



Introduction à Matplotlib

Qu'est-ce que Matplotlib ?

- Bibliothèque Python pour créer des visualisations 2D
- Permet de tracer des graphiques, courbes, histogrammes, etc.

Pourquoi utiliser Matplotlib ?

- Analyser et visualiser des données en cybersécurité, réseaux ou IoT.
- Simplifier la compréhension des résultats complexes.

Installation et Importation



- **Installation**

```
pip install matplotlib
```

ou

```
pip3 install matplotlib
```

- **Importation de base**

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

ou

Graphique simple



- Tracer une courbe représentant le trafic réseau.

```
import matplotlib.pyplot as plt

heures = [1, 2, 3, 4, 5]
trafic = [120, 200, 150, 300, 250]

plt.plot(heures, trafic)
plt.title("Trafic réseau (en paquets)")
plt.xlabel("Heures")
plt.ylabel("Nombre de paquets")
plt.grid(True)
plt.show()
```

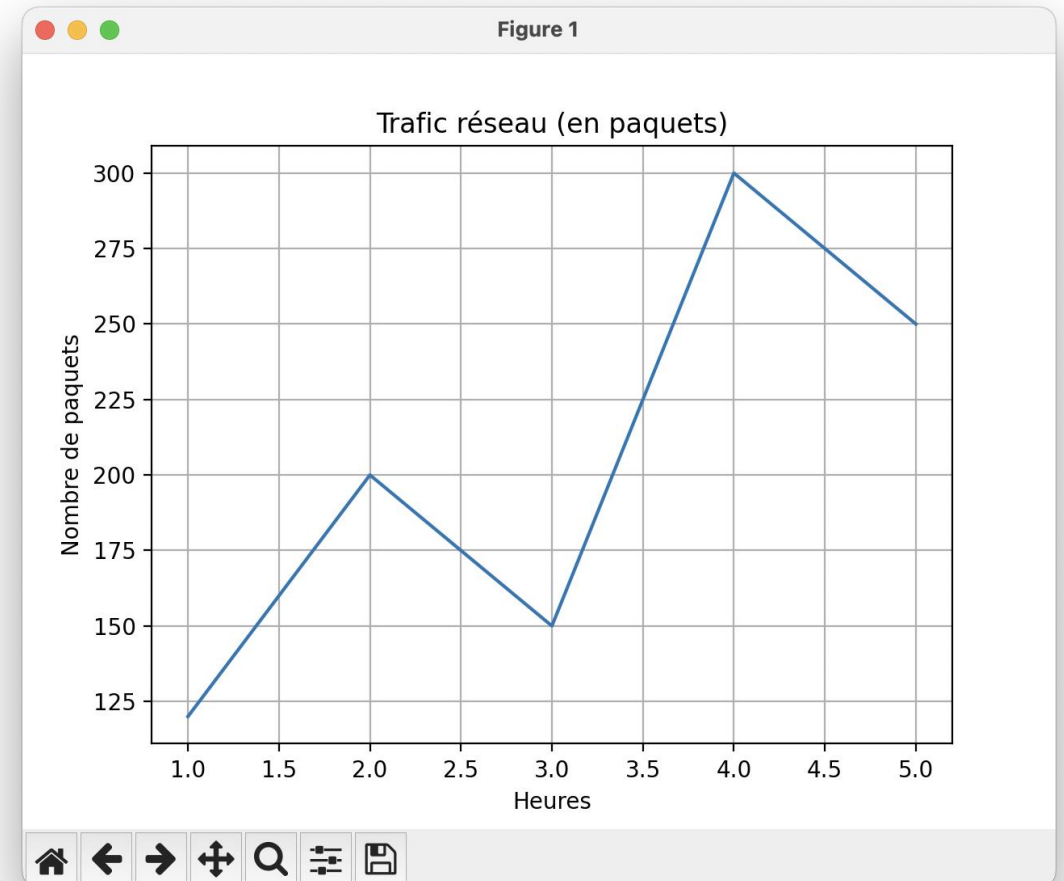


Diagramme en barre

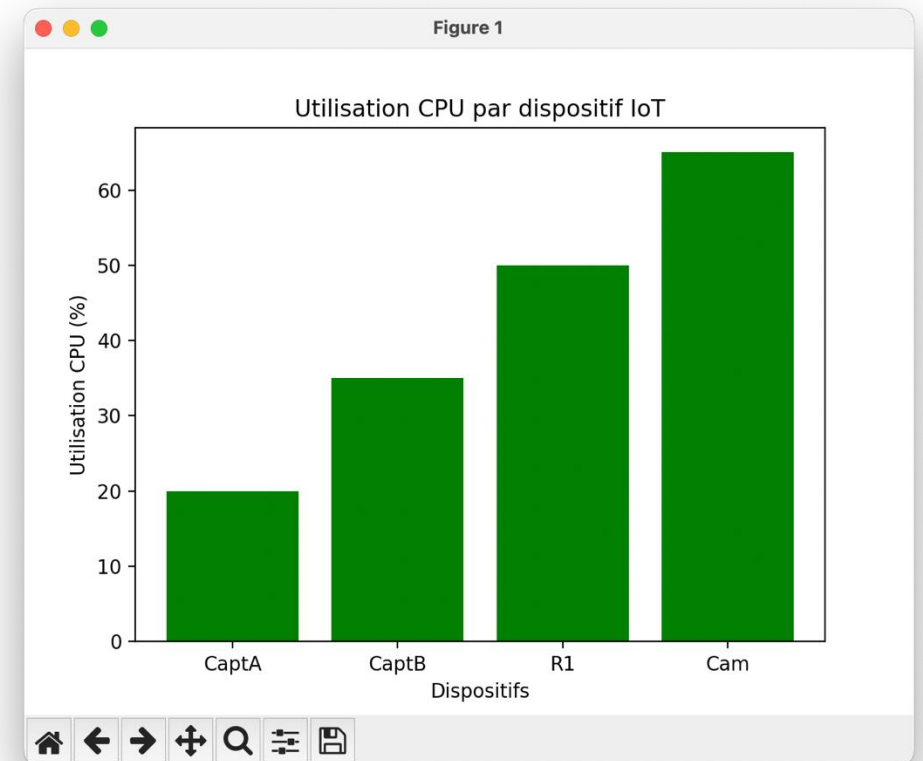


- Comparer l'utilisation CPU de différents dispositifs IoT.

```
import matplotlib.pyplot as plt

dispositifs = ['CaptA', 'CaptB', 'R1', 'Cam']
utilisation_cpu = [20, 35, 50, 65]

plt.bar(dispositifs, utilisation_cpu, color='green')
plt.title("Utilisation CPU par dispositif IoT")
plt.xlabel("Dispositifs")
plt.ylabel("Utilisation CPU (%)")
plt.show()
```



Graphique en secteurs (camembert)



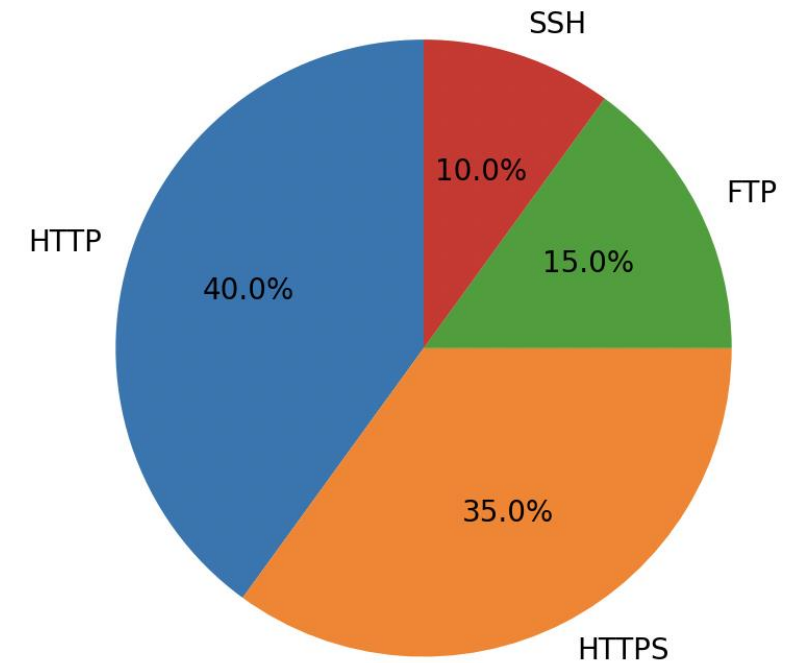
- Visualiser le pourcentage des types de protocoles détectés sur le réseau..

```
import matplotlib.pyplot as plt

protocoles = ['HTTP', 'HTTPS', 'FTP', 'SSH']
parts = [40, 35, 15, 10]

plt.pie(parts, labels=protocoles,
        autopct='%1.1f%%', startangle=90)
plt.title("Répartition des protocoles réseau")
plt.show()
```

Répartition des protocoles réseau



Histogramme



- Visualiser la répartition des tailles de paquets.

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

# Simuler des données de temps de réponse (en ms)
response_times = np.random.normal(loc=50, scale=10, size=1000)

plt.hist(response_times, bins=100, color='blue', alpha=0.7)
plt.title("Distribution des temps de réponse")
plt.xlabel("Temps (ms)")
plt.ylabel("Fréquence")

plt.show()
```

