ASI (L2) : TP7 Fichiers et graphiques sous *Rstat*

Objectifs du TP :

Savoir importer des fichiers-texte et lire des fichiers `.DBF` ; savoir faire des tris à plat et des tris croisés ; enfin, savoir faire des graphiques de courbes et d'histogrammes, des diagrammes de tige et feuille et des boxplots avec *Rstat*.

1. Importer des fichiers avec *Rstat*

1.1 Importation de fichiers-texte

Rstat sait lire les fichiers-texte à l'aide de l'instruction `read.table`. Il y a de nombreuses options, comme

- *header* qu'il faut mettre à $TRUE$ si la ligne 1 du fichier contient le nom des colonnes,
- *skip* qui permet d'ignorer les n premières lignes du fichier,
- *nrows* qui dit combien de lignes on veut lire,
- *blank.lines.skip* qu'il faut mettre à $TRUE$ si on veut ignorer les lignes vides.

Lire le fichier `ELF.DAR` qui contient en ligne 1 le nom des colonnes. Combien y a-t-il de lignes ? de colonnes ? Comment faire pour mettre dans la variable `age` les données de la colonne correspondante du fichier ?

Après avoir lu l'aide sur la fonction `read.table` du logiciel *Rstat* donner l'instruction (unique) qui met dans la variable *cc* la matrice des données associées à la page *Web*

`http://lib.stat.cmu.edu/DASL/Datafiles/cigcancerdat.html`

sachant qu'on peut lire directement cette page *Web* avec *Rstat*, qu'il faut sauter les 35 premières lignes, que les entêtes (noms des colonnes) sont présentes dans le fichier, qu'il y a 44 lignes de données à lire et qu'il faut ignorer les lignes vides. On vérifiera qu'alors la ligne 1 correspond à AL et la ligne 44 à AK.

1.2 Importation de fichiers *Dbase*

Rstat ne lit pas directement les fichiers *Dbase*. Toutefois il n'est pas très difficile d'écrire une fonction `read.dbf` ou d'en trouver une sur Internet. Par exemple, on pourra utiliser celle du fichier `fonctions.r` car l'auteur de cette fonction nous a officiellement autorisés à l'utiliser. L'intérêt principal de cette foncion réside dans le fait qu'elle fonctionne aussi sous *Unix*.

Cette fonction `read.dbf` renvoie toutes les informations de la base de données, sous forme d'une liste et d'un *"dataframe"* – ce que fait aussi `read.table`. On accède aux informations comme *dbf* et *header* en ajoutant le symbole dollar puis le nom de l'information à la variable issue de la lecture. Ainsi

```
mesvaleurs <- read.dbf("elf.dbf")
mesvaleurs$dbf
```

affiche l'ensemble de la base de données `ELF`.

Lire le fichier `VINS.DBF` après avoir chargé avec `source` le texte de la fonction `read.dbf` ; combien y a-t-il de lignes ? de colonnes ? comment faire pour mettre dans la variable `rfa` les données de la colonne correspondante du fichier ?

2. Comptages, tris à plat et tris croisés

On fera toutes les manipulations avec le fichier `ELF.DAR`.

Sachant que la fonction *table* fait le calcul des effectifs absolus, effectuer les calculs suivants :

- le tri à plat de `SEXE` en ligne avec les effectifs absolus,
- le tri à plat de `ETUDE` en ligne avec les effectifs absolus,
- le tri croisé `SEXE / ETUDE` en effectifs absolus.

Comment obtenir le le tri à plat de `SEXE` en ligne avec les effectifs relatifs exprimés en pourcentages entiers ?

Comment nommer les modalités de ce tri à plat ?

Comment avoir un vecteur des effectifs ?

Comment obtenir le tri croisé `SEXE / ETUDE` divisé par les effectifs de `ETUDE` ? et divisé par les effectifs de `SEXE` ?

3. Graphiques de courbes sous *Rstat*

On fera toutes les manipulations avec le fichier `VINS.DBF`.

Sachant que la commande de tracé est *plot*, tracer la colonne `UK` (c'est la numéro 6) directement puis par valeurs triées. On utilisera *sort*.

A l'aide de l'option `main`, ajouter un titre.

A l'aide de `par(mfrow=c(...)` essayer de mettre ces deux graphiques ensemble, verticalement puis horizontalement. Comment revenir à un graphe par page graphique ?

Tracer `UK` en fonction de `RFA` en direct puis trié par ordre croissant. On utilisera la fonction *order*.

Reprendre avec

- des segments de droite seuls,
- des points reliés par des segments de droite,
- des points seuls et la droite de régression linéaire sachant que l'équation est `UK` = 0,5131 * `RFA` + 3903,6 ; on utilisera *abline*.

Comment retrouver ces coefficients avec *lm* ? et ρ ?

Comment tracer la droite de régression en bleu ?

Et mettre les points avec des ronds en orange entourés de vert ?

Voir la page de zoonek pour plus de détails :

`http://zoonek2.free.fr/UNIX/48_R/03.html`

4. Histogrammes de fréquences avec *Rstat*

On fera toutes les manipulations avec le fichier `ELF.DBF`.

La commande *hist* permet de tracer des histogrammes. Essayer d'afficher les histogrammes de `SEXE` en absolu puis ceux de `ETUDE`. Comment faire pour bien séparer les barres (ni trop, ni trop peu) ? Comment afficher en relatif ?

Si on dispose des effectifs comme `eff <- c(35,64)` comment peut-on utiliser *hist* ?

Pour des histogrammes de tri croisés, on utilise *barplot* avec l'option `beside`. Faire tracer les histogrammes à partir des deux tris croisés Sexe / Etude puis Etude / Sexe sans se préoccuper des labels.

5. Tracés en tige et feuille

La commande est *stem*. Donner le graphique pour la variable `AGE` de `ELF` toutes personnes confondues puis pour les hommes seulement et enfin pour les femmes seulement.

6. Tracés de boites à moustaches

La commande est *boxplot*. Tracer pour le dossier `VINS`. Essayer de mettre de la couleur.

Tracer les 10 variables du dossier `LOGEMENT` puis uniquement les mimima puis que les maxima (ce sont respectivement les colonnes impaires et paires).

7. Tracés multiples de courbes

Comment passer en revue toutes les courbes de `VINS` ? de `LOGEMENT` ? On utilisera la fonction *pairs*.

8. Tracés complémentaires (s'il reste du temps)

8.1 Histogrammes probabilistes

Est-il simple de tracer l'histogramme de p_i en fonction des x_i pour la loi binomiale avec n=10 ? Mettre 4 graphiques sur une seule page avec les 4 cas : $p = 0.1$, 0.2, 0.5 et 0.9.

Ajouter la droite de régression à chaque fois, faire une fonction pour que ce soit plus rapide à écrire, mettre la valeur de ρ dans le titre.

8.2 Transposition de dossier et boxplot

La fonction t transpose les données. Comment afficher un `boxplot` des catégories de vins ?

8.3 Couleur de points dans les tracés

Comment afficher un graphique de l'age du dossier `ELF` avec un rond bleu pour les garçons et rose pour les filles ?

SOLUTION

```
# lecture de elf.dar

elf_dar <- read.table("elf.dar",header=TRUE)

# dimensions

dim(elf_dar)
nbl <- dim(elf_dar)[1]
nbc <- dim(elf_dar)[2]

# stockage de l'age (colonne 3)

age <- elf_dar[,3]

# lecture de cigcancerdat.html

cc <- read.table("http://lib.stat.cmu.edu/DASL/Datafiles/cigcancerdat.html",
                          skip=35,header=TRUE,nrows=44,blank.lines.skip=TRUE)
dim(cc)
cc[1,]
cc[44,]

# lecture de vins.dbf

source("fonctions.r")
vins <- read.dbf("vins.dbf")
vins_data <- vins$dbf
dim(vins_data)

# rfa est la colonne 4

rfa <- vins_data[,4]

# au passage, montrer

summary(rfa)
```

```
# tris à plat (effectifs absolus)

sexe <- elf_dar[,2]
etud <- elf_dar[,5]

table(sexe)

table(etud)

table(sexe,etud)

table(etud,sexe)

# tris à plat (effectifs relatifs)

round(100*table(sexe)/length(sexe))

# modalités nommées

tap_sexe <- table(sexe)
rownames(tap_sexe) <- c("Homme","Femme")

# conversion en vecteur

eff <- as.integer(tap_sexe)

# tris croisés en relatif

round(100*table(sexe,etud)/as.integer(table(sexe)))

round(100*table(etud,sexe)/as.integer(table(etud)))

# tracés de courbes élémentaires

vins <- read.dbf("vins.dbf")
vins
vins_data <- vins$dbf
uk <-  vins_data[,6]
uk
```

```
plot(uk)
plot(sort(uk))

# ajout du titre

plot(sort(uk),main="croissant")

# deux graphiques par page en vertical

par(mfrow=c(2,1))
plot(uk)
plot(sort(uk),main="croissant")

# deux graphiques par page en horizontal

par(mfrow=c(1,2))
plot(uk)
plot(sort(uk),main="croissant") par(mfrow=c(2,1))

# on remet un seul graphique par page

par(mfrow=c(1,1))
plot(sort(uk),main="croissant") par(mfrow=c(2,1))

# tracé de uk en fonction de rfa (ordre de lecture du fichier)

rfa <-  vins_data[,4]
plot(rfa,uk)

# tracé trié

idx <- order(rfa)
plot(rfa[idx],uk[idx])

# tracés avec points, lignes et les deux ("both" donc "b")

plot(rfa[idx],uk[idx],type="p")
plot(rfa[idx],uk[idx],type="l")
plot(rfa[idx],uk[idx],type="b")
```

```
# ajout d'une droite

abline(3903.6,0.5131)
abline(coef=c(3903.6,0.5131))

# régression (modèle linéaire donc lm)

lm( uk ~ rfa)

# coefficient de corrélation (linéaire)

cor(uk,rfa)

# un peu de couleur pour la droite et la courbe

abline(3903.6,0.5131,col="blue")
plot(uk~rfa,col="green",bg="orange",cex=1.2,pch=21)

# histogrammes

hist(sexe)

# on gère l'espacement ("breaks" en anglais)

hist(sexe,col="blue")
vmin    <- min(sexe)
vmax    <- max(sexe)
brk     <- c(vmin - 0.75 + 0.5*(0:(2*vmax+2)))
hist(sexe,col="red",br=brk)

# idem pour la variable etud

hist(etud)
vmin    <- min(etud)
vmax    <- max(etud)
brk     <- c(vmin - 0.75 + 0.5*(0:(2*vmax+2)))
hist(etud,col="red",br=brk)

# utilisation des pourcentages

hist(etud,col="red",br=brk,freq=FALSE)
```

```
# une fonction pour ne pas se fatiguer

histo <- function(v) {
   vmin    <- min(v)
   vmax    <- max(v)
   brk     <- c(vmin - 0.75 + 0.5*(0:(2*vmax+2)))
   hist(v,col="red",br=brk)
} ; # fin de fonction histo

# pour les effectifs seuls

histo( rep(1:length(eff),eff) )

# histogrammes de tris croisés

tcr1 <- table(sexe,etud)
barplot(tcr1) # incorrect car empilés
barplot(tcr1,beside=TRUE)

# et dans l'autre sens

tcr2 <- table(etud,sexe)
barplot(tcr2,beside=TRUE)

# tige et feuilles

age <- elf_dar[,3]
stem(age)
agehom <- age[ sexe == 0 ]
stem(agehom)
stem(age[sexe==1])

# boxplots pour vins

nblv <- dim(vins_data)[1]
nbcv <- dim(vins_data)[2]
vins_num <- vins_data[ 1:nblv,2:nbcv]
boxplot(vins_num)
boxplot(vins_num,col="yellow",bg="red",pch=21)
```

```
# boxplots pour logement

loge <- read.dbf("logement.dbf")
loge_data <- loge$dbf
nl <- dim(loge$dbf)[1]
nc <- dim(loge$dbf)[2]
loge_num <- loge_data[1:nl,2:nc]
boxplot(loge_num,col="yellow",bg="red",pch=21)

# que les minimum

boxplot(loge_num[1:nl,1+ 2*0:4],col="yellow",bg="red",pch=21)

# que les maximum

boxplot(loge_num[1:nl,2*1:5],col="yellow",bg="red",pch=21)
```