Plan

1 Financial time-series in Python

① Time series:

t	x_t	r_t
0	x_0	
1	x_1	$\ln(x_1/x_0)$
2	x_2	$\ln(x_2/x_1)$
3	x_3	$\ln(x_3/x_2)$
4	x_4	$\ln(x_4/x_3)$
⋮	⋮	⋮
N-1	x_{N-1}	$\ln(x_{N-1}/x_{N-2})$
N	x_N	$\ln(x_N/x_{N-1})$

Ex: t = time (in days) from some given starting day $t=0$

x_t = price (at close) for AAPL (Apple) at day t

Return at day t:

$$R_t = \frac{x_t - x_{t-1}}{x_{t-1}} = x_t/x_{t-1} - 1$$

$$\Leftrightarrow \quad x_t/x_{t-1} = R_t + 1$$

$$x_t = x_{t-1}\cdot(1+R_t)$$

Continuous return at day t:

$$r_t = \ln(x_t/x_{t-1})$$

$$\Leftrightarrow \quad x_t/x_{t-1} = e^{r_t}$$

$$x_t = x_{t-1}\cdot e^{r_t}$$

Note: $r_1 + r_2 + \dots + r_N$

$$= \ln(x_1/x_0) + \ln(x_2/x_1) + \dots + \ln(x_N/x_{N-1})$$

$$= \ln\left(\frac{x_1}{x_0}\cdot\frac{x_2}{x_1}\cdots\frac{x_N}{x_{N-1}}\right) = \ln(x_N/x_0)$$

$$\bar{r} = \frac{1}{N}(r_1 + \dots + r_N) = \frac{1}{N}\ln(x_N/x_0)$$

$$= \ln(x_N/x_0)^{1/N} \quad \leftarrow \text{CAGR = compounded agregated growth rate}$$

Using R_i: $\bar{R} = \left[(1+R_1)(1+R_2)\cdots(1+R_N)\right]^{1/N} - 1$

Monthly continuous returns for n securities (AAPL, ...)

t	AAPL r_t	MSFT r_t	AMZN r_t	.....
0				
1	r_{11}	r_{21}	r_{31}	--
2	r_{12}	r_{22}	r_{32}	--
3	r_{13}	r_{23}	r_{33}	--
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
N	r_{1N}	r_{2N}	r_{3N}	--
Mean:	μ_1	μ_2	μ_3	--
Std.dev: (sample)	σ_1	σ_2	σ_3	--
Covariance:	σ_{11}	σ_{12}	σ_{13}	--
	σ_{21}	σ_{22}	σ_{23}	--
	σ_{31}	σ_{32}	σ_{33}	--

$$\mu_i = \frac{r_{i1}+r_{i2}+\dots+r_{iN}}{N}$$

$$\sigma_i^2 = \frac{(r_{i1}-\mu_i)^2+\dots+(r_{iN}-\mu_i)^2}{N-1}$$

$$\sigma_{ij} = \frac{(r_{i1}-\mu_i)(r_{j1}-\mu_j)+\dots}{N-1}$$

$$\underline{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{pmatrix}$$

expected returns

Covariance matrix

$\mu_i = E(r_i)$

$\sigma_{ij} = Cov(r_i, r_j)$

$\underline{e} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

$\underline{e}^T \cdot \underline{w} = 1$

$(w_1, \dots, w_n)$: $w_1 + w_2 + \dots + w_n = 1$

w_i : share of capital invested in sec. #i.

$R = w_1 r_1 + w_2 r_2 + \dots + w_n r_n$

$E(R) = E(w_1 r_1 + \dots + w_n r_n) = w_1 \mu_1 + \dots + \mu_n \cdot w_n = \underline{\mu}^T \cdot \underline{w}$

$Var(R) = Cov(w_1 r_1 + \dots + w_n r_n, \; w_1 r_1 + \dots + w_n r_n)$

$= w_1^2 \, Cov(r_1, r_1) + w_1 w_2 \cdot Cov(r_1, r_2) + \dots + w_n^2 \, Cov(r_n, r_n) = \underline{w}^T \Sigma \underline{w}$

Assumptions:

(i) Σ symmetric, positive def. matrix

a) Σ clearly symmetric : $\text{Cov}(r_i, r_j) = \text{Cov}(r_j, r_i)$

b) Σ positive semidefinite : $\underline{w}^T \Sigma \underline{w} = \text{Var}(w_1 r_1 + \dots + w_n r_n) \geq 0$ for all $\underline{w}$

c) Σ positive def. $\iff$ $\underline{w}^T \Sigma \underline{w} > 0$ for all $\underline{w} \neq \underline{0}$.

If there is such a choice $\underline{w}$ of portfolio weight that $\underline{w}^T \Sigma \underline{w} = 0$, then

$w_1 r_1 + w_2 r_2 + \dots + w_n r_n = C$ is constant (zero variance)

$C \neq 0$: arbitrage opportunity

$C = 0$: a security has returns that are a linear comb. of returns of the other securities

(ii) $\{\underline{\mu}, \underline{e}\}$ are linearly independent

otherwise, $\underline{\mu} = c \cdot \underline{e}$

$\Rightarrow \underline{\mu} = \begin{pmatrix} c \\ c \\ \vdots \\ c \end{pmatrix}$ (all securities have the same exp. return)

Optimization problem

(A) $\min \underline{w}^T \Sigma \underline{w}$ when $\underline{e}^T \cdot \underline{w} = 1$

$\underline{w}^T \Sigma \underline{w} = \text{Var}(r)$

$\underline{e} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$

(B) $\min \underline{w}^T \Sigma \underline{w}$ when $\begin{cases} \underline{e}^T \cdot \underline{w} = 1 \\ \underline{\mu}^T \cdot \underline{w} = \alpha \end{cases}$

for a given value α of the expected portfolio return.

	MSFT	AAPL	SBUX	NKE	AMZN
Return	0.0147	0.0227	0.0114	0.0154	0.0257
Stdev	0.0654	0.0909	0.0767	0.0654	0.0938
Oct-06	0.0485	0.0519	0.1032	0.0494	0.1705
Nov-06	0.0224	0.1226	-0.0674	0.0741	0.0574
Dec-06	0.0203	-0.0773	0.0037	0.0008	-0.0221
Jan-07	0.0329	0.0104	-0.0136	-0.0003	-0.0464
Feb-07	-0.0912	-0.0132	-0.1229	0.0557	0.0383
Mar-07	-0.0072	0.0936	0.0148	0.0170	0.0165
Apr-07	0.0716	0.0715	-0.0109	0.0154	0.4327
May-07	0.0247	0.1942	-0.0739	0.0523	0.1199
Jun-07	-0.0373	0.0070	-0.0934	0.0268	-0.0106
Jul-07	-0.0164	0.0766	0.0166	-0.0287	0.1381
Aug-07	-0.0090	0.0497	0.0321	-0.0020	0.0173
Sep-07	0.0286	0.1028	-0.0502	0.0404	0.1533
Oct-07	0.2227	0.2133	0.0182	0.1252	-0.0439
Nov-07	-0.0912	-0.0415	-0.1316	-0.0092	0.0157
Dec-07	0.0611	0.0835	-0.1333	-0.0217	0.0227
Jan-08	-0.0880	-0.3807	-0.0793	-0.0429	-0.1759
Feb-08	-0.1811	-0.0795	-0.0504	-0.0186	-0.1867
Mar-08	0.0463	0.1379	-0.0271	0.1218	0.1007
Apr-08	0.0049	0.1924	-0.0753	-0.0140	0.0979
May-08	-0.0070	0.0817	0.1140	0.0232	0.0373
Jun-08	-0.0253	-0.1198	-0.1447	-0.1371	-0.1071
Jul-08	-0.0673	-0.0520	-0.0690	-0.0123	0.0402
Aug-08	0.0593	0.0644	0.0575	0.0324	0.0569
Sep-08	-0.0183	-0.3998	-0.0454	0.0987	-0.1049
Oct-08	-0.1784	-0.0549	-0.1244	-0.1454	-0.2399
Nov-08	-0.0993	-0.1493	-0.3855	-0.0790	-0.2931
Dec-08	-0.0326	-0.0823	0.0577	-0.0432	0.1831
Jan-09	-0.1283	0.0545	-0.0021	-0.1148	0.1372
Feb-09	-0.0572	-0.0091	-0.0312	-0.0858	0.0967
Mar-09	0.1356	0.1630	0.1941	0.1214	0.1253
Apr-09	0.0979	0.1798	0.2635	0.1183	0.0920
May-09	0.0306	0.0763	-0.0049	0.0837	-0.0319
Jun-09	0.1355	0.0476	-0.0354	-0.0969	0.0702
Jul-09	-0.0106	0.1373	0.2424	0.0940	0.0248
Aug-09	0.0469	0.0291	0.0703	-0.0223	-0.0548
Sep-09	0.0481	0.0970	0.0838	0.1554	0.1397
Oct-09	0.0752	0.0169	-0.0843	-0.0350	0.2411
Nov-09	0.0588	0.0588	0.1431	0.0427	0.1345
Dec-09	0.0401	0.0527	0.0516	0.0180	-0.0103
Jan-10	-0.0785	-0.0928	-0.0566	-0.0316	-0.0701
Feb-10	0.0172	0.0633	0.0501	0.0586	-0.0575
Mar-10	0.0261	0.1384	0.0577	0.0837	0.1369
Apr-10	0.0418	0.1053	0.0681	0.0363	0.0097
May-10	-0.1687	-0.0163	0.0007	-0.0476	-0.0887
Jun-10	-0.1099	-0.0210	-0.0634	-0.0691	-0.1383

Plan

1 Portfolio optimization in Python

We take the theoretical models from Lecture 9 Part 1 as a starting point (see attached summary of solution for $\underline{w}$ - minimal variance portfolio given constraints; these solutions follow from Lecture 9).

[PA] Chap. 5, 10 includes material on DataFrames, Series and how to use them.

Python script from this lecture in attached.

Problem A:

$$\min \quad \underline{w}^T \Sigma \underline{w} \qquad \text{wlm} \qquad \underline{e}^T \cdot \underline{w} = 1$$

Problem B:

$$\min \quad \underline{w}^T \Sigma \underline{w} \qquad \text{wlm} \left\{ \begin{array}{l} \underline{e}^T \underline{w} = 1 \\ \underline{\mu}^T \underline{w} = \kappa \end{array} \right.$$

Solution:

$$\underline{A} : \quad \underline{w}_M^* = \frac{1}{\underline{e}^T \Sigma^{-1} \underline{e}} \cdot \Sigma^{-1} \cdot \underline{e}$$

$$\underline{B} : \quad \underline{w}^* = \Sigma^{-1} \left(\frac{\lambda_1}{2} \underline{e} + \frac{\lambda_2}{2} \underline{\mu} \right)$$

$$= \Sigma^{-1} \cdot \left(\underline{e} \,|\, \underline{\mu} \right) \cdot \begin{pmatrix} \underline{e}^T \Sigma^{-1} \underline{e} & \underline{e}^T \Sigma^{-1} \underline{\mu} \\ \underline{\mu}^T \Sigma^{-1} \underline{e} & \underline{\mu}^T \Sigma^{-1} \underline{\mu} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \kappa \end{pmatrix}$$

```python
import pandas as pd
import datetime as dt
import numpy as np

tickers = ['AAPL', 'BAKKA']

ticker = tickers[0]
tickers = tickers[1:]

daily = pd.read_csv(ticker+'.csv', parse_dates = True, usecols=[0,5], header=0, names=['Date','Price'], index_col=0)
monthly = daily.resample('M').last()
returns = np.log(monthly/monthly.shift(1))

for ticker in tickers:
    daily = pd.read_csv(ticker+'.csv', parse_dates = True, usecols=[0,5], header=0, names=['Date','Price'], index_col=0)
    monthly = daily.resample('M').last()
    returns[ticker] = np.log(monthly.Price/monthly.Price.shift(1))

mu = np.array(returns.mean()).reshape(len(tickers)+1,1)
Sigma = np.array(returns.cov())



# Compute minimum variance portfolio
u = np.ones((mu.shape[0],1))
SigmaInv = np.linalg.inv(Sigma)
n = SigmaInv.dot(u)
omega_M = SigmaInv.dot(u)/u.transpose().dot(SigmaInv.dot(u))
mu_M = mu.transpose().dot(omega_M)[0,0]
sigma_M = omega_M.transpose().dot(Sigma.dot(omega_M))[0,0]**0.5



# Plot minimum variance portfolio with expected return alpha
def stddev(alpha,mu,Sigma):
    u = np.ones((mu.shape[0],1))
    U = np.concatenate((u,mu),axis=1)
    SigmaInv = np.linalg.inv(Sigma)
    B = U.transpose().dot(SigmaInv).dot(U)
    BInv = np.linalg.inv(B)
    lvec = np.array([[1],[alpha]])
    omega = SigmaInv.dot(U).dot(BInv.dot(lvec))
    sigma = omega.transpose().dot(Sigma.dot(omega))[0,0]**0.5
    return(sigma)

points=[]
for r in np.linspace(0*10*mu_M,3*mu_M,50):
    sigma = stddev(r,mu,Sigma)
    points.append((sigma,r))
p = pd.DataFrame(points, columns=['standard deviation','expected return'])
p.plot(kind='line',x='standard deviation',y='expected return',color='blue')


# Compare with the individual securities
s=[]
for i in range(mu.shape[0]):
    s.append((Sigma[i,i]**0.5,mu[i,0]))
sec = pd.DataFrame(s, columns=['standard deviation','expected return'])
```