UNIVERSITÉ FÉLIX HOUPHOUËT-BOIGNY
ABIDJAN-COCODY

UFR Mathématiques et Informatique

Cours de logiciel R

LICENCE 2

Prof. DOSSO Mouhamadou
Maître de Conférence

E-mail : mouhamadou.dosso@univ-fhb.ci
mouhamadoudoss@yahoo.fr

Table des matières

Chapitre 1

Interface d'utilisation sous windows

1.1 Fonction des menus

Ce logiciel R forme un interface utilisateur simple. Elle est structurée autour d'une barre de menu et de diverses fenêtres.

- Le menu **file** (ou fichier) : contient les outils nécessaires à la gestion de l'espace de travail ;
- Le menu **Edit** (ou Edition) : contient les habituelles commandes de copier-coller et la boîte de dialogue pour la personnalisation de l'apparence de l'interface ;
- Le menu **Misc** traite de la gestion des objets en mémoire et permet permet d'arrêter une procédure en cours de traitement.
- Le menu **Package** : automatise la gestion et le suivi des librairies de fonctions ;
- Les menus **windows** (ou fenêtre) et **Help** (ou Aide) : assurent des fonctions similaires à celles qu'ils occupent dans les autres applications Windows (les fénêtres, l'aide en ligne et manuels de références de R)

1.2 D'autres fenêtres

Parmi les fénêtres, on a la console qui est la fenêtre principale où on réalise par défaut les entrées de commandes et sorties de résultats en mode texte. On y ajoute également un certain nombre de fenêtres telles que les fenêtre graphiques et les fenêtres d'informations (historique des commandes, aide, visualisation de fichier, etc...).
Ces dernières sont toutes appelées à partir de la console par des commandes spécifiques.

Chapitre 2

Les objets : Vecteurs, Matrices, Matrices à plus de deux dimensions, liste et structures de données

2.1 Objets

Les éléments de base du language R sont des objets qui peuvent être des données (vecteurs, matrices,...), des fonctions, des graphiques,...
Les objets se différencient en mode (qui sont :**null (objet vide), logical, numeric, complex, character**) qui décrivent leur contenu et leur classe.
Les principales classes d'objets sont : vector, matrix, array, factor, time-series, data.frame, list
N.B. : On peut avoir des vecteurs, matrices, tableaux, variables,... de mode **null (objet vide)**, **logical**, **numeric**, **complex**, **character**.

2.2 Les vecteurs

C'est l'objet de base dans R.

$>a = c(5, 5.6, 1, 4, -5)$ Création de l'objet a recevant un vecteur numérique de dimension 5 et de cordonnées $5, 5.6, 1, 4, -5$.

$>a$ Affichage du vecteur a

$>a[1]$ Affichage de la première coordonnée du vecteur a

$>b = a[2:4]$ création du vecteur numérique b de dimension 3 et de coordonnées $5, 1, -5$

$>d = a[c(1, 3, 5)]$ Création d'un vecteur numérique d de dimension 3 et de coordonnées $5, 1, -5$

$>sum(d)$ Calcul de la somme de d

$>length(d)$ Affichage de la dimension de d

$>t(d)$ Transposition du vecteur d

$>t(d)\%*\%e$ Produit scalaire entre les vecteurs d et e

2.3 Les matrices

Les matrices comme les vecteurs, sont de modes quelconque, mais elles ne peuvent pas contenir des éléments de nature différente. La syntaxe de création d'une matrice est :

matrix(vec,nrow=n,ncol=p,byrow=T)

où vec est le vecteur contenant, qui seront rangés en colonne (sauf si "byrow=T" est choisie) les éléments de la matrice.

$>a = 1:20$

$>b = sample(1:10, 10)$

$>x1 = matrix(a, nrow = 5)$ Création d'une matrice numérique $x1$ de dimesinsion 5×4 ayant pour prémière ligne $1, 6, 11, 16$

$>x2 = matrix(a, nrow, byrow = T)$ Création d'une matrice numérique $x2$ de dimension 5×4 ayant pour prémière ligne $1, 2, 3, 4$

$>x3 = t(x2)$ Tranposition de la matrice $x2$

$>b = x3\%*\%x2$ produit de matriciel entre $x3$ et $x2$

$>dim(x1)$ Affichage de la dimension de $x1$

2.4 Les matrices à plus de deux dimensions

Les matrices à plus de deux dimensions sont crées à l'aide de la commande suivante :

array(vec,c(n,p,q,...))

où

- **vec** est le vecteur contenant les éléments de la matrice
- **c(n,p,q,...)** désigne les dimensions :
  - **n** est le nombre de lignes,
  - **p** le nombre de colonnes,
  - **q** le nombre de lignes,
  - ⋮

$>x = array(1:50, c(2,5,5))$
$>x$
$>x[1,2,2]$
$>dim(x)$
$>aperm(x)$ Transposition généralisée de **x, x[i,j,k]** devient **x[k,j,i]**

2.5 Les listes

La construction d'une liste passe par la fonction **list(nom1=el1,nom2=el2,...)** , l'utilisation des noms étant étant facultative.

$>li = list(num = 1:5, y = "couleur", a = T)$
$>li$
$>li\$num$
$>li\$a$
$>li[[1]]$
$>li[[3]]$
$>a = matrix(c(6,2,0,2,6,0,0,0,36), nrow = 3)$
$>res = eigen(a, symmetric = T)$ Diagonalisation de a
$>res\$values$
$>res\$vectors$

2.6 Les structures de données

Les tableaux de données (**data.frame**) constituent une classe partitulière de listes consacrée au stockage des données destinées à l'analyse. Pour créer un tableau de données, on peut régrouper les tableaux de même longueur à l'aide de la commande

data.frame(nom1=var1,nom2=var2,...).

On peut ainsi transformer une matrice en tableau de données en utilisant la commande **as.data.frame(mat**

Exemple 1

$>v1 = sample(1:12, 30, rep = T)$ *Echatillonnage avec remise dans les entiers de 1 et 12*

$>v2 = sample(LETTERS[1:10], 30, rep = T)$

$>v3 = runif(30)$ *30 réalisations indépendantes d'une loi uniforme sur* $[0, 1]$

$>v4 = rnorm(30)$ *30 réalisations indépendentes d'une loi normale de moyenne 0 et variance 1*

$>v1; v2; v3; v4$

$>xx = data.frame(v1, v2, v3, v4)$

$>xx$

Chapitre 3

Quelques fonctions usuelles

3.1 Fonctions statistiques

Loi	Nom	Paramètres	Valeurs par défaut
Beta	beta	shape1,shape2	
Binomiale	binom	size,prob	
Cauchy	cauchy	location,scale	0,1
Khi-Deux	chisq	df	
Explonentielle	exp	1/mean	1
Fisher	f	df1,df2	
Gamma	gamma	shape,1/scale	-1
Géométrique	geom	prob	
Hypergéométrique	hyper	m,n,k	
Log-Normale	lnom	mean,sd	0,1
Logistique	logis	location,scale	0,1
Normale	norm	mean,sd	0,1
Poisson	pois	lambda	
Student	t	dt	
Uniforme	unif	min,max	0,1
Weibull	weibull	shape	

Pour chacune de ces distributions, on dispose de quatre commandes préfixées par une des lettres **d,p,q,r** et suivi du nom de distribution :

- **dnomdist** : fonction de densité pour une distribution probabilité continue et de la fonct. de probabilité ($\mathbb{P}(X = k)$) ;
- **pnomdist** :fonction de répartition ($\mathbb{P}(X \leq x)$) ;
- **qnomdist** : fonction de quantité ;
- **rnomdist** :génère des réalisations aléatoires indépendantes de la distribution **nomdist**.

3.2 Quelques fonctions de statistique exploratoire

$>data(women)$
$>names(women)$
$>attach(women)$
$>mean(height)$ Calcul de la moyenne empirique de
variable quantitative **height**
$>var(height)$ Calcul de la varance empirique de **height**
estimateur non biaisé (diviseur $n-1$)
$>sd(height)$ Calcul de l'écart-type de **height**
$>median(height)$ Calcul de la médiane empirique de **height**
$>quantile(height)$ Calcul des quantiles empiriques de **height**
$>summary(weight)$ Résumé de **weight**
$>summary(women)$ Résumé de **women**
$>hist(weight, nclass = 15)$ Histogramme de **weight** constitué de 15 classes
$>cor(height, weight)$ Calcul du coefficient de corrélation linéaire
$>$ empirique entre **weight** et **height**
$>v1 = rnorm(100)$
$>hist(v1)$
$>v2 = factor(saple(letters[1:4], 100, rep = T))$
$>table(v2)$ Résumé de la variable qualitative **v2**
$>barplot(table(v2))$ Diagramme en barre de **v2**
$>sd(height)$ Diagramme en secteur **v2**

Chapitre 4

Construction d'une nouvelle fonction

On peut définir une nouvelle fonction soit directement à partir de la console, soit via un éditeur de texte externe grâce à la commande $fix(nom\_fonction)$. La seconde possibilité permet la correction du code en cours d'édition, tandis que la prémière s'éffectue ligne par ligne, sans retour en arrière.

4.1 Syntaxe générale

La syntaxe générale de la définition d'une nouvelle fonction par l'expression

nom_fonction-function($arg1[-expr1], arg2[-expr2], ...$)
{
bloc d'intructions
}

Les accolades définissent le début et la fin du code source de la fonction ; les crochets ne font pas partie de l'expression mais indiquent le caractère facultatif des valeurs par défaut des arguments. On peut créer également une fonction personnalisée à partir d'une fonction existente grâce

nom_fonction2-edit(nom_function1); fix(nom_fontion2)

Exemple

$>x = 2$
$>carre = function(x)\ \{\ x - x * x;\ x\ \}$
$>carre(2)$
$>x$
$>fix(carre)$

On peut ajouter des commentaire au code en les faisant précéder du symbole #.

4.2 Quelques éléments de programmation

Cette partie concerne les commandes **if**, **while**, **for**

Exemple 2

$>bool = T$
$>i = 0$
$>while(bool == T)\ \{i = i + 1;\ if\ (i > 10)\ \{bool = F\}\}$
$>i$
$>s = 0$
$>for\ (i\ in\ 1 : 1000)\ \{s = s + x[i]\}$
$>s$
$>un = rep(1, 10000)$
$>t(un)\% * \%x$
$>s = 0$
$>system.time(for\ (i\ in\ 1 : 10000)\ \{s = s + x[i]\})[3]$
$>system.time(t(un)\% * \%x)[3]$

N.B. : Le logiciel **R** peut avoir des problème de mémoire de ce fait, il est préférable de les remplacer par les outils de calcul matriciel.

Chapitre 5

Les entrées/Sorties et gestion des objets créés

5.1 Les entrées/Sorties

Les données que l'on souhaite analyser proviennet de source externes sous forme de fichiers. Les objets crées doivent pouvoir être sauve gauvegardés dans des fichiers afin d'être transportables.

- **Formats propriétaire** : **save()** autorise la sauvegarde de n'importe quelle liste d'objet en mémoire ;
- **Fichiers textes ASCII** : ils sont prisent en charge par la commande **scan()**
- **Logiciels statistiques** : La librairie **foreign** offre ces outils pour une sélection des logiciels statistiques plus courants, à savoir MINITAB, S-PLUS,SAS, SPSS et STATS.

5.2 gestion des objets créés

Lors de l'exécution de **R**, les fichiers **.RData** et **.Rhistory** sont automatiquement crées dans le répertoire de travail. Losqu'on quitte **R** à l'aide de la commande **q()**, les nouveaux objets crées peuvent être sauvegardé dans les fichier : **.Rdta** ou **.Rhistory** en mode texte.

- **history()** : permet de visualiser la suite de commandes que l'on a tapée
- **getwd()** : permet de conaitre le répertoire de travail courant au cours d'une session ;
- **ls()** : permet de visualiser la liste des objets crées
- **rm()** : permet de détruire des objets

Chapitre 6

Graphisme avec R

Le fenêtre graphiques sous Windows sont nommées **windows**. On peut ouvrir une fenêtre graphique avec la commande $x11()$.
Pour ouvrir un fichier, on utilise : **postscript("mesgraphiques.eps")** ou **pdf("mesgraphiques.pdf")**.
La commande **dev.list()** permet d'afficher la liste des dispositifs graphiques ouvertes. Cependant les graphiques seront tracés dans le dispositif graphique actif ; ainsi, les commandes (ou fonctions) :

- **dev.cur()** permet de connaître ce dispositif.
- **dev.set(i)** permet de rend le ième dispositif actif
- **dev.off()** ferme le dispositif actif,
- **dev.off(i)** ferme le ième dispositif.

D'autre part, on peut partitionner la fenêtre graphique active à l'aide des commandes **split.screen(x)** (x est un vecteur) ou **layout(m.widths=w.heights)h)** (m matrice, w vecteurs, h vecteur).
Pour visualiser une fenêtre graphique partitionnée avec **layout**, on utilise **layout.show(n)** où n est le nombre de sous-fenêtres.

6.1 Les principales commandes graphiques

Les principales commande graphiques sont données dans le tableau suivant.

plot(x)	Tracer de graphe des valeurs de x ordonnées sur l'axe des abscisses.
plot(x,y)	Tracer le graphe de y en fonction de x
sumflowerplot(x,y)	Idem mais les points superposés sont dessinés sous forme de freurs dont le nombre de pétales correspond au nombre de points
pie(c)	Tracer un graphe en camembert.
boxplot(x)	Tracer le graphe en "boîtes et moustaches" de x.
stripplot(x)	Tracer le graphe des valeurs de x sur une ligne
interaction.plot(f1,f2,x,fun=mean)	Tracer le graphe des moyennes de x en fonction des valeurs des facteurs $f1$ (sur l'axe des abscisses) et $f2$ (plusieurs graphes).
coplot(x$\sim$ y|z)	Tracer le graphe bivarié de x et y pour chaque de x
matplot(x,y)	Tracer le graphe bivarié de la 1ére colonne de x contre la 1ère colonne de y, la 2ème colonne de x contre la 2ème colonne de y...
pairs(x)	Si x est une matrice ou un data.frame, tracer tous les graphes bivariés entre les colonnes de x
plot.ts(x),ts.plot(x)	Tracer le graphe d'une série temporelle x en fonction du temps.
hist(x.freq=T)	Tracer un histogramme des fréquences de x.
barplot(x)	Tracer un histogramme des valeurs de x.
qqnorm(x)	Tracer les quantiles de x en fonction de ceux attendus d'une loi normale.
qqplot(x,y)	Tracer les quantiles de y en fonction de ceux de x.
contour(x,y,z)	Tracer des courbes de niveau.
filled.contour(x,y,z)	Idem mais les aires entre contours sont colorées
image(x,y,z)	Idem mais en couleur.
persp(x,y,z)	Idem mais en 3D
symbols(x,y,...)	Dessiner aux coordonnées données par x et y des symboles (étoiles, cercles, boxplots...).

6.2 Les options

Pour les fonctions, les options sont les suivantes :

axes=TRUE ou FASTE	Si TRUE, les axes et le cadre sont tracé
Tpe="n"."p"."1"."b."h"."s"...	Précisé le type de graphe dessoné. type="n" supprimé le graphe.
col="blue",col.axis,col.main..	Précise la couleur du graphe, des axes, du titre...
bg="yellow"	Précise la couleur du fond.
xlim=c(0,10),ylim=c(0,20)	Précise les limites des axes.
xlab="abscisse",ylab="ordonnée"	Précise les annotations des axes.
main="title"	Précise le titre du graphe.
sub="sstitle"	Précise le sous-titre du graphe.
bty="n","o"...	Contrôle comment le cadre est tracé. btc="n" supprime le cadre
cex=1.5, cex.axis, cex.main...	Contrôle la taille des caractères.
font=1. font.axis...	Précise la police du texte.
las="0"	Contrôle comment sont orientéss le annotations des axes.
lty="1"	Contrôle le type de lignes tracées.
lwd=1.5	Contrôle la largeur des lignes.
pch="+","o"	Contrôle le type de symbole utilisé pour le tracé des points.
pe=1.5	Contrôle la taille en points du texte et des symboles.
tck,tc1	Précise la longueur des graduations sur les axes.
mfcil=c(3,2),mfrow=c(3,2)	Partitionne le graphe en 3 lignes et 2 colonnes. figures sont remplies colonnes par colonnes ou lignes par lignes.

Chapitre 7

Exercices de Travaux Pratiques sous R

Exercice 1

1) Créer les vecteurs suivants :

- y_0 *est constitué de la suite des entiers de* 1 *à* 9
- *On pose* $d = 4$, y_1 *contient trois fois la valeur de* d, *puis trois fois celle de* d^2, *puis trois fois celle de* $\sqrt{d}$,
- y_2 *est une suite arithmétique prenant ses valeurs entre* 1 *et* 20 *avec un pas de deux.*
- y_3 *contient* 10 *chiffres compris entre* 1 *et* 30 *avec un intervalle constant.*

2) Extraite de y_3

- *le* $3^{ième}$ *élément*
- *tous les éléments sauf le* $3^{ième}$
- *tous le éléments inférieurs à* 12

3) Comparer les commandes suivantes

- *matrix(y3,nrow=2)*
- *matrix(y3,nrow=2,byrow=T)*

4) Construire une matrice qui contient sur la première ligne y_10 *et sur la seconde* y_1

Exercice 2

1) Construire la matrice Z *suivante :*

$$Z = \begin{pmatrix} 2 & 4 & 1 & 2 \\ 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 \\ 11 & 12 & 35 & 7 \end{pmatrix}$$

2) Affiche l'élément de Z *contenu dans*

- *La première ligne et toisième colonne*

- *La première ligne de Z*
- *La troisième colonne de Z*
- *la sous-matrice après avoir enlevé la première ligne et la première colonne de Z*

Exercice 3

1) Construire une fonction qui calcule la valeur de la fonction $f : x \mapsto sin(x)^2 + \sqrt{|x-3|}$.

2) Tracer la courbe représentative de la fonction f sur le doamaine $[-6, 3]$

3) reprendre les mêmes questions pour

$$g : x \mapsto \begin{cases} sin(x)^2 log(x) & x > 0 \\ sin(x)^2 x & x \leq 0 \end{cases}$$

Exercice 4

X suit une loi binomiale de pramètres $(50, 1/3)$. Calculer la probabilité des événements suivants

$$[X = 1]\ ;\ [X \leq 5]\ ; [X \geq 15]\ ;\ [X \notin \{15, 3, 4, 10\}]\ ;\ [X \in 2\mathbb{N}].$$

Exercice 5

1) Créer une fonction qui pour un couple donné $(n, p) \in \mathbb{N} \times [0, 1]$, évalue le maximum de l'erreur commise lorsque l'on approche la loi binomiale par la loi de Posson

$$M_{n,p} = \max_{k=0,\ldots,n} |P(X_n = k) - P(Y_n = k)|$$

où X_n suit une loi binomiale de paramètres (n, p) et Y_n suit une loi de poisson de paramètre np.

2) Pour $p = 1/2$, représenter graphiquement l'erreur en fonction de n

3) Pour $n = 40$, représenter graphiquement l'erreur en fonction de p

Exercice 6

Soit X une variable aléatoire qui suit une loi normale standard $\mathcal{N}(0, 1)$. Calculer la probabilité des événements suivants

1) $[X \leq 1]\ ;\ [X \geq 2.6]\ ;\ [0.5 < X \leq 1]$;

2) Calculer le quantile d'ordre $a = 0.75$, c'est-à-dire la valeur de x telle que

$$P(X \leq x) = a$$

3) Représenter graphiquement la densité et la fonction de répartition de la loi de X

4) Simuler un échatillon $(x_1; ...; x_{100})$ de taille $n = 100$ suivant la loi de X

5) Representer les valeurs de l'échatillon simulé.

6) Créer une liste qui contient la moyenne empirique $\frac{1}{100}\sum_{i=1}^{100} x_i$, le minimum de $(x_1; ..., x_{100})$ et le maximum de $(x_1; ..., x_{100})$.

Exercice 7 (Sans utiliser les boucles FOR)
On se donne $x = (x_1, ..., x_n)$ une séquence de longueur n.

1) Construire une fonction qui à pour paramètre d'entrée x et retourne le scalaire

$$S_n = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - (\frac{1}{n}\sum_{i=1}^{n} x_i)^2$$

2) Construire une fonction qui a pour paramètre d'entrée x et retourne le vecteur $(S_1, ..., S_n)$

3) Representer graphiquement $(S_1, ..., S_n)$ pour le vecteur x constitué de nombres aléatoires iid suivant la loi uniforme sur $[0, 1]$.

Exercice 8 (Sans utiliser les boucles FOR)
Soit $N = (N_{i,j})$ une matrice. On note N_i la somme des termes de la i-ème ligne, N_i la somme des termes de la i-ème colonne et n la somme des termes de la matrice. Construire une fonction qui retourne la quantité suivante

$$\sum_i \sum_j \frac{(N_{i,j} - \frac{N_i N_j}{n})^2}{\frac{N_i N_j}{n}}.$$

Exercice 9
On construit une partition de l'intervalle $]0, 1]$ en prenant $\bigsqcup_{i=1}^{p} A_i$ avec $A_1 =]0, a_1]$, $A_1 =]a_{i-1}, a_i]$ pour $i = 2...p-1$ et $A_p =]a_{p-1}, 1]$.

1) En utilisant une boucle **while**, *construire une fonction qui pour un réel donné x et une suite a retourne*

- *s'il existe, l'entier i tel que $x \in A_i$,*
- *un message d'avertissement sinon*

2) Expliquer le code suivant

```
ind = function(x, a)
{
if (x <= 0 | x > 1) stop('x n'est pas dans ]0, 1]')
sum(x - a > 0) + 1
}
```

```
find = function(x, a)
{
sapply(x, ind, a)
}
```

Exercice 10 *Pour* $(n = 100, p = 0.5)$, *puis* $(n = 1000, p = 0.5)$, $(n = 10000, p = 0.5)$, $(n = 1000, p = 0.3)$, $(n = 1000, p = 0.8)$

1. *Simuler un échatillon de n variables aléatoires de Bernouilli, de paramètre p,*
   (a) *en utilisant la fonction* **rbinom**
   (b) *en utilisant la fonction* **runif**
   (c) *en utilisant la fonction* **sample**
2. *Calculer les fréquences de* 0 *et de 1 dans l'échatillon,*
   (a) *en utilisant la fonction* **sum**
   (b) *en utilisant la fonction* **which**
   (c) *en utilisant la fonction* **table**
3. *Représenter les fréquences de* 0 *et de* 1 *par un diagramme en barres (fonction* **barplot***). Représeter par un double diagramme en barre, les fréquences empiriques de* 0 *et de* 1 *en bleu et les probabiltés théoriques* $(1 - p)$ *et* p *en rouge.*
4. *Utiliser votre échatillon pour simuler n parties d'un jeu de pile et face où la probabilité de gagner* 1 *euro est p, la probabilité de perdre* 1 *euro est* $1 - p$. *Calculer les valeurs successives de la fortune d'un joueur dont la fortune initiale est nulle (***fonction cumsum***). Représenter graphiquement ces valeurs (fonction* **plot***).*
5. *Calculer les valeurs successives de la moyenne empirique des i premières valeurs de l'échantillon initial, pour i allant de* 1 *à n. Représenter graphiquement ces valeurs en bleu et superposer sur le même graphique la droite horizontale d'ordonnée p en rouge (fonction* **abline***).*