

Introduction aux logiciels pour les statistiques (IS2A3) – 2024/2025**TP1 : Manipulations de base en R****Exercice 1 : Les objets****Variables**

Variables numériques

- (a) Créez une variable nombre à laquelle vous affecterez π .
- (b) Arrondissez la variable nombre à 2 chiffres après la virgule (\Rightarrow *round*).

Variables caractères

- (a) Affectez à une variable *mot1* le mot « Aiguille ».
- (b) Affectez à une variable *mot2* le mot « de ».
- (c) Affectez à une variable *mot3* le mot « BUFFON ».
- (d) Quel est le rôle des packages *stringi* et *stringr* ?
- (e) En déduire la variable *experience* valant « AIGUILLE De buffon ».

Fonction (niveau 1)

Une valeur approchée de π par l'expérience de Buffon est donnée par : $\pi \approx \frac{2na}{xL}$

n est le nombre d'aiguilles lancées.

a est la longueur d'une aiguille.

x est le nombre d'aiguilles qui tombent sur 2 lattes à la fois.

L est la largeur des lattes du parquet.

Écrivez une fonction *Buffon*(*n,a,x,L*) qui renvoie la valeur de π grâce à la formule sus-citée.

Vecteurs

1. Effacez le contenu de l'environnement.
2. Créez un vecteur A comprenant les valeurs 7, 26, 7, 18, 22, 18, 7.
3. Créez un vecteur B comprenant les valeurs 8, 4, 8, 2, 4, 5 et 13.
4. Créez un vecteur C comprenant les valeurs 10 et 20.
5. Créez un vecteur D comprenant 50 fois l'entier 7 (\Rightarrow *rep*).
6. Créez un vecteur E comprenant tous les entiers de 0 à 50 en allant de 2 en 2 (\Rightarrow *seq*).
7. Quel est le but de la commande « *A[2]* » ?
8. Saisissez « *i ← which(A >= 25)* » puis indiquez ce que représente la variable *i*.
9. Quel est le but de la commande « *A[i] ← 0* » ?
10. Saisissez « *A + 4* » puis « *2*A + 4* » et enfin « *exp(A)* ». Commentez.
11. Saisissez « *sum(A)* », « *prod(A)* », « *range(A)* », « *length(A)* » puis « *summary(A)* ». Commentez.
12. Saisissez « *A + B* ». Commentez.
13. Saisissez « *A + C* ». Commentez.

14. Saisissez « `cbind(A, B)` » puis « `rbind(A, B)` ». Commentez.
15. Saisissez « `c(A,B)` ». Commentez.
16. Déterminez les modalités du vecteur A et les effectifs associés (\Rightarrow *table*).
17. Triez les valeurs du vecteur A par ordre croissant (\Rightarrow *sort*).

Fonction (niveau 2)

Écrivez une fonction *Quadratique(x,w)* qui prend en paramètres 2 vecteurs : une série statistique *x* et les pondérations *w* associées. Elle renvoie la moyenne quadratique pondérée de la série.

Matrices

1. Créez la matrice M en tapant « `M ← matrix(c(1, 2, 3, 4), nrow=2, ncol=2)` » .
2. Tapez « `M` », « `M[1,2]` », « `M[,1]` » et « `M[1,]` ». Commentez.
3. Créez la matrice N en tapant « `N ← matrix(c(10, 20, 30, 40), nrow=2, ncol=2)` » .
4. Créez la matrice P en tapant « `V ← c(1, 0, 3, 4, 5, 5, 0, 4, 5, 6, 3, 4, 0, 1, 3, 2)` » puis « `P ← matrix(V, nrow=4, ncol=4)` » .
5. Tapez « `nrow(M)` », « `ncol(M)` » et « `dim(M)` » .
6. Saisissez « `M*N` » puis « `M%*%N` ». Commentez.
7. Créer une matrice de dimension 4 dont les éléments diagonaux sont 12, 45, 17 et 6.
8. Créez la matrice P1 composée des colonnes de P dont tous les éléments sont inférieurs à 3.5 (\Rightarrow *apply* et *all*) .
9. Créez la matrice P2 composée des lignes de P dont tous les éléments sont différents de 0 (\Rightarrow *apply* et *all*) .

Data.frame

On peut importer des données en R de plusieurs manières.

1. Créez une table de données directement grâce au langage R.
2. Importez la même table de données grâce à la commande « *Import Dataset* » .
3. Importez la table de données au format « *RData* » .
4. Importez un jeu de données d'un package de votre choix.
5. Exécutez le script *mystere* et commentez.

Manipuler un jeu ou une table de données

1. Rappelez la propriété fondamentale des tables de données en R.
2. Sans utiliser la syntaxe de dplyr, sélectionnez la 2^{ème} colonne de la table *iris*.
3. Sans utiliser la syntaxe de dplyr, sélectionnez la 2^{ème} modalité de la 3^{ème} colonne de la table *iris*.

Exercice 2 : Factorielles

1. Calculez la factorielle de 10 à l'aide d'une boucle for.
2. Calculez la factorielle de 10 sans boucle for.
3. À partir du code de la question précédente, écrivez une fonction factorielle qui retourne la factorielle de l'entier strictement positif passé en unique argument. Avant d'effectuer le calcul, la fonction vérifie le type de l'argument. S'il n'est pas conforme, alors elle retourne un message d'erreur.
4. En déduire une fonction *coeffbinomial* qui prend en paramètre des entiers strictement positifs n et p. Elle renvoie le coefficient binomial associé.
5. Donnez le plus petit entier strictement positif n dont la factorielle est supérieure à 2023 (\Rightarrow *while*) .

Exercice 3 : Suite de Fibonacci

La suite de Fibonacci est définie par la relation de récurrence suivante :

$$\begin{cases} F_0 = 0 \\ F_1 = 1 \\ F_n = F_{n-1} + F_{n-2} \end{cases}$$

Écrivez une fonction qui prend en paramètre un entier n supérieur ou égal à 2 et affiche la suite de Fibonacci jusqu'à son $n^{\text{ème}}$ terme.

Exercice 4 : Divisible

Écrivez une fonction *Divisible* qui prend en paramètres K et Q . Votre fonction vérifie si K et Q sont bien des entiers strictement positifs. Elle considère ensuite la séquence des entiers de 1 à K . Elle renvoie alors le nombre exact d'éléments de la séquence qui sont divisibles par Q .

Exercice 5 : Un premier graphique

1. Écrivez une fonction f définie par $f(x) = \sin(x)^2 |x-3|^{1/2}$
2. Tracez la courbe représentative de f pour $x \in [-6, 3]$ (\Rightarrow plot).
3. Ajoutez sur le graphique précédent la courbe de g définie par :

$$g(x) = \begin{cases} \sin(x)\ln(x) & \text{si } x > 0 \\ 3 + \sin(x)^2 x & \text{sinon} \end{cases}$$

4. Ajoutez un titre et une légende à votre graphique (\Rightarrow plot).

Exercice 6 : Résolution d'un système matriciel

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 3 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

1. Générez la matrice A et le vecteur b .
2. Calculez le déterminant de la matrice A . Est-elle inversible ?
3. Résolvez le système linéaire $Ax=b$.
4. Saisissez la commande « `eigen(A, symmetric=FALSE, only.values = FALSE)` ». Elle renvoie une liste dont vous extrairez :
 - (a) les valeurs propres dans une matrice D .
 - (b) les vecteurs propres dans une matrice de passage P .
5. Calculez P^{-1} grâce à la commande `solve(P)` et retrouvez A à partir de P , D et P^{-1} .