

Année : 2008-2009 1er semestre
Niveau : MASTER IS 1ère année
Cours : Logiciel R
Enseignant : A. Illig

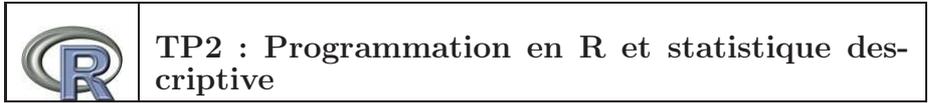


Table des matières

1	Boucles et vectorisation dans R	3
2	Ecrire une fonction, un programme en R	5
2.1	Fonction de R	5
2.2	Organisation du travail	5
3	Exercices	7
3.1	Exercice 1	7
3.2	Exercice 2	7

1 Boucles et vectorisation dans R

R est un logiciel qui permet de programmer simplement l'exécution successive de plusieurs commandes. Comme les autres langages, il dispose de boucles et de structures de contrôle (*if...else, for, while, repeat...*) analogues à celles du langage C. Les quelques exemples suivants illustrent l'usage de ces différentes structures.

Exemple 1. Formons à partir d'un vecteur x un autre vecteur y qui prend la valeur 1 pour les valeurs de x égales à 3 et la valeur 0 sinon :

```
> x=c(3,2,-5,3,4,3)
> y=numeric(length(x))
> for (i in 1:length(x)){
+ if (x[i]==3){
+ y[i]=1}
+ else {y[i]=0}}
```

Exemple 2. Ajoutons les vecteurs x et y pour former un nouveau vecteur z :

```
> z=numeric(length(x))
> for (i in 1:length(x)){
+ z[i]=x[i]+y[i]}
```

Exemple 3. Augmentons de manière aléatoire la valeur d'un réel a jusqu'à dépasser la valeur arbitraire 10 :

```
> a=0
> bool=T
> while (bool==T){
+ a=a+runic(1)
+ if (a>10){bool=F}}
```

Dans ces exemples, on constate que le prompt $>$ de R est remplacé par $+$ après l'ouverture d'une boucle par l'accolade $\{$ et ceci jusqu' à la fermeture de la boucle par l'accolade $\}$.

Dans la plupart des cas, les boucles peuvent être évitées au moyen d'une écriture vectorielle comme dans le cas de l'*Exemple 2.* :

```
> z=x+y
```

ou par une indexation logique comme dans le cas de l'*Exemple 1.* :

```
> y[x==3]=1
> y[x!=3]=0
```

Il est également possible de faire appel à la commande `apply(X,margin,fun)` où X est une matrice, `margin` indique si l'opération doit être appliquée sur les lignes (1), sur les colonnes (2) ou sur les deux (`c(1,2)`) et `fun` est une fonction (c.f. paragraphe suivant) ou une opération saisie entre guillemets doubles :

```
> X=cbind(x,y)
> A=apply(X,2,median)
> B=apply(X,1,sum)
> C=apply(X,2,sum)
> bool2=logical(1)
> bool2[length(B)==length(C)]=T
```

2 Ecrire une fonction, un programme en R

2.1 Fonction de R

Les manipulations en R se font principalement au moyen de fonctions de R dont les arguments sont indiqués entre parenthèses. Tout utilisateur de R peut aussi créer ses propres fonctions qui viendront se rajouter le temps de la session à celles déjà enregistrées dans le logiciel. La fonction suivante permet de calculer l'IMC d'une personne de poids p et de taille t :

```
> IMC=function(p,t){  
+ # Fonction calculant l'IMC d'une personne  
+ # de poids p et de taille t  
+ x=p/t^2  
+ ;x}
```

On appelle ensuite la fonction IMC :

```
> IMC(60,1.7)
```

2.2 Organisation du travail

En pratique, il est préférable d'enregistrer les nouvelles fonctions dans un fichier texte *Nouvelles_fonctions.R* et de l'exécuter ensuite dans R par la commande

```
> source('Nouvelles_fonctions.R')
```

Si les commandes du TP2 enregistrées dans le fichier *tpR2.R* font appel aux fonctions du fichier *Nouvelles_fonctions.R*, il est nécessaire d'exécuter le fichier *Nouvelles_fonctions.R* préalablement au fichier *tpR2.R*.

3 Exercices

3.1 Exercice 1

D'après les définitions introduites dans le TP1, créer (sans faire appel aux fonctions `mean`, `sd`, `quantile`, `min` et `max`) une ou plusieurs fonction(s) R permettant pour une série statistique donnée de :

1. calculer la moyenne,
2. calculer l'écart-type,
3. déterminer les valeurs minimale et maximale (voir la fonction `sort` pour ordonner les composantes d'un vecteur),
4. déterminer les quartiles au moyen de la procédure décrite ci-dessous.

Procédure de calcul du quartile d'ordre q d'une série statistique de taille n :

- Pour $q = 0.25, 0.5, 0.75$, on cherche d'abord la position ind du quartile dans la série ordonnée : $ind = q(n - 1) + 1$.
- On introduit ensuite les indices $i = [ind]$ et $i + 1 = [ind] + 1$.
- Puis, on recherche les valeurs de la série x_i, x_{i+1} d'indices i et $i + 1$.
- Enfin, le quartile est obtenu par interpolation :

$$quant = (x_{i+1} - x_i)(ind - i) + x_i.$$

Cette procédure correspond dans le cas de la médiane à la convention que nous avons faite au TP1. Dans le cas des premier et troisième quartiles, retrouve-t-on la convention de R ?

3.2 Exercice 2

Considérons la série de notes obtenues à un examen de statistiques en L3 :

19.5 12 14 9.5 8.5 15.5 13 19 18 8.5 12 9 17 15.5 1 14 15 5 9 17

Créer un programme R pour :

1. Enregistrer la série de notes dans un vecteur `x` puis créer un vecteur `y` dont les composantes correspondent aux notes ordonnées de manière croissante.

Remarque : Lors de la partie 1 du TP2, vous avez défini des vecteurs `x` et `y` différents de ceux intervenant dans cette question. Pour les supprimer taper dans la fenêtre R la commande

```
> rm(x,y)
```

Plus généralement, pour effacer tous les objets en mémoire :

```
> rm(list=ls())
```

2. Calculer la moyenne, l'écart-type, les quartiles, la note minimale et la note maximale aux moyens des fonctions créées dans l'Exercice 1.
3. Comparer les quartiles obtenus par la fonction `quantile` aux quartiles obtenus par la fonction de l'Exercice 1.
4. Tracer la boîte à moustaches de la série statistique.
5. Tracer un histogramme comportant 4 classes.