decorrelation

January 8, 2019

1 1A.data - Décorrélation de variables aléatoires

On construit des variables corrélées gaussiennes et on cherche à construire des variables décorrélées en utilisant le calcul matriciel.

```
In [1]: from jyquickhelper import add_notebook_menu
        add_notebook_menu()
```

```
Out[1]: <IPython.core.display.HTML object>
```

Ce TD appliquera le calcul matriciel aux vecteurs de variables normales corrélées ou aussi décomposition en valeurs singulières.

1.1 Création d'un jeu de données

1.1.1 Q1

La première étape consiste à construire des variables aléatoires normales corrélées dans une matrice $N \times 3$. On cherche à construire cette matrice au format numpy. Le programme suivant est un moyen de construire un tel ensemble à l'aide de combinaisons linéaires. Complétez les lignes contenant des . . . . .

```
In [2]: import random
        import numpy as np

        def combinaison () :
            x = random.gauss(0,1) # génère un nombre aléatoire
            y = random.gauss(0,1) # selon une loi normale
            z = random.gauss(0,1) # de moyenne null et de variance 1
            x2 = x
            y2 = 3*x + y
            z2 = -2*x + y + 0.2*z
            return [x2, y2, z2]

        # mat = [ ............ ]
        # npm = np.matrix ( mat )
```

1.1.2 Q2

A partir de la matrice npm, on veut construire la matrice des corrélations.

```
In [3]: npm = ...               # voir question précédente
        t   = npm.transpose ()
        a   = t * npm
        a  /= npm.shape[0]
```

A quoi correspond la matrice a ?

1.1.3 Corrélation de matrices

1.1.4 Q3

Construire la matrice des corrélations à partir de la matrice a. Si besoin, on pourra utiliser le module copy.

```
In [4]: import copy
        b = copy.copy (a)     # remplacer cette ligne par b = a
        b[0,0] = 44444444
        print(b)              # et comparer le résultat ici


        ---------------------------------------------------------------------------

        NameError                                 Traceback (most recent call last)

        <ipython-input-6-e4c5a44882a1> in <module>()
          1 import copy
    ----> 2 b = copy.copy (a)    # remplacer cette ligne par b = a
          3 b [0,0] = 44444444
          4 print(b)             # et comparer le résultat ici


        NameError: name 'a' is not defined
```

1.1.5 Q4

Construire une fonction qui prend comme argument la matrice npm et qui retourne la matrice de corrélation. Cette fonction servira plus pour vérifier que nous avons bien réussi à décorréler.

```
In [5]: def correlation(npm):
            # ..........
            return "....."
```

1.2 Un peu de mathématiques

Pour la suite, un peu de mathématique. On note M la matrice npm. $V = \frac{1}{n}M'M$ correspond à la matrice des *covariances* et elle est nécessairement symétrique. C'est une matrice diagonale si et seulement si les variables normales sont indépendantes. Comme toute matrice symétrique, elle est diagonalisable. On peut écrire :

$$\frac{1}{n}M'M = P\Lambda P'$$

P vérifie $P'P = PP' = I$. La matrice Λ est diagonale et on peut montrer que toutes les valeurs propres sont positives ($\Lambda = \frac{1}{n}P'M'MP = \frac{1}{n}(MP)'(MP)$).
On définit alors la racine carrée de la matrice Λ par :

$$\begin{aligned} \Lambda &= diag(\lambda_1, \lambda_2, \lambda_3) \\ \Lambda^{\frac{1}{2}} &= diag\left(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3}\right) \end{aligned}$$

On définit ensuite la racine carrée de la matrice V :

$$V^{\frac{1}{2}} = P\Lambda^{\frac{1}{2}}P'$$

On vérifie que $\left(V^{\frac{1}{2}}\right)^2 = P\Lambda^{\frac{1}{2}}P'P\Lambda^{\frac{1}{2}}P' = P\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}P' = V = P\Lambda P' = V$.

1.3 Calcul de la racine carrée

1.3.1 Q6

Le module numpy propose une fonction qui retourne la matrice P et le vecteur des valeurs propres L :

```
L,P = np.linalg.eig(a)
```

Vérifier que $P'P = I$. Est-ce rigoureusement égal à la matrice identité ?

1.3.2 Q7

Que fait l'instruction suivante : `np.diag(L)` ?

1.3.3 Q8

Ecrire une fonction qui calcule la racine carrée de la matrice $\frac{1}{n}M'M$ (on rappelle que M est la matrice `npm`). Voir aussi Racine carrée d'une matrice.

1.4 Décorrélation

`np.linalg.inv(a)` permet d'obtenir l'inverse de la matrice `a`.

1.4.1 Q9

Chaque ligne de la matrice M représente un vecteur de trois variables corrélées. La matrice de covariance est $V = \frac{1}{n}M'M$. Calculer la matrice de covariance de la matrice $N = MV^{-\frac{1}{2}}$ (mathématiquement).

1.4.2 Q10

Vérifier numériquement.

1.5 Simulation de variables corrélées

1.5.1 Q11

A partir du résultat précédent, proposer une méthode pour simuler un vecteur de variables corrélées selon une matrice de covariance V à partir d'un vecteur de lois normales indépendantes.

1.5.2 Q12

Proposer une fonction qui crée cet échantillon :

```
In [6]: def simultation (N, cov) :
            # simule un échantillon de variables corrélées
            # N : nombre de variables
            # cov : matrice de covariance
            # ...
            return M
```

1.5.3 Q13

Vérifier que votre échantillon a une matrice de corrélations proche de celle choisie pour simuler l'échantillon.