Structure des données en R

Joël Kazadi

2022-09-13

Cette section aborde les différentes structures de données disponibles sous le langage R. On énumère principalement les classes suivantes :

- les vecteurs ;
- les matrices ;
- les arrays ;
- les dataframes.

1 Création d'un vecteur

Un vecteur est un objet de dimension égale à 1 (ligne ou colonne). Les éléments d'un vecteur doivent nécessairement être de nature identique, i.e. de même mode.

1.1 Données numériques

L'objet `vector1` est de la classe "vecteur". Il est de taille égale à 3 et est composé des données de mode "numérique".

```
vector1 <- c(1,2,3)
print(vector1)
```

```
## [1] 1 2 3
```

```
is.vector(vector1)
```

```
## [1] TRUE
```

```
length(vector1)
```

```
## [1] 3
```

```
str(vector1)
```

```
##  num [1:3] 1 2 3
```

1.2 Chaînes de caractères

L'objet `vector2` est de la classe "vecteur". Il est de taille égale à 2 et est composé des données de mode "chaîne de caractères".

```
vector2 <- c("Joe","Kaz")
print(vector2)
```

```
## [1] "Joe" "Kaz"
```

```
is.vector(vector2)
```

```
## [1] TRUE
```

```
length(vector2)
```

```
## [1] 2
```

```
str(vector2)
```

```
##  chr [1:2] "Joe" "Kaz"
```

2 Création d'une matrice

Une matrice est un objet de dimension égale à 2 (lignes et colonnes). Comme pour les vecteurs, les éléments d'une matrice doivent nécessairement être de même mode.

2.1 Opérations sur les matrices

Créons une matrice carrée de taille égale à 3.

```
matrix1 <- matrix(1:9, nrow=3, ncol=3, byrow = TRUE)
print(matrix1)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

Créons une matrice unitaire de taille égale à 3.

```
identite <- diag(x = 1, nrow = 3)
print(identite)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

Effectuons le produit matriciel des deux précédentes matrices.

```
matrix1%*%identite
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

Trouvons la transposée de la matrice `matrix1`.

```
t(matrix1)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

Calcul des sommes et des moyennes à partir d'une matrice par ligne et par colonne.

```
matrix2 <- matrix(c(25,7,6,13), nrow=2, ncol=2, byrow = FALSE)
print(matrix2)
```

```
##      [,1] [,2]
## [1,]   25    6
## [2,]    7   13
```

```
rowSums(matrix2)
```

```
## [1] 31 20
```

```
colSums(matrix2)
```

```
## [1] 32 19
```

```
rowMeans(matrix2)
```

```
## [1] 15.5 10.0
```

```
colMeans(matrix2)
```

```
## [1] 16.0  9.5
```

Calcul du déterminant, de l'inverse, de la décomposition de Cholesky ainsi que des valeurs propres et vecteurs propres de la matrice `matrix2`.

```
det(matrix2)
```

```
## [1] 283
```

```
solve(matrix2)
```

```
##             [,1]        [,2]
## [1,]  0.04593640 -0.02120141
## [2,] -0.02473498  0.08833922
```

```
chol(matrix2)
```

```
##      [,1] [,2]
## [1,]    5  1.2
## [2,]    0  3.4
```

```
eigen(matrix2)
```

```
## eigen() decomposition
## $values
## [1] 27.83176 10.16824
##
## $vectors
##           [,1]       [,2]
## [1,] 0.9043401 -0.3750138
## [2,] 0.4268125  0.9270192
```

2.2 Manipulation des matrices

Créons une matrice de dimension 3x2 dont les entrées sont du mode "chaîne de caractères".

```
matrix3 <- matrix(c("A","B","C","d","e","f"), nrow=3, ncol=2)
print(matrix3)
```

```
##      [,1] [,2]
## [1,] "A"  "d"
## [2,] "B"  "e"
```

```
## [3,] "C"  "f"
```

```
dim(matrix3)
```

```
## [1] 3 2
```

Renommons les lignes et les colonnes de la matrice `matrix3`.

```
rownames(matrix3)=c("L1", "L2", "L3")
colnames(matrix3)=c("C1", "C2")
print(matrix3)
```

```
##    C1  C2
## L1 "A" "d"
## L2 "B" "e"
## L3 "C" "f"
```

Extrayons les éléments des deux dernières lignes sur la première colonne de la matrice `matrix3` suivant des approches :

- Approche par "index" ;
- Approche par "étiquette".

```
matrix3[c(2,3),1]             #approche par index
```

```
##  L2  L3
## "B" "C"
```

```
matrix3[c("L2", "L3"),"C1"]   #approche par etiquette
```

```
##  L2  L3
## "B" "C"
```

Modifions toutes les entrées sur la dernière colonne de la matrice `matrix3`.

```
matrix3[c(1:3),2]=c("D","E","F")
print(matrix3)
```

```
##    C1  C2
## L1 "A" "D"
## L2 "B" "E"
## L3 "C" "F"
```

3 Création d'un array

Un array est un objet de dimension égale à $n > 2$. Comme pour les vecteurs et les matrices, les éléments d'un array doivent nécessairement être de même mode, i.e. soit numérique, soit chaîne de caractère.

```
V <- vector(mode = "integer", length = 24)    #creation d'un vecteur nul de taille 24
A <- array(data = V, dim = c(3,4,2))          #creation d'un array compose de 2 matrices de dimension 3
print(A)
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    0    0
## [2,]    0    0    0    0
## [3,]    0    0    0    0
##
```

```
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    0    0
## [2,]    0    0    0    0
## [3,]    0    0    0    0
```

4 Création d'un dataframe

Dans les vecteurs ou les matrices, les éléments doivent être de même nature, i.e. il y a homogéneité des données. Les dataframes permettent d'avoir des données de nature variée, i.e. à la fois de mode numerique et de mode caractère.

```
data <- data.frame(Econometrie = c(18,16,17,18),
                   Statistique = c(14,13,15,19),
                   Niveau = c("Bon","Moyen",NA,"Excellent"),
                   row.names = c("Kadima","Kazadi","Nsamba","Malu"))
print(data)
```

```
##        Econometrie Statistique    Niveau
## Kadima          18          14       Bon
## Kazadi          16          13     Moyen
## Nsamba          17          15      <NA>
## Malu            18          19 Excellent
```

Réalisons le résumé statistique du dataframe, puis inspectons la présence de valeurs manquantes dans ce dataframe.

```
summary(data)
```

```
##   Econometrie     Statistique       Niveau
##  Min.   :16.00   Min.   :13.00   Length:4
##  1st Qu.:16.75   1st Qu.:13.75   Class :character
##  Median :17.50   Median :14.50   Mode  :character
##  Mean   :17.25   Mean   :15.25
##  3rd Qu.:18.00   3rd Qu.:16.00
##  Max.   :18.00   Max.   :19.00
```

```
summary(is.na(data))
```

```
##  Econometrie     Statistique       Niveau
##  Mode :logical   Mode :logical   Mode :logical
##  FALSE:4         FALSE:4         FALSE:3
##                                  TRUE :1
```