

ASI (L2) : TP1 Généralités *Excel* et *Rstat*

Objectifs du TP :

Savoir utiliser Excel et Rstat (entrer, sortir, faire des calculs simples, utiliser les menus), savoir où il y a des aides sur internet.

Généralités à transmettre

Présentation du contenu de l'ensemble des TP à savoir l'utilisation de deux logiciels, *Excel* et *Rstat* pour faire des calculs statistiques.

Nous insistons sur le fait qu'il y a bien d'autres logiciels : citer au moins *StatBox* (présent sur les machines) et *Sas*. Il faut rappeler que le plus important est le fond, la méthodologie et la méthode, les logiciels n'étant considérés que comme "machines à calculer".

La rédaction d'interprétation et de commentaires écrits dans un français soigné après la production des tableaux de chiffres sont très importants en statistiques. Présentation ensuite des pages

<http://www.info.univ-angers.fr/pub/gh/Wstat/>
<http://www.info.univ-angers.fr/pub/gh/Wstat/asi.htm>

Signalons qu'il y a une version *Web* de *Rstat* pour les gens qui ont *Internet* mais qui ne veulent pas installer *Rstat*. Cette version est référencée au bas de la page

<http://www.info.univ-angers.fr/pub/gh/tuteurs/tutrstat.htm>

qui contient des cours et des références de cours pour Rstat.

Des tuteurs *Excel* existent partout dont

<http://www.er.uqam.ca/merlin/xw999998/fr/fexcel.htm>
<http://dj.joss.free.fr/xllien.htm>
<http://www.derochebelle.qc.ca/ticnerveux/excel/e01.htm>

(ne pas taper les liens, ils sont sur la page asi).

Rappel : *OpenOffice* est un tableur gratuit équivalent à *Excel*.

1. Utilisation d'Excel

Expliquer la différence entre

- un tableur dont l'unité de base est la cellule (ou case)
- un gestionnaire de base de données dont l'unité est le champ (ou colonne)
- un langage qui n'a pas d'unité de base mais des variables, des tableaux...

Montrer comme on entre et comment on sort d'*Excel*. Indiquer que *.xls* est par défaut le type pour *Excel*.

Rappeler les composants d'une fenêtre :

- barre de titre
- barre de menu
- barre d'outils (icônes)
- barre des onglets
- barre d'état (ou de status, en bas)

Montrer comment on adresse une cellule (les deux styles en ligne/colonne et bataille navale) avec utilisation du menu Outils / Options onglet général et encoche nommé Style de Référence L1C1.

Exemples de questions :

- à quel nombre correspond la lettre Z ?
- qu'y-at-il après Z ?

Montrer ce qu'est une liste de cellule séparées par des points-virgules et ce qu'est une plage simple comme `B1:C12\verb`.

Montrer comment on calcule dans une cellule :

- on entre la valeur numérique
- on entre la valeur texte
- on écrit = et on fait un calcul
- on écrit = et on utilise une fonction comme somme.

Montrer comment on référence une cellule en relatif et absolu comme B2 et `=$2\verb`. Exemple classique : conversion francs/euros d'une liste de valeurs, le taux est dans une case spéciale.

Montrer comment on formate une cellule : ajout, retrait de décimales, couleur pour négatif. Montre que les cases 0.3 et 0.3 affichées sans décimales valent 0 et que leur somme affichée sans décimales vaut 1 d'où prudence car $0 + 0$ ne fait pas 1.

Montrer que 1 est cadré à droite et "1" est cadré à gauche car c'est une chaîne de caractères.

Montrer comment on formate une ensemble de cellules avec le menu Format et la mise en forme automatique. Montrer comment on utilise la poignée de recopie, qu'elle recopie formules et format. Montrer la liste des fonctions sans détailler mais leur dire qu'ils peuvent s'entraîner à essayer...

Montrer comment on ajoute, supprime une ligne, une colonne, comment on annule une modification.

Expliquer que copier/coller copie la formule mais qu'avec collage spécial on peut faire d'autres choses (copie seule des valeurs et transposition notamment).

Montrer comment on ajoute, supprime, renomme un onglet. Montrer la notation `Onglet!A1` etc. pour écrire des formules avec des cellules venant de divers onglets.

Montrer comment on trie des données (trier une colonne, trier un bloc de colonnes et montrer la différence).

Si l'heure n'est pas passée : montrer les menus, leur faire réaliser une table de multiplication de la cellule A1 sous forme d'un tableau bien beau, bien propre, bien formaté.

2. Utilisation de Rstat

Montrer comme on entre et comment on sort sous *Windows*. Indiquer que *R* fonctionne aussi sous *Linux* (machine *Deneb*, commande : *R*.)

Faire écrire `2+3 <enter>`.

Montrer les menus (dont "Clear console") et l'aide nommée *Html help*.

Indiquer qu'il n'y a pas de type par défaut pour les fichiers de sortie mais qu'on peut garder une trace de ce qui a été fait en `.txt` (car ce sera lisible par *Notepad* avec un double-clic dans l'Explorateur de *Windows*).

Signaler que tout ce qu'on écrit peut être mis dans un fichier `.r` et chargé ensuite via le menu `File / Source R code` ou en ligne de commande par `source("...")` – ne pas faire de manipulation mais le dire seulement et faire une manipulation lorsqu'il auront vu quelques expressions en *R* (c'est rappelé un peu plus loin).

Montrer un calcul direct puis l'affectation.

Montrer l'arithmétique classique avec `+` `-` `*` `/`. Montrer la puissance avec `**` ou `^` et montrer que `3.2 ** 2.9` existe. Leur faire écrire et tester des expressions comme

```
a <- 17/3
a
round(a)
round(a,3)
n <- 10
n
5 -> x
n + x
b <- 1e5
b
b <- 10**6
b
# montrer le symbole # comme commentaire
a <- 2
a
# a <- 3
a
ls() # et ls.str()
```

Les faire chercher comment on dit *modulo* et *reste_entier*. Par exemple, quel est le quotient de 178 divisé par 9 et son reste entier ? Utiliser `help("%%")`.

Montrer comment fonctionne l'aide en ligne et l'aide *Web*. Indiquer la carte de référence (page asi). Utiliser ensuite `help.start()`.

Chercher dans la page *Web* affichée **Search Engine & Keywords** puis aller sur le menu **Mathematics** ; cliquer sur **arith** et à nouveau sur **Arithmetic** dans la page des résultats.

Attention à l'erreur classique qu'on leur fera faire :

```
a <-- 2
```

interprétée comme

```
a <- (-2)
```

Montrer le formatage :

```
n <- 35
n
print(n)
format(n)
x <- 171/7
format(x)
print(x,digits=1)
format(x,digits=1)
format(x,decimal.mark=",")
sprintf("%5.1f euros",x)
```

Montrer la fonction `seq` et son raccourci comme `1:n`.

```
1:10
a <- 5
b <- 11
a:b
n <- 10
seq(1,n)
help(seq)
v <- seq(from=1.1,by=0.3,length=5)
ls.str()
```

Montrer les vecteurs avec `c()` et les fonctions attenantes, tester l'arithmétique des tableaux :

```
w <- c( 1,50,8,6,11) # attention : il faut meme dimension que v
length(v)
length(w)
v + w
2*v + 1
w / 10
v/w
v[1]
v[1:3]
v[-2]
v[ c(1,3) ]
v[ - (2:4) ]
```

Présenter la fonction `sum`. Faire des essais avec `sum` et `seq`.

```
v
sum(v)
v*v
sum(v*v)
sum(v*w) # c'est le produit scalaire
n <- 5
sum(1:n) # c'est n*(n+1)/2
sum( (1:n) **2 ) # formule ?
```

Montrer comment on formate et arrondit un vecteur, comment on le transpose avec `t()`.

```
v <- (1:10) / 7
v
print(v,digits=2)
format(v,digits=2)
dim(v)
dim( t(v) )
length( v )
length( t(v) )
```

Reprendre `c()` avec des chaînes de caractères. Utiliser avec `rep()` :

```
p <- c( "oui","non")
length(p)
rep( p , c(10,5) ) -> v
length(v)
```

Mettre avec *Notepad* deux ou trois instructions dans un fichier. Montrer `source()` avec ses options et discuter sur la permanence des variables.

Montrer comment on écrit une fonction courte comme

```
somme <- function( x ) { sum(x) }
```

Reprendre l'exemple classique de conversion francs/euros d'une valeur, d'une liste de valeurs. Parler de variables globales, locales, paramètres.

```
# rm(x) pour enlever x ; rm(x,f) pour enlever x et f
f <- function(x) { x**2 + 1 }
f(1)
f(2)
f( c(1,8) )
f( 1:10 )
```

S'il reste du temps : montrer les sélections par test logique comme :

```
v <- c( 1,50,8,6,11,8)
v < 10
v[ v < 10]
v == 8
w <- c("oui","non","non","oui","oui","non")
w == "oui"
v[ v < 10 ] # 1 8 6 8
v[ w == "oui"] # 1 6 11
v[ (v < 10) & (w == "oui") ] # 1 6
```