dtype('int64')
```

```
s.values
```

```
array([ 10,  13,  12,  17,  56,  36,  29,  14,  10,  24, 110,  28,  17,
         8,  30,   9,  47,  35,  29,  14,  56,  14,  11,  46,  11,  23,
        65,  26,  69,  61,  94,  10,  18,   9,  10,  28,  31,  26,  29,
        31,  16])
```

```
s.describe()
```

```
count     41.000000
mean      30.048780
std       23.472272
min        8.000000
25%       13.000000
50%       26.000000
75%       35.000000
max      110.000000
Name: Contaminacion_SO2, dtype: float64
```

```
s.mean()
```

```
30.048780487804876
```

```
s.sum()
```

```
1232
```

```
s.unique()
```

```
array([ 10,  13,  12,  17,  56,  36,  29,  14,  24, 110,  28,   8,  30,
         9,  47,  35,  11,  46,  23,  65,  26,  69,  61,  94,  18,  31,
        16])
```

```
s.value_counts()
```

```
10     4
14     3
29     3
9      2
31     2
28     2
26     2
11     2
```

```
56     2
17     2
61     1
46     1
69     1
8      1
12     1
13     1
23     1
16     1
18     1
24     1
94     1
30     1
35     1
36     1
110    1
47     1
65     1
Name: Contaminacion_SO2, dtype: int64
```

```
s.sort_values(ascending=True).head(10)
```

```
13     8
33     9
15     9
0     10
34    10
31    10
8     10
24    11
22    11
2     12
Name: Contaminacion_SO2, dtype: int64
```

```
s[9:16]
```

```
9      24
10    110
11     28
12     17
13      8
14     30
15      9
Name: Contaminacion_SO2, dtype: int64
```

```
datos[9:16]
```

	Contaminacion_SO2	Temperatura	Fabricas	Habitantes	Velocidad_viento	Lluvi
9	24	61.5	368	497	9.1	48.3
10	110	50.6	3344	3369	10.4	34.4
11	28	52.3	361	746	9.7	38.7
12	17	49.0	104	201	11.2	30.8

```
datos.iloc[3:8,:]
```

	Contaminacion_SO2	Temperatura	Fabricas	Habitantes	Velocidad_viento	Lluvia
3	17	51.9	454	515	9.0	12.95
4	56	49.1	412	158	9.0	43.37
5	36	54.0	80	80	9.0	40.25
6	29	57.3	434	757	9.3	38.89
7	14	68.4	136	529	8.8	54.47

```
datos.iloc[3:8,2:6]
```

	Fabricas	Habitantes	Velocidad_viento	Lluvia
3	454	515	9.0	12.95
4	412	158	9.0	43.37
5	80	80	9.0	40.25
6	434	757	9.3	38.89
7	136	529	8.8	54.47

```
datos.iloc[:,2:6]
```

	Fabricas	Habitantes	Velocidad_viento	Lluvia
0	213	582	6.0	7.05
1	91	132	8.2	48.52
2	453	716	8.7	20.66
3	454	515	9.0	12.95
4	412	158	9.0	43.37
5	80	80	9.0	40.25
6	434	757	9.3	38.89
7	136	529	8.8	54.47
8	207	335	9.0	59.80
9	368	497	9.1	48.34
10	3344	3369	10.4	34.44
11	361	746	9.7	38.74
12	104	201	11.2	30.85
13	125	277	12.7	30.58
14	291	593	8.3	43.11
15	204	361	8.4	56.77
16	625	905	9.6	41.31
17	1064	1513	10.1	30.96
18	699	744	10.6	25.94
19	381	507	10.0	37.00
20	775	622	9.5	35.89
21	181	347	10.9	30.18
22	46	244	8.9	7.77

```
datos.iloc[:,-3:6]
```

	Velocidad_viento	Lluvia
0	6.0	7.05
1	8.2	48.52
2	8.7	20.66
3	9.0	12.95
4	9.0	43.37
5	9.0	40.25
6	9.3	38.89
7	8.8	54.47
8	9.0	59.80
9	9.1	48.34
10	10.4	34.44
11	9.7	38.74
12	11.2	30.85
13	12.7	30.58
14	8.3	43.11
15	8.4	56.77
16	9.6	41.31
17	10.1	30.96
18	10.6	25.94
19	10.0	37.00
20	9.5	35.89
21	10.9	30.18
22	8.9	7.77
23	8.8	33.36
24	12.4	36.11

```
# Columnas específicas de un DF
df2 = datos.filter(['Temperatura', 'Dias_Lluvia'])
df2
```

	Temperatura	Dias_Lluvia
0	70.3	36
1	61.0	100
2	56.7	67
3	51.9	86
4	49.1	127
5	54.0	114
6	57.3	111
7	68.4	116
8	75.5	128
9	61.5	115
10	50.6	122
11	52.3	121
12	49.0	103
13	56.6	82
14	55.6	123
15	68.3	113
16	55.0	111
17	49.9	129
18	43.5	137
19	54.5	99
20	55.9	105
21	51.5	98
22	56.8	58
23	47.6	135
24	47.1	166
25	54.0	132
26	49.7	155
27	51.5	134
28	54.6	115

```
df2 = datos.drop(['Temperatura', 'Dias_Lluvia'], axis=1)
df2
```

```
     Contaminacion_SO2  Fabricas  Habitantes  Velocidad_viento  Lluvia
```

```
s = datos.Dias_Lluvia
s.values
```

```
array([ 36, 100,  67,  86, 127, 114, 111, 116, 128, 115, 122, 121, 103,
        82, 123, 113, 111, 129, 137,  99, 105,  98,  58, 135, 166, 132,
       155, 134, 115, 147, 125, 105, 119,  78, 103,  89, 116, 115, 164,
       148, 123])
```

```
s.value_counts()
```

```
115    3
105    2
123    2
116    2
103    2
111    2
127    1
148    1
147    1
82     1
100    1
78     1
137    1
89     1
135    1
134    1
132    1
67     1
129    1
86     1
99     1
155    1
98     1
125    1
36     1
166    1
58     1
113    1
114    1
119    1
164    1
121    1
122    1
128    1
Name: Dias_Lluvia, dtype: int64
```

```
grupo = datos.groupby('Dias_Lluvia')
type(grupo)
```

```
pandas.core.groupby.generic.DataFrameGroupBy
```

```
 29                   61        347          520                    9.4    36.22
```

```
grupo.mean()
grupo.mean()
```

	Contaminacion_SO2	Temperatura	Fabricas	Habitantes	Velocidad_vi

```
grupo.sum().head(20)
```

	Contaminacion_SO2	Temperatura	Fabricas	Habitantes	Velocidad_vien
Dias_Lluvia					
36	10	70.3	213	582	
58	11	56.8	46	244	
67	12	56.7	453	716	
78	9	66.2	641	844	1
82	8	56.6	125	277	1
86	17	51.9	454	515	
89	28	51.0	137	176	
98	14	51.5	181	347	1
99	14	54.5	381	507	1
100	13	61.0	91	132	
103	27	117.9	825	1434	2
105	66	117.5	1112	1246	1
111	76	112.3	1059	1662	1
113	9	68.3	204	361	
114	36	54.0	80	80	
115	119	173.9	2257	2746	2
116	45	127.7	232	837	1
119	18	59.4	275	448	
121	28	52.3	361	746	
122	110	50.6	3344	3369	1

128	10.000000	75.500000	207.000000	335.000000	9.00

```
s_grupo = grupo.sum()
cond = s_grupo.Temperatura >= 65
cond
```

```
Dias_Lluvia
36      True
58     False
67     False
78      True
```

```
82     False
86     False
89     False
98     False
99     False
100    False
103     True
105     True
111     True
113     True
114    False
115     True
116     True
119    False
121    False
122    False
123     True
125    False
127    False
128     True
129    False
132    False
134    False
135    False
137    False
147    False
148    False
155    False
164    False
166    False
Name: Temperatura, dtype: bool
```

```
s_grupo[cond]
```

```
# Numpy: utilizar arreglos y matrices "a la matlab"
import numpy as np
```

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

```
numpy.ndarray
```

105 66 117.5 1112 1246 1

```
v.shape
```

```
(5,)
```

115 119 173.9 2257 2746 2

```
v.ndim
```

```
1
```

```
v.dtype
```

```
dtype('int64')
```

```
v + v
```

```
array([ 2,  4,  6,  8, 10])
```

```
7.18 * v
```

```
array([ 7.18, 14.36, 21.54, 28.72, 35.9 ])
```

```
v * v
```

```
array([ 1,  4,  9, 16, 25])
```

```
v @ v # producto punto
```

```
55
```

```
np.zeros(5)
```

```
array([0., 0., 0., 0., 0.])
```

```
np.ones(5)
```

```
array([1., 1., 1., 1., 1.])
```

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

```
m = np.array([[1,2,3],[4,5,6],[7,8,9]])
type(m)
```

```
numpy.ndarray
```

```
m.shape
```

```
(3, 3)
```

```
m + m
```

```
array([[ 2,  4,  6],
       [ 8, 10, 12],
       [14, 16, 18]])
```

```
7.12 * m
```

```
array([[ 7.12, 14.24, 21.36],
       [28.48, 35.6 , 42.72],
       [49.84, 56.96, 64.08]])
```

```
m * m
```

```
array([[ 1,  4,  9],
       [16, 25, 36],
       [49, 64, 81]])
```

```
m@m
```

```
array([[ 30,  36,  42],
       [ 66,  81,  96],
       [102, 126, 150]])
```

```
np.zeros((2,3))
```

```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

```
np.ones((3,2))
```

```
array([[1., 1.],
       [1., 1.],
       [1., 1.]])
```

```
r = np.random.random((3,5))*100
r
```

```
array([[28.59190643, 88.89008955, 28.96653224, 53.43865308, 82.23642151],
       [23.77256156, 20.94847528, 58.07835104, 40.52839005, 71.72227041],
       [11.74241858, 98.29272273, 29.30936364, 52.25047165, 74.84800268]])
```

```
rt = r.T
rt
```

```
array([[28.59190643, 23.77256156, 11.74241858],
       [88.89008955, 20.94847528, 98.29272273],
       [28.96653224, 58.07835104, 29.30936364],
       [53.43865308, 40.52839005, 52.25047165],
       [82.23642151, 71.72227041, 74.84800268]])
```

```
rt[1,2]
```

```
98.29272272774045
```

```
rt[1,:]
```

```
array([88.89008955, 20.94847528, 98.29272273])
```

```
rt[:,2]
```

```
array([11.74241858, 98.29272273, 29.30936364, 52.25047165, 74.84800268])
```

```
rt[2:5,1:3]
```

```
array([[58.07835104, 29.30936364],
       [40.52839005, 52.25047165],
       [71.72227041, 74.84800268]])
```

```
rt > 45
```

```
array([[False, False, False],
       [ True, False,  True],
       [False,  True, False],
       [ True, False,  True],
       [ True,  True,  True]])
```

```
rt[ rt>45 ]
```

```
array([88.89008955, 98.29272273, 58.07835104, 53.43865308, 52.25047165,
       82.23642151, 71.72227041, 74.84800268])
```

```
rt.sum()
```

```
763.6166304334438
```

```
rt.sum(axis=0)
```

```
array([282.12360281, 215.05004834, 266.44297929])
```

```
rt.sum(axis=1)
```

```
array([ 64.10688656, 208.13128755, 116.35424692, 146.21751479,
       228.80669461])
```

```
rt.std()
```

```
26.403182085068394
```

```
rt.std(axis=0)
```

```
array([25.52558848, 19.56730934, 30.95915239])
```

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

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

```
a2 = a.reshape(2,5)
a2
```

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

```
a.reshape(5,2)
```

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

```
a
```

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

```
v = np.array([7,8,9])
m = np.array([[1,2,3],[4,5,6]])
v.shape
```

```
(3,)
```

```
m.shape
```

```
(2, 3)
```

```
v+m
```

```
array([[ 8, 10, 12],
       [11, 13, 15]])
```

```
v = np.array([8,9])
m = np.array([[1,2,3],[4,5,6]]).T
m.shape
```

```
(3, 2)
```

```
m
```

```
array([[1, 4],
       [2, 5],
       [3, 6]])
```

```
v+m
```

```
array([[ 9, 13],
       [10, 14],
       [11, 15]])
```

```
v * m
```

```
array([[ 8, 36],
       [16, 45],
       [24, 54]])
```

```
v @ m
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-80-33886e1f3833> in <module>()
----> 1 v @ m

ValueError: matmul: Input operand 1 has a mismatch in its core dimension 0, with
gufunc signature (n?,k),(k,m?)->(n?,m?) (size 3 is different from 2)
```

SEARCH STACK OVERFLOW

```
m @ v
```

```
array([44, 61, 78])
```

```
# Tiempo Numpy vs Python
import numpy as np
import numpy.random as rnd
```

```
from operator import matmul
from time import time
```

```
def mult_mat(m1, m2):
  if m1.shape[1] != m2.shape[0]:
    return None
  mat = np.zeros([m1.shape[0], m2.shape[1]])
  for i in range(mat.shape[0]):
    for j in range(mat.shape[1]):
      for k in range(mat.shape[0]):
        mat[i][j] += m1[i][k] * m2[k][j]
  return mat
```

```
def time_fun(m1, m2, func):
  t_ini = time()
  func(m1, m2)
  return time() - t_ini
```

```
m1 = np.random.random((300,400))*10
m2 = np.random.random((400,200))*10
```

```
print("tiempo numpy : ", time_fun(m1, m2, np.dot))
print("tiempo matmul : ", time_fun(m1, m2, matmul))
print("tiempo python : ", time_fun(m1, m2, mult_mat))
```

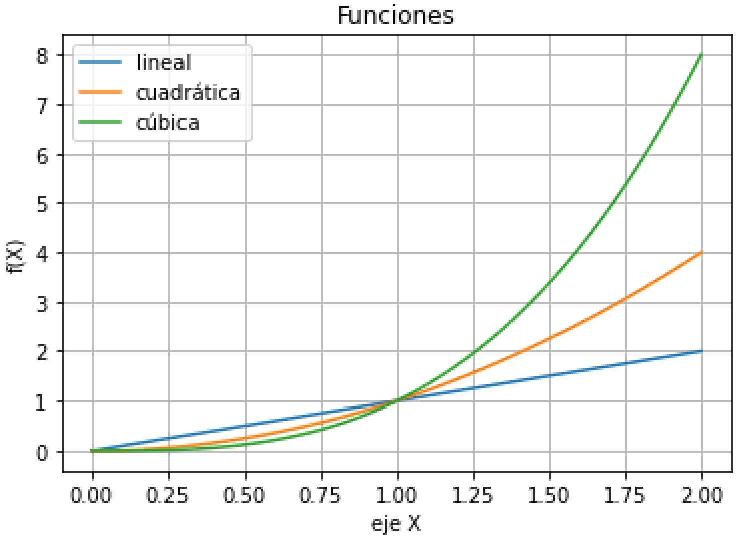
```
tiempo numpy :  0.010650634765625
tiempo matmul :  0.0018911361694335938
tiempo python :  29.771321058273315
```

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

```
x = np.linspace(-5, 5, 50)
y = np.sin(x)

plt.plot(x, y, 'y*', linestyle='dashed', markersize=10)
```

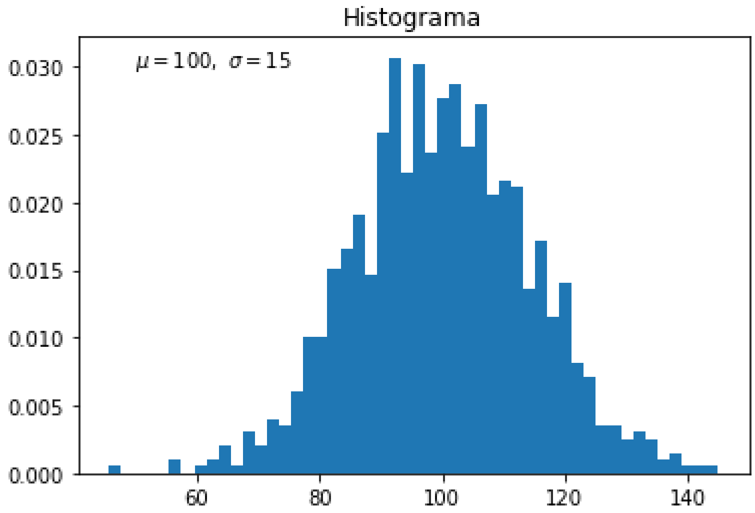
```
[<matplotlib.lines.Line2D at 0x7f499d910210>]
```

```
x = np.linspace(0, 2, 100)
plt.plot(x, x, label='lineal')
plt.plot(x, x**2, label='cuadrática')
plt.plot(x, x**3, label='cúbica')
plt.xlabel('eje X')
plt.ylabel('f(X)')
plt.title('Funciones')
plt.grid()
plt.legend()
plt.show()
```



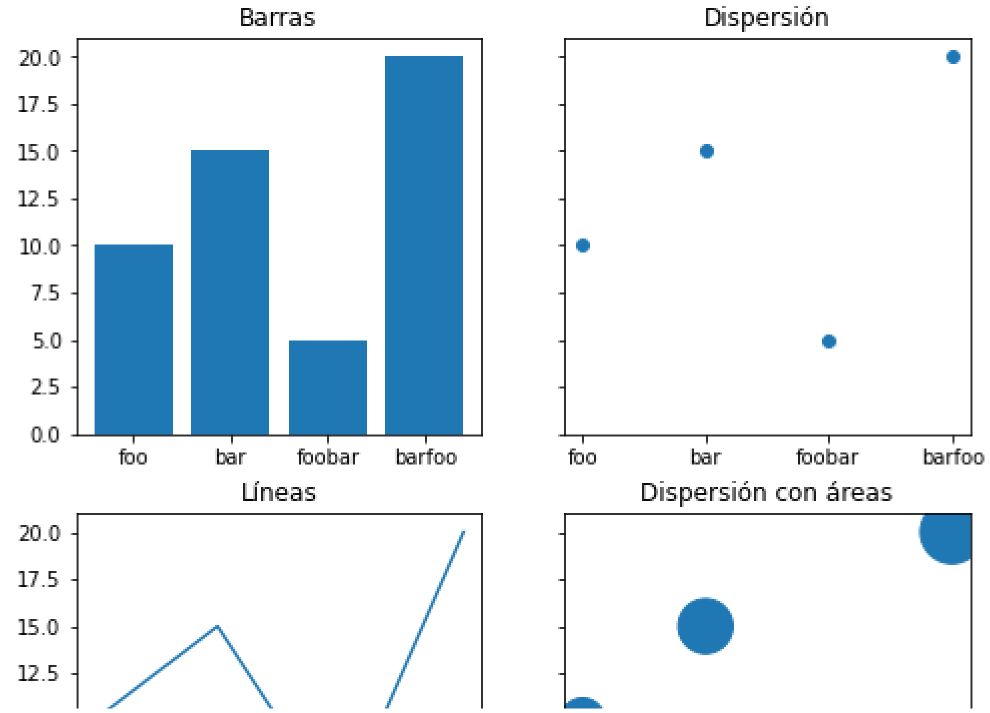
```
n = 50
x = np.random.rand(n)
y = np.random.rand(n)
colores = np.random.rand(n) * 20
areas = (30 * np.random.rand(n))**2
plt.scatter(x,y,s=areas,c=colores,alpha=0.4)
plt.show()
```

```
mu,sigma = 100,15
x = mu+sigma * np.random.randn(1000)
plt.hist(x, 50, density=True)
plt.title('Histograma')
plt.text(50, 0.030, r'$\mu=100,\ \sigma=15$')
plt.show()
```



```
data={'foo':10,'bar':15,'foobar':5,'barfoo':20}
names=list(data.keys())
values=list(data.values())
fig,((ax1,ax2),(ax3,ax4)) = plt.subplots(2,2,figsize=(8,8),sharey=True)

ax1.bar(names,values)
ax1.set_title('Barras')

ax2.scatter(names,values)
ax2.set_title('Dispersión')

ax3.plot(names, values)
ax3.set_title('Líneas')

areas = [v*50 for v in values]
ax4.scatter(names,values,s=areas)
ax4.set_title('Dispersión con áreas')
```

```
Text(0.5, 1.0, 'Dispersión con áreas')
```



```
type(fig)
```

```
matplotlib.figure.Figure
```

```
type(ax1)
```

```
matplotlib.axes._subplots.AxesSubplot
```