



VI. Initiation à la programmation sous R

Comme nous l'avons constaté tout au long du document, R offre de nombreuses fonctionnalités. En tant que nouvelles utilisateurs de R il vous est désormais PRESQUE possible de participer à l'évolution de R en proposant des nouveaux outils d'analyse. Il faut avant cela revenir au langage et à la programmation. Nous allons voir des idées simples et susceptibles d'être mises en pratique aisément.

Le point fort de R par rapport à un logiciel à menu déroulant type SPSS est dans la possibilité de programmer, de façon simple une suite d'analyses qui seront exécutées successivement. Cette possibilité est propre à tout langage informatique, mais R possède des particularités qui rendent la programmation accessible à des non-spécialistes.

VI.1 Boucles et vectorisation

La boucle for

```
for (i in 1 : n)
{
  instruction(s)
}
```

La boucle while

```
while (condition)
{
  instruction(s)
}
```

L'utilisation du if – else

```
if (condition)
{
  instruction1(s)
}
else
{
  instruction2(s)
}
```

Remarque □ On peut exécuter plusieurs instructions si elles sont encadrées par des accolades.

```
y = rnorm(20); y
[1] 1.13088474 -0.32506118 -0.21300185 -0.40149019 -0.73160051 -0.76896992
[7] 0.20388507 1.72975780 0.58476297 1.34209095 -0.58889570 -0.33435842
[13] -0.77073270 1.29515298 -0.66902960 0.43934088 -0.06785511 -0.41360195
[19] 1.39177097 -1.87708783
```

```
for (i in 1:length(y))
{
  if (y[i] < 0)
    y[i] = 0
  else
    y[i] = 1
}
```

```
y
[1] 1 0 0 0 0 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0
```

Il est préférable d'éviter, les boucles et/ou les structures de contrôle. Ceci est rendu possible grâce à une caractéristique du langage R □ «La vectorisation». Ainsi, les boucles et structures de contrôle de l'exemple précédent peuvent être évité de la manière suivante □

```
x = y; x[y < 0] = 0; x[y >= 0] = 1
x
[1] 1 0 0 0 0 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0
```

VI.2 Quelques petits programmes sympathiques



Premier exemple

On souhaite d'une part évaluer si un nombre, rentré par l'utilisateur, est pair ou impair d'autre part calculer son carré.

```
pair_carre = fonction(nombre)
{
  if (nombre%%2 == 0)
    print(paste("Le nombre", nombre, "est pair"))
  else
    print(paste("Le nombre ", nombre, " est impair"))
  carre = nombre^2
  print(paste("qu'il soit pair ou impair, une chose est sure, le carre de",
    nombre, "est ", carre))
}
```

```
pair_carre(8)
[1] "Le nombre 8 est pair"
[1] "qu'il soit pair ou impair, une chose est sure, le carre de 8 est 64"
```

```
pair_carre(15)
[1] "Le nombre 15 est impair"
[1] "qu'il soit pair ou impair, une chose est sure, le carre de 15 est 225"
```



Deuxième exemple

On souhaite calculer le factoriel d'un nombre rentré par l'utilisateur. Plusieurs manières de procéder.

Utilisation de la boucle for

```
factoriel1 = fonction(nombre)
{
  facto = 1
  for (i in 1 : nombre)
    facto = facto*i

  print(paste(" Le factoriel de", nombre, "est", facto))
}

factoriel1(5)
[1] " Le factoriel de 5 est 120"
factoriel1(10)
[1] " Le factoriel de 10 est 3628800"
```

Utilisation de la boucle while

```
factoriel2 = fonction(nombre)
{
  facto = 1
  i = 1
  while (i <= nombre)
  {
    facto = facto*i
    i = i+1
  }
  print(paste(" Le factoriel de", nombre, "est", facto))
}

factoriel2(3)
[1] " Le factoriel de 3 est 6"
factoriel2(5)
[1] " Le factoriel de 5 est 120"
```

Notons que dans le cadre de la fonction `factoriel1`, une seule instruction est contenue dans la boucle `for` si bien que les accolades ne sont pas indispensables. En revanche, les accolades de la boucle `while` sont indispensables car le nombre d'instructions à effectuer est égal à 2.

Notons qu'il est utile de s'assurer que la fonction à programmer n'est pas disponible dans un package. En effet, la fonction `factorial()`, disponible dans le package `base` calcule le factoriel d'un nombre

```
factorial(5)
[1] 120
factorial(10)
[1] 3628800
```



Troisième exemple

Considérons un jeu de données X défini par n observations et p variables. Soient x et y deux observations. La fonction $K(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$ fournit un indice de similarité entre les observations x et y . L'objectif est de décrire une fonction qui à toute matrice X associe la matrice K .

$$\begin{array}{cccc}
 \begin{array}{c} \boxed{x_{11}} \\ \boxed{x_{21}} \\ \boxed{\cdot} \\ \boxed{\cdot} \\ \boxed{\cdot} \\ \boxed{x_{n1}} \end{array} & \begin{array}{c} x_{12} \\ x_{22} \\ \cdot \\ \cdot \\ \cdot \\ \dots \end{array} & \dots & \begin{array}{c} x_{1p} \\ x_{2p} \\ \cdot \\ \cdot \\ \cdot \\ x_{np} \end{array} \\
 & & \blacksquare & \blacksquare \\
 \begin{array}{c} \boxed{K(x_1,x_1)} \\ \boxed{K(x_1,x_2)} \\ \boxed{\cdot} \\ \boxed{\cdot} \\ \boxed{\cdot} \\ \boxed{K(x_n,x_1)} \end{array} & \begin{array}{c} \boxed{K(x_1,x_2)} \\ \boxed{K(x_2,x_2)} \\ \boxed{\cdot} \\ \boxed{\cdot} \\ \boxed{\cdot} \\ \dots \end{array} & \dots & \begin{array}{c} \boxed{K(x_1,x_n)} \\ \boxed{K(x_2,x_n)} \\ \boxed{\cdot} \\ \boxed{\cdot} \\ \boxed{\cdot} \\ \dots \end{array}
 \end{array}$$

Afin d'alléger le code, on peut remarquer que la matrice K est symétrique....

```

noyau_gaussien = fonction(X, sigma)
{
  K = matrix(0, dim(X)[1], dim(X)[1])
  for (i in 1 : dim(X)[1])
  {
    for (j in i : dim(X)[1])
    {
      K[i, j] = (sum((abs(X[i, ] - X[j, ]))^2))^0.5
      K[j, i] = K[i, j]
    }
  }
  K = exp(-K / sigma);
  return(K)
}

```

Pour appeler la fonction `noyau_gaussien` et retourner la matrice K , on procède de la manière suivante

```
K = noyau_gaussien(X, 1)      où sigma = 1
```

Quatrième exemple

On souhaite écrire une fonction qui à toute matrice X associe la matrice K à travers la fonction suivante $K(x_i, x_j) = (\langle x_i, x_j \rangle + c)^n$ où c et n sont des paramètres à fixer par l'utilisateur.

```

noyau_polynomial = fonction(X, c, n)
{
  K = matrix(0, dim(X)[1], dim(X)[1])
  for (i in 1 : dim(X)[1])
  {
    for (j in i : dim(X)[1])
    {
      K[i, j] = (sum(X[i, ] * X[j, ]) + c)^n
    }
  }
}

```



```
        K[j, i] = K[i, j]
    }
}
return(K)
}
```

Pour appeler la fonction `noyau_polynomial` et retourner la matrice `K` on procède de la manière suivante□

`K = noyau_polynomial(X, 1, 2)` où $c = 1$ et $n = 2$.

Remarque□ Un programme avec boucle est lent par rapport à un programme sans boucle. L'utilisation de la vectorisation est systématiquement préférée ... mais ce n'est pas toujours possible.



VII. Conclusion

R est un logiciel libre offrant un panel d'outil statistique de visualisation et d'analyse très complet. La communauté des utilisateurs de R augmente très rapidement si bien que les méthodes les plus récentes sont disponibles très vite via les packages fournis par le CRAN.

Le site du CRAN recueille également toutes les informations utiles à la compréhension et l'utilisation de R. On peut y trouver une liste de publications (livres et articles) liées à R ainsi que des documents d'explications des méthodes statistiques développées :

www.R-project.org/doc/bib/R-publications.html et www.cran.r-project.org/other-docs.html

On trouve également sur le site du CRAN un FAQ (Frequently Asked Questions). Le «R-FAQ» est régulièrement mise à jour et est consultable à l'adresse suivante : www.cran.r-project.org/doc/FAQ/R-FAQ.html

Les listes de discussions de «R-help» sont une source intéressante d'informations pour les utilisateurs. On y trouve les solutions à de nombreux problèmes rencontrés par les utilisateurs de R, les annonces de nouvelles versions et de nouveaux packages. <http://www.r-project.org/mail.html>

Remarque : De nombreux utilisateurs ont envoyé sur «R-help» des fonctions ou des programmes disponibles dans les archives. Il est donc important si l'on a un problème avec R de procéder dans l'ordre avant d'envoyer un message à «R-help» :

1. Consulter attentivement l'aide-en-ligne
2. Consulter le «R-FAQ»
3. Chercher dans les archives de «R-help»
4. Envoyer un message à «R-help»

Ajoutons que la revue électronique «R News» comble l'espace entre les listes de discussion électroniques et les publications scientifiques traditionnelles. Le premier numéro a été publié en janvier 2001 et le rythme de sortie est de trois numéros par an. Kurt Hornik et Friedrich Leisch sont les éditeurs de cette revue. <http://cran.r-project.org/doc/Rnews/>

Enfin, si vous mentionnez R dans une publication, il faut citer l'article original :

Ihaka R. & Gentleman R. *R: a language for data analysis and graphics*. *Journal of Computational and Graphical Statistics* 5 : 299–314, 1996.