

Finance quantitative

Formation pour débutants

Que va-t-on apprendre ?

- Qu'est ce que la finance quantitative ?
- Utiliser Python pour manipuler des données financières
- Modéliser des portefeuilles d'investissement
- Backtester des stratégies de trading
- Appliquer des concepts statistiques aux marchés financiers

Planning de la formation

- Semaine 1 : Introduction à la finance quantitative & Python
- Semaine 2 : Bases des séries temporelles & analyse de données
- Semaine 3 : Modèle de portefeuille d'investissement (théorie moderne du portefeuille)
- Semaine 4 : Introduction au Trading Algorithmique & Backtesting
- Semaine 5 : Introduction aux modèles statistiques en finance
- Semaine 6 : Optimisation de portefeuille avec Python
- Semaine 7 : Introduction au Machine Learning en finance
- Semaine 8 : Projet final - Construction de portefeuille & Backtest

Semaine 1 : Introduction à la finance quantitative & Python

Objectifs du jour :

- Comprendre ce qu'est la finance quantitative
- S'initier à Python et aux bibliothèques essentiellement utilisées en finance
- Récupérer des données financières en utilisant Python
- Analyser les prix des actions et afficher des graphiques simples

1.1 Théorie

Qu'est ce que la finance quantitative ?

- Appliquer des méthodes mathématiques / statistiques à la finance
- Quants : utilisent ces modèles pour analyser les marchés financiers, évaluer les risques, ...
- Différents types de quants

Quant Finance

A hand is shown writing various mathematical and physical equations on a chalkboard. The equations include:

- $Q = mc\Delta T$
- $\log_a(\frac{1}{x}) = -\log_a x$
- $\lim_{x \rightarrow 0} \frac{(1+x)^n - 1}{x} = n$
- $v^2 - v_0^2 = 2a(x - x_0)$
- $T = \frac{2\pi}{\omega}$
- $v = v_0 + at$
- $P = mv$
- $v = \omega r$
- $\sin \alpha = \frac{\sin \beta}{b} = \frac{\sin r}{c}$
- $a^2 + b^2 - 2ab \cos R = c^2$
- $E = mc^2$
- $F = \frac{\Delta P}{\Delta t}$
- $\sin^2 + \cos^2 = 1$
- $E_k = \frac{1}{2}mv^2$
- $y = x^2 + a$
- $v = f\lambda$
- $PV = nRT$
- $P = IV$
- $= \frac{V^2}{R}$
- $= I^2 R$

1.1 Théorie

Concepts clés

- Modélisation des prix des actifs financiers
(Ex : Black-Scholes)

- Gestion des risques (estimer et gérer les risques liés aux investissements)

A hand is writing mathematical formulas on a pink paper. The formulas are:

$$V_c = P_0 N_{d_2} - P_0 e^{-k_{RF} t} N_{d_1}$$
$$d_1 = \frac{\ln\left(\frac{P_0}{X}\right) + (k_{RF} + .5\sigma^2)t}{\sigma\sqrt{t}}$$
$$d_2 = d_1 - \sigma\sqrt{t}$$

The handwritten notes include:

- P_0 : Price of the stock
- X : Exercise price
- k_{RF} : Risk-free rate
- σ : Standard deviation
- t : time in years
- N : cumulative normal distribution function
- d_1 and d_2 : standard normal variables

- Optimisation de portefeuille (avoir la meilleure allocution d'actifs → maximiser le rendement et minimiser le risque)

Outil utilisé : Python

1.2 Théorie

Python et ses bibliothèques

- **Pandas** : manipuler et analyser des données financières
- **NumPy** : effectuer des calculs numériques et des manipulations de matrice
- **Matplotlib/Seaborn** : visualisation des données sous forme de graphiques
- **yfinance** : récupérer les données financières (prix des actions, indices, etc.) directement depuis Yahoo Finance

1.3 Pratique

Installation des bibliothèques

```
(base) samaberkane@MacBook-Air-de-Sam ~ % pip install pandas numpy matplotlib seaborn yfinance
```

- Premier projet : Récupérer les données financières avec **yfinance** —> prix d'action d'Apple (AAPL) sur les 6 derniers mois et afficher un graphique des prix de clôture

1.4 Pratique

Importation des données & visualisation

- Code :

```
* semainel.py
1 #Importation des bibliothèques nécessaires
2 import yfinance as yf #Pour récupérer les données financières
3 import pandas as pd #Pour manipuler les données
4 import matplotlib.pyplot as plt #Pour tracer des graphes
5
6 data = yf.download('AAPL', start='2024-07-01', end='2024-12-31') #Stockage des données
dans la variable data
7
8 print(data.head()) #Affichage des 5 premières lignes histoire de vérifier
9
```

- Console :

```
>>> (executing file "semainel.py")
[*****100%*****] 1 of 1 completed
   Price          Close      High       Low      Open     Volume
   Ticker          AAPL        AAPL       AAPL       AAPL      AAPL
   Date
2024-07-01  216.261475  217.019756  211.442359  211.611974  60402900
2024-07-02  219.773544  219.883297  214.615199  215.662820  58046200
2024-07-03  221.050659  221.050659  218.536335  219.504150  37369800
2024-07-05  225.829849  225.939602  221.150418  221.150418  60412400
2024-07-08  227.306534  227.336465  222.746827  226.578168  59085900
```

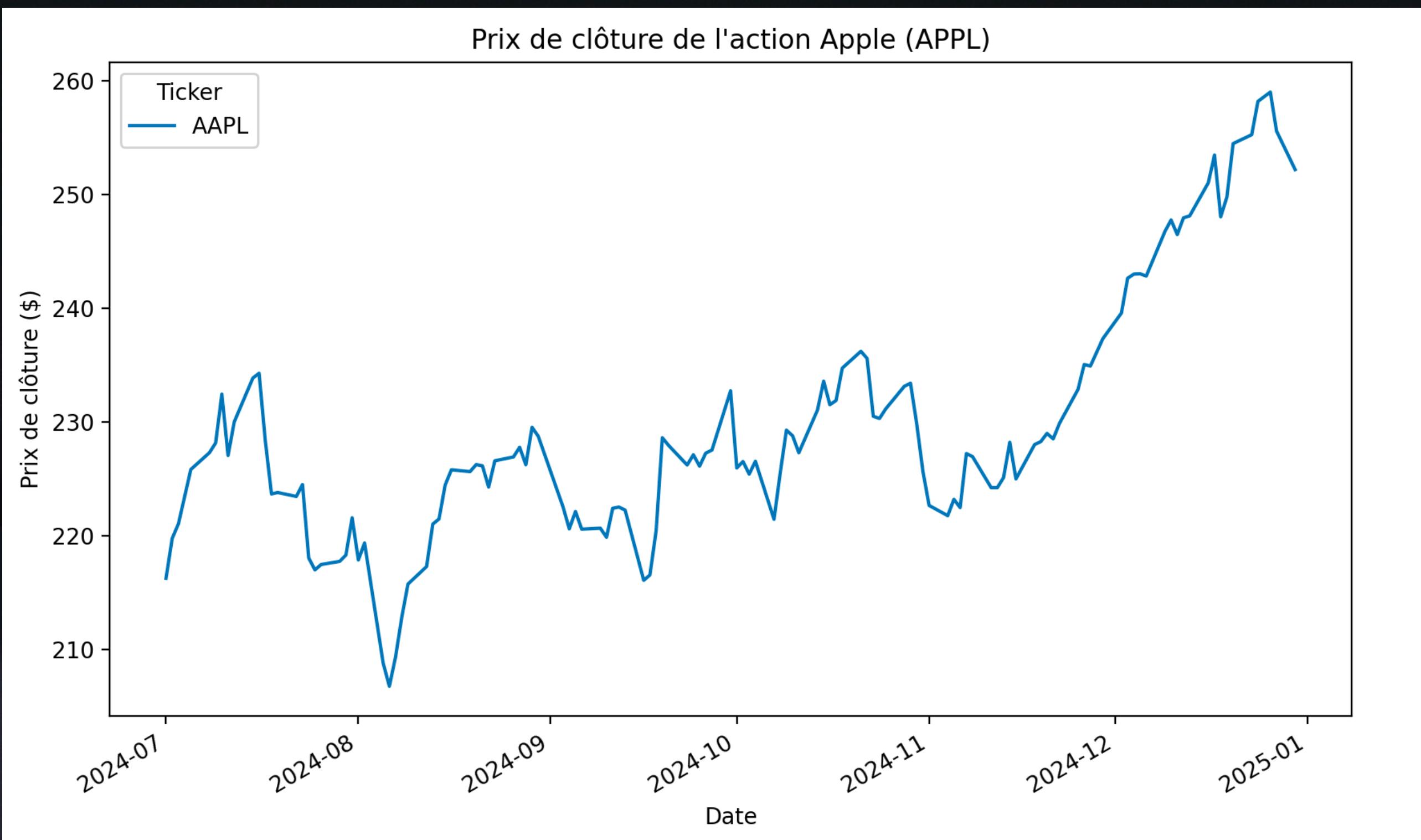
1.4 Pratique

Tracer du graphe des prix de clôture

- Code :

```
10 #Tracer du prix de clôture de l'action d'Apple
11 data['Close'].plot(title="Prix de clôture de l'action Apple (APPL)", figsize=(10,6))
12 plt.xlabel('Date')
13 plt.ylabel('Prix de clôture ($)')
14 plt.show()
```

- Console :



1.4 Pratique

Commandes utiles

- Importer des données financières : `data = yf.download(...)`
- Les visualiser : `print(data.head())`
- Tracer un graphe :
 - `plt.figure(figsize=..., ...)`
 - `plt.plot(axe des X, axe des Y, titre, couleur)`
 - `plt.show()` (possibilité de donner un titre à l'axe X, Y, ajouter une grille au graphique etc...)

1.5 Exercices

À vous de jouer !

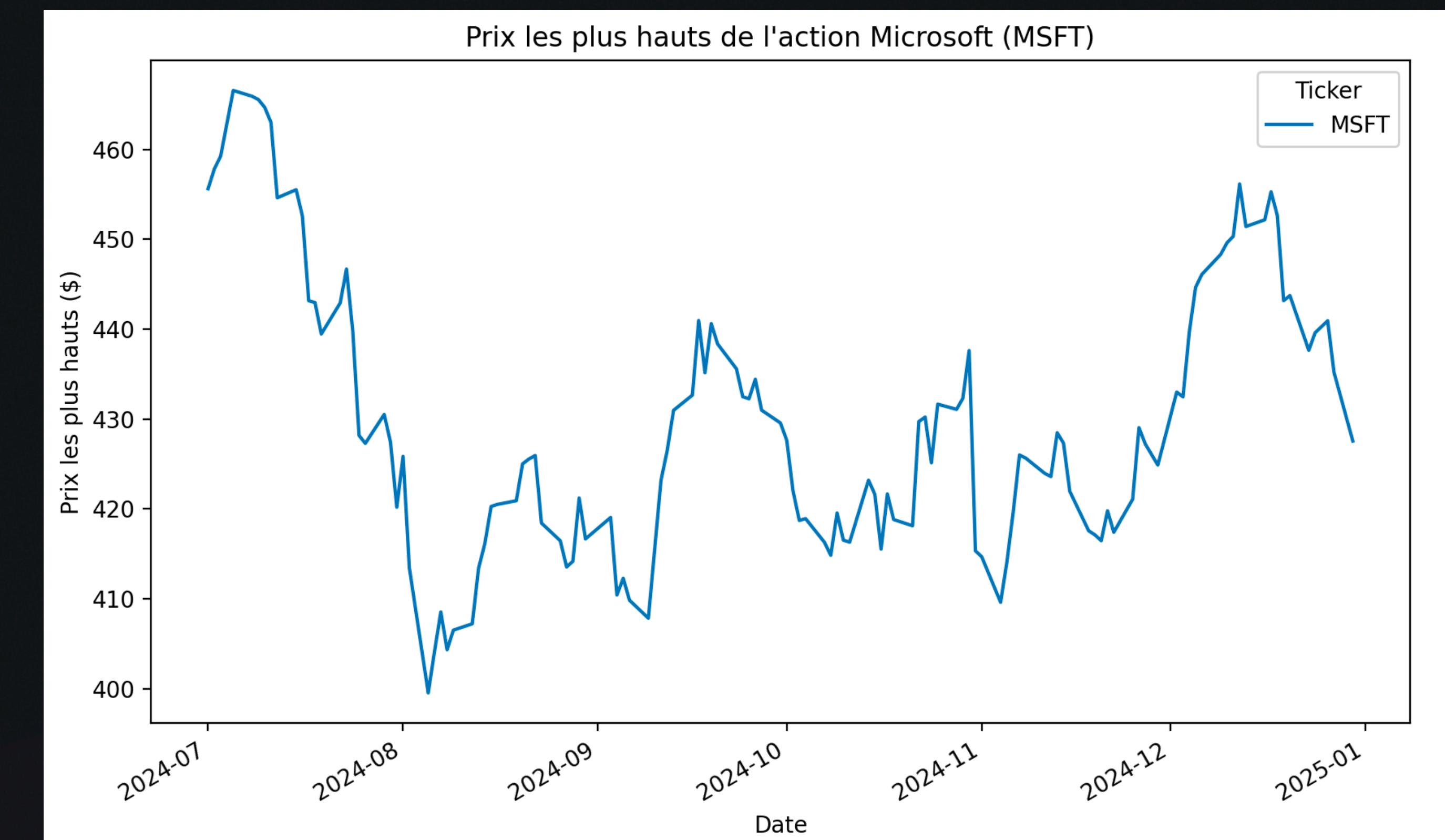
- Exo 1: Récupérer les données d'une autre action (par exemple Microsoft (MSFT), Tesla (TSLA), ...) et afficher un graphique des prix les plus hauts
- Exo 2: Ajouter un graphique avec les prix d'ouverture et de clôture (sur un même graphique, différentes couleurs)
- Exo 3: Calculer les rendements → créer une nouvelle colonne dans le DataFrame qui calcule le rendement quotidien de l'action et afficher les rendements sous forme de graphique. Utiliser `pct_change()` pour le calcul du rendement quotidien

Remarque :

$$\text{Rendement} = \frac{\text{Prix de clôture d'aujourd'hui} - \text{Prix de clôture d'hier}}{\text{Prix de clôture d'hier}}$$

1.5 Exercices

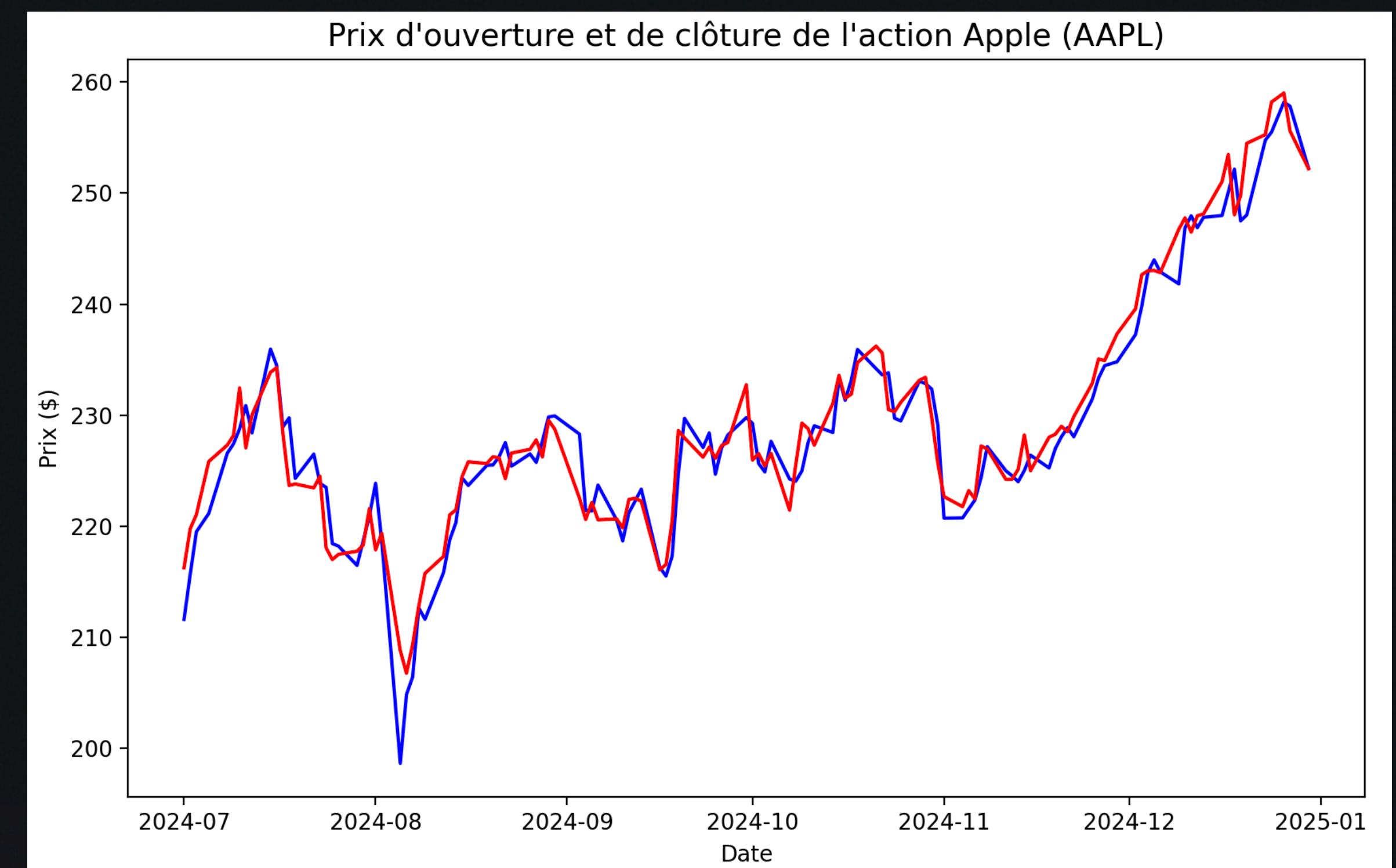
Exo 1 :



```
20 #Exo 1 :  
21  
22 data_microsoft = yf.download('MSFT', start='2024-07-01', end='2024-12-31')  
23 print(data_microsoft.head())  
24  
25 data_microsoft['High'].plot(title="Prix les plus hauts de l'action Microsoft (MSFT)",  
26 figsize=(10,6))  
27 plt.xlabel('Date')  
28 plt.ylabel('Prix les plus hauts ($)')  
29 plt.show()
```

1.5 Exercices

Exo 2 :

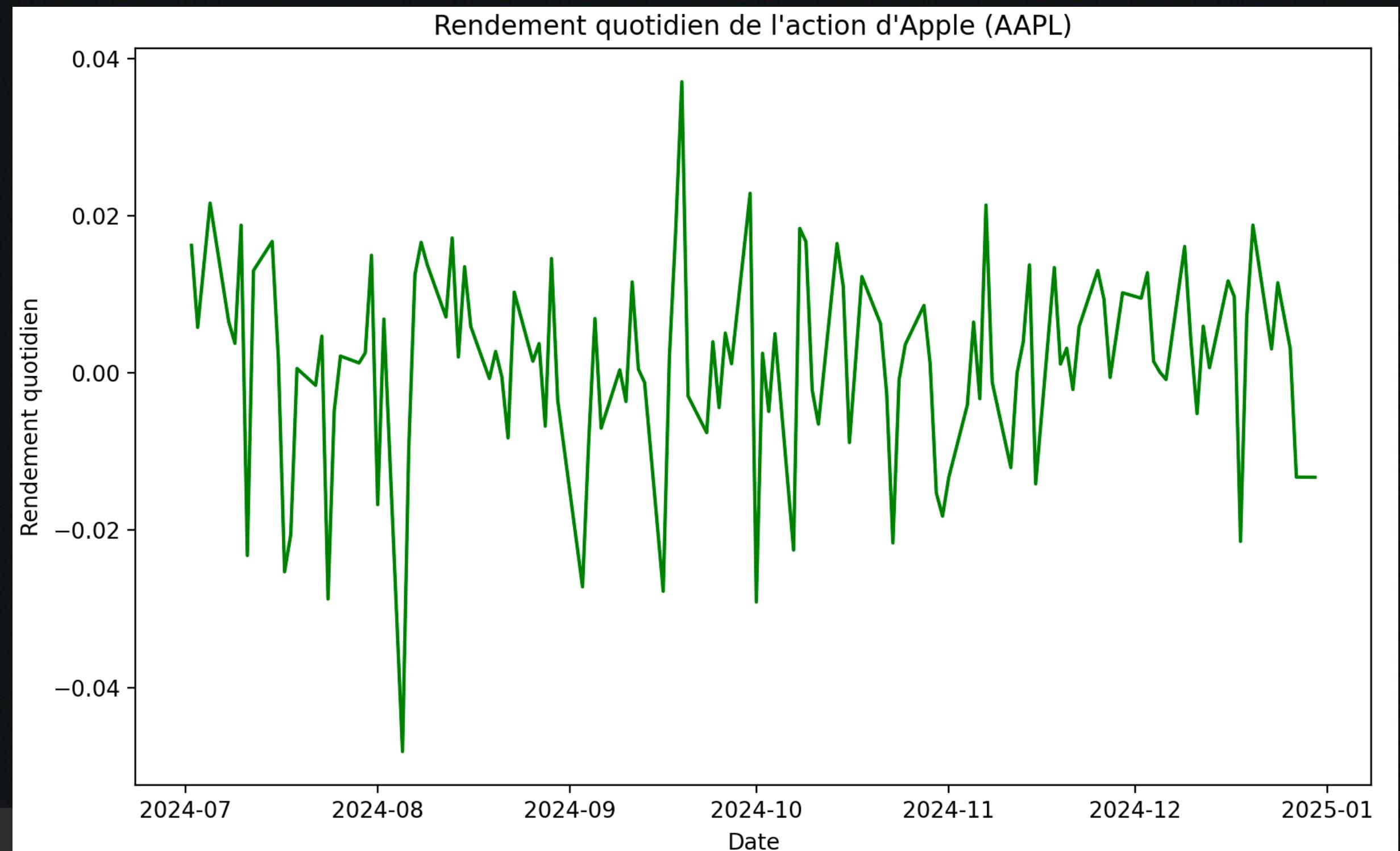


```
32 #Exo 2 :
33
34 data = yf.download('AAPL', start='2024-07-01', end='2024-12-31')
35
36 print(data[['Open', 'Close']].head()) #Afficher à la fois prix d'ouverture et de clôture
37
38 plt.figure(figsize=(10, 6))
39
40 plt.plot(data.index, data['Open'], label="Prix d'ouverture", color="blue")
41 plt.plot(data.index, data['Close'], label="Prix de clôture", color="red")
42
43 plt.title("Prix d'ouverture et de clôture de l'action Apple (AAPL)", fontsize=14)
44 plt.xlabel("Date")
45 plt.ylabel("Prix ($)")
46 #plt.grid(True) #Ajoute une grille au graphile (améliore le visuel)
47 plt.show()
```

1.5 Exercices

Exo 3 :

```
51 #Exo 3 :  
52  
53 data = yf.download('AAPL', start='2024-07-01', end='2024-12-31')  
54 print(data.head())  
55  
56 #Calcul des rendements quotidiens  
57  
58 data["Daily Return"] = data['Close'].pct_change() #Utilisation de pct_change() pour le  
59 #calcul du rendement quotidien en pourcentage  
60 print(data[["Daily Return", "Close"]].head())  
61  
62 #Afficher les rendements sous forme de graphique  
63 plt.figure(figsize=(10, 6))  
64  
65 plt.plot(data.index, data["Daily Return"], label="Rendement quotidien", color="green")  
66  
67 plt.title("Rendement quotidien de l'action d'Apple (AAPL)")  
68 plt.xlabel("Date")  
69 plt.ylabel("Rendement quotidien")  
70 plt.show()
```



Merci de votre attention !