Algèbre Linéaire et Analyse de Données
Licence 2 MIASHS (2022-2023)

Guillaume Metzler

Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr

Résumé

Ces premiers travaux Pratiques ont pour objectif de vous faire prendre en main le logiciel R via des manipulations simples mobilisant quelques outils présentés en Algèbre Linéaire pour résoudre des problèmes élémentaires mais aussi à vous faire manipuler les objets courants de l'Analyse de Données.

1 Introduction

Le logiciel R, disponible en téléchargement à l'adresse suivante :

http://www.r-project.org/

est un langage de programmation spécifiquement dédiée aux statisticiens mais également à toutes personnes souhaitant pratiquer l'Apprentissage Machine ou l'Analyse de Données.
Il est également utilisé pour la richesse de ses options graphiques.
Son fonctionnement repose essentiellement sur l'utilisation de *packages* ou de *librairies* comme c'est le cas pour Python. Ces *packages* sont créés par la communauté et peuvent être mis à disposition pour l'ensemble de la communauté d'utilisateurs de R.

Bien que R soit utilisé dans un cadre universitaire, il est aussi employé dans le professionnel par certaines entreprises qui utilisent grandement les statistiques dans le cadre de leur activité.
Dans un cadre plus orienté *Machine Learning* qui n'est pas éloigné de l'*Analyse de Données*, R se retrouve parfois mis de côté afin de privilégier l'utilisation de *Python* dont l'usage est croissant parmi les informaticiens.

On va cependant privilégier l'usage de Rdans le cadre de ce cours, pour sa simplicité d'utilisation et surtout pour son usage encore prépondérant par des non informaticiens, *i.e.* sociologues, géographes, journalistes, biologistes, ... à des fins statistiques.

Il est possible d'utiliser R de deux façons différentes :

- en l'installant directement sur votre machine,
- directement en ligne via RStudio Cloud.

Il est disponible sur Mac/Windows/Linux

1.1 R et RStudio

Le logiciel R est un logiciel gratuit et OpenSource. Vous pouvez l'installer gratuitement à l'adresse suivante :

http://www.r-project.org/

Une fois que le logiciel est installé, vous pouvez maintenant l'utiliser. Cette première version est cependant assez "sobre" d'un point de vue esthétique et on préférera souvent installer un deuxième logiciel qui viendra se greffer sur R dont l'usage sera plus agréable et qui fournira une interface plus riche. Pour cela, vous pouvez installer RStudio à l'adresse suivante :

https://rstudio.com/products/rstudio/download/#download

1.2 RStudio Cloud

Dans le cas où vous ne souhaitez pas installer RStudio sur votre ordinateur ou que vus ne l'avez pas en votre possession, il est toujours possible d'accéder aux fonctionnalités de R directement en ligne. Vous pouvez faire cela dans le cadre de ce TP si vous souhaitez ensuite pouvoir avoir accès à votre fichier chez vous.
Une version gratuit de l'utilisation de cette version en ligne peut se faire à l'adresse suivante :

https://rstudio.cloud/plans/free

Vous devrez cependant vous créer un compte pour cela et veillez à bien choisir la création d'un compte **gratuit**.

2 Prise en main

Remarquez que votre interface Rse compose de quatres parties dont vous découvrirez tous les usages au fur et à mesure que vous vous familiariserez avec le logiciel

1. **Supérieure gauche :** c'est l'emplacement de votre script (*i.e.* fichier code) ou vous pourrez **écrire** mais aussi **commenter** votre code et l'enregistrer à toute fin utile.
   Ce code peut directement être exécuté via la commande *run* se trouvant en haut à droite.
2. **Inférieure gauche :** il s'agit de la *console* dans laquelle vous pourrez écrire du code directement et l'exécuter en appuyant sur *Entrée*.
3. **Supérieure droite :** vous y trouverez toutes les informations relatives à votre environnement de travail, notamment les objets que vous avez créés ainsi que leurs valeurs. C'est également ici que vous aurez la possibilité d'importer des jeux de données.
4. **Inférieure droite :** vous pourrez y trouver l'interface graphique de R, *i.e.* l'endroit où sont générés les graphes. Vous pourrez également, dans cette partie là, accéder à l'interface d'aide des fonctions de R ou encore installer les packages dont vous aurez besoin dans le cadre de votre travail.

2.1 Préliminaires

1. Ouvrez un nouveau fichier en allant sur *File* puis *New File*, fichier dans lequel vous écrirez votre code pour cette première séance.

2. Enregistrez votre fichier sous le nom *PriseEnMain* dans un répertoire de votre ordinateur (ou espace). Cela vous permettra de retrouver et de manipuler votre code à n'importe quel moment.
3. Entrer la commande suivante dans la console et l'exécuter

```
# Création d'une variable x
x = 1
```

Que remarquez-vous ?

Exécuter ensuite la commande

```
x
```

La variable x ainsi créée est une variable dite globale à laquelle on a affecté la valeur 1. On peut maintenant réutiliser cette variable aux différentes étapes de notre code.

4. Déterminer la valeur de $(x+3)^5 - 5x^4 + 2 \times x + 2$.

On peut aussi créer des objets plus complexes qui ne contiennent pas qu'une seule valeur.

5. Dans votre script, entrer les commandes suivantes et les exécuter

```
# Création d'une variable x
s = seq(1,10)
r = rep(0,10)
```

Que font ces différentes commandes ?

2.2 Vecteurs et Matrices

On va maintenant se concentrer sur les objets comme les vecteurs et les matrices afin de voir comment effectuer les opérations sous R.

1. On peut créer un vecteur à l'aide de commande suivante

```
# Création d'un vecteur v1
v1 = c(1,2,3)
```

où c désigne la concaténation des éléments $1, 2$ et 3. Créer les vecteurs suivants, on les appellera $v2$ et $v3$

$$\mathbf{v}_2 = \begin{pmatrix} 2 \\ 8 \\ 0 \end{pmatrix} \quad \text{et} \quad \mathbf{v}_3 = \begin{pmatrix} 3 \\ -1 \\ 5 \end{pmatrix}.$$

2. Que font les opérations suivantes ?

```
v1+v2
v1*v2
```

3. Ecrire la somme des vecteurs $\mathbf{v}_2$ et $\mathbf{v}_3$.
4. Déterminer la valeur de $3\mathbf{v}_3$.
5. On peut aussi extraire la composante d'un vecteur à l'aide de la commande suivante

```
# Extraction de la première composante du vecteur v1
v1[1]

## [1] 1
```

Calculer la valeur du vecteur $\mathbf{v}_1 - 3\mathbf{v}_2 + 2\mathbf{v}_3$ et en extraire la deuxième composante

6. Il est possible de concaténer des vecteurs à l'aide de la commande

```
# Concaténation des vecteurs v1 et v2
u = c(v1,v2)
u

## [1] 1 2 3 2 8 0
```

et d'accéder à un sous ensemble de ses composantes, par exemple 2 et 5, par

```
# Extraction des première et cinquième coordonnées
u[c(1,5)]

## [1] 1 8
```

Concaténer les vecteurs $\mathbf{v}_1, \mathbf{v}_2$ et $\mathbf{v}_3$ dans un objet $\mathbf{w}$ et extraire les coordonnées paires de ce vecteur

7. Créer des les vecteurs $\mathbf{u}_1, \mathbf{u}_2$ et $\mathbf{u}_3$ définis par

$$\mathbf{u}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad \mathbf{u}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{et} \quad \mathbf{u}_3 = \begin{pmatrix} 2 \\ -2 \end{pmatrix}$$

Exécuter le code suivant et le commenter. Que peut-on dire des vecteurs $\mathbf{u}_1, \mathbf{u}_2$ et $\mathbf{u}_3$

```
# Création d'un plan et ajout d'une grille
plot(NA, xlim=c(-2,3), ylim=c(-3,2), xlab="x1", ylab="x2")
grid(col = "gray", lty = "dotted", lwd = 1)

# Représentation des vecteurs dans le plan
arrows(0,0,u1[1],u1[2],col="blue")
arrows(0,0,u2[1],u2[2],col="red")
arrows(0,0,u3[1],u3[2],col="brown")
```

8. Représenter graphiquement le vecteur $\mathbf{u} = \mathbf{u}_1 - \mathbf{u}_2 + \mathbf{u}_3$ dans le plan précédent et le colorant en vert.
9. Exécuter les commandes suivantes et comparer les résultats pour la création de matrices

```
V1 = matrix(1:9, nrow = 3)
V2 = matrix(1:9, nrow = 3, byrow = TRUE)
```

Quid si l'on exécute la commande suivante ?

```
matrix(1:9, nrow = 2)
```

10. Exécuter les commandes suivantes et comparer les résultats pour la création de matrices

```
V3 = rbind(v1,v2,v3)
V4 = cbind(v1,v2,v3)
```

11. Exécuter et commenter les opérations suivantes sur les matrices

```
V1+V3
V1-V3
V1*V3
V1%*%V3
3*V1
V1 + v1
```

12. Sachant que la transposée et le déterminant d'une matrice A sont données par les commandes suivantes

```
t(A)
det(A)
```

Calculer le déterminant de la matrice V_1 précédente ainsi que le produit $Z = 4V_1^T V_3$.

13. Enfin identifier les fonctions des lignes de code suivantes

```
V1
V1[1,3]
V1[1,3] = 100
V1[1,3]
V1

V2
```

```r
V2[,2]
V2[3,]
V2[1,] = c(-3,-2,-1)
V2
```

14. On considère l'application linéaire $\theta : \mathbb{R}^2 \to \mathbb{R}^2$ définie par

$$\theta(\mathbf{x}) = \left(\frac{\sqrt{2}}{2}(x_1 - x_2), \frac{\sqrt{2}}{2}(x_1 + x_2) \right).$$

    (a) Donner la matrice O de cette application linéaire et la définir dans R.

    (b) Calculer les images des vecteurs de base $\mathbf{f}_1 = \theta(\mathbf{e}_1)$ et $\mathbf{f}_2 = \theta(\mathbf{e}_2)$ par l'application θ et représenter ces 4 vecteurs dans un plan.

    (c) Que pouvez vous-dire sur l'application θ ?

15. On considère enfin les matrice P et D définie par

$$P = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1/3 \\ -1 & 1 & 1/2 \end{pmatrix} \quad \text{et} \quad D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}.$$

    (a) Définir ces deux matrices sous R.

    (b) La matrice P est-elle inversible ?

    (c) Calculer le produit $A = PDP^{-1}$.

    (d) Que peut-on dire des vecteurs colonnes de P et des éléments de D vis à vis de la matrice A ?

2.3 Systèmes Linéaires

On se propose ici de voir comment on peut résoudre certains systèmes linéaires directement à partir de R. Plus précisément, les systèmes linéaires de Cramer. Pour cela, on considère le système d'équations défini par

$$\begin{cases} x - 2y + 2z &= 3 \\ 3x - 2z &= -7 \\ -x + y + z &= 6 \end{cases}$$

On rappelle que l'on peut également écrire ce système sous la forme matricielle suivante

$$A\mathbf{x} = \mathbf{b}$$

1. Définir la matrice A et le vecteur b associés à ce système et créer les objets correspondants sur R.

2. Montrer que ce système est un système de Cramer.
3. On se propose maintenant de résoudre ce système linéaire. Nous pouvons le faire directement à l'aide de la commande suivante

```
solve(A,b)
```

Résoudre le système.

4. Retrouver les solutions de ce système à l'aide des formules de Cramer.
5. Sachant que la solution d'un système de Cramer $A\mathbf{x} = \mathbf{b}$ est donnée par $\mathbf{x} = A^{-1}\mathbf{b}$ et que la matrice inverse de A est donnée par

```
solve(A)
```

Déterminer une troisième façon de trouver les solutions de ce système avec R.

2.4 Diagonalisation

On va maintenant regarder comment on peut aisément déterminer les valeurs et vecteurs propres d'une matrice sous R.
On considère la matrice B donnée par

$$B = \begin{pmatrix} 2 & 0 & 4 \\ 3 & -4 & 12 \\ 1 & -2 & 5 \end{pmatrix}$$

1. Définir la matrice B dans R.
2. Vérifiez que les vecteurs suivants sont des vecteurs propres de B

$$\mathbf{w}_1 = \begin{pmatrix} -4 \\ 3 \\ 2 \end{pmatrix}, \quad \mathbf{w}_2 = \begin{pmatrix} -4 \\ 0 \\ 1 \end{pmatrix} \quad \text{et} \quad \mathbf{w}_3 = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}$$

3. Exécuter les lignes de codes suivantes et commenter

```
# Réduction d'un endomorphisme
eig  = eigen(B)
eig$values
eig$vectors
```

4. Calculer la norme des vecteurs propres de B. Que constatez-vous ?

2.5 Décomposition en valeurs singulières

On considère la matrice T suivante

$$T = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

1. Créer la matrice T
2. Exécuter et comprendre les commandes suivantes

```
n = nrow(T)
d = ncol(T)
dim(T)
```

3. Exécuter et comprendre les commandes suivantes

```
svdT = svd(T)
svdT
sigma = svdT$d
U = svdT$u
V = svdT$v
```

4. En faisant le lien avec votre cours, dire ce que ce représente les objets $T1$ et $T2$ définis ci-dessous

```
T1=sigma[1]*U[,1]%*%t(V[,1])
T2=T1+sigma[2]*U[,2]%*%t(V[,2])
```

5. Donner les meilleures approximations (au sens de la distance de Frobenius) de rang 3 et de rang 4 de la matrice T. Que remarquez vous en particulier ce qui concerne la meilleure approximation de rang 4.
6. Stocker dans la variable Z, la matrice issue du calcul TT^T. Puis stocker dans la variable *eigZ* le résultat de la décomposition spectrale de cette matrice. Etudier les éléments propres de cette matrice.
7. Faire de même avec la matrice T^TT que vous stockerez dans une matrice Y.
8. Faire le lien entre les éléments propres de X et Y et la décomposition en valeurs singulières de T.

2.6 Un peu de statistiques et de géométrie

Cette dernière section se propose de faire quelques manipulations statistiques sous R afin de préparer le prochain TP.

	Q1	Q2	Q3	Q4	Q5
Individu 1	3	3	3	5	5
Individu 2	2	3	1	5	4
Individu 3	2	3	3	4	5
Individu 4	1	1	1	1	1
Individu 5	5	5	4	3	3
Individu 6	4	5	5	2	3
Individu 7	5	5	5	3	3
Individu 8	1	1	1	1	1

TABLE 1 – Résultats du questionnaire sur un ensemble de 8 individus.

Pour cela, on considère le jeu de données présenté en Table 1. C'est le jeu de données présenté dans l'introduction de ce cours. Pour rappel, pour obtenir ces données, nous avons demandé aux étudiants qui suivent un cours s'ils en sont satisfaits, à travers 5 critères sur lesquels ils doivent se positionner sur une échelle de 1 à 5, 1 correspondant à "très insatisfait" et 5 à "très satisfait". Voici ces 5 critères :

1. Clarté du cours écrit
2. Fluidité de la lecture du cours écrit
3. Facilité à comprendre les exemples du cours
4. Clarté des vidéos
5. Satisfaction vis à vis de l'enseignant dans la vidéo

1. Créer la matrice correspondante à ce jeu de données sous R
2. La fonction *mean* permet de calculer la moyenne des éléments d'un vecteur, (*i.e.* d'un échantillon. Calculer la moyenne obtenue à chaque question
3. La fonction *var* permet de calculer la variance des éléments d'un vecteur, (*i.e.* d'un échantillon. Calculer la variance de l'échantillon associée à la variable
4. à l'aide de la fonction *var* et à l'aide de la définition de la variance. Que constatez vous ?
5. Déterminer la matrice de variance-covariance de votre jeu de données (c'est une matrice de taille 5×5) à l'aide de la fonction *cov*.
   Attention ! R calcule à nouveau la covariance **débiaisée**. Recalculer cette matrice là à l'aide des opérations précédentes
6. Déterminer la matrice de corrélation de votre jeu de données (c'est une matrice de taille 5×5) à l'aide de la fonction *cor*. Recalculer cette matrice là à l'aide des opérations précédentes
7. Calculer la distance entre les individus 1 et 2 puis entre les individus 1 et 6.