

matplotlib

Matplotlib est une bibliothèque Python dédiée à la **visualisation de données**.

Elle permet de créer :

- des graphiques en lignes
- des histogrammes
- des nuages de points
- des diagrammes circulaires
- des graphiques scientifiques avancés
- etc.

C'est l'une des bibliothèques les plus utilisées en data science et en analyse scientifique. pyplot est un **module** de matplotlib qui fournit une interface simple pour créer des graphiques. Il permet notamment de :

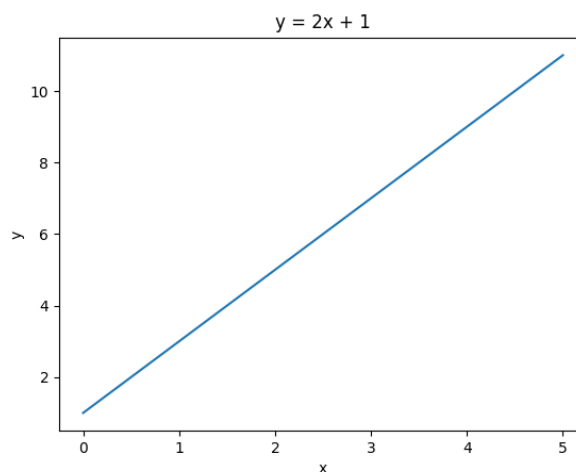
- créer une figure
- tracer des courbes
- ajouter des titres
- ajouter des labels aux axes
- afficher le graphique

Activité 1 - En mode éditeur, taper le code suivant, puis analyser son contenu :

```
import matplotlib.pyplot as plt
x = [0, 1, 2, 3, 4, 5]
y = [1, 3, 5, 7, 9, 11]

fig, ax = plt.subplots()
ax.plot(x, y)
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_title("y = 2x + 1")
plt.savefig("figure_1.png")
plt.show()
```

Figure 1



Activité 2 - En mode éditeur, compléter le code de l'activité 1, puis analyser son contenu :

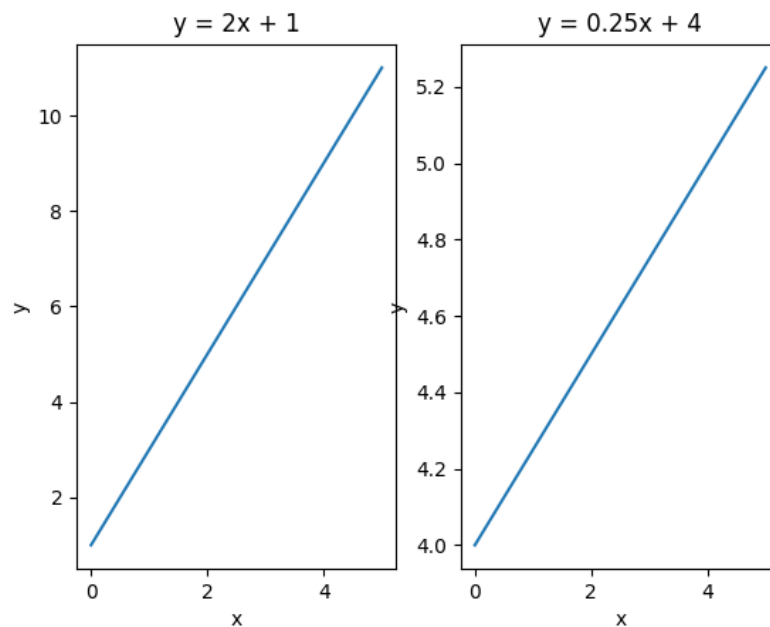
```
import matplotlib.pyplot as plt
x = [0, 1, 2, 3, 4, 5]
y1 = [1, 3, 5, 7, 9, 11]
y2 = [4, 4.25, 4.5, 4.75, 5, 5.25]

fig, ax = plt.subplots(1,2) # crée la zone de dessin : fig crée la fenêtre
                             # complète# ax crée les axes où l'on trace le graphique
ax[0].plot(x, y1) # trace la courbe
ax[0].set_xlabel("x") # donne un nom à l'axe horizontal
ax[0].set_ylabel("y") # donne un nom à l'axe vertical
ax[0].set_title("y = 2x + 1") # ajoute un titre au graphique

ax[1].plot(x, y2) # trace la courbe
ax[1].set_xlabel("x") # donne un nom à l'axe horizontal
ax[1].set_ylabel("y") # donne un nom à l'axe vertical
ax[1].set_title("y = 0.25x + 4") # ajoute un titre au graphique

plt.savefig("figure_2.png")
plt.show() # affiche la fenêtre contenant le graphique
```

Figure 2



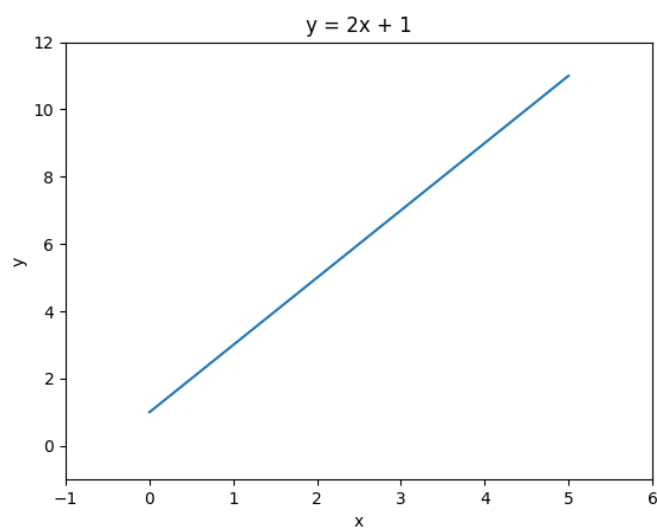
Activité 3 - En mode éditeur, modifier le code de l'activité 1 :

```
import matplotlib.pyplot as plt

x = [0, 1, 2, 3, 4, 5]
y = [1, 3, 5, 7, 9, 11]

fig, ax = plt.subplots()
ax.plot(x, y)
ax.set_xlim(-1, 6)      # limites de l'axe
ax.set_ylim(-1, 12)    # limites de l'axe vertical
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_title("y = 2x + 1")
plt.savefig("figure_3.png")
plt.show()
```

Figure 3



Activité 4 - En mode éditeur, écrire le code :

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

x = [0, 1, 2, 3, 4, 5]
y = [1, 3, 5, 7, 9, 11]

fig, ax = plt.subplots()

ax.plot(x, y)

ax.set_xlim(-1, 6)
ax.set_ylim(-1, 12)

# axes passant par l'origine
ax.spines["left"].set_position("zero")
ax.spines["bottom"].set_position("zero"
)

# supprimer les bordures inutiles
ax.spines["right"].set_color("none")
ax.spines["top"].set_color("none")

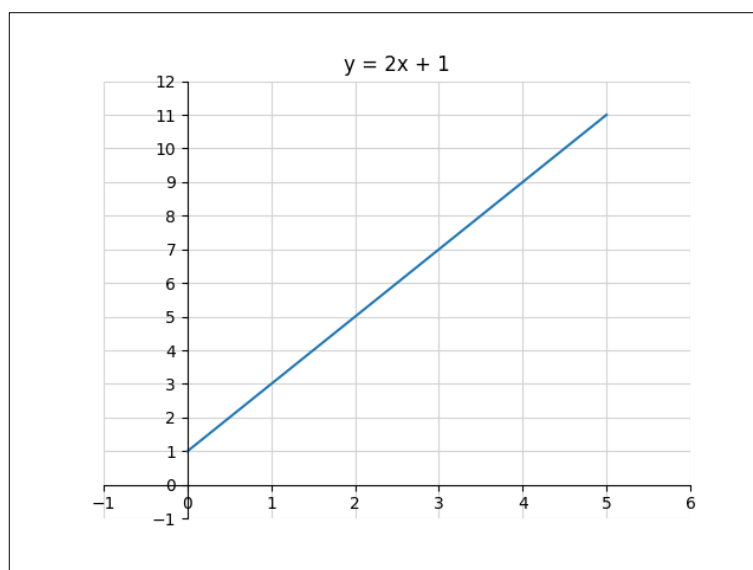
# graduations entières
ax.set_xticks(np.arange(-1, 7, 1))
ax.set_yticks(np.arange(-1, 13, 1))

# grille
ax.grid(True, color="lightgray")

ax.set_title("y = 2x + 1")

plt.savefig("figure_4.png")
plt.show()
```

Figure 4



Activité 5

Représentation de la suite numérique (u_n) définie par : $u_{n+1} = u_n + 2$ avec $u_0 = 1$.
L'affichage des termes de la suite est réalisé à l'aide de la méthode `scatter()`.

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

x = [0, 1, 2, 3, 4, 5]
y = [1, 3, 5, 7, 9, 11]

fig, ax = plt.subplots()

ax.scatter(x, y, s=10, color="red", zorder=3) # points plus petits et au premier
plan
ax.set_xlim(-1, 6)
ax.set_ylim(-1, 12)

# axes passant par l'origine
ax.spines["left"].set_position("zero")
ax.spines["bottom"].set_position("zero")

# supprimer les bordures inutiles
ax.spines["right"].set_color("none")
ax.spines["top"].set_color("none")

# graduations entières
ax.set_xticks(np.arange(0, 7, 1))
ax.set_yticks(np.arange(0, 13, 1))

# grille
ax.grid(True, color="lightgray", zorder=0)

ax.set_title(r"$u_{n+1} = u_n + 2 \quad (u_0 = 1)$")

plt.savefig("figure_5.png")
plt.show()
```

Figure 5

