

# TP : Introduction au logiciel R

## **1. Tableaux et matrices**

### **Exercice 1 :**

Différence entre un tableau `array()` et une matrice `matrix()` ?

### **Exercice 2 :**

Comment se fait le remplissage d'une matrice ?

### **Exercice 3 :**

Créer une matrice par concaténation de plusieurs vecteurs (`cbind()`, `rbind()`).

### **Exercice 4 :**

Sélection de la deuxième colonne d'une matrice. Suppression de colonnes ? De lignes ?

### **Exercice 5 :**

Soit A une matrice de dimension (3,4) et B une matrice de dimensions (4,3).

Que donne  $A*B$  ? Conclusion ?

BONUS : Comment faire un produit matriciel ?

### **Exercice 6 :**

Inverse d'une matrice.

## **2. Listes et data frames ; lecture de fichier**

### **Exercice 1 :**

Différence entre une liste `list()` et un data frame `data.frame()` ?

### **Exercice 2 :**

Créer une liste contenant 3 collections de type et de longueur différents.  
Donner un nom à chaque collection.

### **Exercice 3 :**

Accéder à la première puis la deuxième collection.  
Accéder au deuxième élément de la troisième collection.

### **Exercice 4 :**

A quoi servent les commandes `setwd()` et `getwd()` ?

### **Exercice 5 :**

Charger les données fournies dans un data frame « data ».

Le tableau donne, pour l'année 1982, le pourcentage des communes qui disposent d'une boulangerie en fonction de la taille de la commune. (Source INSEE-SCEES).

Calculer le pourcentage de communes qui disposent d'une boulangerie sur l'ensemble du territoire.

## **3. Ecriture d'une fonction**

Soit  $f$  la fonction définie par :

$$f(x) = \frac{k}{1 + \frac{k - n_0}{n_0} e^{-rx}}$$

où  $k$ ,  $n_0$  et  $r$  sont des paramètres.

### **Exercice 1 :**

Ecrire les commandes permettant de créer cette fonction.

### **Exercice 2 :**

Comment faire si on veut définir les paramètres  $n_0$  et  $r$  par défaut à 2, 1 et 1 respectivement ?

### **Exercice 2 :**

Soit  $x=c(1,10)$ . Que retourne cette fonction si on l'appelle avec les paramètres par défaut ?

## **4. Boucles for, while**

### **Exercice 1 :**

Parcourir les entiers de 1 à 20 à l'aide d'une boucle for en affichant dans la console à chaque itération si le nombre courant est pair.

### **Exercice 2 :**

À l'aide de la fonction `while()`, créer une boucle qui permet de calculer la factorielle d'un nombre.

### **Exercice 3 :**

Réutiliser le code de la question précédente pour en faire une fonction qui, lorsqu'on lui donne un nombre, retourne sa factorielle. Comparer le résultat avec la fonction `factorial()`.

### **Exercice 4 :**

Choisir un nombre mystère entier entre 1 et 100, et le stocker dans un objet que l'on nommera `nombre_mystere`. Ensuite, créer une boucle qui à chaque itération effectue un tirage aléatoire d'un entier compris entre 1 et 100. Tant que le nombre tiré est différent du nombre mystère, la boucle doit continuer. À la sortie de la boucle, une variable que l'on appellera `nb_tirages` contiendra le nombre de tirages réalisés pour obtenir le nombre mystère.

### **Exercice 5 :**

Utiliser le code de la question précédente pour réaliser la fonction `trouver_nombre`, qui, lorsqu'on lui donne un nombre compris entre 1 et 100, retourne le nombre de tirages aléatoires d'entiers compris entre 1 et 100 nécessaires avant de tirer le nombre mystère.

### **Exercice 6 :**

En utilisant une boucle `for`, faire appel 1000 fois à la fonction `trouver_nombre()` qui vient d'être créée. À chaque itération, stocker le résultat dans un élément d'un vecteur que l'on appellera `nb_essais_rep`. Enfin, afficher la moyenne du nombre de tirages nécessaires pour retrouver le nombre magique.  
`nb_essais_rep <- rep(NA, 1000)`

### **Exercice 7 :**

Utiliser une boucle `for` pour reproduire la suite de Fibonacci jusqu'à son cinquantième terme.