# Scénario: Exploration, classification des encours boursiers parisiens

## Résumé

*Scénario d'analyse d'un jeu de données : l'ensemble des séries des encours boursier à Paris. Description, lissage et classification de ces courbes. Utilisation des différentes techniques descriptives uni, bi et multidimensionnelles : ACP, k-means,CAH.*

## 1 Objectif

Le travail proposé s'intéresse aux cours des actifs boursiers sur la place de Paris de 2000 à 2009. Seules 252 cours d'entreprises ou indices sont considérés, ceux qui ont été régulièrement cotés sur la période considérée. Les autres, présentant trop de données manquantes car introduits ou exclus au cours de la période, par exemple à la suite d'une fusion, ont été éliminées. Le travail ne se veut pas exhaustif mais illustratif.

L'objectif principal est la réalisation d'une classification des entreprises au regard du comportement de leur titre au cours de la période et plus particulièrement autour des difficultés rencontrées en 2002 et 2008. Il s'agit donc de déterminer des groupes ou classes homogènes quant à ce comportement. Ce travail est relativement original car absent des analyses financières classiques qui se focalisent sur le comportement d'un titre avec des indicateurs très sophistiqués associés à une série chronologique. L'originalité vient de ce que ce sont toutes les séries qui sont simultanément étudiées. Ce pourrait être le travail préalable d'un analyste qui, voulant constituer un portefeuille « équilibré », chercherait des classes de comportement homogènes afin de compléter une information plus spécifique sur chaque entreprise et leur secteur d'activité. Il ne s'agit donc d'une étude exploratoire sans se préoccuper des qualités prédictives de modèles.

Après un descriptif de l'origine des données, la première partie décrit un traitement préalable des données, la deuxième une description multidimensionnelle par analyse en composantes principales, la troisième, la recherche d'une classification non supervisée, de sa représentation et de son interprétation. La spécificité de l'analyse est qu'il s'agit ici de données temporelles. La même variable est observée à différents instants.

## 2 Données

### 2.1 Présentation

Beaucoup de sites, dont les principaux moteurs de recherche, proposent des historiques des grandes places boursières. Ces historiques fournissent, pour une action, (définie par son code), pour une période donnée et pour chaque jour : le cours d'ouverture, le maximum, le minimum, le cours de fermeture et le nombre de titres échangés.

La première tâche est un travail de compilation afin de récupérer l'ensemble des titres gérés, ici par la bourse de Paris. La deuxième est un filtrage afin de ne conserver que les titres cotés sur toute une période (2000 à 2009). La troisième nécessite un choix de « granularité ». En effet, nous nous intéressons au comportement global sur la période en négligeant les fluctuations du titre à court terme ainsi que la volatilité (la variance) associée. Le choix fait est celui de la simplicité : on s'intéresse aux moyennes mensuelles des cours. Cet indicateur étant relativement "grossier", il n'est plus important de savoir s'il s'agit de la moyenne du cours à l'ouverture, la fermeture, le min ou le max, c'est le montant à l'ouverture qui a été choisi. En complément du calcul de cette moyenne mensuelle, l'opportunité d'un lissage est prise en compte afin de débruiter les données pour mieux se focaliser sur l'analyse des grandes tendances des comportements.

### 2.2 Lecture

Les donnés sont disponibles dans le répertoire "data" : fichier `paris2010.dat`.

```
paris=read.table("paris_10.txt",
   row.names="Nom",header=TRUE,sep=",")
isin=paris[,121] # codes des cours
paris=paris[,-121]
summary(paris)
```

# 3 Traitements préalables

Quels prétraitements semblent indispensables. Les justifier en commentant les graphiques.

```
ts.plot(t(paris))
boxplot(t(paris))
lparis=log(paris)
ts.plot(t(lparis))
lparis0=lparis-apply(lparis,1,mean)
ts.plot(t(lparis0))
```

La réduction des variables semble-t-elle nécessaire ?

On s'interroge sur l'opportunité ou l'intérêt d'un lissage spine pour débruiter les données. La fonction ci-dessous permet de lisser chaque ligne ou séries de cours d'une entreprise.

```
lsm=function(y,spar) {
n <- nrow(y)
p <- ncol(y)
ychap <- y
for(i in 1:n) {
ychap[i, ] <- smooth.spline(1:p,y[i, ],spar=spar)$y
}ychap}
```

Exécuter le lissage pour différentes valeurs du paramètres et représenter une des séries des cours :

```
sm01.lparis0=lsm(lparis0,0.1)
sm04.lparis0=lsm(lparis0,0.4)
sm08.lparis0=lsm(lparis0,0.8)
sm1.lparis0=lsm(lparis0,1)
# Comparaison des graphes plus ou moins lissés
ts.plot(t(rbind(lparis0[100,],sm01.lparis0[100,],
   sm04.lparis0[100,],sm08.lparis0[100,])),
         col=1:4,lwd=2)
legend("bottomleft",legend=c("spar=0","spar=0.1",
   "spar=0.4", "spar=0.8"),col=c(1:4),pch="_")
```

Le choix du paramètre de lissage est basée sur les résultats de l'analyse en composantes principales.

# 4 Analyse en composantes principales

## 4.1 ACP réduite ou non ?

Comparer les résultats ci-dessous.

```
# Acp non réduite
pca=princomp(lparis0)
plot(pca)
biplot(pca)
# ACP réduite
pcar=princomp(lparis0,cor=TRUE)
summary(pcar)
plot(pcar)
biplot(pcar)
boxplot(data.frame(pcar$score[,1:10]))
```

Voir que les représentatinos des individus sont inchangées alors que celle des variables est plus lisible lors de l'acp réduite. C'est ce choix qui est retenu.

## 4.2 Quel lissage

Comparer les résultats afin de choisir à la fois une dimension et le paramètre de lissage.

```
boxplot(data.frame(princomp(lparis0,cor=TRUE)$
   score[,1:10]))
boxplot(data.frame(princomp(sm01.lparis0,
   cor=TRUE)$score[,1:10]))
boxplot(data.frame(princomp(sm04.lparis0,
   cor=TRUE)$score[,1:10]))
boxplot(data.frame(princomp(sm08.lparis0,
   cor=TRUE)$score[,1:10]))
boxplot(data.frame(princomp(sm1.lparis0,
   cor=TRUE)$score[,1:10]))
```

Vecteurs ou plutôt “fonctions propres” de l’ACP retenue.

```
pcars=princomp(sm08.lparis0,Scale=TRUE)
summary(pcars)
plot(pcars)
biplot(pcars)
plot.ts(pcars$loadings[,1:6],
   main="Fonctions propres")
```

S’aider de ce dernier graphe pour interpréter les 3 premiers axes factoriels. Mettre en relation ces interprétations avec les séries de 4 cours remarquables.

```
ts.plot(t(sm08.lparis0[c("Somfy","Fala","Billon",
   "Trigano"),]),gpars=list(col=c(1:4)),lwd=2)
legend("bottomleft",legend=c("Somfy","Fala",
   "Billon","Trigano"),col=c(1:4),pch="_")
```

### 4.3 MDS ou ACP de distances

Comparer la représentation de l’ACP avec celle du positionnement multidimensionnel (MDS) ou ACp du tableau des distances.

```
# centrage et réduction des variables temporelles
actions=scale(sm08.lparis0)
dN=dimnames(lparis0)[[1]]
# distances euclidiennes
d=dist(actions)
# MDS
mdparis= cmdscale(d, k=3)
plot(mdparis, type="n", xlab="", ylab="",main="")
text(mdparis,dN)
abline(v=0,h=0)
```

## 5 Classification des cours

### 5.1 Classification ascendante hiérarchique

Attention, il est important en classification de calculer à partir des variables réduites car cette transformation n’est pas implicite.

```
# CAH
hc.d <- hclust(d,method="ward")
# dendrogramme
plot(hc.d)
# choix du nombre de classes
plot(hc.d$height[252 :240],type="b")
```

Fixer le nombre de classes et couper l’arbre.

```
classif.6a = cutree(hc.d,k=6)
# répartition des actions en classes
sort(classif.6a)
# actions de la 2ème classe
names(classif.6a[classif.6a==2])
```

Représenter avec des couleurs dans les coordonnées du MDS.

```
coul = classif.6a
mds=cmdscale(d,k=3)
# avec les noms des actions
plot(mds, type="n",xlab="Dim1",ylab="Dim2")
text(mds,dN,col=coul)
# avec des points
plot(mds, type="p",col=coul,pch=19,cex=1,
   xlab="Dim1", ylab="Dim2", main="")
```

### 5.2 Algorithme de réallocation

Utiliser l’algorithme kmeans avec 6 classes.

```
km.actions=kmeans(actions,centers=6)
# comparaison des deux classifications
table(classif.6a,km.actions$cluster,
   dnn=c("cah","kmeans"))
```

Les classes obtenues sont relativement différentes. La stratégie suivante consiste à initialiser kmeans avec les barycentres de la CAH précédente.

```
# matrice nulle
mat.init.km.actions=matrix(nrow=6,ncol=120)
```

```
# calcul des barycentres des classes
for (i in 1 :6)
mat.init.km.actions[i,]=
   apply(actions[classif.6a==i,],2,mean)
# kmeans après initialisation
# par les barycentres
km.actions.init=kmeans(actions,centers=
   mat.init.km.actions)
# comparaisons des classifications
table(classif.6a,km.actions.init$cluster,
   dnn=c("cah","kmeans2"))
```

Comparer à nouveau les classes. Que dire de la convergence de l'algorithme kmeans ?

Représentation dans le MDS.

```
coul1 = km.actions.init$cluster
plot(mds, type="n", xlab="Dim1",
   ylab="Dim2", main="")
text(mds,dN,col=coul1)
```

Quelle serait la "meilleure" classification ?

Il est possible d'obtenir les représantations dans les autres plans :

```
plot(mds[,c(1,3)], type="p",col=coul1,pch=pt,
   cex=1, xlab="Dim1", ylab="Dim3", main="")
```

Mais celle-ci n'apporte pas grand chose sur une meilleure représentation des classes qui se discriminent bien dans le premier plan.

Un graphique permet de représenter la bonne homogénéité des classes de courbes.

```
klasse=km.actions.init$cluster
par(mfcol=c(2,3))
ts.plot(t(sm08.lparis0[klasse==1,]),
   ylim=c(-3,3),col=1)
ts.plot(t(sm08.lparis0[klasse==2,]),
   ylim=c(-3,3),col=2)
ts.plot(t(sm08.lparis0[klasse==3,]),
   ylim=c(-3,3),col=3)
ts.plot(t(sm08.lparis0[klasse==4,]),
   ylim=c(-3,3),col=4)
ts.plot(t(sm08.lparis0[klasse==5,]),
   ylim=c(-3,3),col=5)
ts.plot(t(sm08.lparis0[klasse==6,]),
   ylim=c(-3,3),col=6)
```

ainsi que les moyennes des classes.

```
x11() # autre fenêtre
ts.plot(cbind(apply(sm08.lparis0[klasse==1,],
   2,mean),
   apply(sm08.lparis0[klasse==2,],2,mean),
   apply(sm08.lparis0[klasse==3,],2,mean),
   apply(sm08.lparis0[klasse==4,],2,mean),
   apply(sm08.lparis0[klasse==5,],2,mean),
   apply(sm08.lparis0[klasse==6,],2,mean)),
   ylim=c(-2,2),col=1:6,lwd=2)
```

Interpréter les classes obtenues en se basant principalement sur la représentation de l'ACP ou de celle du MDS.

## Annexe : Programme SAS

Ce programme a été utilisé pour convertir les historiques journaliers des cours fournis trouvés sur le web en une matrice (une ligne par entreprise) des cours moyens sur la période considérée.

```
data sasuser.bourse09;
infile 'Historique_bourse_paris-1993-2009.txt'
 dlm='09'x ;
input code $char12. date $ v1 ;
/* se limiter aux cotations à partir de 2000 */
if date < 20000000 then delete;
run;
data sasuser.boursem09 (keep = codem datem n som moy);
```

```
/* Calcul des moyennes mensuelles des cotations */
/* Ne sont stockées que les moyennes pour des mois
*/
/* présentant au moins 10 cotations journalières. */
set sasuser.bourse09 end=fin;
retain  n 0 som 0;
if _N_=1 then do;
   datem=date ;
   codem = code; end;
if fin then do;
   som=som+v1; n=n+1;
   moy=som/n; output; end;
if (substr(date,5,2) EQ substr(datem,5,2))
   AND (codem EQ code) then do;
         som=som+v1;n=n+1;end;
   else do;moy=som/n;
                  if n>10 then output;
        som=v1;n=1;end;
codem=code; datem=substr(date,1,6);
retain codem datem som n;
run;
data sasuser.paris_09 (drop = n som datem code i moy);
/* Formatter les données : une ligne par société */
/* comprenant autant de valeurs que de
   cotations moyennes mensuelles */
/* Limiter aux sociétés cotées sur les 10 ans
  avec donc 120 valeurs ou variables */
array courm{120} (120*0);
set sasuser.boursem09;
retain i 0;
if _N_=1 then code = codem;
if codem NE code then do;
   if i > 119 then output; i=0; end;
i=i+1;
courm{i}=moy;
code=codem;
retain code coursm;
run;
PROC EXPORT
   DATA = sasuser.paris_09
   OUTFILE = 'paris_10.txt'
   DBMS = DLM ;
DELIMITER = ',' ;
RUN;
```