

# 1. Installation et Configuration

- **Installation de R** : Téléchargez et installez R à partir du site officiel [CRAN](#).
- **RStudio** : Il est recommandé d'utiliser RStudio, un environnement de développement intégré (IDE) pour R. Vous pouvez le télécharger sur [RStudio](#).

# 2. Structure de base d'un script R

Un script R est constitué de commandes que vous pouvez écrire dans RStudio ou dans un fichier `.R`.

Exemple :

```
# Ceci est un commentaire
x <- 10 # Affectation de la valeur 10 à la variable x
y <- 5  # Affectation de la valeur 5 à la variable y
z <- x + y # Addition de x et y, résultat stocké dans z
print(z) # Affiche le résultat
```

# 3. Types de données

R prend en charge différents types de données :

- **Numérique** : 5 , 3.14
- **Entier** : 1L , 2L (ajoutez L pour spécifier un entier)
- **Logique** : TRUE , FALSE
- **Chaîne de caractères** : "Bonjour"
- **Vecteurs** : Liste de valeurs (même type)

```
vecteur <- c(1, 2, 3, 4, 5) # Créer un vecteur de nombres
```

- **Matrices et dataframes** :
  - **Matrice** : `matrix()`
  - **Dataframe** : `data.frame()`

# 4. Opérations de base

- **Opérations arithmétiques** :

```
x <- 10
y <- 5
addition <- x + y # Addition
```

```
multiplication <- x * y # Multiplication
division <- x / y # Division
```

- Logiques :

```
x > y # Vrai si x est supérieur à y
x == y # Vrai si x est égal à y
x != y # Vrai si x est différent de y
```

## 5. Manipulation de données

- Vecteurs / Matrices

```
# Initialize a 2x3 matrix with custom values
mat3 <- matrix(c(5, 8, 2, 1, 6, 3), nrow = 2, ncol = 3)
# Vector
v <- c(1, 2, 3, 4, 5)
sum(v) # Somme des éléments du vecteur
mean(v) # Moyenne des éléments
```

- Dataframe :

```
df <- data.frame(
  Nom = c("Alice", "Bob", "Charlie"),
  Age = c(25, 30, 35)
)
print(df) # Afficher le dataframe
summary(df) # Statistiques descriptives du dataframe
```

## 6. Contrôle de flux

- Conditionnel :

```
if (x > y) {
  print("x est plus grand que y")
} else {
  print("x n'est pas plus grand que y")
}
```

- Boucles :

```
for (i in 1:5) {
  print(i) # Affiche les nombres de 1 à 5
}
```

## 7. Visualisation des données

```
plot(sin, -pi, 2*pi)

data <- data.frame(
  Age = c(18, 22, 25, 30, 35, 40, 45, 50, 55, 60),
  Score = c(78, 85, 88, 92, 90, 80, 70, 65, 60, 58)
)
plot(data)
```

## 8. Déclarer une fonction

```
additionner <- function(a, b) {
  somme <- a + b
  return(somme)
}
```

## 9. Packages

Pour étendre les capacités de R, vous pouvez installer des packages supplémentaires :

- Installation d'un package :

```
install.packages("ggplot2")
```

- Charger un package :

```
library(ggplot2)
```

## 10. Aide et documentation

- Pour obtenir de l'aide sur une fonction, utilisez `?` ou `help()` .

```
?mean # Affiche l'aide sur la fonction mean()
help(mean) # Autre méthode
```