

Portfolio management: EuroFund

09/12/2024

Loïc Blocquel, Anthony Cohen, Ludovic Deschamps

Introduction

- Vision: To offer **attractive returns** in a complex economic environment by investing in **French and European companies**.
- Target investors: **Retail investors** wishing to invest in **France and Europe, seeking a higher return** than European indices associated with an acceptable risk.

Summary

1. Objectives
2. Issues and constraints
3. Detailed strategy
4. Market & key indicators
5. Stock selection
6. Performance & characteristic

Goals

- **Performance:** higher return than the Euronext 100 index (5%)
- **Risk:** Volatility equal to the benchmark index
- **Sector allocation:** Sector diversification & focus on promising sectors (technology, energy, health, industry)
- **Optimization:** Little rebalancing to minimize turnover and associated costs

Constraints

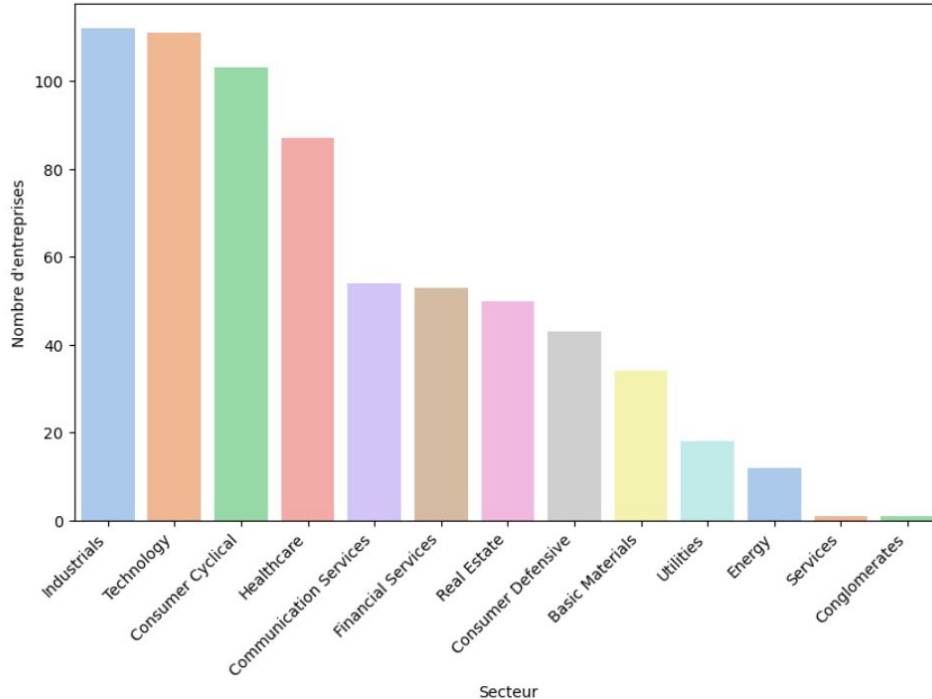
- **Geographic**-> European stock market with focus on France (Euronext exchange)
- **Minimize risk** -> investment in stable and counter-cyclical companies & in bonds
- **Diversification** -> Varied sectors and stock type (growth stocks, value stocks, counter-cyclical stocks and bonds)

Detailed Strategy

- **20% investment in growth companies:** volatile sectors such as technology or biotechnology, potential for **high returns** over the long term
- **40% investment in mature ‘value’ companies:** search for undervalued stocks for regular and predictable gains (e.g. Engie)
- **40% investment in stabilizing assets:** counter-cyclical companies (20%), government bonds to provide liquidity when needed (20%)

Investment universe

Nombre d'Entreprises par Secteur

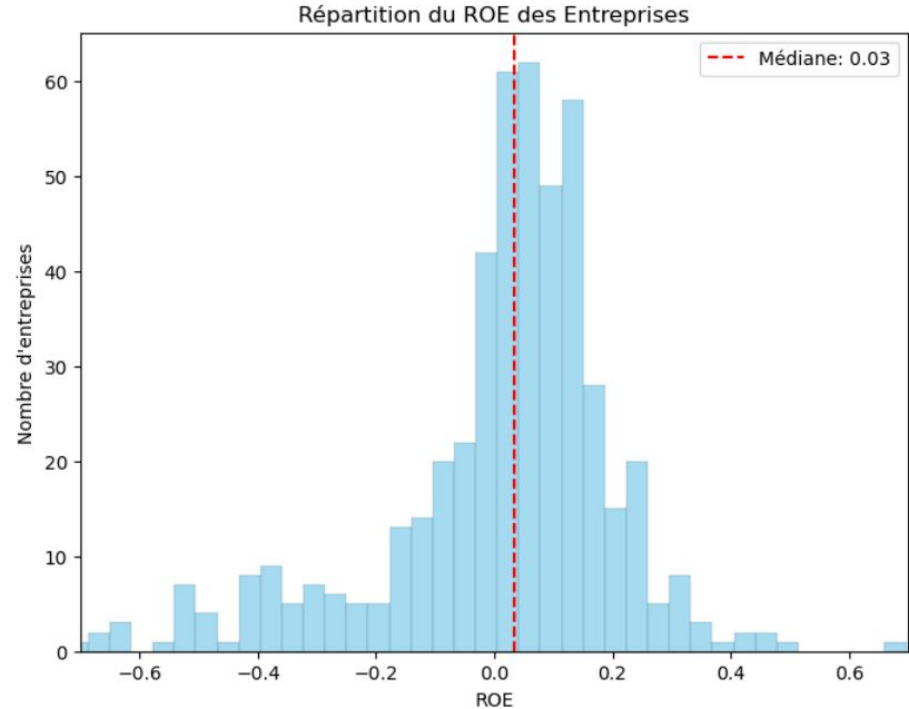


- **900 companies** from various sectors listed on **Euronext**
- Looking for the best stock in the 3 categories:
 - **growth** stocks
 - **value** stocks
 - **stable**/countercyclical stocks

Key indicators

$$\text{Return on Equity} = \frac{\text{Net Income}}{\text{Equity}}$$

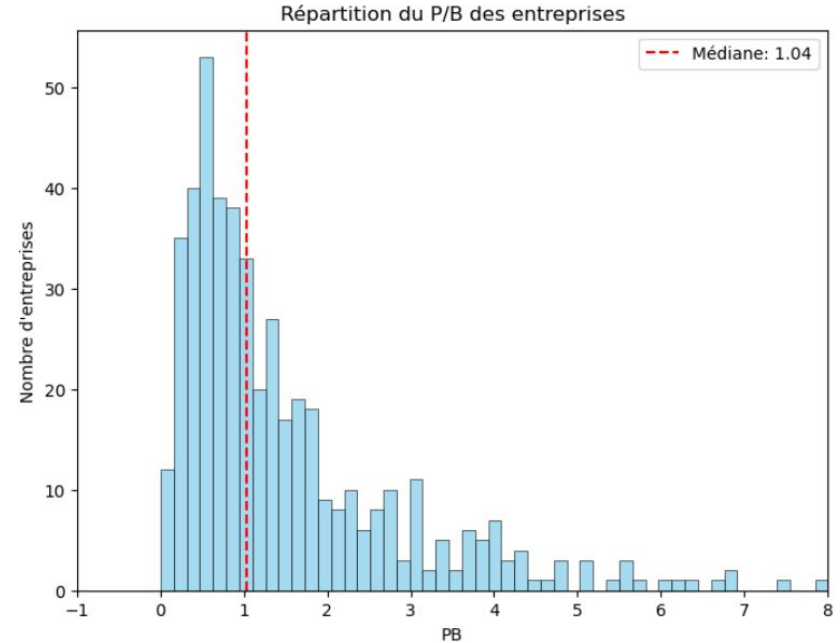
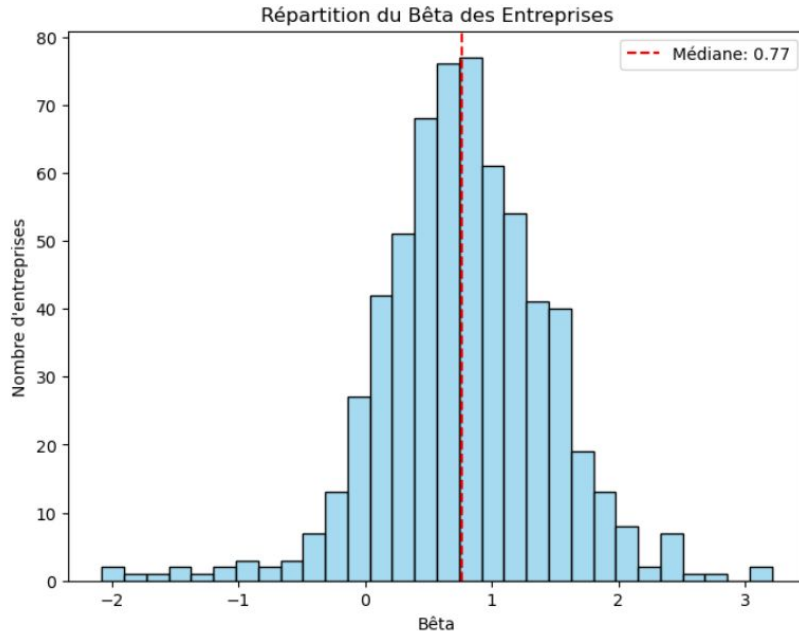
$$\text{Price-Earnings Ratio (P/E)} = \frac{\text{Stock Price}}{\text{Earnings Per Share (EPS)}}$$



Key indicators

$$\beta_p = \frac{\text{Cov}(r_p, r_b)}{\text{Var}(r_b)}$$

$$\text{Price to Book Ratio (P/B)} = \frac{\text{Market Capitalization}}{\text{Book Value of Equity (BVE)}}$$



Stock selection

Growth stocks:

- $P/E > 20$
- $\text{Rev growth} > 10\%$
- $\text{ROE} > 15\%$

Ex: Bourse direct,
L'oreal, Sword group

Value stocks:

- $P/E < 15$
- $P/B < 1.5$
- $\text{ROE} > 10\%$
- $0.5 < \text{Beta} < 1.5$

Ex: Eiffage, Engie, Mr.
Bricolage, TF1

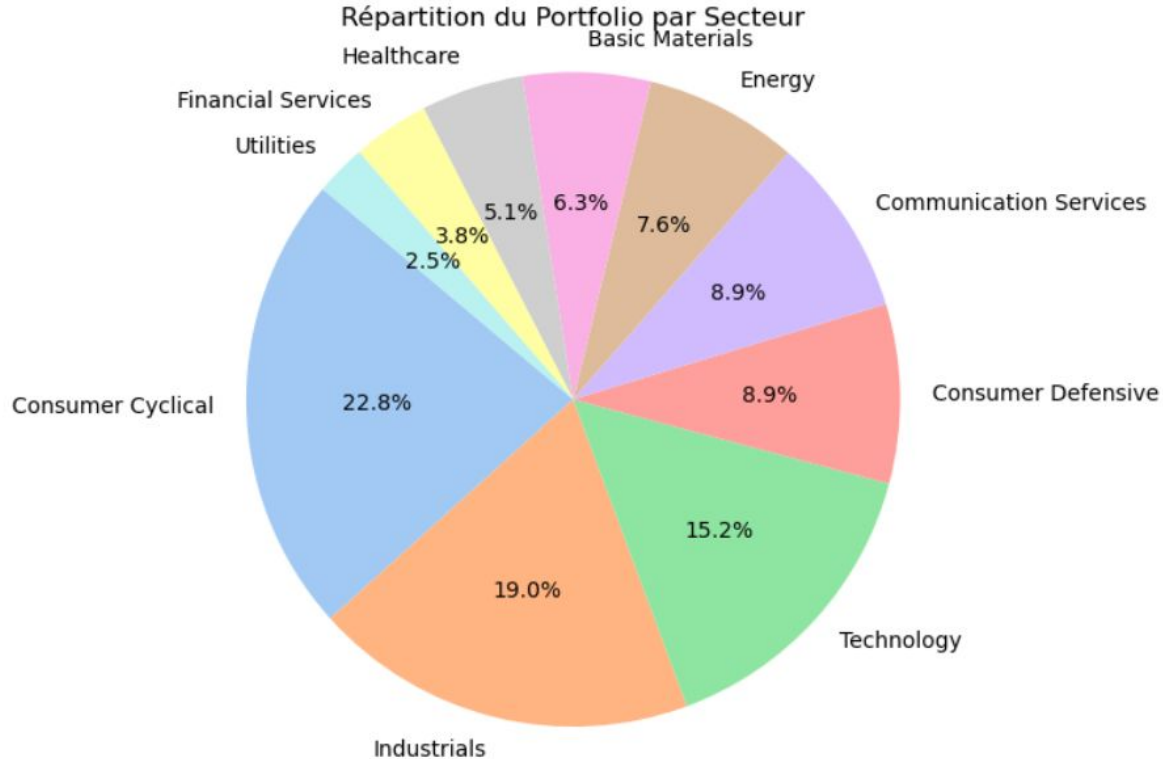
Counter-cyclical stocks:

- $-1 < \text{Beta} < 0.5$
- $\text{ROE} > 10\%$
- Sectors: healthcare,
utility, consumer
defensive

Ex: Bic, Euromedis,
Biomerieux

Stocks in the portfolio

- 900 -> 75 firms
- Sectoral balance in the portfolio
- Asset repartition
 - 20% growth (29 stocks)
 - 40% value (35 stocks)
 - 20% stable (11 stocks)
 - (20% bonds)



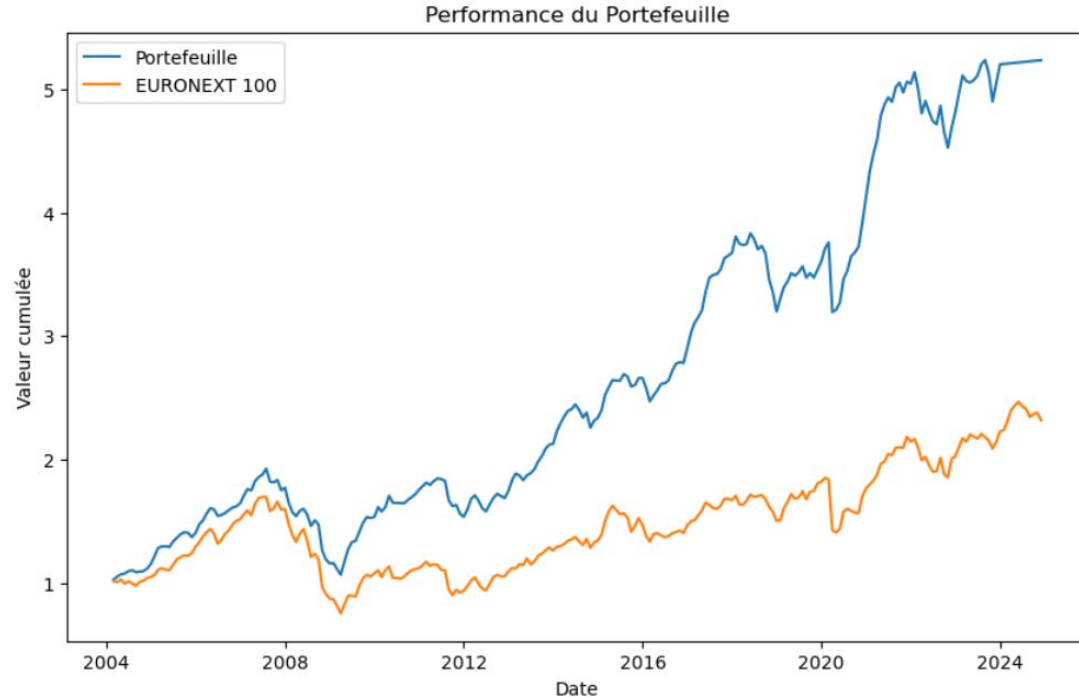
Portfolio characteristics

Portfolio (2004 - 2024)

- Annualized return: 8.2%
- Volatility: 10.5%
- Sharpe: 0.62
- VaR : - 5.7 %

Euronext 100 (2004 - 2024)

- Annualized return: 4.5%
- Volatility: 13.7%
- Sharpe: 0.22
- VaR : - 6.8%



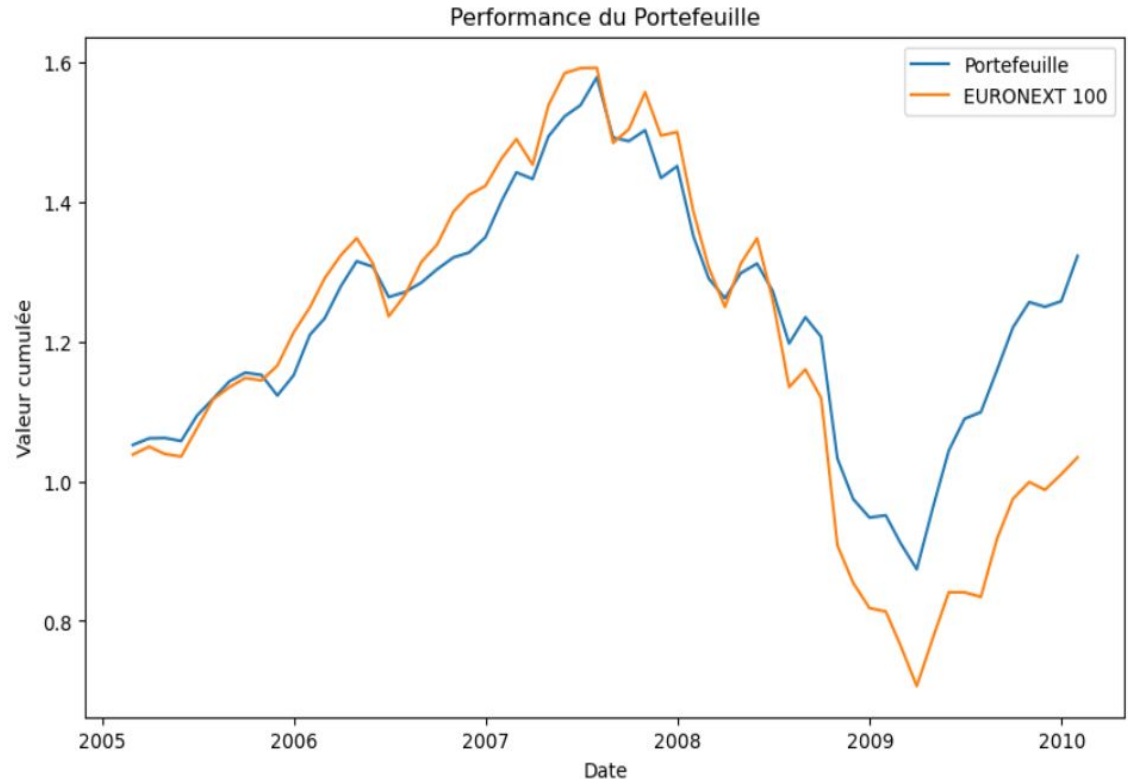
Performance in time of crisis

Portfolio (2005 - 2010)

- Return: 5.5%
- Volatility: 14%
- Sharpe: 0.33
- VaR : -7%

Euronext 100

- Return : 0.7 %
- VaR : - 8.1 %



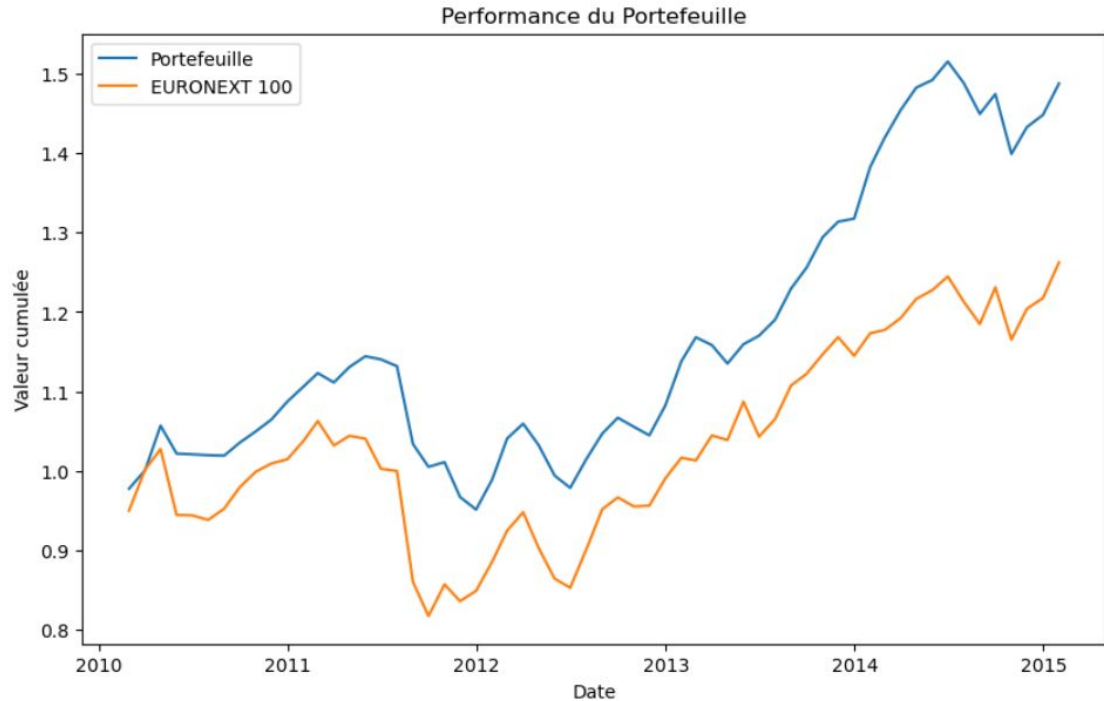
Performance under favorable conditions

Portfolio (2010 - 2015)

- Performance: 8.3%
- Volatility: 9%
- Sharpe: 0.72
- VaR : - 5%

Euronext 100

- Return : 4.8 %
- VaR : - 6.3%



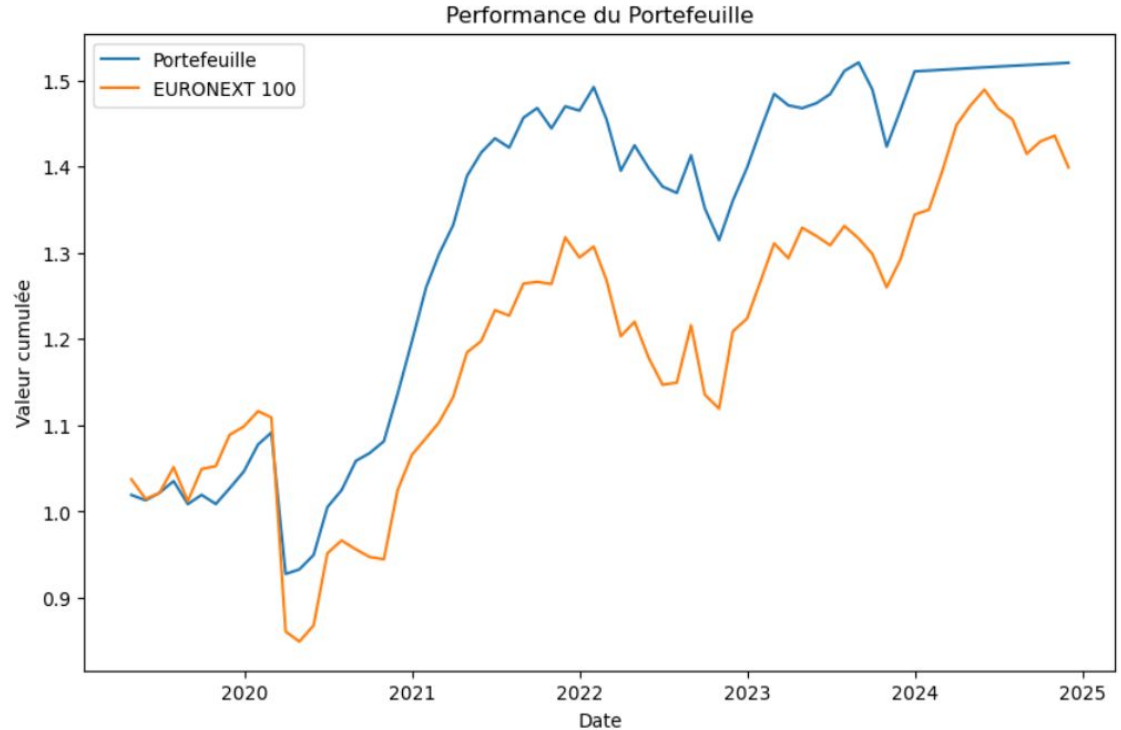
Recent Performance

Portfolio (2019 - 2024)

- Performance: 7.8%
- Volatility: 10%
- Sharpe: 0.60
- VaR : - 5.5 %

Euronext 100

- Return: 6.1 %
- VaR : - 7.3 %



Regulations

1. European regulatory framework

- **UCITS Directive :**

- Mandatory diversification (-10% in one asset)
- Eligible assets only (EuroNext)

2. MiFID II and Risk management:

- Advice adapted to each investor's risk profile
- Controlled volatility (-14% all times)
- VaR communicated (lower than EuroNext all times)
- Stress test to assess the impact of crises

3. Transparency and communication

- Communication of performance (yield, Sharp ratio, VaR)

Conclusion

- **Performance and volatility targets achieved :**
 - 8% Yield ($> 5\%$ Euronext 100)
 - 11% Volatility ($< 14\%$ Euronext 100)
 - 9% VaR ($< 18\%$ Euronext 100)
- **High sensitivity to the European market :** impact of local regulations and economic fluctuations
- **Limited turnover:** low costs but hinders exploitation of short-term opportunities

Thanks for your attention!

Questions ?

Code

```
1 import yfinance as yf
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from sklearn.linear_model import LinearRegression
6 import requests
7 from bs4 import BeautifulSoup
8 import math
9 from tqdm import tqdm
10 from itertools import islice
11 import seaborn as sns
```

```
1 def get_stock(dico,final_tab):
2     names,miss = list(dico.keys()), []
3     for firm in tqdm(names):
4         try:
5             code = dico[firm]
6             tick = yf.Ticker(code)
7             hist = tick.history(period = 'max')
8             hist.index = hist.index.strftime('%Y-%m-%d')
9             hist = hist.loc['2000-01-03':'2023-12-31','Close']
10            hist = hist.to_frame(name =firm)
11            final_tab = pd.concat([final_tab,hist],axis = 1)
12        except:
13            miss.append(firm)
14    return final_tab,miss
```

```
1 #On considère les 850 stocks de euronext Paris
2 stock_list = pd.read_excel(r'C:\Users\ludo1\OneDrive\Documents\Centrale\Rapport finance>List stock Paris.xlsx')
3 all_stocks = {}
4 for i in range(len(stock_list)):
5     all_stocks[stock_list.iloc[i,1]] = stock_list.iloc[i,2] + ".PA"
6
7 len(all_stocks)
```

```

1  ## La première fois on reconstruit une table de valeurs de stocks à partir de yahoo finance
2  tick = yf.Ticker('^N100')
3  hist = tick.history(period = 'max')
4  hist.index = hist.index.strftime('%Y-%m-%d')
5  hist = hist.loc['1999-12-29':'2024-11-28', 'High']
6  final_tab = hist.to_frame(name = 'EURONEXT 100')
7  final_tab, miss = get_stock(all_stocks, final_tab)
8  final_tab.index = pd.to_datetime(final_tab.index)
9  moyennes_mois = final_tab.resample('M').mean()

```

```

1  donnees = []
2
3  for s in tqdm(stocks):
4      try :
5          ticker = all_stocks[s]
6          tick = yf.Ticker(ticker)
7          info = tick.info
8
9          # Extraire Les paramètres
10         PE = info.get('trailingPE', None)
11         ROE = info.get('returnOnEquity', None)
12         growth = info.get('revenueGrowth', None)
13         div = info.get('dividendYield', None)
14         sec = info.get('sector', None)
15         PB = info.get('priceToBook', None)
16         beta = info.get('beta', None)
17
18
19         donnees.append([s, PE, ROE, growth, div, sec, PB, beta])
20
21     except Exception as e:
22         print(f"Erreur avec {s}: {e}")
23
24 param = pd.DataFrame(donnees, columns=['Entreprise', 'PER', 'ROE', 'Croissance CA', 'Dividende', 'Secteur', 'PB', 'beta'])
25 param.set_index('Entreprise', inplace=True)

```

```
: 1 # Sélection des actions contracyclique
2 safe_sectors = ['Consumer Defensive', 'Healthcare', 'Utilities']
3
4 actions_contracy = param[(param['beta'] < 0.5) &
5                           (param['beta'] >= -1) &
6                           (param['ROE'] >= 0.10) &
7                           (param['Secteur'].isin(safe_sectors))
8                           ].index.tolist()
9 len(actions_contracy)
```

: 7

```
: 1 # Sélection des actions growth
2
3 actions_growth = param[
4     (param['ROE'] > 0.15) &
5     (param['Croissance CA'] > 0.07)
6 ].index.tolist()
7
8 len(actions_growth)
```

: 34

```
: 1 #Selection des actions value
2 actions_value = param[
3     (param['PB'] <= 1.5) &
4     (param['ROE'].between(0.10, 0.20)) &
5     (param['beta'].between(0.5, 1.5))
6 ].index.tolist()
7
8 len(actions_value)
```

: 38

```
1 obligations = ['Obli']
```

```
1 #Selection de la fenetre de temps  
2 rendements.index  
3 date = rendements.index[230:298]  
4 print (f'Etude du {date[0]} au {date[-1]}')
```

Etude du 2019-04-30 00:00:00 au 2024-11-30 00:00:00

```
1 poids = {  
2     "growth": 0.2,  
3     "value": 0.4,  
4     "contracyclique": 0.2,  
5     "obligations": 0.2  
6 }
```

```
1 rendement = (  
2     rendements.loc[date, actions_growth].mean(axis=1) * poids["growth"] +  
3     rendements.loc[date,actions_value].mean(axis=1) * poids["value"] +  
4     rendements.loc[date,actions_contracy].mean(axis=1) * poids["contracyclique"] +  
5     rendements.loc[date,obligations].mean(axis=1) * poids["obligations"]  
6 )
```

```

1 valeur = (1 + rendement).cumprod()
2
3 #calcul du rendement annualisé
4 perf = valeur.iloc[-1] - 1
5 perf_an = (1 + perf) ** (12 / len(date)) - 1
6 print(f'rendement annualisé : {perf_an}')
7
8 #calcul de la volatilité annualisé
9 vol = rendement.std() * np.sqrt(12)
10 print (f'volatilité : {vol}')
11
12 #ratio de sharpe
13 taux_sans_risque = 0.02 # Exemple annuel
14 sharpe = (rendement.mean() * 12 - taux_sans_risque) / vol
15 print (f'Sharpe : {sharpe}')

```

rendement annualisé : 0.07673372017878721
volatilité : 0.10163994356216932
Sharpe : 0.5850523653693505

```

1 euro100 = rendements.loc[date, 'EURONEXT 100']
2 v100 = (1+euro100).cumprod()
3 perf100 = v100.iloc[-1] - 1
4 perf_an100 = (1 + perf100) ** (12 / len(date)) - 1
5
6 vol100 = euro100.std() * np.sqrt(12)
7 print(f'rendement annualisé : {perf_an100}')
8 print (f'volatilité : {vol100}')
9
10 taux_sans_risque = 0.02 # Exemple annuel
11 sharpe100 = (euro100.mean() * 12 - taux_sans_risque) / vol100
12 print (f'Sharpe : {sharpe100}')

```

rendement annualisé : 0.061067356903222514
volatilité : 0.14277275958191066
Sharpe : 0.3515258743565519