

---

## TP 1: Prise en main du logiciel R

---

Le document de référence pour les TPs est le cours d'initiation d'Anne Philippe disponible à l'adresse suivante :

<http://www.math.sciences.univ-nantes.fr/~philippe/download/Anne-Philippe-cours-R-par4.pdf>

Pour un manuel sur R plus complet, nous recommandons celui de Vincent Goulet à l'adresse suivante :

[https://cran.r-project.org/doc/contrib/Goulet\\_introduction\\_programmation\\_R.pdf](https://cran.r-project.org/doc/contrib/Goulet_introduction_programmation_R.pdf)

L'utilisation de l'aide en ligne `help` est également recommandée.

Les feuilles de TP suppose que l'étudiant utilise RStudio, disponible en salle de TP sous Linux ; des autres possibilités existent.

**Exercice 1.1** — On définit (les " = " sont interchangeables avec les " < - ")

```
x = c(1,3,5,7)
y <- c(2,3,5,7,11,13)
z = c(4,-1)
b <- y < 5
```

Expliquez la sortie de chacune des commandes R suivantes.

47 / 8	(w = c(x,y))	sum( x>5   x<3)
47 %/% 8	x+z	x > 5 & x < 3
47 %% 8	x+y	sum(x > 5 & x < 3)
x+2	b	y[3]
y*3	!b	y[-3]
length(x)	y>7	y[x]
z	y[y>7]	class(x)
c(z)	y[b]	class(b)
c(z,z)	x[b]	as.integer(b)
w = c(x,y)	sum(x>5)	is.numeric(x)
	sum(x[x>5])	

**Exercice 1.2** —

- 1) Créer les vecteurs suivants avec `rep` et `seq` :
  - a) `y0` constitué de la suite des entiers de  $-3$  à  $20$  par pas de  $1$ .
  - b) `y1` contient tout les entiers pairs entre  $2$  et  $18$
  - c) `y2` est constitué de  $8$  fois de suite la valeur  $4$ .
  - d) `y3` contient  $8$  nombres entre  $0$  et  $16$ , par pas égaux. (Pourquoi est-ce que `y3=seq(0, 16, by = 16/8)` ne fait pas l'affaire ?)
- 2) Extraire de `y3` :
  - a) un vecteur composé du 4ème puis le 7ème élément.
  - b) tous les éléments sauf le 7ème

3) Comparer les commandes suivantes :

<code>matrix(y0, 3, 8)</code>	<code>m[c(1,3),]</code>
<code>matrix(y0, 3, 8, byrow=TRUE)</code>	<code>rbind(y2, y3)</code>
<code>(m=matrix(y0, nrow = 4))</code>	<code>rbind(y1, y2)</code>
<code>m[3,3]</code>	

### Exercice 1.3 —

- 1) Créer un vecteur  $x$  qui contient les réels compris entre 0 et 1 par pas de 0,1.
- 2) Calculer  $n$  la longueur de  $x$
- 3) En utilisant les opérations vectorielles, créer un vecteur  $y = 4x(1 - x)$ .
- 4) avec `plot`, tracer la courbe rejoignant les points  $(x_i, y_i)$ .
- 5) Calculer le maximum des  $y_i$
- 6) En quel point le maximum est-il atteint?
- 7) Tracer la courbe de la fonction  $f : x \mapsto 4x^2(1 - x)$  sur l'intervalle  $[-2, 1]$ , en rouge avec le logiciel.

### Exercice 1.4 — En utilisant les commandes `factorial` et `choose`, calculer

- 1) la réponse à l'exercice TD 1.6
- 2) la réponse à l'exercice TD 1.9
- 3) Combien de chances a-t-on de gagner le super jackpot à l'euromillion? (5 numéros parmi 49, et 2 numéros étoilés parmi 10, l'ordre ne compte pas).
- 4) Pour  $k = 1, 2, 3, 4, 5, 6$ , taper `choose(k, 0:k)` puis `sum(choose(k, 0:k))`. À quel énoncé du cours correspondent ces sommes ?

### Exercice 1.5 — Boucles

- 1) Taper `x = c(1)`. Utiliser la boucle `for` contenant la commande `x = c(x,1)` pour créer un vecteur de taille 20 ne contenant que de 1. Comment créer le même vecteur avec `rep` ?
- 2) Même question, avec une commande du genre `x = i*x`, pour calculer  $20! = \text{factorial}(20)$ .
- 3) Vérifier l'identité dans l'exercice TD 1.13 pour  $n = 8$ .

### Exercice 1.6 — Fibonacci (script et boucle sous R) On souhaite calculer avec R les 100 premiers termes de la suite de Fibonacci $u_{n+2} = u_{n+1} + u_n$ .

- 1) Créer un script dans le menu "fichier -- > créer un script" et le nommer "fib.R"
- (a) Sur la première ligne : mettre en commentaire (la ligne commence par `#`) le nom du programme, par exemple `# suite de Fibonacci`
- (b) Créer un vecteur  $u$  de taille 100 ne contenant que de 1
- (c) En utilisant la boucle `for`, assigner à  $u_{n+2}$  la valeur  $u_{n+1} + u_n$ .
- 2) Lancer le script (p.ex. taper ctrl+maj+entrée).
- 3) En utilisant la commande `plot` représenter la suite  $(u_n)$  sur un graphique.
- 4) Représenter également la suite  $(v_n) = (\ln u_n)$  sur un graphique.