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

TP1 : Premières manipulations dans R et introduction à la statistique descriptive

Table des matières

1 Les objets de R

1.1 Objets, modes et longueurs

Le logiciel R manipule plusieurs *objets* : `vector, factor, array, matrix, data.frame, time-series, list` décrits dans le tableau ci-dessous. Chaque objet dispose de deux attributs intrinsèques : *mode*, *longueur*. Le *mode* correspond au type des éléments d'un objet. Il existe quatre *modes* principaux : `numeric, complex, logical, character`.

Objet	Description	Exemples

`vector`	Vecteur au sens classique (éléments de même *mode*)	`> x=c(0,1,-9,7)`
		`> y=c(4 :7)`
		`> z=seq(-3,2,0.2)`
		`> t=rep(x,times=2)`
		`> text=c(‘‘grand’’,’’petit’’)`

`factor`	Variable catégorique (éléments de *modes* identiques : `numeric` ou `character`)	`> f=factor(1 :3,levels=1 :5, exclude=4)`
`array`	Tableau k-dimensionnel (éléments de même *mode*)	`> A=array(1 :20,dim=c(5,4))`
`matrix`	Cas particulier de `array` avec $k = 2$	`> B=matrix(c(1,-1,8,3), nrow=1,ncol=4)` `> C=matrix(0,nrow=8,ncol=2)`
`data.frame`	Tableau de données composé d'un ou plusieurs vecteur(s) et/ou facteur(s) ayant même longueur mais des *modes* pouvant être différents	`> Donnees=data.frame(D1=x, D2=y)` `> Donnees$D1`
`ts`	Données de type série temporelle	`> Serie1=ts(z,start=1942)` `> Serie2=ts(z,freq=12, start=c(1942,3))`
`list`	Liste d'éléments de tout *mode*	`> L=list(x,z,text)`

Les commandes `mode(x)` et `length(x)` affichent le *mode* et la *longueur* de l'objet `x` :

```
> x=c(0,1,-9,7)
> x
> mode(x)
> length(x)
```

Afin de savoir si `x` est un vecteur, une matrice, ... on utilise les fonctions `is.vector(x)`, `is.matrix(x)`, ... :

```
> bool=c(is.vector(x),is.matrix(x))
> bool
```

```
> mode(bool)
> length(bool)
```

Les commandes `ls.()` et `objects()` ont le même rôle et affichent une liste des objets enregistés dans la mémoire courante. La commande `ls.str()` permet d'afficher les différents objets en mémoire et leurs caractéristiques. par exemple, après avoir enregistrer les différents objets du tableau précédent, la commande `ls.str()` affiche :

```
> ls.str()
A :  int [1:5, 1:4] 1 2 3 4 5 6 7 8 9 10 ...
bool :  logi [1:2]  TRUE FALSE
B :  num [1, 1:4] 1 -1 8 3
C :  num [1:8, 1:2] 0 0 0 0 0 0 0 0 0 0 ...
Donnees : 'data.frame': 4 obs. of  2 variables:
 $ D1: num  0 1 -9 7
 $ D2: int  4 5 6 7
f :  Factor w/ 4 levels "1","2","3","5": 1 2 3
L : List of 3
 $ : num [1:4] 0 1 -9 7
 $ : num [1:26] -3.0 -2.8 -2.6 -2.4 -2.2 ...
 $ : chr [1:2] "grand" "petit"
Serie1 :  Time-Series [1:26] from 1942 to 1967: -3.0 -2.8 -2.6 -2.4 -2.2 ...
Serie2 :  Time-Series [1:26] from 1942 to 1944: -3.0 -2.8 -2.6 -2.4 -2.2 ...
t :  num [1:8] 0 1 -9 7 0 1 -9 7
text :  chr [1:2] "grand" "petit"
x :  num [1:4] 0 1 -9 7
y :  int [1:4] 4 5 6 7
z :  num [1:26] -3.0 -2.8 -2.6 -2.4 -2.2 ...
```

Enfin, la commande `rm(x,y)` efface les objets `x` ety. Pour effacer tous les objets en mémoire, on utilise la commande `rm(ls())`.

1.2 Vecteurs, matrices et tableaux

La fonction `c()` est utilisée pour créer un vecteur. Vous pouvez taper

```
> x=c(0,4,7)
```

et afficher le résultat

```
> x
[1] 1 4 7
```

D'autres méthodes peuvent être utilisées pour créer des vecteurs :

```
> # Suites croissantes ou décroissantes d'entiers
> nombresde20a13=20:13
> # Concaténation de vecteurs
```

```
> c(x,nombresde20a13)
> pleindenombres=c(nombresde20a13,4:10)
```

Maintenant, comment extraire certains éléments d'un vecteur ? Voici quelques exemples :

```
> # Le 10 ème élément de pleindenombres
> pleindenombres[10]
> # Certains éléments bien choisis
> pleindenombres[c(1,6,4)]
> # Tous les éléments sauf le 3 ème
> pleindenombres[-3]
> # Tous les éléments sauf les 3 derniers
> pleindenombres[-(length(pleindenombres)-3:length(pleindenombres))]
```

Pour remplir une matrice, on utilise la fonction `matrix()` :

```
> m=matrix(1:6,nrow=2,ncol=3)
> m
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Comme pour les vecteurs, on extrait facilement certains éléments d'une matrice :

```
> m[1,3]
[1] 5
> m[,2]
[1] 3 4
```

Les tableaux jouent le même rôle que les matrices à la différence près qu'ils peuvent avoir plus de deux indices :

```
> a=array(1:12,c(3,2,2))
> a
, , 1

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

, , 2

     [,1] [,2]
[1,]    7   10
[2,]    8   11
[3,]    9   12
```

Afin de savoir si `x`, `m` ou `a` est un vecteur, une matrice, un tableau, ... on utilise les fonctions `is.vector(x)`, `is.matrix(x)`, `is.array(x)` ... :

```
> bool=c(is.vector(x),is.matrix(x),is.array(m))
> bool
> mode(bool)
> length(bool)
```

2 Statistique descriptive : paramètres de position et de dispersion

Soit $x_1 \leq \cdots \leq x_n$ une série statistique quantitative discrète ordonnée. Si la série admet les modalités $y_1 < \cdots < y_m$, on note n_j l'effectif de la modalité y_j et $f_j = \frac{n_j}{n}$ la fréquence de la modalité y_j. La fonction de répartition des fréquences est définie sur $\mathbb{R}$ par :

$$F(y) = \sum_{j \,|\, y_j \leq y} f_j.$$

2.1 Paramètres de position

On définit les paramètres de position : modalité la plus fréquente, moyenne, médiane, quantiles :

- La **modalité la plus fréquente** correspond à la modalité dont l'effectif est le plus grand.
- La **moyenne** est la moyenne arithmétique des données :

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i = \frac{1}{n} \sum_{j=1}^{m} n_j \, y_j$$

  et s'obtient en R par la fonction `mean`.
- Tout réel M_e tel qu'il ait autant de valeurs de la série inférieures ou égales à M_e que de valeurs supérieures ou égales à M_e est appelé **médiane**. Cette définition ne permettant pas de définir la médiane de manière unique dans tous les cas, nous adoptons la convention suivante qui est aussi celle de R :
  - si n est impair $M_e = x_{\frac{n+1}{2}}$,
  - si n est pair $M_e = \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}$.
- On appelle **quantile** d'ordre $\alpha\%$ tout réel z_α tel qu'il y ait au moins $\alpha\%$ de valeurs de la série inférieures ou égales à z_α et au moins $(100 - \alpha)\%$ de valeurs supérieures ou égales à z_α. Les **quartiles** sont les quantiles $Q_1 = z_{25}$, $M_e = z_{50}$ et $Q_3 = z_{75}$.
  Comme dans le cas de la médiane cette définition ne permet pas de définir de manière unique le quantile d'ordre $\alpha\%$. On peut convenir que z_α est la modalité y_j telle que
  $$F(y_{j-1}) < \alpha \leq F(y_j)$$
  ou bien utiliser comme c'est le cas de la fonction `quantile` dans R, l'algorithme de Hyndman & Fan (1996) qui coïncide dans le cas de la médiane avec la convention que nous avons faite.

2.2 Paramètres de dispersion

Les paramètres de dispersion fréquement utilisés sont l'étendue, l'intervalle interquartile, la variance et l'ecart-type :

- L'**étendue** est la différence entre la valeur minimale x_{min} et la valeur maximale x_{max} de la série.
- L'**intervalle interquartile** est la différence $Q_3 - Q_1$.
- La **variance** s_x^2 se calcule selon la formule suivante :

$$s_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 = \frac{1}{n}\sum_{j=1}^{m} n_j\,(y_j - \overline{x})^2$$

  et est obtenue en R au moyen de la fonction `var` qui utilise la normalisation en $n-1$.
- L'**écart-type** est la racine carrée de la variance (voir `sd` en R).

2.3 Commandes R

La fonction `summary` de R appliquée à un vecteur recense nombre de ces paramètres de position et de dispersion. Conçernant la visualisation graphique de ces paramètres, la fonction `boxplot` permet de tracer la boîte à moustaches. A titre d'exemple, voici une petite étude de la distance de freinage des données `cars` :

```
> attach(cars)
> summary(cars)
> boxplot(cars,main=''Boites a moustaches des donnees cars'')
> detach(cars)
```

Ces commandes produisent la figure Fig. 1. Sur le graphique, le trait central d'une boîte représente la médiane, les extrémités de chaque boîte correspondent aux premier et troisième quartiles, la patte supérieure (resp. inférieure) de chaque boîte indique la plus grande donnée inférieure (resp. la plus petite donnée supérieure) à 1.5 fois la distance interquartile mesurée à partir de la médiane et les cercles situent les données aberrantes.

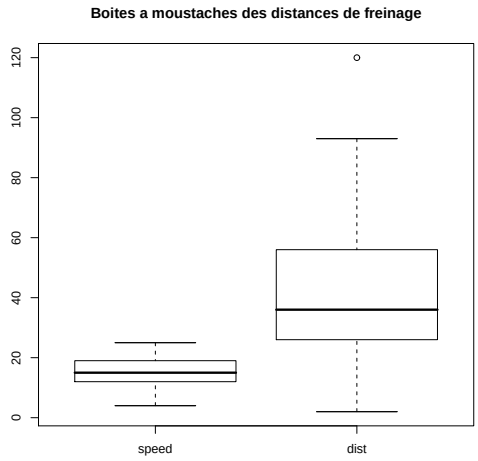

FIG. 1 – Boxplot des données `cars`

3 Indice de Quételet

On s'intéresse à l'indice de Quételet, aussi appelé indice de masse corporelle (IMC) ou body mass index (BMI) :

$$IMC = \frac{poids}{(taille)^2}$$

où la taille est exprimée en mètres et le poids en kilogrammes. Cet indice permet de mesurer la corpulence de l'homme ou de la femme adulte entre 18 et 65 ans au moyen des critères de l'Organisation Mondiale de la Santé (OMS) :

$IMC < 18.5$	Maigreur
$18.5 \leq IMC < 25$	Normal
$25 \leq IMC < 30$	Surpoids
$30 \leq IMC < 40$	Obésité
$40 < IMC$	Obésité majeure

Remarque : Cet indice n'a qu'une valeur indicative. Pour déterminer l'existence d'une obésité réelle, il faut faire d'autres mesures destinées à établir exactement la proportion de masse grasse, car c'est l'excès de masse grasse qui présente un facteur de risque. Pour plus de renseignements, se reporter à la page :

`http://fr.wikipedia.org/wiki/Indice_de_masse_corporelle.`

1. Calculer votre IMC.
2. Enregistrer votre IMC dans `monIMC`.

4 Exercices

4.1 Exercice 1

1. Créer un vecteur `Taille1` comportant les tailles des membres du TD. Faites de même pour la variable poids. Calculer l'IMC des membres du TD et stocker le résultat dans le vecteur `IMC1`.
2. Créer un `data.frame` regroupant toutes ces informations.
3. Calculer les paramètres de position et de dispersion pour le vecteur `IMC1`.
4. Les vecteurs

   ```
   Taille2=c(1.62,1.98,1.73,1.83,1.75,1.83,1.70,1.66)
   ```

   et

   ```
   Poids2=c(54,75,67,62,75,72,58,56)
   ```

   regroupent les tailles et les poids de quelques professeurs de maths de l'Université de Versailles. Créer des vecteurs `Taille`, `Poids` et `IMC` regroupant les informations des deux échantillons.
5. Représenter graphiquement le vecteur `Taille` en fonction du vecteur `Poids` à l'aide de la fonction `plot`.
6. Représenter graphiquement le vecteur `IMC` en fonction du vecteur `Poids` puis du vecteur `Taille`.
7. Refaire la question 3. avec le nouvel échantillon et tracer la boîte à moustaches de `IMC` à l'aide de la fonction `boxplot` (Utiliser préalablement la commande `help(boxplot)`).
8. Tracer l'histogramme de `IMC` au moyen de la fonction `hist` après avoir pris des renseignements sur la fonction `hist`.

4.2 Exercice 2

1. Charger le *package* `MASS` puis les données `survey` de ce *package*.
2. Utiliser les commandes `help(survey)` et `names(survey)`.
3. Attacher les données de `survey`.
4. Tracer un histogramme de chacune des données quantitatives de `survey`.
5. A l'aide de la commande `pie(summary(Sex))` tracer un diagramme circulaire des données du vecteur `Sex`. Faire de même pour les autres données qualitatives.