TP n° 1 : Prise en main de R

Exercice 1. On définit trois vecteurs x, y et z par les commandes R suivantes :

```
x = c(1, 3, 5, 7, 9)
y = c(2, 3, 5, 7, 11, 13)
z = c(9, 3, 2, 5, 9, 2, 3, 9, 1)
```

Reproduire et comprendre les résultats des commandes suivantes :

```
x + 2
y * 3
length(x)
x + y
sum(x > 5)
sum(x[x > 5])
sum(x > 5 | x < 3)
```

```
y[3]
y[-3]
y[x]
(y > 7)
y[y > 7]
sort(z)
sort(z, dec = TRUE)
```

```
rev(z)
order(z)
unique(z)
duplicated(z)
table(z)
rep(z, 3)
```

Exercice 2. Créer deux vecteurs de dimensions quelconques. Créer un vecteur en insérant le second vecteur entre les 2-ème et 3-ème éléments du premier vecteur.

Exercice 3.

1. Créer les vecteurs suivants :
   (a) $y0$ constitué de la suite des entiers de 0 à 10 par pas de 2,
   (b) $y1$ constitué de tous les entiers pairs entre 1 et 18,
   (c) $y2$ constitué de 20 fois de suite la valeur 4,
   (d) $y3$ constitué de 20 nombres entre 0 et 10.
2. Extraire de $y3$:
   (a) le troisième élément,
   (b) tous les éléments sauf le troisième.
3. Comparer les commandes suivantes :

   ```
   matrix(y3, nrow = 2)
   ```

   ```
   matrix(y3, byrow = TRUE)
   ```
4. Construire une matrice A comportant quatre lignes et trois colonnes remplies par lignes successives avec les éléments du vecteur `1:12`.
5. Construire une matrice B comportant quatre lignes et trois colonnes remplies par colonnes successives avec les éléments du vecteur `1:12`.
6. Extraire l'élément situé en deuxième ligne et troisième colonne de A.
7. Extraire la première colonne de A, puis la deuxième ligne de A.

8. Construire une matrice C constituée des lignes 1 et 4 de A.

Exercice 4. Construire une matrice comportant 9 lignes et 9 colonnes avec des 0 sur la diagonale et des 1 partout ailleurs (on pourra utiliser la commande `diag`).

Exercice 5.

1. Créer un vecteur $x = (x_1, \ldots, x_{11})$ contenant les réels compris entre 0 et 1 par pas de 0.1.
2. Afficher la longueur de x.
3. En utilisant les opérations vectorielles, créer un vecteur $y = 4x(1-x)$.
4. Tracer la courbe rejoignant les points $(x_1, y_1), \ldots, (x_{11}, y_{11})$ avec la commande `plot`.
5. Calculer le maximum des $y_1, \ldots, y_{11}$.
6. En quel point le maximum est-il atteint ?
7. Tracer la courbe de la fonction $f(x) = 4x^2(1-x)$, $x \in [-2, 1]$, en rouge.

Exercice 6. Utiliser R pour donner les valeurs numériques attendues :

1. Combien y-a-t-il de carrés (4 cartes de même valeur) dans un jeu de 32 cartes ?
2. Combien d'anagrammes peut-on faire avec le mot "dinosaure" ?
3. Combien de chances a-t-on de gagner le super jackpot à l'euromillion ? (donc d'avoir 5 bons numéros parmi 49, et 2 bons numéros étoilés parmi 10).
4. Chaque pièce d'un nouveau jeu de domino est de la forme: $\boxed{a \,\|\, b}$ avec $(a, b) \in \{0, \ldots, 9\}^2$ en sachant qu'un domino reste le même si on le tourne à 180 degrés (par exemple, $\boxed{ \,\|\, 8} = \boxed{8 \,\|\, }$ est un, et un seul domino). Déterminer le nombre de pièces différentes que contient un jeu complet de dominos.

Exercice 7. On souhaite calculer avec R les 100 premiers termes de suite de Fibbonacci :

$$u_{n+2} = u_{n+1} + u_n.$$

1. Créer un script dans le menu "fichier -créer un script", le nommer "fib.R".
2. Sur la première ligne: mettre en commentaire (la ligne commence par ♯) le nom du programme, par exemple `♯ suite de Fibonnacci`.
3. Créer un vecteur u de taille 100 ne contenant que des 1.
4. En utilisant la boucle `for`, assigner à u_{n+2} la valeur $u_{n+1} + u_n$.
5. En utilisant la commande `plot` représenter la suite (u_n) sur un graphique.