

Pour réaliser des représentations graphiques sous Python, nous utiliserons la librairie `matplotlib.pyplot` avec le raccourci : **`import matplotlib.pyplot as plt`**

Nuages de points

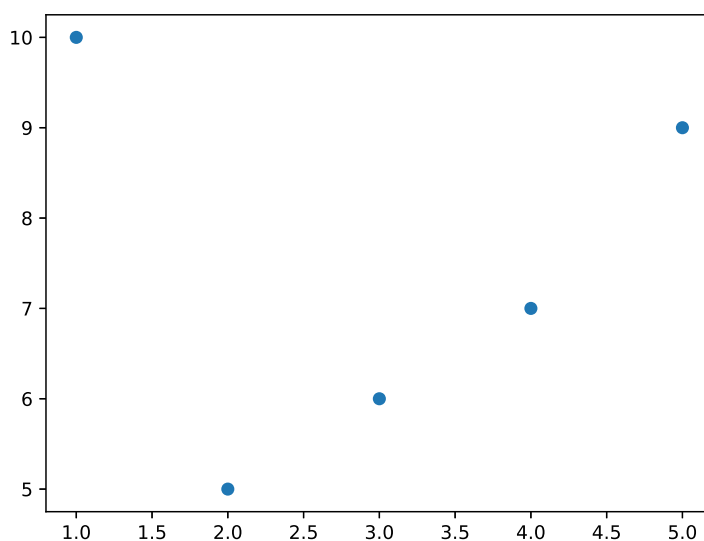
Principe de base des graphiques sous Python : Python trace en fait des nuages de points.

Par exemple : on veut tracer les points $A(1, 10)$, $B(2, 5)$, $C(3, 6)$, $F(4, 7)$, $E(5, 9)$.

On crée deux listes l'une pour les abscisses x et la seconde y pour les ordonnées, puis l'instruction **`plt.plot(x,y,'o')`** avec éventuellement des options comme les couleurs, les tailles de points...

L'affichage se fait par l'instruction **`plt.show()`**.

```
x=[1,2,3,4,5]
y=[10,5,6,7,9]
plt.plot(x,y,'o')
plt.show()
```



Si on veut comparer deux nuages de points représentant deux séries :

série y						série z					
x	1	2	3	4	5	x	1	2	3	4	5
y	10	5	6	7	9	z	9	8	5	11	10

```
import matplotlib.pyplot as plt

x=[1,2,3,4,5]
y=[10,5,6,7,9]
z=[9,8,5,11,10]

plt.axis([0,6,0,13])
plt.plot(x,y,'ro')
plt.plot(x,z,'ob')
plt.grid()
plt.legend(['y','z'])
plt.title("Nuage de points")
plt.show()
```

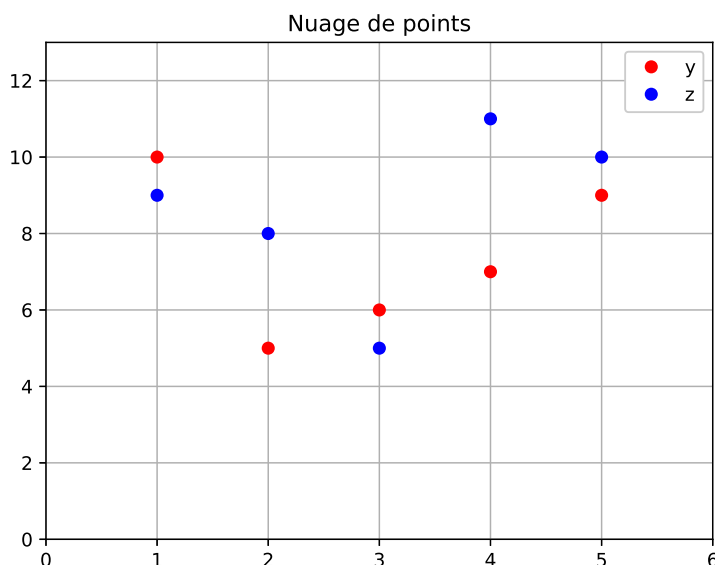


Diagramme en bâtons

On peut vouloir représenter ces séries par un diagramme en bâtons :

```
import matplotlib.pyplot as plt

x=[1,2,3,4,5]
y=[10,5,6,7,9]
z=[9,8,5,11,10]

axe2=plt.subplot()

axe2.spines['top'].set_visible(False)
# efface la
# ligne supérieure du cadre

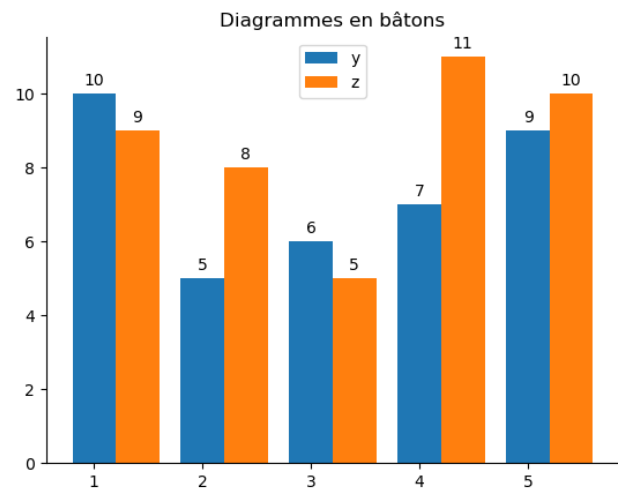
axe2.spines['right'].set_visible(False)
# efface la ligne de droite du cadre

width=0.4# épaisseur du bâton
x1=[i for i in x]
#on décale les abscisses pour la série
x2=[i+width for i in x]

# bâtons pour la série y
p1=plt.bar(x1,y,width)

# bâtons pour la série z
p2=plt.bar(x2,z,width)
plt.bar_label(p1, padding=3)
plt.bar_label(p2, padding=3)
plt.title("Diagrammes en bâtons")
plt.legend(['y','z'])

plt.show()
```



Histogrammes

```
import matplotlib.pyplot as plt
```

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

```
y=[10,5,6,7,9]
```

```
z=[9,8,5,11,10]
```

```
def SeriePonderee(list,effectif):
```

```
    n= len(list)
```

```
    T=[]
```

```
    for i in range(n):
```

```
        m=effectif[i]
```

```
        for j in range(m):
```

```
            T.append(list[i])
```

```
    return T
```

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

```
y=[10,5,6,7,9]
```

```
z=[9,8,5,11,10]
```

```
Y=SeriePonderee(x,y)
```

```
Z=SeriePonderee(x,z)
```

```
axe3=plt.subplot()
```

```
axe3.spines['top'].set_visible(False)
```

```
axe3.spines['right'].set_visible(False)
```

```
Y=SeriePonderee(x,y)
```

```
Z=SeriePonderee(x,z)
```

```
plt.hist([Y,Z], bins = [1,3,4,5], color = ['yellow','blue'],  
         edgecolor = 'red')
```

```
plt.xlabel('x')
```

```
plt.ylabel('y')
```

```
plt.title('Exemple d\' histogramme simple')
```

```
plt.legend('y')
```

```
plt.show()
```

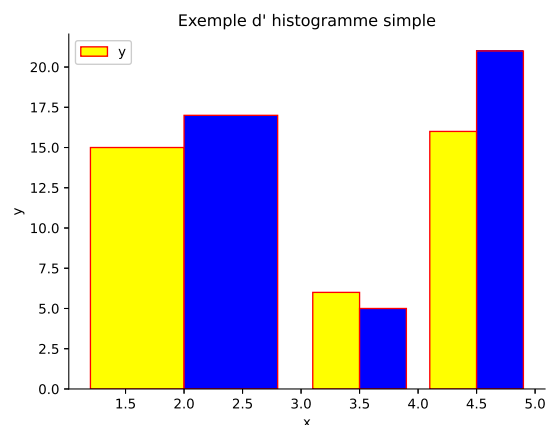


Diagramme circulaire

```
import matplotlib.pyplot as plt
```

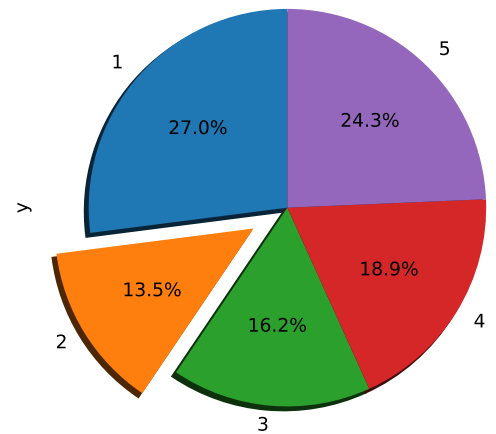
```
x=[1,2,3,4,5]  
y=[10,5,6,7,9]  
z=[9,8,5,11,10]
```

```
axe4=plt.subplot()  
explode = (0, 0.2, 0, 0, 0)  
#On isole seulement la deuxième part
```

```
axe4.spines['top'].set_visible(False)  
axe4.spines['right'].set_visible(False)
```

```
plt.pie(y, labels=['1','2','3','4','5'], explode=explode,  
autopct='%1.1f%%', shadow=True, startangle=90)  
plt.ylabel('y')  
plt.title('Exemple de diagramme circulaire ')  
  
plt.show()
```

Exemple de diagramme circulaire



Représentations de fonctions

On peut commencer par des instructions sans options :

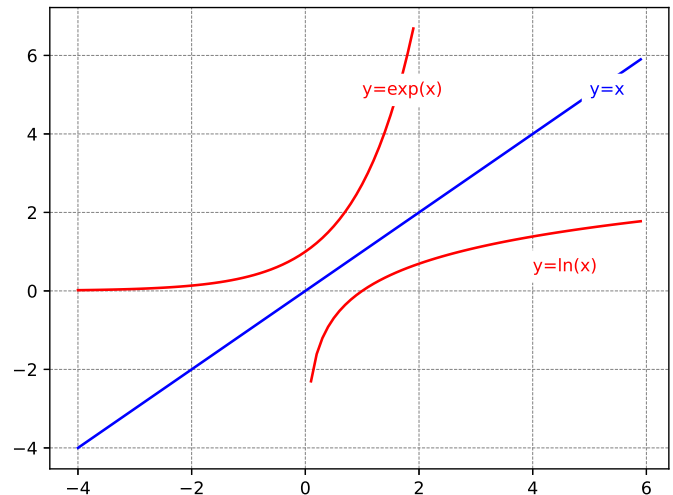
```
import matplotlib.pyplot as plt
import numpy as np
```

```
axe5=plt.subplot()
```

```
x1=np.arange(-4,2,0.1)
x2=np.arange(0,6,0.1)
x3=np.arange(-4,6,0.1)
```

```
y1=[np.exp(t) for t in x1]
y2=[np.log(t) for t in x2]
y3=[t for t in x3]
```

```
plt.plot(x1,y1,'r-')
plt.plot(x2,y2,'r-')
plt.plot(x3,y3,'b-')
axe5.grid(color='gray', linestyle='dashed',
          linewidth=.5,which='both')
```



```
plt.text(1,5,'y=exp(x)',color='red',backgroundcolor='white')
plt.text(5,5,'y=x',color='blue',backgroundcolor='white')
plt.text(4,0.5,'y=ln(x)',color='red',backgroundcolor='white')
```

```
plt.show()
```

Puis avec quelques options pour améliorer l'affichage :

```
import matplotlib.pyplot as plt

axe5=plt.subplot()

axe5.xaxis.labelpad = 0.5
axe5.yaxis.labelpad = 1

axe5.spines['top'].set_visible(False)
axe5.spines['right'].set_visible(False)
axe5.spines['left'].set_position('zero')
axe5.spines['bottom'].set_position('zero')
axe5.set_aspect('equal')

major_ticks = np.arange(-5,5,1)
minor_ticks = np.arange(-5,5, 1)

axe5.set_xticks(major_ticks)
axe5.set_xticks(minor_ticks, minor=True)
axe5.set_yticks(major_ticks)
axe5.set_yticks(minor_ticks, minor=True)

axe5.grid(which='minor', alpha=0.2)
axe5.grid(which='major', alpha=0.5)
plt.yticks(np.arange(-4,7,1))
plt.xticks(np.arange(-4,6,1))

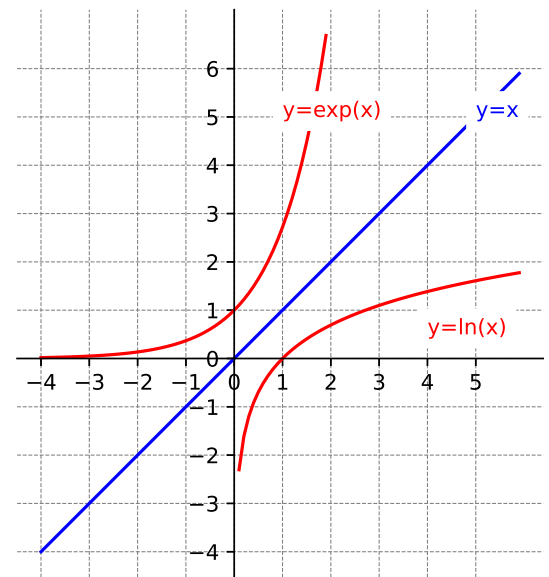
x1=np.arange(-4,2,0.1)
x2=np.arange(0,6,0.1)
x3=np.arange(-4,6,0.1)

y1=[np.exp(t) for t in x1]
y2=[np.log(t) for t in x2]
y3=[t for t in x3]

plt.plot(x1,y1,'r-')
plt.plot(x2,y2,'r-')
plt.plot(x3,y3,'b-')
axe5.grid(color='gray', linestyle='dashed', linewidth=.5,which='both')

plt.text(1,5,'y=exp(x)',color='red',backgroundcolor='white')
plt.text(5,5,'y=x',color='blue',backgroundcolor='white')
plt.text(4,0.5,'y=ln(x)',color='red',backgroundcolor='white')

plt.show()
```



Boîte à moustaches ou boîte de Tukey

```
import matplotlib
import matplotlib.pyplot as plt

import numpy as np

def SeriePonderee(list,effectif):
    n= len(list)
```

```
    T=[]
    for i in range(n):
        m=effectif[i]
        for j in range(m):
            T.append(list[i])
```

```
    return T
```

```
x=[1,2,3,4,5]
y=[10,5,6,7,9]
z=[9,8,5,11,10]
```

```
Y=SeriePonderee(x,y)
Z=SeriePonderee(x,z)
```

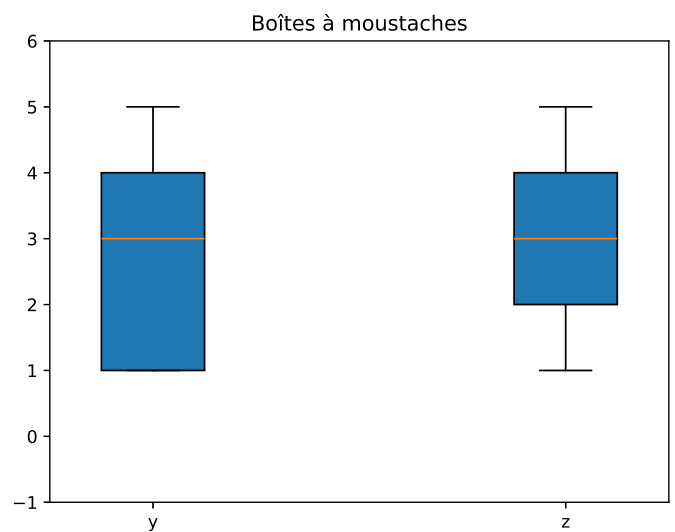
```
plt.clf()
axe6=plt.subplot()
```

```
plt.ylim(-1, 6)
# plt.boxplot(Y,Z,positions = [1, 3 ])
```

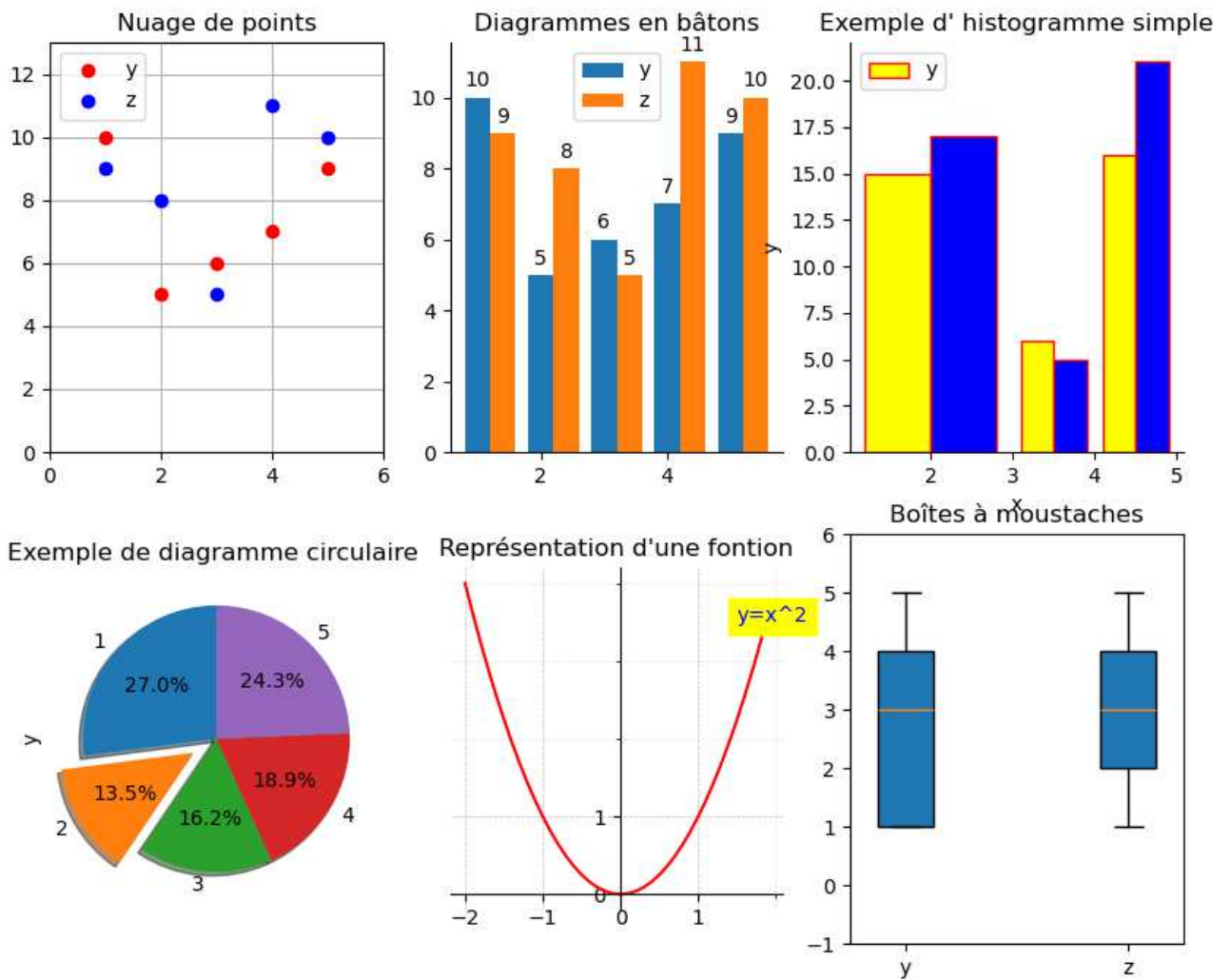
```
plt.boxplot([Y,Z], sym = 'g*', whis = 2,
             widths = [ 0.5,0.5], positions = [1, 3],
             patch_artist = True)
```

```
axe6.xaxis.set_ticklabels(['y', 'z'])
plt.title("Boîtes à moustaches")
```

```
plt.show()
```



On peut aussi représenter tous ces graphiques dans une même fenêtre :



Avec le code accessible sur le site du prof.