

Exercices simples avec corrigés sur le logiciel R

Exercice 1 : Création et manipulation de vecteurs

Question :

1. Créez un vecteur v contenant les nombres de 1 à 10.
2. Multipliez chaque élément de ce vecteur par 2.
3. Filtrez uniquement les éléments supérieurs à 10.

Exercice 2 : Création et manipulation de matrices

Question :

1. Créez une matrice m de dimension 3×3 avec les nombres de 1 à 9.
2. Calculez la somme des éléments de chaque colonne.
3. Remplacez les éléments de la deuxième colonne par des zéros.

Exercice 3 : Manipulation de data frames

Question :

1. Créez un data frame nommé `df` avec deux colonnes :
 - `Nom` contenant les noms "Alice", "Bob", "Charlie".
 - `Age` contenant les âges 25, 30, 35.
2. Ajoutez une nouvelle colonne `Ville` contenant "Paris", "Lyon", "Marseille".
3. Filtrez les lignes où l'âge est supérieur à 28.

Exercice 4 : Graphiques simples

Question :

1. Tracez un histogramme des 100 premières valeurs aléatoires tirées d'une loi normale.
2. Ajoutez un titre et des labels aux axes.

Exercice 5 : Statistiques descriptives

Question :

1. Créez un vecteur `x` contenant les valeurs suivantes : 4, 8, 15, 16, 23, 42.
2. Calculez la moyenne, la médiane et l'écart-type.
3. Identifiez la valeur minimale et maximale.

Exercice 6 : Boucles

Question :

1. Utilisez une boucle `for` pour calculer les carrés des nombres de 1 à 10 et stockez les résultats dans un vecteur.
2. Réécrivez le même calcul en utilisant une boucle `while`.

Exercice 7 : Génération de nombres aléatoires

Question :

1. Générez 10 nombres aléatoires uniformes dans l'intervalle $[0, 1]$.
2. Générez 10 nombres aléatoires uniformes dans l'intervalle $[10, 20]$.
3. Calculez la moyenne et l'écart-type des nombres générés.

Exercice 8 : Distribution normale

Question :

1. Générez 1000 nombres aléatoires suivant une loi normale $N(0, 1)$.
2. Tracez un histogramme des données.
3. Générez 1000 nombres aléatoires suivant une loi normale $N(5, 2)$.
4. Tracez un histogramme des nouvelles données.

Solutions

Solution exercice 1 :

```
1 # 1. Creer un vecteur
2 v <- 1:10
3
4 # 2. Multiplier chaque element par 2
5 v2 <- v * 2
6
7 # 3. Filtrer les elements superieurs a 10
8 v_filtered <- v2[v2 > 10]
9
10 v_filtered
```

Solution exercice 2 :

```
1 # 1. Creer une matrice
2 m <- matrix(1:9, nrow = 3, ncol = 3)
3
4 # 2. Somme des elements de chaque colonne
5 col_sums <- colSums(m)
6
7 # 3. Remplacer la deuxieme colonne par des zeros
8 m[, 2] <- 0
9
10 list(matrice = m, col_sums = col_sums)
```

Solution exercice 3 :

```
1 # 1. Creer un data frame
2 df <- data.frame(
3   Nom = c("Alice", "Bob", "Charlie"),
4   Age = c(25, 30, 35)
5 )
6
7 # 2. Ajouter une colonne
8 df$Ville <- c("Paris", "Lyon", "Marseille")
9
10 # 3. Filtrer les lignes ou l'age est superieur a 28
11 df_filtered <- df[df$Age > 28, ]
12
13 df_filtered
```

Solution exercice 4 :

```

1 # 1. Generer des donnees
2 data <- rnorm(100)
3
4 # 2. Tracer un histogramme
5 hist(data, main = "Histogramme de donnees aleatoires",
      xlab = "Valeurs", ylab = "Frequence")

```

Solution exercice 5 :

```

1 # 1. Creer un vecteur
2 x <- c(4, 8, 15, 16, 23, 42)
3
4 # 2. Calculs statistiques
5 moyenne <- mean(x)
6 mediane <- median(x)
7 ecart_type <- sd(x)
8
9 # 3. Valeurs minimale et maximale
10 val_min <- min(x)
11 val_max <- max(x)
12
13 list(moyenne = moyenne, mediane = mediane, ecart_type =
      ecart_type, min = val_min, max = val_max)

```

Solution exercice 6 :

```

1 # Utiliser une boucle for
2 result_for <- c()
3 for (i in 1:10) {
4   result_for <- c(result_for, i^2)
5 }
6
7 # Utiliser une boucle while
8 result_while <- c()
9 i <- 1
10 while (i <= 10) {
11   result_while <- c(result_while, i^2)
12   i <- i + 1
13 }
14
15 list(result_for = result_for, result_while =
      result_while)

```

Solution exercice 7 :

```

1 # Generer 10 nombres aleatoires uniformes dans [0, 1]
2 uniform_01 <- runif(10, min = 0, max = 1)
3
4 # Generer 10 nombres aleatoires uniformes dans [10, 20]
5 uniform_10_20 <- runif(10, min = 10, max = 20)
6
7 # Calculer la moyenne et l'ecart-type
8 mean_01 <- mean(uniform_01)
9 sd_01 <- sd(uniform_01)
10 mean_10_20 <- mean(uniform_10_20)
11 sd_10_20 <- sd(uniform_10_20)
12
13 list(mean_01 = mean_01, sd_01 = sd_01, mean_10_20 =
      mean_10_20, sd_10_20 = sd_10_20)

```

Solution exercice 8 :

```

1 # 1. Generer des donnees suivant N(0, 1)
2 data_normal_01 <- rnorm(1000, mean = 0, sd = 1)
3
4 # 2. Tracer un histogramme des donnees N(0, 1)
5 hist(data_normal_01, main = "Histogramme des donnees
      N(0, 1)", xlab = "Valeurs", ylab = "Frequence")
6
7 # 3. Generer des donnees suivant N(5, 2)
8 data_normal_52 <- rnorm(1000, mean = 5, sd = 2)
9
10 # 4. Tracer un histogramme des donnees N(5, 2)
11 hist(data_normal_52, main = "Histogramme des donnees
      N(5, 2)", xlab = "Valeurs", ylab = "Frequence")

```

Syntaxe des Structures de Contrôle et Fonctions en R

```
if (prédicat) {  
    Bloc d'instructions  
} else {  
    Bloc d'instructions  
}
```

La syntaxe d'une structure conditionnelle est donnée comme suit :

```
for (itérateur in une_séquence) {  
    Bloc d'instructions  
}
```

La syntaxe d'une structure itérative simple est donnée comme suit :

```
while (prédicat) {  
    Bloc d'instructions  
}
```

La syntaxe d'une structure itérative conditionnelle est donnée comme suit :

```
Nom_Fonction <- function (Arg1, Arg2, ...) {  
    Bloc d'instructions  
}
```

La syntaxe d'une fonction est donnée comme suit :

Exemples

Exemple 1 : Utilisation de if-else

```
# Exemple simple de condition if-else  
x <- 5  
if (x > 0) {  
    print("Nombre positif")  
} else {  
    print("Nombre non positif")  
}
```

Exemple 2 : Utilisation d'une boucle for

```
# Boucle for pour parcourir une séquence
for (i in 1:5) {
  print(paste("Itération :", i))
}
```

Exemple 3 : Utilisation d'une boucle while

```
# Exemple d'une boucle while
x <- 1
while (x <= 5) {
  print(x)
  x <- x + 1
}
```

Exemple 4 : Définition d'une fonction

```
# Fonction pour additionner deux nombres
additionner <- function(a, b) {
  return(a + b)
}

# Appel de la fonction
resultat <- additionner(3, 4)
print(resultat)
```

Exemple 5 : Demande de saisie utilisateur en R

Dans cet exemple, le programme demande à l'utilisateur de saisir un nombre, vérifie si la saisie est valide, puis effectue des calculs avec la valeur fournie.

```
# Demande à l'utilisateur de saisir un nombre
cat("Veuillez entrer un nombre : ")
user_input <- as.numeric(readline()) # Lire la saisie de l'utilisateur et la convertir en nombre
# Ou nous pouvons utiliser
# Demande à l'utilisateur de saisir un nombre
user_input <- as.numeric(readline(prompt = "Veuillez entrer un nombre : ")) # Lire la saisie de l'utilisateur et la convertir en nombre

# Vérifie si l'entrée est valide
if (!is.na(user_input)) {
  # Utilise la valeur fournie
  print(paste("Vous avez entré le nombre :", user_input))
  print(paste("Le carré de ce nombre est :", user_input^2)) # Affiche le carré du nombre
```

```

} else {
  # Message d'erreur si l'entrée n'est pas un nombre
  cat("Entrée non valide. Veuillez saisir un nombre la prochaine fois.\n")
}

```

Exemple 6 : Quitter une boucle et créer un vecteur de zéros

Dans cet exemple, le programme :

- Crée un vecteur de zéros avec `numeric(n)`.
- Remplit ce vecteur avec des valeurs saisies par l'utilisateur.
- Interrompt la boucle si une valeur négative est saisie.

```

# Définir la taille du vecteur
n <- as.integer(readline(prompt = "Entrez la taille du vecteur : "))
vec <- numeric(n) # Crée un vecteur de zéros de taille n

# Remplir le vecteur avec des valeurs
cat("Entrez", n, "valeurs (entrez une valeur négative pour quitter) :\n")
for (i in 1:n) {
  val <- as.numeric(readline(prompt = paste("Valeur", i, ": ")))

  if (val < 0) {
    cat("Valeur négative détectée. Sortie de la boucle.\n")
    break # Quitte la boucle
  }

  vec[i] <- val # Remplit l'élément i du vecteur avec la valeur entrée
}

# Afficher le vecteur résultant
cat("Le vecteur résultant est : ", vec, "\n")

```