

1. Matplotlib

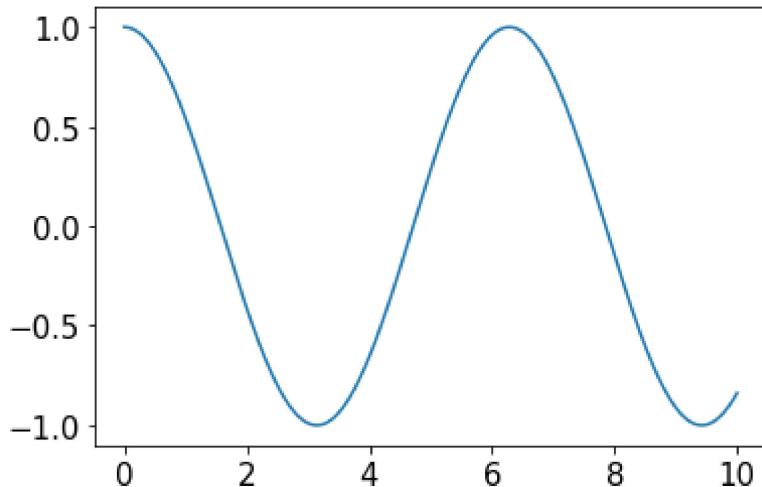
Matplotlib permettre de générer directement des graphiques à partir de Python

```
%matplotlib inline #permet d'ajouter des tracés à l'interface du navigateur
import matplotlib.pyplot as plt
import numpy as np # Le package de référence pour le calcul scientifique
```

UsageError: unrecognized arguments: #permet d'ajouter des tracés à l'interface du navigateur

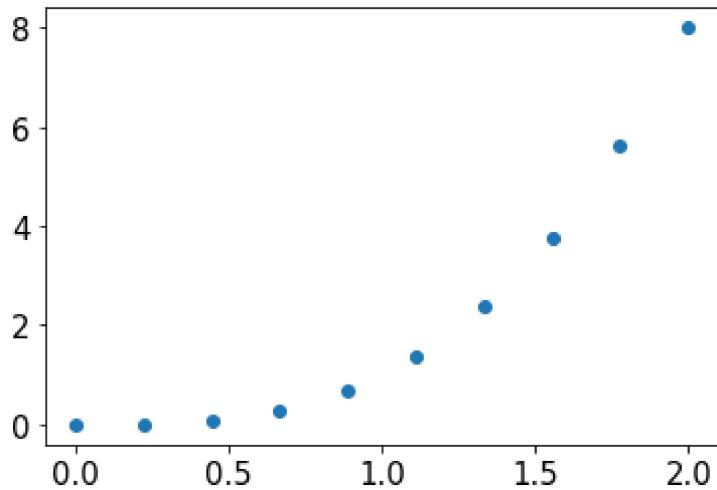
Graph Simple

```
x= np.linspace(0, 10, 1000)
y =np.cos(x)
plt.plot(x, y)
plt.show()
```



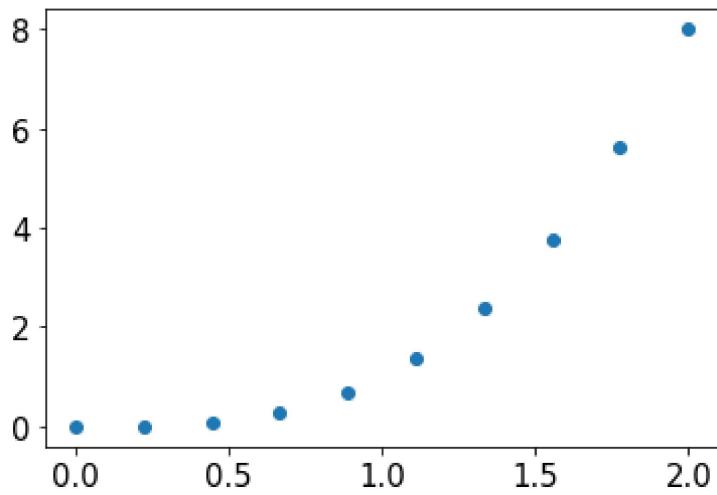
Nuage des points

```
X=np.linspace(0,2,10)
y=X**3
plt.scatter(X, y)
plt.show()
```

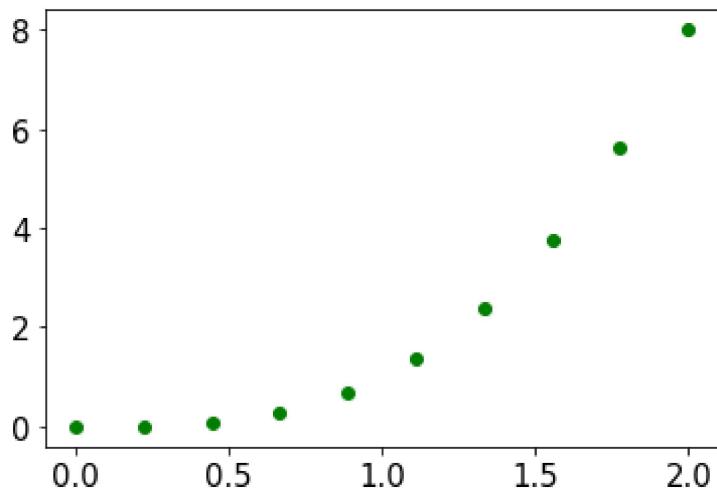


autrement

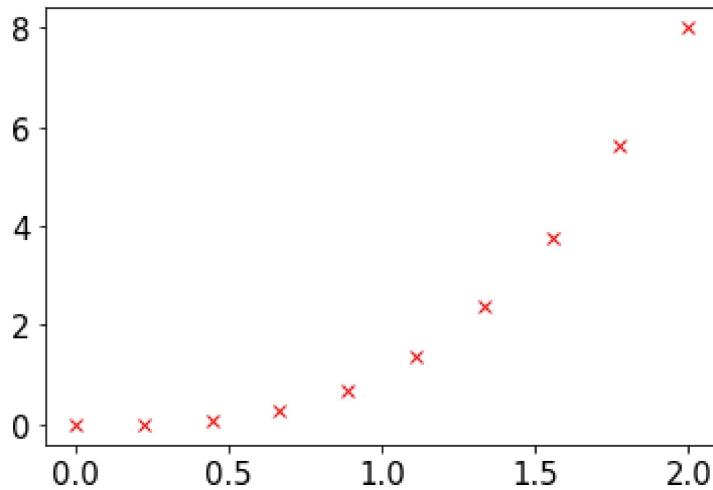
```
plt.plot(x, y, 'o')
plt.show()
```



```
plt.plot(x, y, 'go')
plt.show()
```



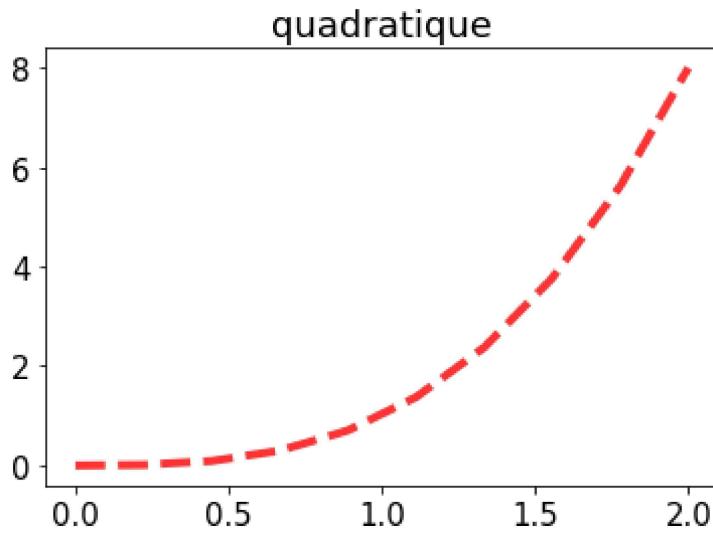
```
plt.plot(x, y, 'rx')
plt.show()
```



Style du graphe

- **c** : couleur de la ligne
- **lw** : epaisseur de la ligne (pour les graphiques plot)
- **ls** : style de la ligne (pour les graphiques plot)
- **size** : taille du point (pour les graphiques scatter)
- **marker** : style de points (pour les graphiques scatter)
- **alpha** : transparence du graphique

```
plt.plot(X, y, c='red', lw=4, ls='--', alpha=0.8)
plt.title('quadratique')
plt.show()
```



Structure d'une figure

Pour créer des figures on doit respecter la structure suivante :

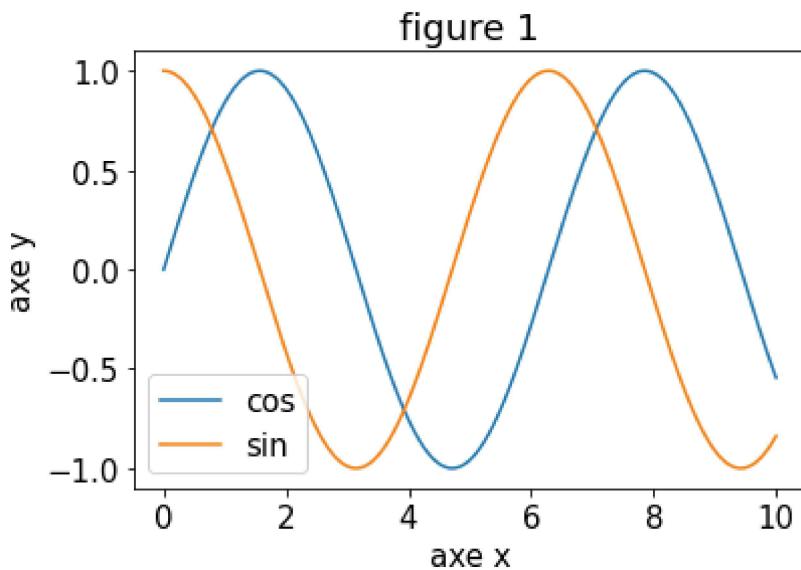
1. **plt.figure(figsize())**
2. **plt.plot()**
3. Extras (titre, axes, legendes)

4. plt.show()

```
# Changer la taille par défaut
# plt.rcParams.update({'font.size': 15})

X = np.linspace(0, 10, 1000)

plt.figure() # Création d'une figure
plt.plot(X, np.sin(X), label='cos') # première courbe
plt.plot(X, np.cos(X), label='sin') # deuxième courbe
# Extra information
plt.title('figure 1') # titre
plt.xlabel('axe x') # axes
plt.ylabel('axe y') # axes
plt.legend() # Legend
plt.savefig('figure.png') # sauvegarde la figure dans le répertoire de travail
plt.show() # affiche la figure
```



Subplot

Le subplot nous permet de créer plusieurs graphiques sur une même figure

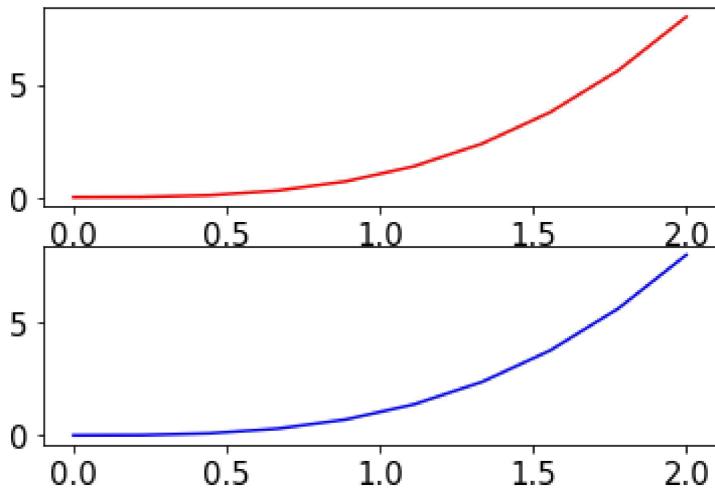
```
X=np.linspace(0,2,10)
y=X**3
```

```
y.shape
```

```
(10,)
```

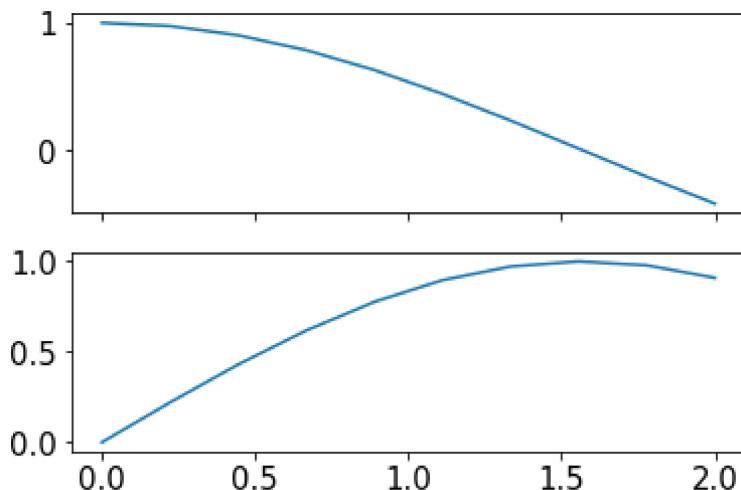
```
plt.subplot(2, 1, 1)
plt.plot(X, y, c='red')
plt.subplot(2, 1, 2)
plt.plot(X, y, c='blue')
```

```
[<matplotlib.lines.Line2D at 0x21398168460>]
```



autrement (orienté objet)

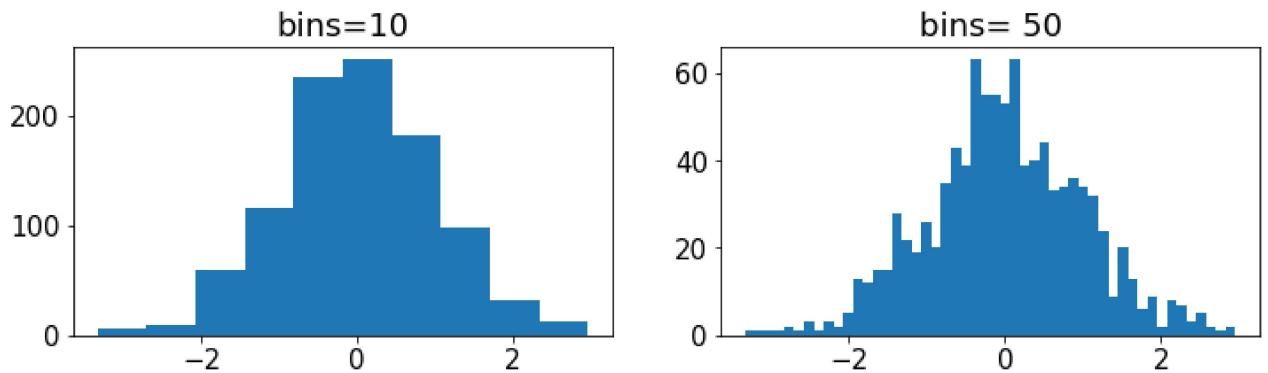
```
fig, ax = plt.subplots(2, 1, sharex=True) # Les deux figures avoir le même axe pour les
ax[0].plot(X, np.cos(X))
ax[1].plot(X, np.sin(X))
plt.show()
```



Histogramme

```
#générer des valeurs aléatoire
val1= np.random.randn(1000)
```

```
plt.figure(figsize=(12, 3))
plt.subplot(121)
plt.hist(val1, bins=10)
plt.title('bins=10')
plt.subplot(122)
plt.hist(val1, bins=50)
plt.title('bins= 50')
plt.show()
```

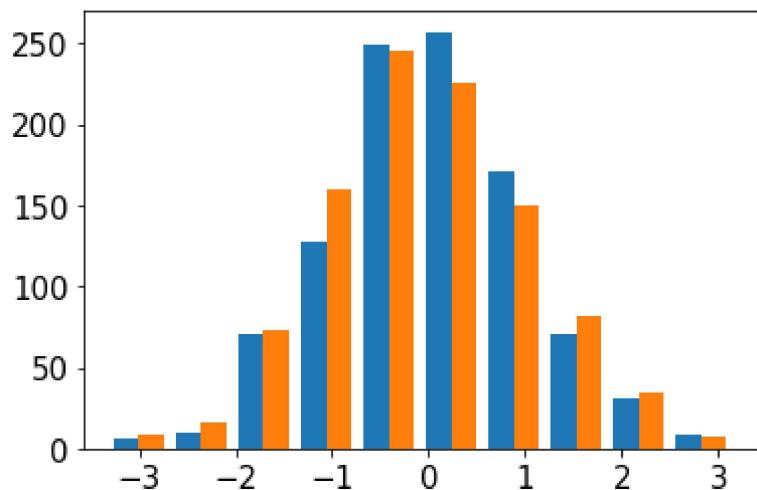


Deux histogramme dans la même liste

```
val2=np.random.randn(1000)
```

```
plt.hist([val1,val2])
```

```
(array([[ 6.,  10.,  70., 127., 249., 257., 171., 71., 31., 8.],
       [ 8., 16., 73., 160., 245., 225., 150., 82., 34., 7.]]),
 array([-3.32821177, -2.68059349, -2.0329752 , -1.38535691, -0.73773863,
        -0.09012034,  0.55749794,  1.20511623,  1.85273451,  2.5003528 ,
        3.14797109]),
 <a list of 2 BarContainer objects>)
```



Exercice et Solution

Créez une fonction "graphique" qui permet de tracer sur une seule et même figure une série de graphiques issue d'un dictionnaire contenant plusieurs datasets

```
data={"experience{i}": np.random.randn(100) for i in range(4)}
```

```
data
```

```
{'experience0': array([-0.03662608, -2.200234 , -1.39705569, -1.13552178, -0.69186953,
 1.08763582,  0.27488106, -0.93875935, -0.25069281,  0.31683472,
 1.03691204,  0.48427301, -0.71682194, -0.63529526, -1.55388339,
 -0.4553198 , -0.644444309, -0.20847143, -2.05457709, -0.11557321,
 -0.91921537, -0.59452879, -0.30319968, -0.62199186,  0.99806386,
 0.0599172 ,  0.58409682, -0.45306311,  0.88903481,  0.48803882,
 -1.4356249 , -1.28261777,  0.60386013, -0.0177244 , -0.7807409 ,
 0.97264578, -1.14800106,  0.6917389 , -0.06453777,  1.03038553,
```

```

-1.68906644, -1.19075291,  0.42220078, -0.69033367, -0.92878689,
-0.26270433, -0.28304778, -0.95483676, -0.45208784, -1.19866772,
-1.33167526, -0.21447382,  0.28314564, -0.44960715, -0.06201268,
-0.62367595, -0.59722883, -1.03066545, -0.84208596,  1.74072181,
-0.19939921,  0.61786495, -0.74977722,  0.9030501 , -0.85920585,
-1.00999797, -1.11608101, -1.6949138 ,  1.15610268,  1.71909845,
1.65058991,  1.08563767, -1.21901473, -0.25076226,  0.37391179,
-1.09974858,  0.8173774 , -0.46894265, -0.51827751,  0.25401537,
-1.37092362,  1.07496981,  0.84421338, -0.53319172,  0.06104729,
-1.29044498, -0.49067773,  1.51209251,  1.06804179,  0.4507436 ,
0.17248915,  0.12017331,  0.75522969,  0.7881124 ,  2.08021023,
-0.71487826,  0.31232889,  0.19252632,  0.91959785,  0.09272522]),

'experience1': array([-1.18860681, -1.67417963, -0.41460178,  2.00804959,  0.00654754,
0.37626294, -0.64368216,  0.38509976,  0.1839485 ,  0.12342644,
0.03825486, -3.14039302,  0.51245136,  0.22177245, -0.41054014,
-0.35933307, -1.61077171,  0.29834986, -0.75930342, -0.58378792,
-1.22264753,  0.27529093, -0.32123672,  0.8462162 , -0.53941332,
1.0716327 , -0.00502736,  1.27753006,  0.05644969, -0.19362843,
1.43399314, -1.27908843, -1.37033615, -0.06421843,  1.46235427,
0.33917251, -0.68320011,  0.38244319,  0.15215613,  0.7206847 ,
0.2169113 , -0.73358406, -1.16819985,  0.26256858, -1.12889901,
-1.54484876,  2.94875047,  0.03593294,  0.88707433, -1.94803956,
-0.11822245, -0.38339937, -0.03834397,  0.32325261, -0.85577133,
-0.79891254,  1.73501405, -3.01616562, -0.82123528,  0.67732934,
1.16010204, -0.29693857,  0.73814999,  0.47655051,  0.55926999,
-1.26200897, -0.1623605 ,  0.60509008,  1.67041963,  0.19799159,
-1.08777401,  1.22851173, -2.12445732, -0.49023157,  0.74927142,
1.45055662,  0.52515951, -1.45989732,  0.78517283, -1.81052567,
-0.32073381, -0.80666462, -0.30090563,  0.71831853, -0.36994574,
-0.34592709, -0.50860937, -0.11788623,  1.33153566,  0.12329666,
0.11163329,  0.63873791, -0.77114743, -1.87049804, -0.30571957,
-1.06953415, -0.38849994, -1.14110588,  0.50603296, -1.84669986]),

'experience2': array([-0.31243041, -0.77104794,  0.60899153, -0.86749373, -0.90514631,
1.18239153, -0.01431953, -0.72483217, -1.34979984,  0.07906504,
0.25368161,  1.05438322, -0.10916281, -1.36465065,  0.14102997,
-1.69012803, -0.10568281,  1.51650795, -2.15606707, -0.83813711,
-0.21886964, -0.19073918, -0.59935413,  0.0848077 , -0.51939315,
-0.91682975, -0.9796685 , -0.39871512,  0.5787158 , -0.97634007,
0.34709681, -1.53928459, -0.5744711 ,  0.32368227,  0.87015039,
1.59877722, -0.65052458,  0.57045043, -0.55750774, -0.29546877,
-0.77625291,  0.16824585,  0.09750127, -0.68705019,  0.09483845,
-0.52671741,  0.3716591 , -0.80030658, -0.65277919,  0.8146354 ,
-0.61619127, -0.70905937, -0.77689292, -0.28608267,  0.03830137,
0.89083581, -0.3228191 , -0.02334214,  0.39899931, -0.02850265,
-0.92575035, -0.19385871, -0.39409595, -0.87025946,  0.83290028,
-1.58656077,  0.88612285,  0.22722327, -0.20025274, -1.2174078 ,
-1.74701133,  0.7367687 , -1.19215979, -0.61228618, -0.12729401,
-1.37751615,  0.61941942,  1.48838273, -0.70038758,  0.59483491,
1.81337249, -0.67887855, -1.14631629,  1.73557181,  0.52612601,
0.20108777,  1.36059396, -0.49822946, -1.95197242, -1.12989691,
-0.95769043,  0.71998064,  0.99017718,  1.41628732,  0.46624644,
-0.71448994,  1.632132 , -1.64840483,  1.64144457, -0.74883522]),

'experience3': array([ 0.18873284, -0.82580275, -1.7229638 , -1.22647966, -0.51818604,
1.33548795, -0.36948709, -0.37825613,  1.28618236,  0.85670368,
0.51082998, -0.07325283, -1.76497399,  1.33941119, -1.15773868,
0.18509107,  0.39532373, -0.22761621, -1.95949679, -1.03288208,
0.58008491,  0.96736178, -0.24043817, -0.09303992,  1.77174048,
0.81566001, -1.27903991,  1.13368662,  1.93924851, -1.02358562,
1.51077304,  2.29025183,  1.3217044 , -1.22294105,  0.45290414,
-0.76779265, -1.04040681, -0.29707563, -0.3702872 ,  0.94156996,
-2.28864375, -1.43698783,  2.62543922, -0.6115174 ,  0.49393595,
0.7298774 , -0.80158279, -0.08823277,  1.46408995, -1.18093623,
-0.43448688,  0.74990854, -0.59590095, -0.18373022, -0.33161894,
0.76899099,  0.338142 , -2.48623539, -0.58002873,  0.94553321,
-1.30705358,  1.81997195, -0.81671946,  1.15654406,  0.24900678,

```

```

0.46759319,  0.15732303, -1.18017792, -0.97277547, -1.18506015,
-0.71867287, -1.46404671, -1.56344423,  1.45808154,  1.76563584,
-0.90289293, -0.13112122, -0.98502201,  0.34883026, -0.94657641,
-0.2862811 ,  0.17940344,  1.55704538, -0.67412747, -1.4140975 ,
-1.36682492, -1.4349017 , -0.43846937,  0.86829391, -1.86699054,
-1.71900641,  0.71724038,  0.09217256,  0.41926816,  0.39241228,
1.61582998, -0.17039591, -1.51363202, -1.8331282 , -0.24673239])
}

```

Solution

```

def graphique(df):
    n = len(data)
    plt.figure(figsize=(12, 20))

    for k, i in zip(df.keys(), range(1, n+1)):
        plt.subplot(n, 1, i)
        plt.plot(df[k])
        plt.title(k)

    plt.show()

```

```
graphique(data)
```

