

# Cours 1 : Introduction - R Programming

Nolan

November 8, 2024

Fonction	Utilisation	Exemple et Résultat
<b>Classe et Types de Données</b>		
<code>class(x)</code>	Retourne la classe de l'objet	<code>class(6.5) ⇒ "numeric"</code>
<code>typeof(x)</code>	Renvoie le type de stockage interne	<code>typeof(6.5) ⇒ "double"</code>
<code>as.integer(x)</code>	Convertit un nombre en entier	<code>as.integer(6.5) ⇒ 6</code>
<code>round(x, digits)</code>	Arrondit un nombre au nombre de décimales spécifié	<code>round(4.567, 2) ⇒ 4.57</code>
<b>Opérations Arithmétiques</b>		
<code>x + y</code>	Additionne deux nombres	<code>4 + 6 ⇒ 10</code>
<code>x * y</code>	Multiplie deux nombres	<code>4 * 6 ⇒ 24</code>
<code>x / y</code>	Divise deux nombres	<code>10 / 2 ⇒ 5</code>
<code>x %% y</code>	Retourne le reste de la division entière	<code>10 %% 3 ⇒ 1</code>
<b>Vecteurs</b>		
<code>c(...)</code>	Crée un vecteur avec les éléments spécifiés	<code>c(1, 2, 3) ⇒ 1, 2, 3</code>
<code>seq(from, to, by)</code>	Crée une séquence avec un intervalle défini	<code>seq(1, 10, by = 2) ⇒ 1, 3, 5, 7, 9</code>
<code>rep(x, times)</code>	Crée un vecteur avec un élément répété	<code>rep(5, times = 3) ⇒ 5, 5, 5</code>
<code>runif(n, min, max)</code>	Génère un vecteur de nombres aléatoires uniformes	<code>runif(3, 1, 5) ⇒ Ex. 2.3, 3.8, 4.1</code>
<code>length(x)</code>	Retourne la longueur du vecteur	<code>length(c(1, 2, 3)) ⇒ 3</code>
<code>sort(x, decreasing = FALSE)</code>	Trie les éléments du vecteur	<code>sort(c(3, 1, 2), decreasing = TRUE) ⇒ 3, 2, 1</code>
<code>rank(x)</code>	Renvoie les rangs de chaque élément du vecteur	<code>rank(c(10, 20, 15)) ⇒ 1, 3, 2</code>
<code>x[i]</code>	Accède à l'élément en position i du vecteur	<code>c(10, 20, 30)[2] ⇒ 20</code>
<code>x[x &gt; value]</code>	Filtre les éléments du vecteur selon une condition	<code>c(10, 20, 30)[c(10, 20, 30) &gt; 15] ⇒ 20, 30</code>
<b>Statistiques Simples sur les Vecteurs</b>		
<code>sum(x)</code>	Calcule la somme des éléments du vecteur	<code>sum(c(1, 2, 3)) ⇒ 6</code>
<code>mean(x)</code>	Calcule la moyenne des éléments du vecteur	<code>mean(c(1, 2, 3)) ⇒ 2</code>
<code>min(x)</code>	Trouve la plus petite valeur du vecteur	<code>min(c(1, 2, 3)) ⇒ 1</code>
<code>max(x)</code>	Trouve la plus grande valeur du vecteur	<code>max(c(1, 2, 3)) ⇒ 3</code>
<code>median(x)</code>	Calcule la médiane des éléments du vecteur	<code>median(c(1, 2, 3)) ⇒ 2</code>
<code>summary(x)</code>	Donne un résumé statistique de base	<code>summary(c(1, 2, 3)) ⇒ Min = 1, Median = 2, Mean = 2, Max = 3</code>
<b>Matrices</b>		
<code>matrix(data, nrow, ncol)</code>	Crée une matrice à partir des données	<code>matrix(1:6, nrow = 2, ncol = 3) ⇒ <math>\begin{bmatrix} 1 &amp; 3 &amp; 5 \\ 2 &amp; 4 &amp; 6 \end{bmatrix}</math></code>
<code>colnames(x) &lt;- names</code>	Définit les noms des colonnes	<code>colnames(mat) &lt;- c("A", "B", "C")</code>
<code>rownames(x) &lt;- names</code>	Définit les noms des lignes	<code>rownames(mat) &lt;- c("Row1", "Row2")</code>
<code>x[i, j]</code>	Accède à un élément spécifique de la matrice	<code>mat[1, 2] ⇒ valeur en ligne 1, colonne 2</code>

Fonction	Utilisation	Exemple et Résultat
<b>Opérations sur les Matrices</b>		
<code>mat1 + mat2</code>	Additionne les éléments correspondants	<code>matrix(1:4, 2, 2) + matrix(5:8, 2, 2)</code>
<code>mat1 * mat2</code>	Multiplie les éléments correspondants	<code>matrix(1:4, 2, 2) * matrix(5:8, 2, 2)</code>
<code>colSums(x)</code>	Calcule la somme des colonnes	<code>colSums(matrix(1:4, 2, 2))</code> ⇒ 3, 7
<code>rowMeans(x)</code>	Calcule la moyenne des lignes	<code>rowMeans(matrix(1:4, 2, 2))</code> ⇒ 1.5, 3.5
<b>Data Frame : Création et Manipulation</b>		
<code>data.frame(...)</code>	Crée un data frame avec les colonnes spécifiées	<code>data.frame(c(1,2), c("A","B"))</code> ⇒ Data frame avec 2 colonnes
<code>colnames(df) &lt;- names</code>	Définit les noms des colonnes d'un data frame	<code>colnames(df) &lt;- c("Age", "Sexe")</code>
<code>rownames(df) &lt;- names</code>	Définit les noms des lignes d'un data frame	<code>rownames(df) &lt;- c("John", "Jane")</code>
<code>df\$colonne</code>	Accède à une colonne par son nom	<code>data_Iris\$Petal.Length</code> ⇒ Colonne <code>Petal.Length</code>
<code>df[row, col]</code>	Accède à un élément spécifique par sa ligne et colonne	<code>data_Iris[1, 2]</code> ⇒ Valeur en ligne 1, colonne 2
<b>Importation de Fichiers de Données</b>		
<code>read.table(file, ...)</code>	Lit un fichier de données en data frame	<code>read.table("iris.csv", sep=",", header=T)</code>
<b>Manipulation des Data Frames</b>		
<code>head(df, n)</code>	Affiche les premières <code>n</code> lignes d'un data frame	<code>head(data_Iris, 6)</code> ⇒ Premières 6 lignes
<code>df[row, cols]</code>	Accède aux lignes et colonnes spécifiées	<code>data_Iris[c(10,40,50), 1:4]</code> ⇒ Lignes 10, 40, 50 des colonnes 1 à 4
<code>\$</code>	Sélectionne une colonne par son nom	<code>data_Iris\$Species</code> ⇒ Colonne <code>Species</code>
<code>df[df\$col == value, ]</code>	Filtre les lignes où une colonne a une valeur donnée	<code>data_Iris[data_Iris\$Species == "setosa", ]</code>
<code>which()</code>	Retourne les indices des éléments vérifiant une condition	<code>which(data_Iris\$Species == "setosa")</code>
<code>%in%</code>	Sélectionne les lignes avec des valeurs dans une liste	<code>data_Iris[data_Iris\$Species %in% c("setosa", "virginica"), ]</code>
<code>subset(df, condition)</code>	Sélectionne les lignes d'un data frame selon une condition	<code>subset(data_Iris, Species == "setosa")</code>
<b>Types de Données Factor</b>		
<code>factor(x)</code>	Transforme un vecteur en facteur	<code>factor(c("m", "f", "m"))</code> ⇒ Factor avec niveaux "m", "f"
<code>levels(factor)</code>	Donne les niveaux d'un facteur	<code>levels(data_Iris\$Species)</code>
<code>class(x)</code>	Retourne la classe de l'objet (ex., "factor")	<code>class(data_Iris\$Species)</code> ⇒ "factor"
<code>summary(x)</code>	Résumé statistique, montre les niveaux d'un facteur	<code>summary(data_Iris)</code>

Fonction	Utilisation	Exemple et Résultat
<b>Manipulations Avancées sur les Data Frames</b>		
<code>min(), mean(), max()</code>	Donne le min, la moyenne, le max d'une colonne	<code>min(data_Iris\$Sepal.Length)</code> ⇒ Min de <code>Sepal.Length</code>
<code>data.frame(col1, col2)</code>	Crée un data frame avec des colonnes spécifiques	<code>data.frame(Age=c(23, 25), Sex=c("M", "F"))</code>
<code>merge(df1, df2, by)</code>	Joint deux data frames par une colonne commune	<code>merge(df1, df2, by="id")</code>
<code>cbind(df1, df2)</code>	Concatène deux data frames par colonnes	<code>cbind(data_iris_quantitative, data_iris_qualitative)</code>
<code>rbind(df1, df2)</code>	Concatène deux data frames par lignes	<code>rbind(data_Iris, data_Iris)</code>
<b>Fonctions de base pour manipuler des datasets</b>		
<code>apply(X, MARGIN, FUN)</code>	Applique une fonction sur les lignes ou colonnes d'une matrice/data frame	<code>apply(iris[, -5], 2, mean)</code> ⇒ Moyennes par colonne
<code>by(X, INDICES, FUN)</code>	Applique une fonction sur les sous-ensembles d'un data frame	<code>by(iris, iris\$Species, summary)</code> ⇒ Résumé statistique par espèce
<code>aggregate(X, BY, FUN)</code>	Calcule des statistiques par groupes	<code>aggregate(iris[, -5], list(iris\$Species), mean)</code> ⇒ Moyennes par espèce
<b>Manipulation avec le package dplyr</b>		
<code>select(df, cols)</code>	Sélectionne des colonnes spécifiques d'un data frame	<code>select(iris, Sepal.Length, Species)</code>
<code>select(df, -cols)</code>	Exclut des colonnes d'un data frame	<code>select(iris, -Species)</code>
<code>select(df, starts_with("prefix"))</code>	Sélectionne les colonnes commençant par un préfixe	<code>select(iris, starts_with("Sepal"))</code>
<code>select(df, ends_with("suffix"))</code>	Sélectionne les colonnes finissant par un suffixe	<code>select(iris, ends_with("Width"))</code>
<code>select(df, contains("str"))</code>	Sélectionne les colonnes contenant une chaîne de caractères	<code>select(iris, contains("al"))</code>
<code>filter(df, condition)</code>	Filtre les lignes d'un data frame selon une condition	<code>filter(iris, Species == "setosa")</code>
<code>filter(df, between(col, a, b))</code>	Filtre les lignes avec des valeurs entre a et b	<code>filter(iris, between(Sepal.Length, 4, 7))</code>
<code>filter_all(df, all_vars(condition))</code>	Filtre sur toutes les colonnes selon une condition	<code>filter_all(iris[, -5], all_vars(. &gt; 4))</code>
<code>%&gt;%</code> (Pipe)	Chaîne les fonctions ensemble	<code>iris %&gt;% select(Species) %&gt;% filter(Sepal.Length &gt; 5)</code>
<code>arrange(df, col)</code>	Trie les lignes par une colonne	<code>arrange(iris, Sepal.Length)</code>
<code>arrange(df, desc(col))</code>	Trie les lignes par ordre décroissant d'une colonne	<code>arrange(iris, desc(Sepal.Length))</code>
<b>Fonctions de résumé statistique avec dplyr</b>		
<code>summarise(df, new_col = FUN(col))</code>	Crée une colonne résumée par des statistiques	<code>summarise(iris, moyenne_Sepal = mean(Sepal.Length))</code>
<code>summarise(df, new_col = n())</code>	Calcule le nombre d'occurrences	<code>summarise(iris, total = n())</code>
<code>summarise_each(df, funs(FUN), cols)</code>	Applique plusieurs fonctions sur des colonnes	<code>summarise_each(iris, funs(mean, min), Sepal.Length, Petal.Length)</code>

Groupement de données et résumé avec dplyr		
<code>group_by(df, col)</code>	Regroupe les données par une colonne	<code>iris %&gt;% group_by(Species)</code>
<code>summarise_each(group_by(df, col), funs(FUN), cols)</code>	Applique des fonctions sur des groupes	<code>iris %&gt;% group_by(Species) %&gt;% summarise_each(funs(mean), Sepal.Length)</code>
<code>filter(df, col &lt; value)</code>	Filtre les lignes selon une condition sur les groupes	<code>iris %&gt;% group_by(Species) %&gt;% filter(Petal.Length &lt; 3)</code>
Mutations avec dplyr		
<code>mutate(df, new_col = expression)</code>	Ajoute des colonnes calculées au data frame	<code>mutate(iris, Total_Petal = Petal.Length + Petal.Width)</code>
<code>mutate(df, col = NULL)</code>	Supprime des colonnes d'un data frame	<code>mutate(iris, Species = NULL)</code>
<code>mutate(df, col = expression)</code>	Modifie les valeurs d'une colonne	<code>mutate(iris, Sepal.Length = Sepal.Length * 2)</code>