TD de la 2°séance

Transformer un fichier à partir d'Excel

Transformer un fichier Excel en un fichier texte (CSV avec séparateur point virgule).
Ouvrir le fichier DataTD2R sous Excel
Choisir l'option enregistrer sous/ autre format et choisir alors l'option (CSV avec séparateur point virgule).
Enregistrer ce fichier dans votre répertoire de travail dans "bureau/AtelieR"

Travailler sous R

1. ouvrir R puis un nouveau script pour les commandes et vérifier votre répertoire de travail
getwd()
si votre répertoire de travail n'est pas le répertoire documents/AtelieR alors changer le répertoire de travail avec la fonction setwd()
setwd(dirname(file.choose()))

2. vérifier le contenu et le format du ficher CSV créé - utiliser la fonction file.show()
file.show(file.choose())
Que contient la première ligne?
Quelle notation est utilisée pour les chiffres décimaux?
Quel est le séparateur des colonnes?
Vous noterez que les valeurs manquantes ont été codées en "999" dans ce fichier

3. Créer un objet R qui contiendra les données du fichier CSV. Utiliser la fonction read.table()
Informer R que la première ligne contient le nom des colonnes. Demander d'afficher les noms des variables avec le paramètre header=T ou header=TRUE
Indiquer les données manquantes codées 999 dans le fichier d'origine, utiliser le paramètre na.string="999"
Nous avons vu que le séparateur décimal était une décimale alors que R utilise l'écriture anglo-saxonne avec un point, Utiliser le paramètre dec="," pour lui demander de transformer les virgules en point.
Enfin rappeler que le séparateur des colonnes est un point virgule avec le paramètres sep = "; "

```
Mydata<-read.table(file.choose(), header=T,na.string="999" ,dec=",", sep=";")
```

4. L'objet créé contient-il bien l'information ?
View(Mydata)
head(Mydata)
On peut le vérifier en demandant à connaître le nom des variables de l'objet dataframe Mydata
names(Mydata)
On peut aussi chercher à connaître le nom des lignes
row.names(Mydata)

Si on veut connaître à la fois le nom des lignes et des colonnes
```
Nom<-dimnames(Mydata)
Nom
```

Si on veut vérifier le nom des colonnes, des lignes et le type de données (ici il s'agit d'une table de données) on utilisera la fonction attributes()
```
attributes(Mydata)
```

5. Pour connaître le nombre de lignes et de colonnes de la table de données

```
dim(Mydata) #nombre de lignes et de colonnes

dim(Mydata)[1] # nombre de lignes

dim(Mydata)[2] # nombre de colonnes
```

Voici une autre commande pour obtenir le nombre de colonnes
```
length(Mydata)
```

6. Renommer une variable (une colonne)
```
names(Mydata)[2]<- "DTENAISS" #modifier le nom de la deuxième colonne
names(Mydata)
```

7. Connaître le type des variables d'une table de données
```
mode(Mydata$SEXE)
```

8. Transformer la variable SEXE en facteur
```
summary(Mydata$SEXE) # qu'observer vous?
Mydata$SEXE<- factor(Mydata$SEXE)
summary(Mydata$SEXE) # comparer avec le résultat de la précédente commande
#Créer une variable sexe avec des lettres
Mydata$SEXE2<-ifelse(Mydata$SEXE==1 , "M",ifelse(Mydata$SEXE==2,"F",NA))
```

9. `summary(Mydata)` *#permet de relever quelles variables sont à factoriser et lesquelles sont à garder en format numérique*
`str(Mydata)` *# permet de voir également les variables à factoriser*
A votre avis quelles sont les variables à factoriser

10. Créer un vecteur contenant les numéros des colonnes à transformer
```
Transvar<-c(3,4,5,8:13,16:19)
```

11. Programmer une boucle qui va transformer chacune des variables choisies en variable qualitative
```
for(i in 1:13){Mydata[,Transvar[i]]<-factor(Mydata[,Transvar[i]])}
```

12. Identifier les colonnes en fonction de leur nature : par exemple les variables numériques
```
sapply(Mydata,is.numeric)
```

13. Vous voulez changer le numéro d'identifiant d'un enregistrement par exemple le nom de la ligne 5
On va d'abord conserver la valeur dans un objet R

```
Nomlign5<-row.names(Mydata)[5]
row.names(Mydata)[5]<- "AtR123456"
```

Vérifier la prise en compte de la modification

```
row.names(Mydata)[5]
```

Redonner sa valeur à la ligne

```
row.names(Mydata)[5]<- Nomlign5
```

14. Vous voulez extraire les données correspondant à une variable

```
Colonn3<- Mydata[,3] #extrait les données de la 3ième colonne
View(Colonn3)
```

15. Pour isoler les données d'un enregistrement

```
Lign2<-Mydata[2,] #enregistrement correspondant à la deuxième ligne
View(Lign2)
```

16. voir ou extraire pour l'enregistrement 2 les données de variable de la colonne 3

```
Mydata[2,3]
```

17. Extraire une partie de la table de données dans une nouvelle table de données

a) Méthode 1 : utiliser la fonction subset()

```
Mydata.2<-subset(Mydata,TEMP> 38)
dim(Mydata.2)
```

b) Méthode 2 :

```
Mydata.3<-Mydata[Mydata$TEMP> 38,]
#la virgule après 38 indique que la recherche porte sur les lignes
dim(Mydata.3)
```

18. Autres exemples :
Extraire les 5 premières lignes avec les variables de 2 à 5

```
Mydata.4<-Mydata[1:5, 2:5]
dim(Mydata.4)
Mydata.4
```

Extraire les 5 premières lignes mais avec les variables 1, 3 et 6

```
Mydata.5<-Mydata[1:5, c(1,3,6)]
dim(Mydata.5)
```

Mydata.5
Extraire toutes les lignes (sauf la 3ème), toutes les variables (sauf la 1ère)
Mydata.6<-Mydata[-3, -1]
dim(Mydata.6)

Extraire les 5 premières lignes, toutes les variables sauf les n° 1, 3, et 5
Mydata.7<-Mydata[1:5, c(-1,-3,-5)]
dim(Mydata.7)

19. Création d'une variable
Créer une variable appelée DUREE dans la base de données Mydata. Cette variable calculera la durée de la maladie.
Il faut d'abord transformer les variables DTEDEBUT et DTEFIN en variable date
Mydata$DTEDEBUT<-as.Date(Mydata$DTEDEBUT , '%d/%m/%Y') # ici Y et non pas y car les années ont 4 chiffres
Mydata$DTEFIN<-as.Date(Mydata$DTEFIN ,'%d/%m/%Y')

Mydata$DUREE<-as.numeric(difftime(Mydata$DTEFIN,Mydata$DTEDEBUT,units="days"))

20. Modifier une variable : ceci n'est pas recommandé, on préférera garder la variable d'origine et en créer une nouvelle

a) Methode 1 : utiliser la fonction transform

Mydata<-transform(Mydata,DUREEMOIS=round(DUREE/30,digits=2))

b) Méthode 2 : créer un objet R puis l'insérer dans la table de donnée

Dureemois.2<-Mydata$DUREE/30 ; Dureemois.2<-round(Dureemois.2, digits = 2)

Puis l'ajouter dans le tableau de données

Mydata<-data.frame(Mydata,DUREEMOIS2=Dureemois.2)

names(Mydata)

c) Methode 3: créer directement la variable dans la table de données

#Affecter des valeurs en fonction des données d'une autre variable

Mydata$FEVER2<-ifelse(Mydata$TEMP< 38,0,1)

21.Transformer une variable numérique en classe

Pour cet exercice nous allons d'abord créer une variable AGEAN

Mydata$DTEXAM <-as.Date(Mydata$DTEXAM, '%d/%m/%Y')

```
Mydata$DTENAISS <-as.Date(Mydata$DTENAISS,'%d/%m/%Y')
Mydata$AGEAN<-as.numeric(round((Mydata$DTEXAM-Mydata$DTENAISS)/365.25 , digits=0))
```

utiliser ensuite la fonction cut()

```
Mydata$TRANCHAGE<-cut(Mydata$AGEAN, breaks=c(-0.1,4,9,14,max(Mydata$AGEAN)))
summary(Mydata$TRANCHAGE)
```

22. Retrouver l'emplacement d'un élément d'après sa valeur

```
which(Mydata == "AW" |Mydata== "Q", arr.ind=TRUE)
```

23. Identifier des valeurs aberrantes

Pour cet exercice, nous allons d'abord créer une donnée aberrante
`Mydata[2,7]<-570`
`boxplot(Mydata$TEMP)` #observer les points aberrants sur la figure
Pour les identifier :
`result<-boxplot(Mydata$TEMP)` #affecte les résultats de boxplot dans un objet
`Aberrant<-result$out` # identifie les valeurs aberrantes
`Aberrant` #voir les valeurs aberrantes
`which(Mydata[,"TEMP"]%in%Aberrant)` #pour connaitre le numéro des lignes avec points aberrants

24. Corriger les points aberrants avec la fonction edit() ou la fonction fix()
`fix(Mydata)`
redonner alors la valeur 37 à cette valeur aberrante
on en profitera pour corriger AW en AA et Q en O pour les coordonnées de la question 22

NB : les fonctions edit() et fix() permettent de modifier les données directement sous R

25. Sauvegarder les données modifiées

a) méthode 1

```
save(Mydata, file="Mydata.Rdata")
```

b) méthode 2 : copier dans le presse papier, séparateur est une tabulation

```
write.table(Mydata,"clipboard",sep="\t",dec=",",row.names=F, col.names=T)
```

dans ce cas là on ouvre ensuite Excel et réalise un coller dans Excel

c) méthode 3 : sauvegarder dans un fichier CSV avec séparateur « ; »

```
write.table(Mydata,file="Mydata.txt",sep=";",dec=",",row.names=F, col.names=T)
write.csv2(Mydata,file= "TDR3.csv", row.names=F)
```