

Pratique de la data science

March 14, 2025

1 TP 1

Librairies à installer :

```
import yfinance as yf
import pandas as pd
```

1.1 Scrapping Ratios financiers

Objectif: Pour chaque ratio, récupérer les données de chaque compagnie dans une liste. Créer un fichier csv à partir du dataframe finalement obtenu.

```
companies = {
    "Apple": "AAPL",
    "Microsoft": "MSFT",
    "Amazon": "AMZN",
    "Alphabet": "GOOGL",
    "Meta": "META",
    "Tesla": "TSLA",
    "NVIDIA": "NVDA",
    "Samsung": "005930.KS",
    "Tencent": "TCEHY",
    "Alibaba": "BABA",
    "IBM": "IBM",
    "Intel": "INTC",
    "Oracle": "ORCL",
    "Sony": "SONY",
    "Adobe": "ADBE",
    "Netflix": "NFLX",
    "AMD": "AMD",
    "Qualcomm": "QCOM",
    "Cisco": "CSCO",
    "JP Morgan": "JPM",
    "Goldman Sachs": "GS",
    "Visa": "V",
    "Johnson & Johnson": "JNJ",
    "Pfizer": "PFE",
    "ExxonMobil": "XOM",
    "ASML": "ASML.AS",
    "SAP": "SAP.DE",
    "Siemens": "SIE.DE",
    "Louis Vuitton (LVMH)": "MC.PA",
    "TotalEnergies": "TTE.PA",
    "Shell": "SHEL.L",
    "Baidu": "BIDU",
    "JD.com": "JD",
    "BYD": "BYDDY",
    "ICBC": "1398.HK",
    "Toyota": "TM",
    "SoftBank": "9984.T",
    "Nintendo": "NTDOY",
    "Hyundai": "HYMTF",
    "Reliance Industries": "RELIANCE.NS",
    "Tata Consultancy Services": "TCS.NS"
}
```

```
ratios = {
    "forwardPE": [],
    "beta": [],
    "priceToBook": [],
    "priceToSales": [],
    "dividendYield": [],
    "trailingEps": [],
    "debtToEquity": [],
    "currentRatio": [],
    "quickRatio": [],
    "returnOnEquity": [],
    "returnOnAssets": [],
    "operatingMargins": [],
    "profitMargins": []
}
```

Fonctions:

- Accéder aux données d'une entreprise : `ticker = yf.Ticker(symbol)`.
- Accéder à un ratio spécifique : `ratio = ticker.info.get('ratio')`

Pipeline:

1. Créer le dictionnaire **"ratios"** : **Clés** = Nom de ratios, **Valeurs** = Liste vide.
2. Récupérer les ratios de chaque compagnie et les stocker dans les listes correspondantes.
3. Convertir le dictionnaire en dataframe avec le nom de la Company en index. Exporter en format csv.

1.2 Scrapping variations des stocks

Objectif: Pour chaque entreprise, récupérer cinq an d'historique de variation des stocks. Exporter dans un csv par entreprise.

Fonction:

- Récupérer les variations historiques: `company_data = yf.download(symbol, start=start_date, end=end_date)`

Pipeline (pour chaque entreprise):

1. Récupérer les données 'Close' de chaque jour dans un dataframe.
2. Créer une colonne 'Next Day Close' (avec la commande shift).
3. Créer une colonne 'Rendement' qui calcule les variations de stocks d'un jour à l'autre.
4. Exporter en format csv (mettre tous les csv dans un même fichier "Companies_historical_data").