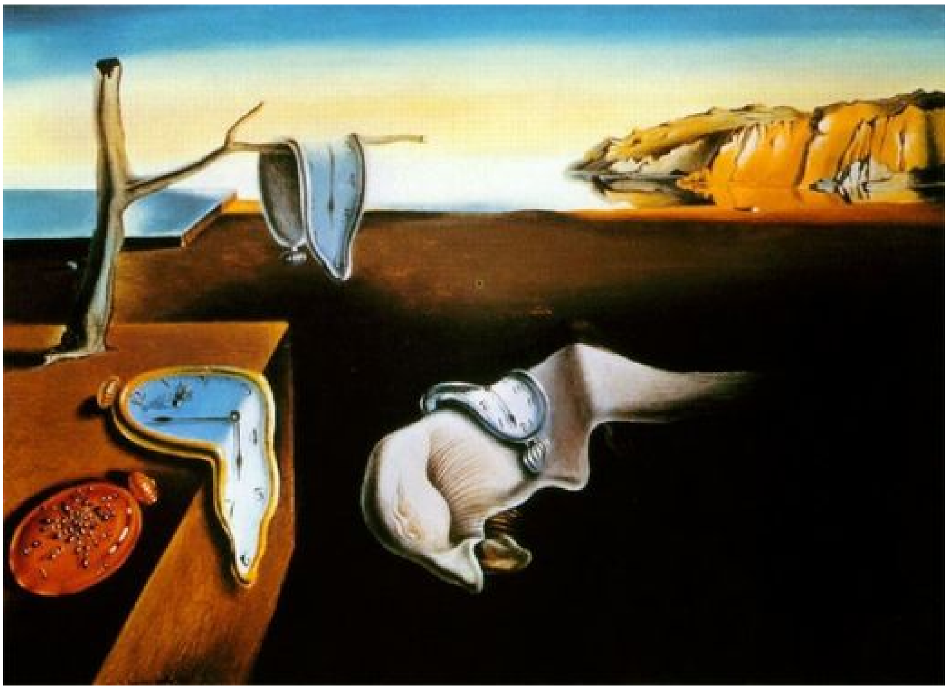
Université de Lorraine
Master d'Economie Appliquée
Programmation en R/Python

Année universitaire 2020/2021
Parcours IFM
Cours de Sophie Béreau-Gnabo

Travail de programmation en R
A rendre au plus tard le vendredi 18/06/2021

Ce travail consiste en trois exercices indépendants nécessitant la mise en œuvre d'un certain nombre concepts abordés en classe. Il vous est demandé de réaliser par groupe de 2 maximum un **code R commenté ligne à ligne** et de le soumettre sous format .R (ou .Rmd, joint avec l'output sous format .pdf) sur la plateforme ARCHE dédiée au plus tard le **vendredi 18 juin 2021 à 23h59**. Dans le préambule, il doit être indiqué la **répartition des tâches** entre chacun des auteurs. Aucune soumission tardive ne sera acceptée.
Vous veillerez par ailleurs, à ce que, pour chacun des exercices, toute **ressource** qui aurait été utilisée pour réaliser votre travail soit **dûment référencée**.

Bon courage !



La persistance de la mémoire - S. Dalì

1 Un exercice de mise en jambes pour commencer

1. Proposez deux versions (une itérative et une récursive) d'une fonction nommée `snpe()` permettant de calculer la somme de n premiers entiers naturels, puis comparez les résultats pour $n = 100$.
2. La suite de Fibonacci est une suite de nombres entiers très connue. Les deux premiers termes de la suite sont 0 et 1 et tous les autres sont la somme des deux termes précédents. Formellement, elle se définit comme suit :

$$u_1 = 0, \; u_1 = 1, u_k = u_{k-1} + u_{k-2} \; \forall k \geq 3 \in \mathbb{N}$$

Consignes. Proposez deux versions (une itérative et une récursive) d'une fonction permettant de calculer les n premiers termes de la suite de Fibonnaci. Comparez les résultats pour $n = 100$.

2 Réplication de la table statistique de la distribution $\mathcal{N}(0, 1)$

L'objectif de cet exercice est de répliquer la table statistique des probabilités d'une distribution Normale centrée réduite ($\mathcal{N}(0, 1)$) à la "force brute", càd sans utiliser la commande `pnorm()`, à l'aide de la seule fonction `integrate()`.

Soit $\Phi(x)$ la fonction de répartition de la loi de distribution $\mathcal{N}(0, 1)$. On rappelle que $\Phi(-x) = 1 - \Phi(x)$. Aussi, peut-on se contenter de construire la table pour les valeurs positives de x.

1. Programmez la fonction `phi()` qui prend un vecteur x de longueur n comme paramètre d'entrée et renvoie le vecteur de valeurs $\frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{x_i^2}{2}}, \; i = 1, \ldots, n$
2. Construisez le vecteur `quantiles` comme une séquence de nombre de 0 à 5.5 d'incrément 10^{-2}, puis utilisez la fonction `integrate()` pour calculer $\Phi(x)$ pour tous les x du vecteur `quantiles` et stockez-les dans un vecteur nommé `probs`. Vérifiez que les probabilités obtenues sont conformes à celles renvoyées par la commande `pnorm`.
3. Tracez la courbe des valeurs de $\Phi(x)$ pour tous les valeurs de -5.5 à 5.5 à l'aide de la fonction `rev()`. Ajoutez sur le même graphique, en bleu, la courbe de la fonction `pnorm()`. Qu'obtenez-vous ?

3 Identification du portefeuille optimal de Markowitz

Dans le cadre de la théorie "moderne" de portefeuille d'Harry Markowitz (Markowitz, 1952), le problème du portefeuille consiste identifier le portefeuille optimal, càd le portefeuille parmi les portefeuilles réalisables [1] qui minimise la variance du portefeuille à cible c^* de rendement espéré donné. Formellement, les poids $x_1, \ldots, x_N$ du portefeuille sont solutions

1. Cette contrainte implique que toute la richesse disponible (pas plus, pas moins) est investie à l'optimum, dans les différents titres disponibles.

du programme d'optimisation (P) suivant :

$$\begin{aligned}
\min_{\{x_1,x_2,\ldots,x_N\}} \sigma_P^2 &= \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij} \\
\text{s.t. } \mu_P = \sum_{i=1}^{N} x_i\mu_i &= c^* \\
\sum_{i=1}^{N} x_i &= 1
\end{aligned}$$

Il est possible de réécrire le problème sous forme matricielle comme suit :

$$\begin{aligned}
\min_x \sigma_{P,x}^2 = x'\Sigma x & \\
s.t.\ \mu_{P,x} = x'\mu &= c^* \\
x'\mathbf{1} &= 1
\end{aligned}$$

avec : $x = (x_1, x_2, \ldots, x_N)'$ le vecteur de poids alloués à chacun des titres indicés par i ; $\mathbf{1} = (1 \ldots, 1)'$ le vecteur unitaire ; $\mu = (\mu_1, \ldots, \mu_N)'$ le vecteur de rendements espérés associés à chacun des actifs du portefeuille ; $\Sigma = (\sigma_{ij})_{i,j\in\{1,\ldots,N\}}$ la matrice de covariances des rendements (avec en diagonale, les variances associés à chacun des N rendements) et c^* la cible de rendement espéré du portefeuille.

Pour identifier x^* le portefeuille de la frontière efficiente ciblant l'espérance de rendement $\mu_{P,x} = c^*$, deux modes de résolution sont possibles :

Résolution 1. Ecrivons tout d'abord le Lagrangien associé au problème (P) :

$$\mathcal{L}(x;\lambda_1,\lambda_2) = x'\Sigma x + \lambda_1(x'\mu - c^*) + \lambda_2(x'\mathbf{1} - 1)$$

Les conditions du premier ordre s'obtiennent alors comme suit [2] :

$$\begin{aligned}
\frac{\partial\mathcal{L}(x;\lambda_1,\lambda_2)}{\partial x} &= 2\Sigma x + \lambda_1\mu + \lambda_2\mathbf{1} = \mathbf{0} \\
\frac{\partial\mathcal{L}(x;\lambda_1,\lambda_2)}{\partial \lambda_1} &= x'\mu - c^* = 0 \\
\frac{\partial\mathcal{L}(x;\lambda_1,\lambda_2)}{\partial \lambda_2} &= x'\mathbf{1} - 1 = 0
\end{aligned}$$

On remarque ici que l'on dispose de $N+2$ équations pour $N+2$ inconnues, à savoir $x_1, \ldots, x_N, \lambda_1$ et λ_2.

En écrivant le système d'équations sous forme matricielle, on obtient la représentation suivante :

$$\underbrace{\begin{pmatrix} 2\Sigma & \mu & \mathbf{1} \\ \mu' & 0 & 0 \\ \mathbf{1}' & 0 & 0 \end{pmatrix}}_{A_x} \underbrace{\begin{pmatrix} x \\ \lambda_1 \\ \lambda_2 \end{pmatrix}}_{z_x} = \underbrace{\begin{pmatrix} \mathbf{0} \\ c^* \\ 1 \end{pmatrix}}_{b}$$

2. $\mathbf{0}$ correspond ici au vecteur de taille N : $(0, \ldots, 0)'$.

L'identification de z_x (qui contient x) nécessite la multiplication à gauche de l'inverse de A_x de part et d'autre de l'égalité :

$$z_x = A_x^{-1} b$$

Finalement, seuls les N premiers éléments de z_x correspondent aux poids optimaux x^* recherchés.

Résolution 2. Par soucis de simplification et sans perte de généralité, il est possible de réécrire le programme (P) comme suit :

$$\begin{array}{ll} \min\limits_{x} & \frac{1}{2} x' \Sigma x \\ s.t. & x'\mu = c^* \\ & x'\mathbf{1} = 1 \end{array} \quad \sim \quad \min_{x;\lambda_1,\lambda_2} \mathcal{L}(x;\lambda_1,\lambda_2)$$

avec : $\mathcal{L}(x;\lambda_1,\lambda_2) = \frac{1}{2}x'\Sigma x + \lambda_1\Big(c^* - x'\mu\Big) + \lambda_2\Big(1 - x'\mathbf{1}\Big)$

Il est possible de montrer[3] que :

$$\lambda_1 = \frac{CE - A}{D}; \ \lambda_2 = \frac{B - AE}{D}; \text{ et } x^* = \frac{EC - A}{D} \cdot \Sigma^{-1}\mu + \frac{B - AE}{D} \cdot \Sigma^{-1}\mathbf{1}$$

avec : $A = \mu'\Sigma^{-1}\mathbf{1}$; $B = \mu'\Sigma^{-1}\mu$; $C = \mathbf{1}'\Sigma^{-1}\mathbf{1}$; $D = BC - A^2$; et $E = c^*$, toutes des grandeurs > 0.

Consignes. Dans cet exercice, il vous est demandé, en usant de la seule force brute[4] de proposer un code permettant :

1. de récupérer à l'aide d'une boucle les cours boursiers de 10 des titres du S&P 500[5] sur la période s'étalant de janvier 2010 à janvier 2020, en fréquence mensuelle sur la plateforme Yahoo !Finance[6] ;
2. d'estimer les paramètres μ_i, σ_i^2, et σ_{ij} $\forall i,j \in \{1,\ldots,10\}$ sur la période considérée à partir des moyennes, variances et covariances d'échantillonnage données par les formulations suivantes :

$$\begin{aligned} \widehat{\mu}_i &= \overline{r}_i = \frac{1}{T}\sum_{t=1}^{T} r_{i,t} \\ \widehat{\sigma}_i^2 &= \frac{1}{T-1}\sum_{t=1}^{T}(r_{i,t} - \overline{r}_i)^2 \\ \widehat{\sigma}_{ij} &= \frac{1}{T-1}\sum_{t=1}^{T}(r_{i,t} - \overline{r}_i)(r_{j,t} - \overline{r}_j) \end{aligned}$$

3. Pour le détail de la preuve, on consultera à profit les pages 10-12 de Danthine and Donaldson (2005).

4. Il est interdit de recourir à des packages pré-programmés calculant les poids optimaux de portefeuilles dans cet exercice. Vous devez faire l'usage du calcul matriciel et des fonctions usuelles en R, à partir des éléments théoriques rappelés ci-dessus.

5. La liste de titres constitutifs de l'indice ainsi que les symboles nécessaires à leur extraction sur la plateforme Yahoo !Finance sont référencés sur la page suivante : `https://www.slickcharts.com/sp500`.

6. Un exemple de code pour la récupération du cours de l'action Apple et le calcul des log-rendements est disponible en Annexe A.

3. et enfin, d'identifier le portefeuille optimal de Markowitz de deux manières distinctes. Pour la cible de rendement c^*, on choisira la moyenne de rendement $\widehat{\mu}_i$ la plus élevée parmi celles calculées à la question précédente.

Références

DANTHINE, J.-P. AND J. B. DONALDSON (2005) : *Intermediate Financial Theory*, Prentice-Hall.

MARKOWITZ, H. (1952) : "Portfolio Selection," *Journal of Finance*, 7, 77–91.

A Exemple de code utile

```
# Code to download your data from Yahoo!Finance.

options(digits=4, width=70)

# Please install first using: install.packages('package_name') command!

library("PerformanceAnalytics")
library("tseries")
library("zoo")

AAPL.prices <- get.hist.quote(instrument='AAPL', start="2009-12-01",
                              end="2020-02-01", quote="AdjClose",
                              provider="yahoo", origin="2000-09-01",
                              compression="m", retclass="zoo")
index(AAPL.prices) <- as.yearmon(index(AAPL.prices))
AAPLr.z <- diff(log(AAPL.prices))
```