G. Hunault
Licence Mass

Angers, novembre 2004

Logiciels Statistiques : exercices en Rbase

1. Un remplacement vite fait

La base de données BullQT.dbf est un extrait de la base de données Bull.dbf. Elle contient les 6 variables quantitatives du dossier BULL et l'identificateur des individus interrogés. Parmi ces variables QT, le palier de déverminage est exprimé en heures et en minutes dans deux champs séparés. Convertir au format texte les données à l'aide de descDbf.exe puis créer une variable *DURPAL* à l'aide des variables PALH et *PALM*. Discuter ensuite comment calculer la moyenne du palier, avec les deux variables de départ puis avec la variable DurPal.

2. Calcul sur des données d'une page Web

On trouve à l'adresse

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

des (vieilles) données concernant l'age et le salaire d'un certain nombre d'employés américains de "petites entreprises". Analyser ces QT avec *Rstat* directement puis avec `statgh.r` ; on essaiera de définir et tracer des graphiques "parlants".

3. Comparaison de moyennes, de pourcentages

Donner les instructions *Rbase* qui permettent de calculer la moyenne et la variance de l'age d'abord pour les hommes et ensuite pour les femmes si on utilise les données de la base de données ELF. On pourra utiliser les fonctions du programme statgh.r après avoir converti les données avec descDbf.exe. Au sens de la comparaison de moyennes, y a-t-il une différence à 5 % ?

Effectuer ensuite une comparaison des pourcentages de personnes ayant fait des études supérieures pour les deux modalités de la variable SEXE dans le dossier ELF. On pourra se limiter à une comparaison à 5 %. Effectuer également une comparaison de moyennes pour l'age de hommes ayant fait des études supérieures et des femmes ayant fait des études supérieures dans ce même dossier ELF. On pourra se limiter à une comparaison à 5 %.

Vous n'oublierez pas de conclure par une phrase simple et lisible à chaque fois.

4. Calcul de médiane

Soit X une variable statistique quantitative de taille n et soit x_1, x_2... x_n les différentes valeurs que prend la variable. La médiane m^* de X qui est une mesure dite "de tendance centrale" est la valeur telle que 50 % des valeurs de X sont au-dessus de m^* et 50 % sont au-dessous de m^* pour un nombre impair de valeurs. Pour un nombre pair de valeurs, on utilise la demi-somme des valeurs du milieu de X lorsqu'elles sont rangées par ordre croissant.

Ainsi la médiane de `-1, 2, 3, 4, 15` est `3` et la médiane de `1, 99` est `50`.

Soit U la variable quantitative définie par les $n = 5$ valeurs

```
12 17 14 16 15
```

correspondant à l'étude du poids de matière séche pour un arbre de type pin du Canada, l'unité de mesure étant le gramme.

Donner les expressions Rbase qui calculent les trois quantités $moy(U)$, $ect(U)$, $m^*(U)$ où moy désigne la moyenne, ect l'écart-type et m^* la médiane. Afin de gagner du temps, on pourra utiliser la base de données nommée mstar.dbf ; les variables U, V et W y sont respectivement nommées XU, XV et XW.

Donner ensuite les instructions qui fournissent avec un seul chiffre après la virgule les valeurs numériques associées.

Donner enfin les valeurs numériques correspondantes pour la série théorique V définie par les valeurs `10 11 12 13 14` puis pour la série théorique W définie par les valeurs `10 11 12 11 10 14`.

Pour les plus fort(e)s, écrire un programme median.r qui demande le nom d'une base de données, d'un champ et qui calcule la médiane de ce champ pour la base de données.

Pourquoi peut-on se passer d'écrire un tel programme ?

5. Coefficients d'asymétrie et d'aplatissement

Soit X une variable statistique quantitative de taille n et soit x_1, x_2... x_n les différentes valeurs que prend la variable. On se propose ici de calculer les coefficients d'asymétrie et d'aplatissement de la variable (nommés aussi *skewness* et *kurtosis*) à l'aide de *Rbase*. On note $moy(X)$ la moyenne de X et $ect(X)$ son écart-type mathématique exact défini comme la racine de la quantité $moy(X^2) - moy(X)^2$.

On nomme valeur centrée réduite pour la valeur numéro i notée d_i la quantité

$$d_i = \frac{x_i - moy(X)}{ect(X)}$$

Le coefficient d'asymétrie de X, noté $sk(X)$ et le coefficient d'aplatissement X, noté $ku(X)$ sont alors définis par

$$sk(X) = \frac{1}{n}\sum_{i=1}^{n} {d_i}^3 \quad \text{et} \quad ku(X) = \frac{1}{n}\sum_{i=1}^{n} {d_i}^4$$

Soit U la variable quantitative définie par les $n = 5$ valeurs

```
12 17 14 16 15
```

correspondant à l'étude du poids de matière séche pour un arbre de type pin du Canada, l'unité de mesure étant le gramme.

Donner les expressions Rbase qui calculent $moy(U)$, $ect(U)$, $sk(U)$, $ku(U)$.

Donner ensuite avec un seul chiffre après la virgule les valeurs numériques associées. Donner enfin les valeurs numériques associée pour la série théorique V définie par les valeurs `10 11 12 13 14` et pour la série théorique W définie par les valeurs `10 11 12 11 10`.

Pour les plus fort(e)s, écrire un programme skku.r qui demande le nom d'une base de données, d'un champ et qui calcule les coefficients d'aplatissement et d'asymétrie de ce champ pour la base de données.

Question annexe : A quoi servent ces coefficients ? Comment s'en sert-on ?

Esquisse de SOLUTION

1. Un remplacement vite fait

A l'aide du logiciel *descDbf*, nous exportons les données de la base `BullQT` au format-texte. Voici les premières lignes de ce fichier :

```
IDNUM ANCIENNETE SALARIES COUT PALH PALM DURANS DURPAL
 I01           23     1200    0    4    0     11    240
 I02            9      700    5    0    0     10      0
 I03           17      320    0    0    0     15      0
 I04           30      950    2    2    0     20    120
 I05           15      500    2    1   10     20     70
 I06           10      160    1   20    0     20   1200
 I07           15      100    0    2   30     15    150
 I08            3      280    2    0    0     10      0
 I09           15      350    2   20    0     15   1200
 I10            8    30000    2    1    0      0     60
...
```

La lecture sous `R` de ce fichier se fait avec `read.table` en validant l'option de `header` puisque la ligne numéro 1 du fichier de données contient le nom des colonnes. Les variables `PALH` et `PALM` correspondent alors aux colonnes numéro 5 et 6 du fichier. Le calcul de `DURPAL` se fait comme s'il s'agissait de nombres car `R` sait additionner les vecteurs et les multiplier. Pour tronquer les valeurs réelles, on peut utiliser la fonction `trunc` et pour formater avec un nombre choisi de décimale, il faut utiliser `sprintf`.

Voici donc les instructions qu'il faut taper :

```
###################################

tval <- read.table("bullqt.dat",header=TRUE) ;

PALH         <- tval[,5] ;
PALM         <- tval[,6] ;

###################################

PALH
```

```
PALM
DURPAL <- 60 * PALH + PALM
DURPAL

mean(DURPAL)
c( mean(PALM),  mean(PALH) )
c( mean(60*PALH+PALM), 60*mean(PALH)+mean(PALM) )

trunc( mean(DURPAL) / 60 )
sprintf("%5.1f",mean(DURPAL) )
```

Si nous mettons ces instructions dans le fichier nommé `palier.r` et si nous exécutons la commande `source("palier.r",echo=TRUE)` alors `R` lit le fichier, exécute les commandes une à une en affichant le texte de la commande avant de l'exécuter. On obtient alors :

```
R : Copyright 2004, The R Foundation for Statistical Computing
Version 1.9.0  (2004-04-12), ISBN 3-900051-00-3

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

>   ###################################
>
>   tval <- read.table("bullqt.dat",header=TRUE) ;
>   dims <- dim(tval) ;
>   nbl  <- dims[1] ;
>   nbc  <- dims[2] ;
>
>   PALH         <- tval[,5] ;
>   PALM         <- tval[,6] ;
```

```
>
>   ###################################
>
>   PALH
 [1]  4  0  0  2  1 20  2  0 20  1  4  0  0  1  4 10  2  0 ...
[26] 96  0  2 15  2  1  0  1  1  1  0  0  0  2  2  0  2  5 ...
>   PALM
 [1]  0  0  0  0 10  0 30  0  0  0  0 25 15 30  0  0  0  0 ...
[26]  0  0  0  0  0 30  0 30  0 30  0 10  0 30  0  0 25  0 ...
>   DURPAL <- 60 * PALH + PALM
>   DURPAL
 [1]  240    0    0  120   70 1200  150    0 1200   60  240 ...
[16]  600  120    0    0    0   30    0    0    0  180 5760 ...
[31]   90    0   90   60   90    0   10    0  150  120    0 ...
>
>   mean(DURPAL)
[1] 305.2222
>   c( mean(PALM),  mean(PALH) )
[1] 6.555556 4.977778
>   c( mean(60*PALH+PALM), 60*mean(PALH)+mean(PALM) )
[1] 305.2222 305.2222
>
>   trunc( mean(DURPAL) / 60 )
[1] 5
>   sprintf("%5.1f",mean(DURPAL) )
[1] "305.2"
>
```

Si on veut en automatique exécuter ce fichier et récupérer les résultats dans disons le fichier `palier.sor`, il suffit d'écrire en ligne de commandes :

```
R < palier.r > palier.sor
```

Attention toutefois au chemin d'accès : sous *Windows* comme sous *Unix*, il faut peut-être indiquer le chemins d'accès à R ; sous *Windows* il faut donc sans doute mettre quelque chose comme

```
"C:\Program Files\Rr\rw1090\bin\Rterm.exe"  < palier.r > palier.sor
```

2. Calcul sur des données d'une page Web

Après avoir enregistré la page Web au format texte, on dispose d'un fichier `ceodat.txt` dont voici un extrait :

```
------------------------------------------------------------
*

Datafile Name:

* CEO Salaries *

Datafile Subjects:

*
Story Names:
* CEO Salaries <../Stories/ceo.html> *
Reference:
* Forbes, November 8, 1993, "America's Best Small Companies,".

*
Authorization:
* free use *
Description:
* Small companies were defined as those with annual sales greater than
five and less than $350 million. Companies were ranked according to
5-year average return on investment. This data covers the first 60
ranked firms. *
Number of cases:
* 60 *
Variable Names:
*

   1. Age: Age of chief executive officer
   2. Sal: Salary of chief executive officer (including bonuses),
      $thousands

*
The Data:
*
```

```
AGE SAL

 53   145
 43   621
 33   262
 45   208
 46   362
 55   424
...
 48   572

 (ici la dernière ligne du fichier, après une ligne vide)
```

Il n'y a donc pas que des lignes de données. On fait le "ménage" à la main en supprimant les 40 et quelques premières lignes de façon à ne conserver que le titre des colonnes et les données associés. On obtient alors

```
AGE SAL

 53   145
 43   621
 33   262
 45   208
 46   362
 55   424
...
 48   572

 (ici la dernière ligne du fichier, après une ligne vide)
```

Il ne reste plus qu'à enlever là encore à la main les lignes vides (en particulier en fin de fichier) pour disposer d'un fichier (nommons-le `ceo.dat`) de 61 lignes qui sera lisible par `read.table` dont voici les premières et dernières lignes.

```
AGE SAL
 53   145
 43   621
..
 62   396
 48   572
```

La lecture de ces données se fait par l'instruction

```
ceo <- read.table("ceo.dat",header=TRUE)
```

ce qui permet de récupérer également le nom des colonnes – à vérifier en tapant `names(ceo)`.

A la suite des instructions `age <- ceo[,1]` et `sal <- ceo[,2]` nous diposons en principe des deux variables `age` et `sal`. En principe, car... il y a une donnée manquante : la trentième ligne de données est `47 *` ce qui fait que *Rbase* considère la variable `sal` comme une variable texte et en fait une `QL` avec plein de modalités (indiqués comme *Levels*...). Après avoir modifié le fichier de départ et corrigé la valeur `*` par la valeur `404` (pourquoi ?), nous pouvons refaire l'import avec désormais deux "belles" variables `age` et `sal` de 60 valeurs chacune – ce qu'on vérifie par `length(age)` et `length(sal)`.

La moyenne arithmétique se calcule par `mean` et l'écart-type estimé par `sd`. La fonction qui calcule le coefficient de variation n'existe pas en *Rbase* ; on peut l'inventer (en la nommant `cv`) par

```
cv <- function( x ) { 100* sd(x)/mean(x) }
```

mais une version plus lisible est sans doute

```
cv <- function( x ) {
      paste( sprintf("%5.2f ",100* sd(x)/mean(x)) ," %")
} ; # fin de xv
```

On obtient donc rapidement les résultats suivants, triés à la main par *cv* décroissant :

Variable	*Moyenne*	$Ecart - type$	*Cv*
`sal`	404 k$	219 k$	54 %
`age`	51 ans	9 ans	17 %

Comme premier graphique, nous allons tracer l'age (il ne faut pas oublier de trier les données), avec indication des valeurs m, $m-\sigma$ et $m+\sigma$ respectivement en rouge, vert et vert, soit les instructions :

```
plot(    sort(age), xlab="",ylab="")
abline(  h=mean(age), col="red")
abline(  h=mean(age)-sd(age), col="green")
abline(  h=mean(age)+sd(age), col="green")
```

Pour conserver le graphique sous forme du fichier *Postscript* nommé `ageCeo.ps`, il suffit d'écrire :

```
postscript("ageCeo.ps") ;
```

et de relancer les instructions de tracé avant de fermer le graphique par

```
dev.off()
```

Comme deuxième graphique, nous traçons les salaires en fonction des ages en fonction de l'age trié. Il nous faut pour cela trier en même temps age et salaire. La fonction de *Rbase* nommée `order` produit les indices de permutation correspondant, d'où les instructions :

```
ordAge <- order[ age ]
plot( age[  ordAge ], sal[ ordAge ] )
```

Le tracé des "boites à moustaches" ne doit pas se faire par

```
boxplot( age, sal )
```

car les deux variables n'ont ni les mêmes unités ni le même ordre de grandeur. On peut toutefois afficher les deux graphiques ensemble (à l'horizontale, soit deux colonnes pour une ligne de graphique) via

```
par( mfrow=c(1,2) )
boxplot( age , xlab = "Ages"     , ylan = "ans" )
boxplot( sal , xlab = "Salaires" , ylab = "k$"  )
```

Pour analyser les données avec nos fonctions définies dans `statgh.r`, il faut commencer par lire le fichier afin de les charger soit :

```
source("statgh.r")
```

On peut traiter les deux variables séparément via

```
decritQT( "Salaires" , sal , " k$" )
decritQT( "Age"      , age , "ans" )
```

et on obtient alors :

```
> decritQT( "Salaires" , sal , " k$" )

VARIABLE  Salaires

 Taille                   60 individus
 Moyenne              404.167         k$
 Ecart-type           216.827         k$
 Coef. de variation        54         %
 Minimum                   21         k$
 Maximum                 1103         k$

> decritQT( "Age"      , age , "ans" )

VARIABLE  Age

 Taille                   60 individus
 Moyenne               51.467        ans
 Ecart-type             8.848        ans
 Coef. de variation        17         %
 Minimum                   32        ans
 Maximum                   74        ans
```

Mais on peut bénéficier de l'automatisation de tous les calculs à l'aide de la fonction `allQT` :

```
allQT( ceo , names(ceo) )
```

ce qui fournit notamment comme résultats

```
1 / 2 : Analyse séparée (univariée) ou "à une dimension"

Par cdv décroissant

   Nom   Num   Taille    Moyenne  Ecart-type Cdv   Minimum   Maximum
     2   SAL       60   404.167     216.827  54.1    21.000  1103.000
     1   AGE       60    51.467       8.848  17.3    32.000    74.000

Par ordre d'entrée

   Nom   Num   Taille    Moyenne  Ecart-type Cdv   Minimum   Maximum
     1   AGE       60    51.467       8.848  17.3    32.000    74.000
     2   SAL       60   404.167     216.827  54.1    21.000  1103.000

Par moyenne décroissante

   Nom   Num   Taille   Moyenne  Ecart-type  Cdv   Minimum   Maximum
     2   SAL      60   404.167     216.827   54.1    21.000  1103.000
     1   AGE      60    51.467       8.848   17.3    32.000    74.000

2 / 2 : Analyse conjointe (bivariée) ou "à deux dimensions"

Matrice des corrélations

         AGE    SAL
    AGE  1.000
    SAL  0.127  1.000

Meilleure corrélation

 rho = 0.1272862  pour  SAL  et  AGE

Formule de corrélation

 AGE  =      0.005 * SAL   +    49.367
```

3. Comparaison de moyennes, de pourcentages

La conversion à l'aide du logiciel *descDbf* produit un fichier-texte des données à traiter. Voici les premières lignes de ce fichier :

```
 IDEN  SEXE AGE PROF ETUD REGI USAL
 M001    1  62    1    2    2    3
 M002    0  60    9    3    4    1
 M003    1  31    9    4    4    1
 M004    1  27    8    4    1    1
 M005    0  22    8    4    1    2
 M006    1  70    4    1    1    1
 M007    1  19    8    4    4    2
 M008    1  53    6    2    2    3
 M009    0  62   16    4    2    2
...
```

Afin de profiter des fonctions de statgh.r on commence par charger ce fichier, soit l'instruction `source("statgh.r")` ; on dispose alors des fonctions `compPourc`, `compMoyData` et `compMoyNoData`. La première de ces fonctions effectue la comparaison de pourcentages si on lui fournit comme paramètres les nombres d'individus marqués et globaux, les deux autres effectuent la comparaison de moyennes soit à partir des données soit directement à partir des moyennes et variances.

Pour utiliser ces fonctions il faut prépaper des variables qui correspondent aux hommes, aux femmes, aux sous-populations définies par les études supérieures, soit le fichier complet des instructions :

```
# chargement des programmes (gH)

source("statgh.r")

# lecture des données

tval <- read.table("elf.dar",header=TRUE) ;
dims <- dim(tval) ;
nbl  <- dims[1] ;
nbc  <- dims[2] ;
```

```
# définition des colonnes

SEXE         <- tval[,2] ;
AGE          <- tval[,3] ;
ETUD         <- tval[,5] ;

# création des variables pour l'étude

indHOM <-  SEXE == 0
indFAM <-  SEXE == 1

ageHOM <- AGE[ indHOM ]
ageFAM <- AGE[ indFAM ]

# comparaison des moyennes d'age
# pour les hommes et les femmes

compMoyData( ageHOM, ageFAM )

nbh    <- length(SEXE[ indHOM ])
nbf    <- length(SEXE[ indFAM ])

# comparaison des pourcentages
# d'hommes et de femmes ayant fait des
# étude supérieures

AHes  <- AGE[ indHOM & (ETUD==4) ]
nbHes <- length( AHes)
AFes  <- AGE[ indFAM & (ETUD==4) ]
nbFes <- length( AFes)

compPourc( nbHes, nbh , nbFes , nbf )

# comparaison des moyennes d'age
# correspondantes

compMoyData( AHes, AFes )
```

On obtient alors des résultats très lisibles, à savoir :

```
R : Copyright 2004, The R Foundation for Statistical Computing
Version 1.9.0  (2004-04-12), ISBN 3-900051-00-3

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

> source("statgh.r")

 statgh.r, version 1.33 Novembre 2004

> # lecture des données

> tval <- read.table("elf.dar",header=TRUE) ;
> dims <- dim(tval) ;
> nbl  <- dims[1] ;
> nbc  <- dims[2] ;

> # définition des colonnes

> SEXE         <- tval[,2] ;
> AGE          <- tval[,3] ;
> ETUD         <- tval[,5] ;

> # création des variables pour l'étude

> indHOM <-  SEXE == 0
> indFAM <-  SEXE == 1

> ageHOM <- AGE[ indHOM ]
> ageFAM <- AGE[ indFAM ]
```

```
> # comparaison des moyennes d'age
> # pour les hommes et les femmes

> compMoyData( ageHOM, ageFAM )

 COMPARAISON DE MOYENNES (valeurs fournies)

    Variable    nbVal      Moyenne    Variance     Ecart-type     Cdv
       A          35       36.400      285.365       16.893        46  %
       B          64       35.516      324.984       18.027        51  %

    différence réduite :  0.2431

    au seuil de 5 % soit 1.96, on peut accepter
    l'hypothèse d'égalité des moyennes.


> nbh    <- length(SEXE[ indHOM ])
> nbf    <- length(SEXE[ indFAM ])

> # comparaison des pourcentages
> # d'hommes et de femmes ayant fait des
> # étude supérieures

> AHes  <- AGE[ indHOM & (ETUD==4) ]
> nbHes <- length( AHes)
> AFes  <- AGE[ indFAM & (ETUD==4) ]
> nbFes <- length( AFes)

> compPourc( nbHes, nbh , nbFes , nbf )

 COMPARAISON DE POURCENTAGES

    pop.   A  17  individus marqués sur 35  soit une proportion de  0.486
    pop.   B  22  individus marqués sur 64  soit une proportion de  0.344
    global :  39  individus marqués sur 99  soit une proportion de  0.394

    écart-réduit :  1.3820

    au seuil de 5 % soit 1.96, on peut accepter
    l'hypothèse d'égalité des pourcentages.
```

```
> # comparaison des moyennes d'age
> # correspondantes

> compMoyData( AHes, AFes )

 COMPARAISON DE MOYENNES (valeurs fournies)

    Variable    nbVal      Moyenne    Variance     Ecart-type     Cdv
       A          17       31.235      128.816       11.350        36  %
       B          22       32.591      151.396       12.304        38  %

    différence réduite :  0.3565

    au seuil de 5 % soit 1.96, on peut accepter
    l'hypothèse d'égalité des moyennes.
```

Voici les sources des fonctions programmées :

```
#################################################################

compPourc <- function(ia,na,ib,nb) {

#################################################################

 pa  <- ia / na ;
 pb  <- ib / nb ;
 p   <- (ia+ib)/(na+nb)
 dp  <- pa - pb
 df  <- abs(dp)
 q   <- p*(1-p)*(1/na+1/nb)
 r <- 1
 eps <- 0
 r   <- sqrt(q)
 eps <- df/r

 cat("\n COMPARAISON DE POURCENTAGES\n\n") ;
 cat("    population A,  ",formatC(ia,width=4)," individus marqués sur ",
     formatC(na,width=4),
     " soit une proportion de ",formatC(pa,format="f",digits=3),"\n")
 cat("    population B,  ",formatC(ib,width=4)," individus marqués sur ",
     formatC(nb,width=4),
     " soit une proportion de ",formatC(pb,format="f",digits=3),"\n")
 cat("    globalisation, ",formatC(ia+ib,width=4)," individus marqués sur ",
```

```
   formatC(na+nb,width=4),
    " soit une proportion de ",formatC(p,format="f",digits=3),"\n")
 cat("\n") ;
 cat("    écart-réduit : ",formatC(eps,format="f",digits=4),"\n\n") ;
 cat("    au seuil de 5 % soit 1.96, on peut ")
 if (eps<1.96) {
    cat("accepter")
 } else {
    cat("refuser")
 } ; # fin de si
 cat(" l'hypothèse d'égalité des pourcentages.\n\n") ;


} ; # fin de fonction  compPourc

######################################################################

compMoyData <- function( varA, varB ) {

######################################################################

 # modification au 23 avril 2004 : utilisation de variance estimée
 # et non pas variance exacte (pour suivre sas)

 na <- length(varA)
 ma <- sum(varA)/na
 va <- (sum(varA*varA)/na-ma*ma)*na/(na-1)
 sa <- sqrt(va)
 ca <- round(100*sa/ma)
 wa <- va/na

 nb <- length(varB)
 mb <- sum(varB)/nb
 vb <- (sum(varB*varB)/nb-mb*mb)*nb/(nb-1)
 sb <- sqrt(vb)
 cb <- round(100*sb/mb)
 wb <- vb/nb

 nu <- abs(ma-mb)
 de <- wa+wb
 r  <- sqrt(de)
 delta <- nu/r
```

```
 cat("\n COMPARAISON DE MOYENNES (valeurs fournies)\n\n") ;

 cat("    Variable    nbVal      Moyenne    Variance     ",
     "    Ecart-type     Cdv\n") ;
 cat("       A  ",formatC(na,format="d",width=9),
                  formatC(ma,format="f",width=12,digits=3),
                  formatC(va,format="f",width=12,digits=3),
                  formatC(sa,format="f",width=12,digits=3),
                  formatC(ca,format="d",width=9)," %\n") ;
 cat("       B  ",formatC(nb,format="d",width=9),
                  formatC(mb,format="f",width=12,digits=3),
                  formatC(vb,format="f",width=12,digits=3),
                  formatC(sb,format="f",width=12,digits=3),
                  formatC(cb,format="d",width=9)," %\n") ;

 cat("\n")
 cat("    différence réduite : ",formatC(delta,format="f",digits=4),"\n\n") ;
 cat("    au seuil de 5 % soit 1.96, on peut ")
 if (delta<1.96) {
    cat("accepter")
 } else {
    cat("refuser")
 } ; # fin de si
 cat(" l'hypothèse d'égalité des moyennes.\n\n") ;

} ; # fin de function compMoyData

#####################################################################

compMoyNoData <- function(na,ma,va,nb,mb,vb) {

#####################################################################

 sa <- sqrt(va)
 ca <- round(100*sa/ma)
 wa <- va/na

 sb <- sqrt(vb)
 cb <- round(100*sb/mb)
 wb <- vb/nb

 nu <- abs(ma-mb)
 de <- wa+wb
```

```
r  <- sqrt(de)
delta <- nu/r

cat("\n COMPARAISON DE MOYENNES (valeurs non fournies)\n\n") ;

cat("    Variable    nbVal      Moyenne    Variance     Ecart-type      Cdv\n") ;
cat("       A  ",formatC(na,format="d",width=9),
                 formatC(ma,format="f",width=12,digits=3),
                 formatC(va,format="f",width=12,digits=3),
                 formatC(sa,format="f",width=12,digits=3),
                 formatC(ca,format="d",width=9)," %\n") ;
cat("       B  ",formatC(nb,format="d",width=9),
                 formatC(mb,format="f",width=12,digits=3),
                 formatC(vb,format="f",width=12,digits=3),
                 formatC(sb,format="f",width=12,digits=3),
                 formatC(cb,format="d",width=9)," %\n") ;
cat("\n")
cat("    différence réduite : ",formatC(delta,format="f",digits=4),"\n\n") ;
cat("    au seuil de 5 % soit 1.96, on peut ")
if (delta<1.96) { cat("accepter")
} else { cat("refuser")
} ; # fin de si
cat(" l'hypothèse d'égalité des moyennes.\n\n") ;

} ; # fin de function compMoyNoData
```

4. Calcul de médiane

Comme l'indique la dernière question de cet exercice, ce n'est pas la peine de se fatiguer car `R` connait la notion de médiane : la fonction `median` en assure le calcul. Pour s'en convaincre, il suffit de taper `median( c(1,99) )` pour constater que `R` répond `50` ce qui est bien la médiane de `1` et `99`.

Il n'y a donc pas grand chose à faire si ce n'est de lire les données. Pour une fois, nous n'allons pas convertir avec *descDbf* mais nous allons utiliser la fonction `read.dbf` disponible dans `statgh.r`, l'auteur de cette fonction nous ayant officiellement autorisé à l'utiliser. L'intérêt principal de cette foncion réside dans le fait qu'elle fonctionne aussi sous *Unix*, ce qui n'est pas le cas de *descDbf*.

La fonction `read.dbf` renvoie toutes les informations de la base de données, sous forme d'une liste et d'un *"dataframe"*. On accède aux informations (*dbf*, *header*) en ajoutant le symbole dollar puis le nom de l'information à la variable issue de la lecture. Ainsi

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

affiche l'ensemble de la base de données.

Voici le détail de l'ensemble des informations récupérées par la lecture de mstar.dbf :

```
>   msv <- read.dbf("mstar.dbf")

>   msv
$dbf
  XU XV XW
1 12 10 10
2 17 11 11
3 14 12 12
4 16 13 11
5 15 14 10

$header
$header$file.version
[1] 3

$header$file.year
[1] 1

$header$file.month
[1] 4

$header$file.day
[1] 24

$header$num.records
[1] 5

$header$header.length
[1] 129
```

```
$header$record.length
[1] 10

$header$fields
  NAME TYPE LENGTH DECIMAL
1   XU    N      4       0
2   XV    N      3       0
3   XW    N      3       0
```

Pour calculer la médiane de `XU` qui est la colonne 1 des données, on peut donc se contenter des instructions

```
ms <- read.dbf("mstar.dbf")
XU <- ms$dbf[,1]
median(XU)
```

et si on veut l'étude de la variable avec sa moyenne, son écart-type, on peut ajouter

```
decritQT("Matière sèche",XU,"g")
```

ce qui nous permet d'obtenir :

```
>   decritQT("Matière sèche",XU,"g")

VARIABLE  Matière sèche

 Taille                 5      individus
 Moyenne               14.800          g
 Ecart-type             1.720          g
 Coef. de variation        12  %
 Minimum                   12          g
 Maximum                   17          g
```

5. Coefficients d'asymétrie et d'aplatissement

Nous préférons écrire directement une fonction nommé `skku` que nous mettons dans le fichier `skku.r` pour effectuer tous les calculs :

```
skku <- function( x ) {

  cnt_x <- length(x)
  moy_x <- sum(x)/cnt_x
  mct_x <- sum(x*x)/cnt_x
  var_x <- mct_x - moy_x**2
  ect_x <- sqrt( var_x)
  dix   <- (x-moy_x)/ect_x
  sk    <- sum(dix**3)/cnt_x
  ku    <- sum(dix**4)/cnt_x

  cat(" pour ",cnt_x," valeurs \n")
  cat("   moyenne    ",moy_x,"\n")
  cat("   écart-type ",ect_x,"\n")
  cat("   skewness   ",sk   ,"\n")
  cat("   kurtosis   ",ku   ,"\n")

} ; # fin de skku
```

Son utilisation est immédiate :

```
>   tval <- read.table("mstar.dat",header=TRUE)
>   XU   <- tval[,1]
>   source("skku.r")
>
> skku( XU )

 pour  5  valeurs

   moyenne     14.8
   écart-type   1.720465
   skewness    -0.3958703
   kurtosis     1.994522
```