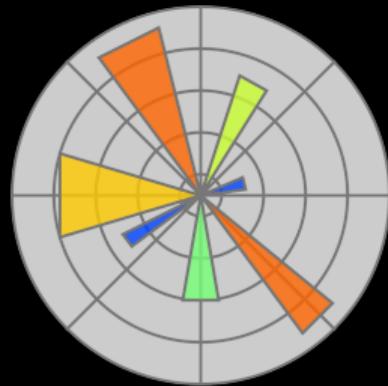


Introduction à Python

Jean-Christophe LOISEAU

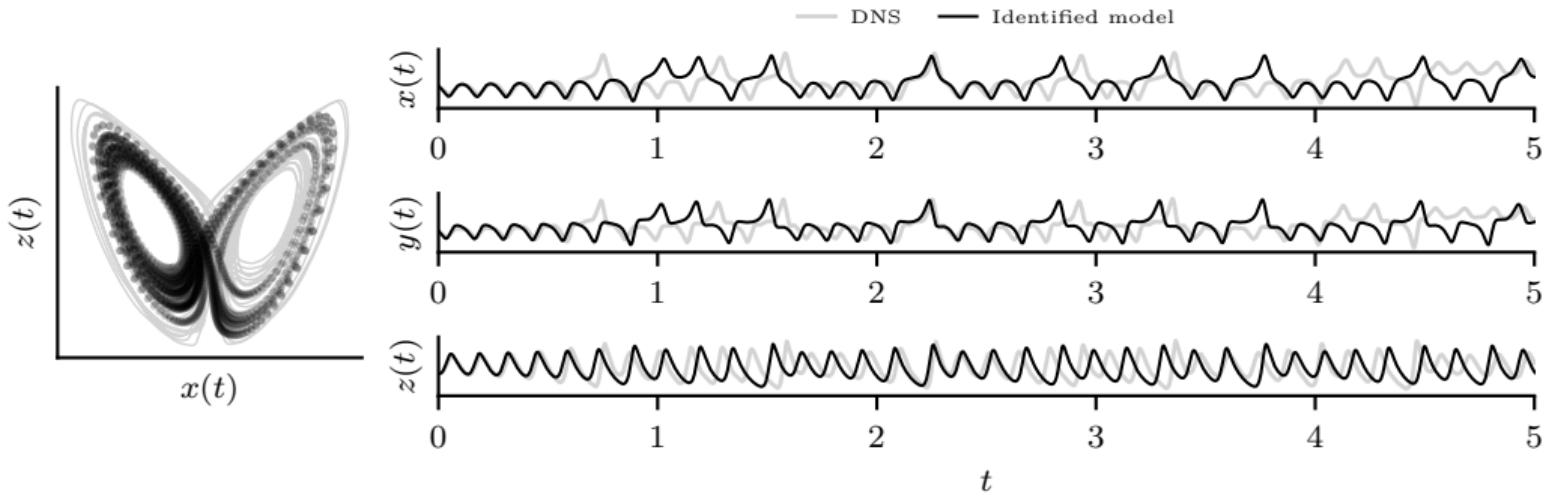
Arts & Métiers Institute of Technology, 2021-2022

Matplotlib : Package Python pour tracer et visualiser des données sous forme de graphiques ou d'animations.

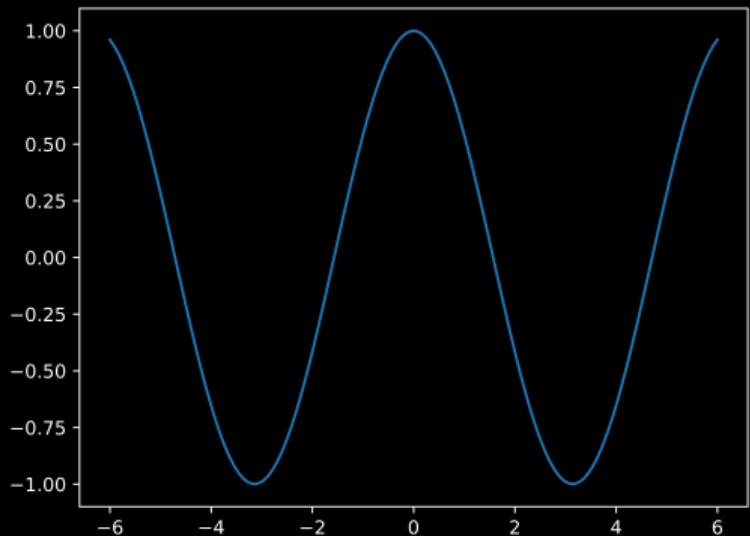



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
```

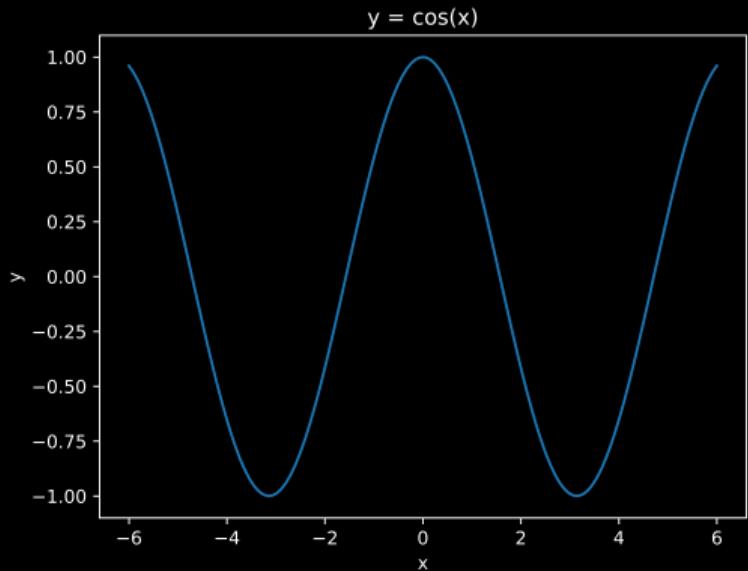
Tracer des courbes



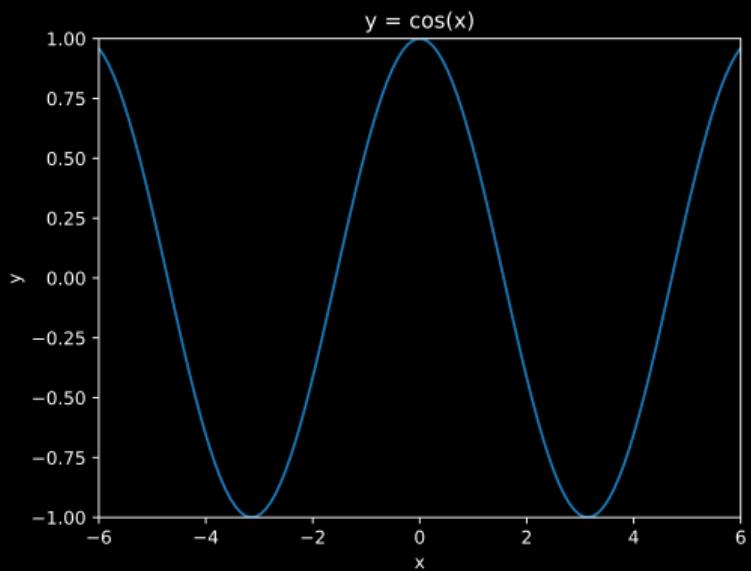
```
1 # Donnees synthetiques.  
2 x = np.linspace(-6, 6, 128)  
3 y = np.cos(x)  
4  
5 # Cree une figure.  
6 plt.figure()  
7  
8 # Plot par defaut.  
9 plt.plot(x, y)  
10  
11 # Affiche le graphe  
12 plt.show()  
13
```



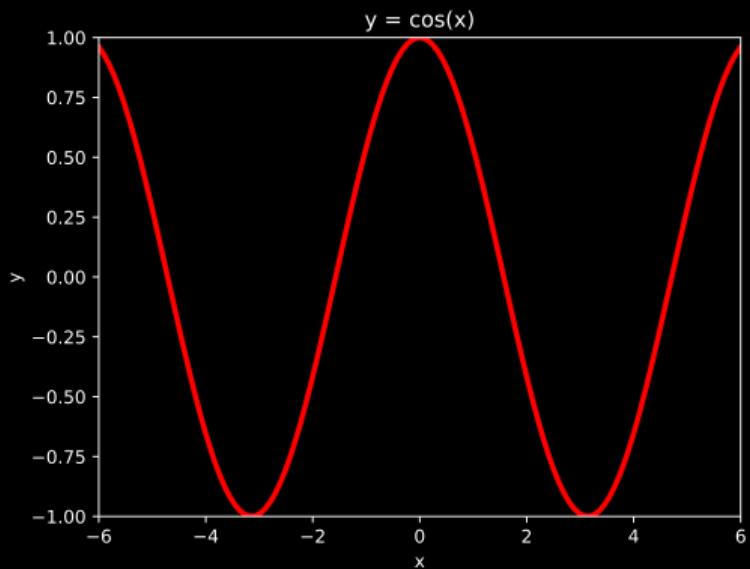
```
1  
2      # Titres des axes.  
3      plt.xlabel('x')  
4      plt.ylabel('y')  
5  
6      # Titre de la figure.  
7      plt.title('y = cos(x)')  
8
```



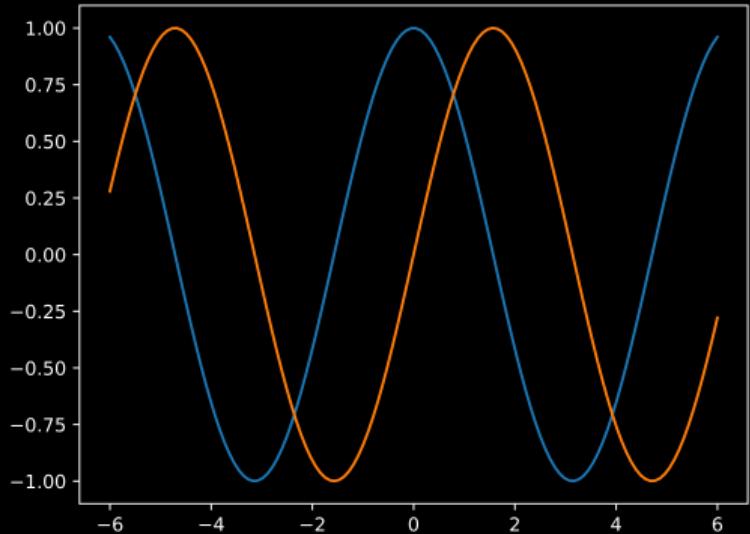
```
1  
2      # Limites des axes.  
3      plt.xlim(-6, 6)  
4      plt.ylim(-1, 1)  
5
```



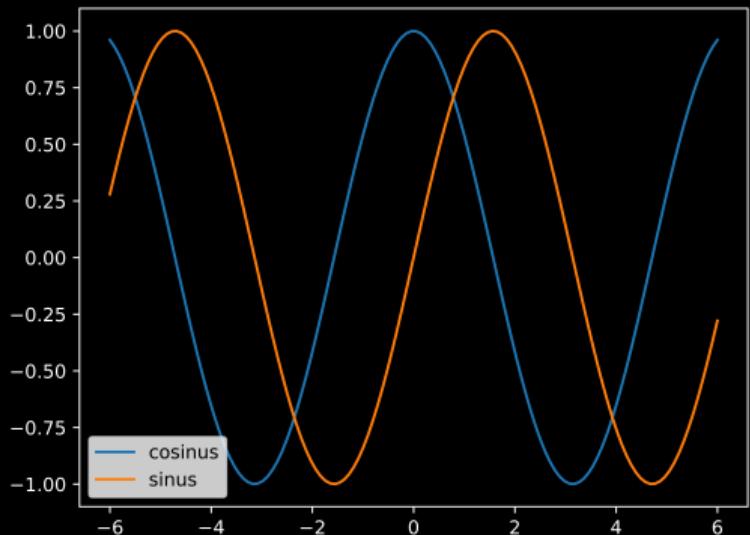
```
1  
2      # Propriétés de la courbe.  
3      plt.plot(x, y, c='r', lw=3)  
4
```



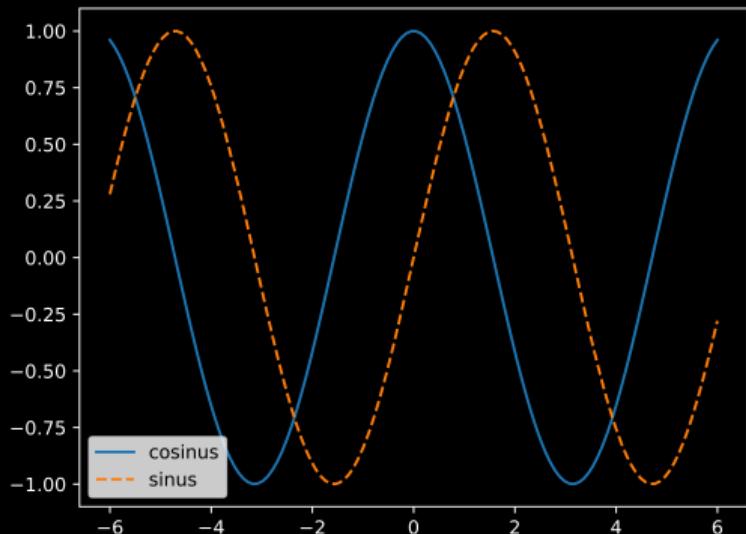

```
1 # Donnees synthetiques.
2 x = np.linspace(-6, 6, 128)
3 y = np.cos(x)
4 z = np.sin(x)
5
6 # Cree une figure.
7 plt.figure()
8
9 # Plot par defaut.
10 plt.plot(x, y)
11 plt.plot(x, z)
12
13 plt.show()
14
```



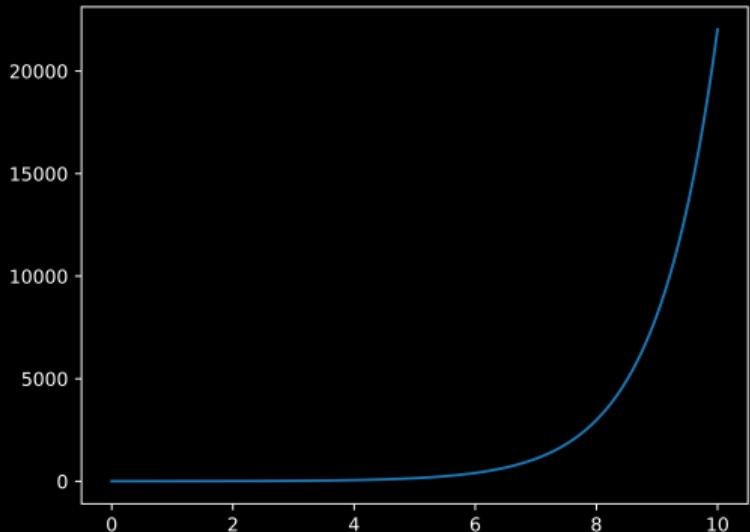
```
1      # Noms des courbes.  
2      plt.plot(  
3          x, y,  
4          label='cosinus')  
5      plt.plot(  
6          x, z,  
7          label='sinus')  
8  
9      # Ajoute la legende.  
10     plt.legend()  
11
```



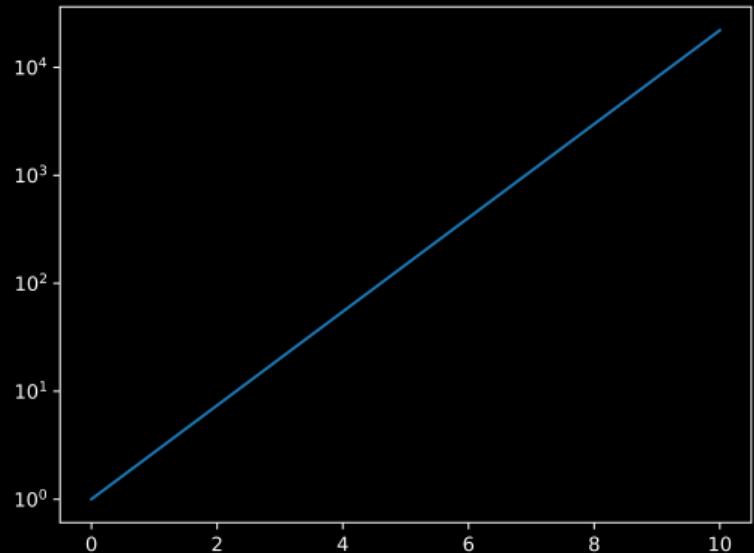
```
1      # Style d'une des courbes.  
2      plt.plot(  
3          x, y,  
4          label='cosinus')  
5      plt.plot(x, z,  
6          label='sinus',  
7          linestyle='--')  
8
```



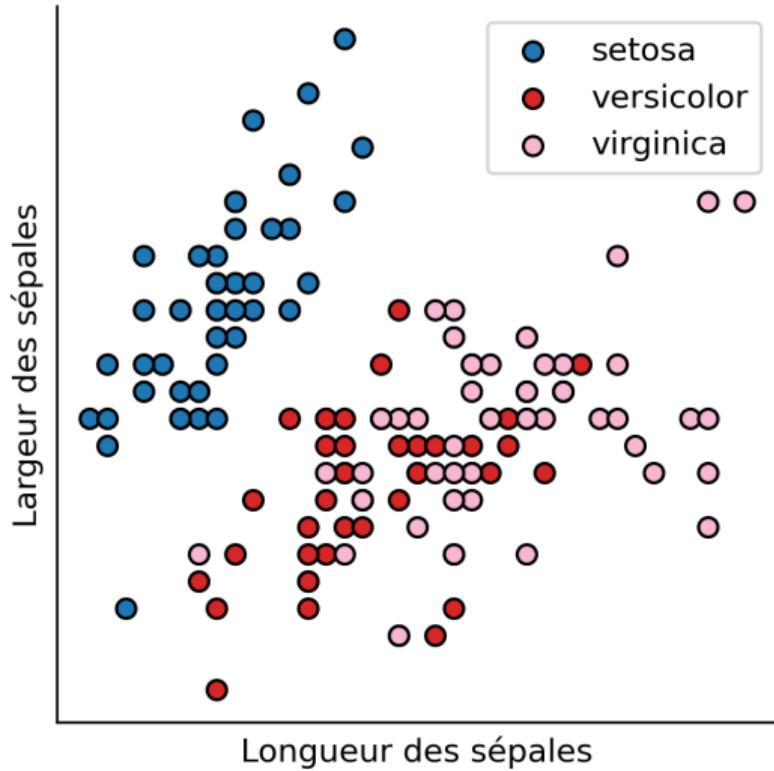

```
1 # Donnees synthetiques.  
2 x = np.linspace(0, 10, 128)  
3 y = np.exp(x)  
4  
5 # Cree la figure.  
6 plt.figure()  
7  
8 # Plot par defaut.  
9 plt.plot(x, y)  
10  
11 plt.show()
```



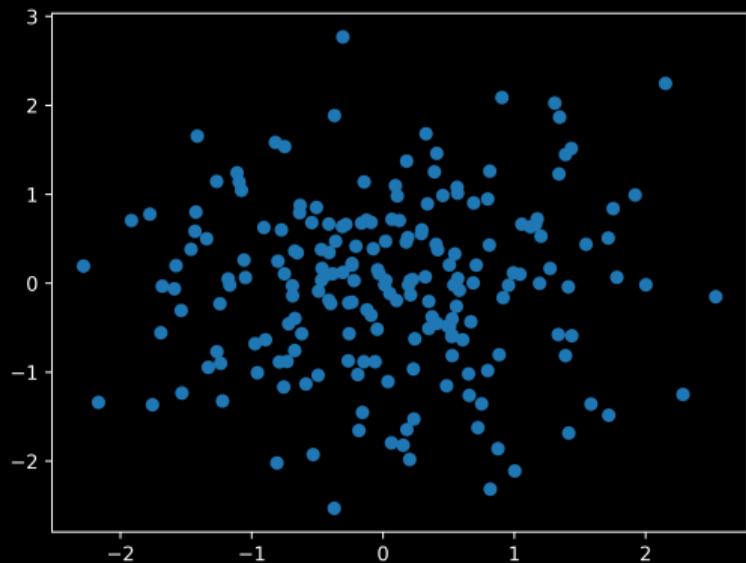
```
1  
2      # Echelle log en x.  
3      plt.semilogx(x, y)  
4  
5      # Echelle log en y.  
6      plt.semilogy(x, y)  
7  
8      # Echelle log-log.  
9      plt.loglog(x, y)  
10
```



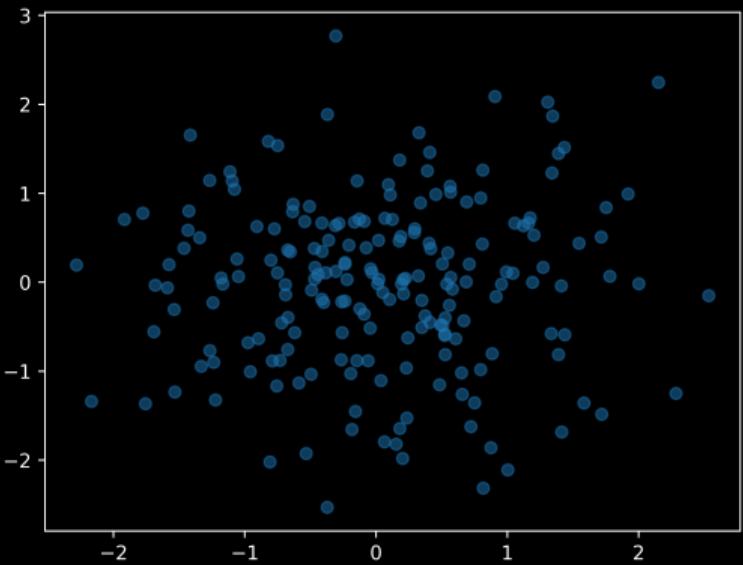
Tracer des nuages de points



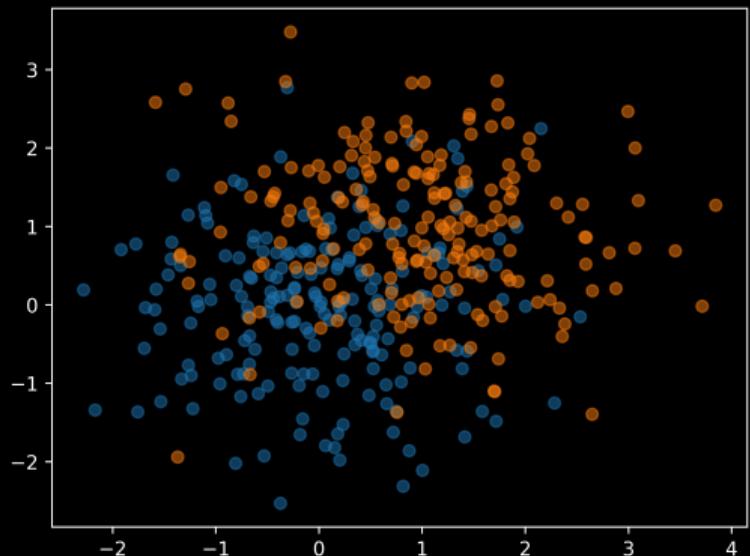
```
1      # Donnees synthetiques.  
2      x = np.random.randn(20, 2)  
3  
4      # Cree la figure.  
5      plt.figure()  
6  
7      # Nuage de points.  
8      plt.scatter(  
9          x[:, 0], x[:, 1])  
10  
11     plt.show()  
12
```



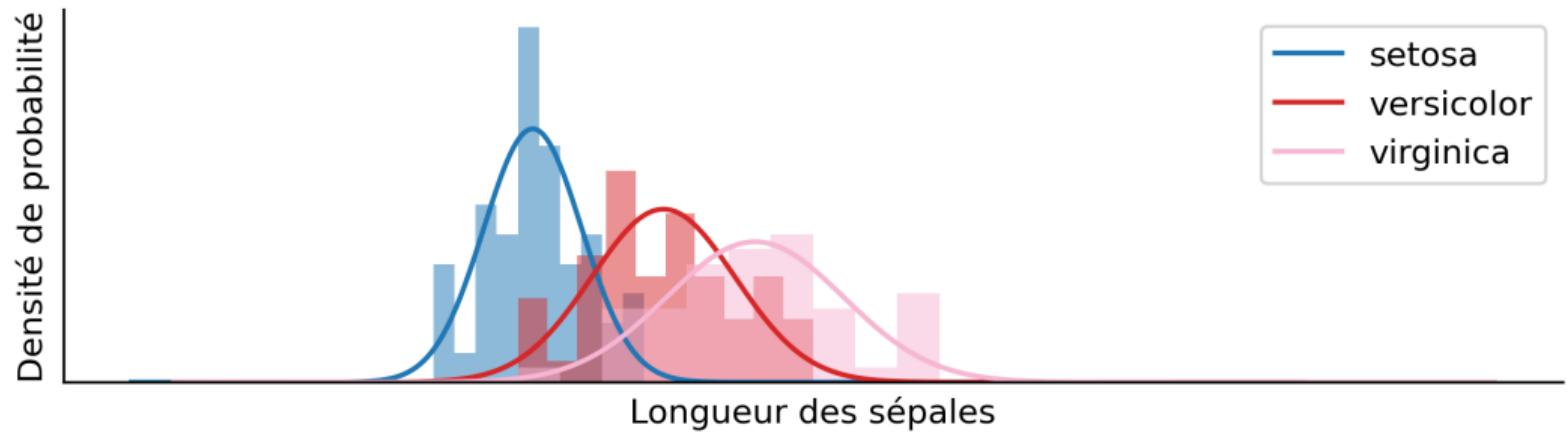
```
1 # Transparency.  
2 plt.scatter(  
3     x[:, 0], x[:, 1],  
4     alpha=0.5)  
5
```



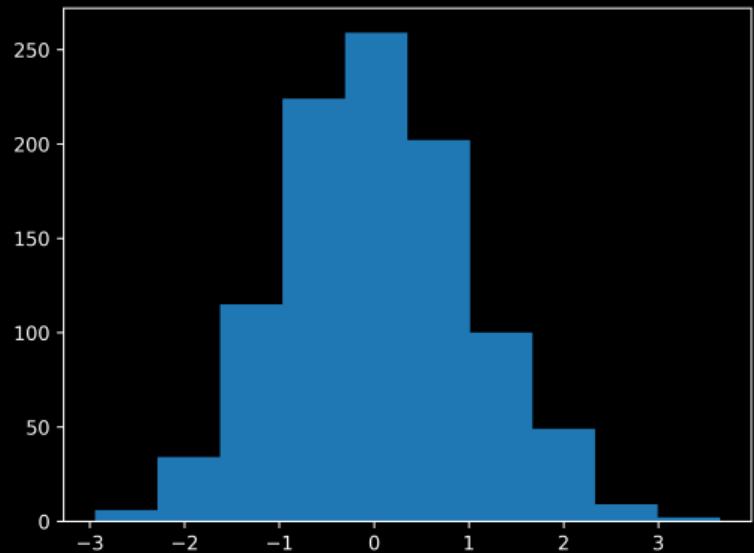
```
1  
2     x = npr.randn(200, 2)  
3     y = npr.randn(200, 2) + 1  
4  
5 # Nuage de points.  
6 plt.scatter(  
7     x[:, 0], x[:, 1],  
8     alpha=0.5)  
9  
10 plt.scatter(  
11     y[:, 0], y[:, 1],  
12     alpha=0.5)  
13
```



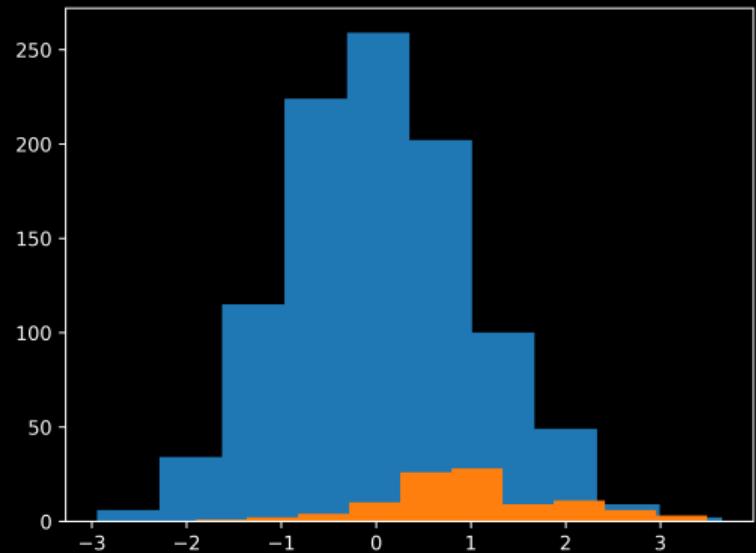
Tracer des histogrammes



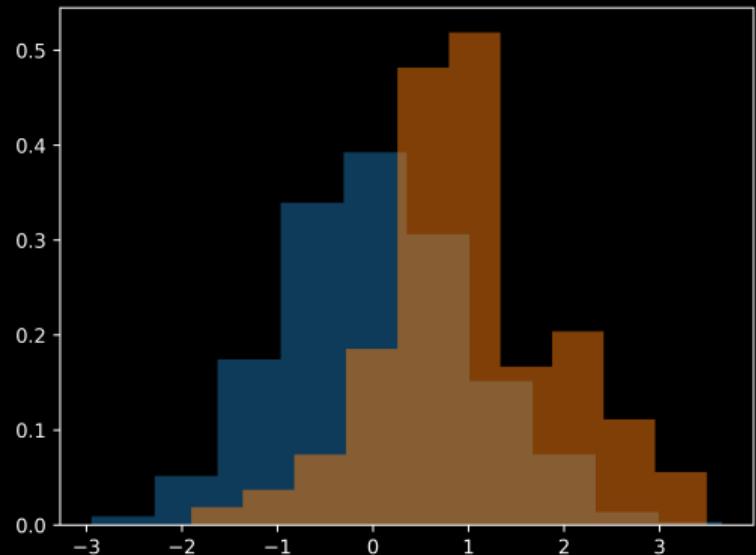
```
1      # Données synthétiques.  
2      x = npr.randn(1000)  
3  
4      # Cree la figure.  
5      plt.figure()  
6  
7      # Plot l'histogramme.  
8      plt.hist(x)  
9  
10     plt.show()
```

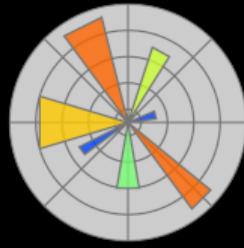


```
1 # Donnees synthetiques.  
2 x = npr.randn(1000)  
3 y = np.random(100) + 1  
4  
5 # Cree la figure.  
6 plt.figure()  
7  
8 # Plot l'histogramme.  
9 plt.hist(x)  
10 plt.hist(y)  
11  
12 plt.show()  
13
```



```
1      # Normalisation.  
2      plt.hist(  
3          x,  
4          alpha=0.5,  
5          density=True)  
6  
7      plt.hist(  
8          y,  
9          alpha=0.5,  
10         density=True)  
11
```





Pour en savoir plus, rendez-vous sur <https://matplotlib.org/>