

Comment réaliser un programme

Un des gros avantages d'un logiciel comme R, outre les très nombreuses fonctions et options qu'il offre, est la possibilité de réaliser des programmes dans un langage très accessible.

Il faut d'abord se rappeler qu'un programme (désigné par *function* sous R) porte un nom et permet de réaliser un traitement dépendant de paramètres. Ces paramètres sont transmis en arguments dans la fonction et mis entre parenthèses, séparés les uns des autres par une virgule. Le programme commence par { et se termine par }. Ce n'est pas plus compliqué que cela. Bien évidemment, ce programme sera introduit dans un script R.

Ainsi, pour créer un programme nommé «conversion», réalisant une simple conversion en euros d'une valeur exprimée en dollar :

(pour plus de clareté, en bleu figurent les instructions définissant le programme, en rouge, les instruction d'appel du programme, en magenta les autres instructions et en noir tous les commentaires commençant par #)

conversion <- function(valeur)

#

Fonction réalisant la conversion en euros d'une valeur exprimée en dollar et affichant le
résultat.

#

{

valeur=valeur*1.50 # taux de conversion au 22 nov 2007

valeur

}

#

Il s'agit maintenant d'appeler la fonction pour réaliser la conversion

#

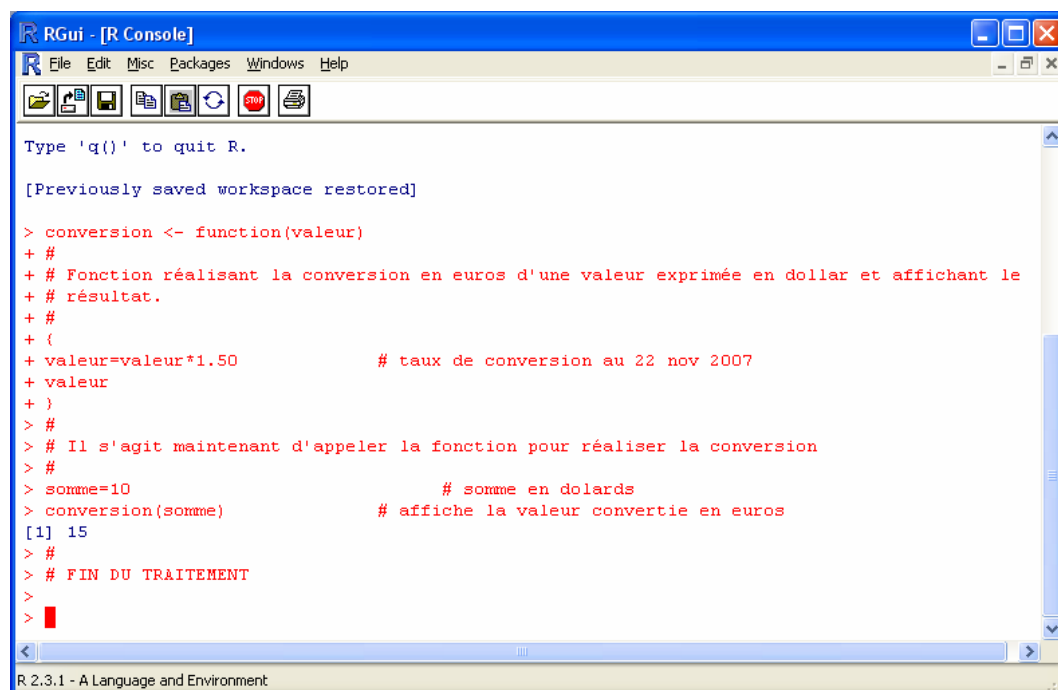
somme=10 # somme en dolards

conversion(somme) # affiche la valeur convertie en euros

#

FIN DU TRAITEMENT

Cette façon d'utiliser le logiciel R permet de gagner du temps et de simplifier l'écriture des scripts (évitant la répétition inutile de lignes qui font le même traitement mais avec des valeurs différentes).



```
RGui - [R Console]
File Edit Misc Packages Windows Help
Type 'q()' to quit R.
[Previously saved workspace restored]

> conversion <- function(valeur)
+ #
+ # Fonction réalisant la conversion en euros d'une valeur exprimée en dollar et affichant le
+ # résultat.
+ #
+ {
+   valeur=valeur*1.50          # taux de conversion au 22 nov 2007
+   valeur
+ }
> #
> # Il s'agit maintenant d'appeler la fonction pour réaliser la conversion
> #
> somme=10                      # somme en dolards
> conversion(somme)             # affiche la valeur convertie en euros
[1] 15
> #
> # FIN DU TRAITEMENT
>
R 2.3.1 - A Language and Environment
```

Voici un nouveau programme qui permet de tracer les courbes de 3 lois normales sur une même page graphique divisée en 3 zones indépendantes :

(en bleu figurent les instructions définissant le programme, en rouge, les instructions d'appel du programme, en magenta les autres instructions et en noir tous les commentaires commençant par #)

```
plot_normal_law <- function(min,max,mu,sigma,color)
#
# Fonction générique dessinant le graphe d'une loi normale
# min et max contiennent les limites de la VA (axe des abscisses)
# mu et sigma sont les arguments recevant la moyenne et l'écart type de la distribution
# enfin, color contient la chaîne de caractère qui définit la couleur souhaitée pour le graphe
#
{
x <- seq(min,max, length=1000)
fx <- dnorm(x,mean=mu,sd=sigma)
plot(fx~x, type='l',col=color)
}
#
# Plot en k frames de k lois Normales, ici k=3
#
nblaws=3
par(mfrow=c(1,nblaws))
#
# Il s'agit en fait d'appeler 3 fois la même fonction, en changeant convenablement les
# paramètres
plot_normal_law(0,6,3.4,0.5,"blue")
plot_normal_law(0,6,3.2,0.5,"red")
plot_normal_law(-2,2,0.2,0.5,"green")
#
# FIN DU TRAITEMENT
```

Pour essayer ces traitements : copier ces lignes puis les coller dans R

... A vous de jouer maintenant !

