

Correction2

Sélectionnez des éléments dans les objets R

Question 1

Quels sont les deux grands principes dans la sélection d'éléments au sein d'un objet R ?

Attention, plusieurs réponses sont possibles.

1. La sélection par position ou indice

2. La sélection par arguments

3. La sélection par conditions

4. La sélection par opérateurs

{ Les deux grands principes de sélection via R sont :

la sélection par indice ou position

la sélection par condition }

Question 2

Quelles affirmations sont vraies concernant la sélection dans un vecteur ?

Attention, plusieurs réponses sont possibles.

1. L'opérateur de sélection est []

2. On peut employer des indices négatif au sein de l'opérateur de sélection

3. On ne peut pas sélectionner dans un vecteur via une condition

{ L'opérateur de sélection est bien [] et on peut sélectionner via des indices négatifs !

En revanche, comme on peut sélectionner via des indices négatifs, si n correspond au nombre d'éléments du vecteur, ces derniers doivent être des entiers entre $-n$ et n , à l'exception du 0 !

On peut également sélectionner dans un vecteur via une condition : c'est une des grandes forces de R !}

Question 3

Quel est l'affichage correspondant à l'exécution du code suivant ?

```
x = c(1, 3, 4, 5, 10)
```

```
x[-c(1:4)]
```

1. 1 3 4 5

2. 1 5

3. 10

4. 1 2 3 4

{La sélection par indice négatif prend tous les éléments, sauf les indices spécifiés. Or ici, l'opérateur `-` indique que nous sélectionnons tous les indices de 1 à 4 : donc au final on sélectionne tout, sauf les éléments de 1 à 4. Il ne reste donc que le dernier : 10}

Question 4

Soit une matrice m . Que permet de sélectionner le code suivant :

```
m[,c(i, 10)]
```

1. Il permet de sélectionner un élément à la ligne i et la colonne 10

2. Il permet de sélectionner l'ensemble des éléments des colonnes i et 10

3. Il permet de sélectionner l'ensemble des éléments des lignes i et 10

4. Il permet de sélectionner les éléments i et 10 de notre matrice

{ Ici notre code permet :

de sélectionner toutes les lignes, car l'emplacement pour sélectionner des lignes (avant la virgule) est laissé vide

de sélectionner les colonnes i et 10

Ainsi, on sélectionne l'ensemble des éléments des colonnes i et 10, peu importe la valeur de i. }

Question 5

Soit une matrice `m` ayant 10 lignes et 4 colonnes. Quel code permet de sélectionner une "sous-matrice" contenant les lignes 2 à 5, de la 3ème colonne ?

1. `m[2:5, 3, drop=F]`

2. `m[-c(1:5), 3]`

3. `m[3, 2:5, drop=F]`

4. `m[2:5, 3]`

{ Les deux codes permettant de sélectionner les lignes 2 à 5, de la 3ème colonne sont :

`m[2:5, 3, drop=F]`

et

`m[2:5, 3]`

Or, comme nous souhaitons conserver le format matrice (et non avoir un résultat sous forme de vecteur), la première écriture est donc la plus adéquate ! }

Question 6

Quelles affirmations parmi les suivantes sont vraies concernant la sélection dans une liste ?

Attention, plusieurs réponses sont possibles.

1. On peut accéder à l'objet R correspondant au second élément d'une liste `L` via l'écriture : `L[2]`

2. Si on connaît le nom d'un élément de la liste, on peut y accéder via l'opérateur `$`

3. On peut sélectionner des éléments d'une liste via une sélection par index (ou indice)

4. Le résultat d'une sélection au sein d'une liste sera forcément une autre liste

{L'écriture `L[2]` permet d'accéder au second élément de la liste et non l'objet correspondant. Pour ce faire, on utilisera l'écriture `[[]]`, soit : `L[[2]]`}

On peut sélectionner via l'opérateur `$`, si on connaît le nom associé à un élément au sein d'une liste. Mais on peut très bien sélectionner les différents éléments via des indices (`L[c(1:3)]` par exemple)

En revanche, le résultat d'une sélection n'est pas forcément une liste. Lorsqu'on sélectionne par exemple l'objet correspondant à un élément, le résultat sera du type de l'objet en question, qui n'est pas forcément une liste ! Par exemple, avec le code ci dessous, on sélectionne un vecteur classique de caractères : }

Question 7

Soit la liste suivante :

```
x <- c(1:10)
```

```
X <- matrix(1:8,ncol=4)
```

```
y <- c(T,T,T,F,F)
```

```
z <- matrix(c("A","B","C","D"),ncol=2)
```

```
maliste <- list(comp1=x,comp2=X,comp3=y,element4=z)
```

Quel(s) code(s) permet d'accéder à l'objet associé au nom `comp3`, c'est à dire un vecteur de booléen ?

Attention, plusieurs réponses sont possibles.

1. `maliste['comp3']`

2. `maliste[3]`

3. `maliste[[3]]`

4. `maliste$comp3`

{ `maliste[3]` et `maliste['comp3']` permettent juste d'accéder à l'élément et non à ce qu'il contient. Les codes corrects pour faire ce que l'on cherche à faire sont donc :

`maliste[[3]]`

`maliste$comp3`

Question 8

Quelles affirmations parmi les suivantes sont vraies concernant les dataframes ?

Attention, plusieurs réponses sont possibles.

1. On peut sélectionner un élément via son numéro de ligne et son numéro de colonne

2. On peut sélectionner selon une condition

3. On peut sélectionner dans un data.frame via l'opérateur %

4. Le résultat d'une sélection sur un data.frame est forcément un autre data.frame

{ La sélection dans les dataframes peut se faire comme la sélection dans les matrices ou comme la sélection dans les listes. Ainsi, on peut très bien sélectionner via l'indice ligne/colonne (comme avec une matrice) ou via une condition (comme avec une liste). On peut également sélectionner une colonne en particulier si on connaît le nom qui y est associé, via l'opérateur \$.

Lorsque l'on sélectionne une seule colonne, le résultat sera par exemple un vecteur : donc le résultat d'une sélection sur un dataframe n'est pas forcément un data.frame. }

Question 9

Soit le dataframe `df` suivant, représentant la température mesurée dans 3 villes différentes, sur deux jours :

```
temp = c(20, 21, 17.5, 32, 30, 25)
```

```
jour = c('Lu', 'Lu', 'Lu', 'Ma', 'Ma', 'Ma')
```

```
lieu = c('Nice', 'Lyon', 'Paris', 'Nice', 'Lyon', 'Paris')
```

```
df = data.frame(temp, jour, lieu)
```

Quel code permet de sélectionner les villes qui ont eu une température strictement supérieure à 20°C, le lundi ?

1. `df[df$jour = 'Lu' & df$temp > 20, 'lieu']`

2. `df[df$jour == 'Lu' & df$temp >= 20, 'lieu']`

3. `df$lieu[df$jour == 'Lu' and df$temp > 20]`

4. `df$lieu[df$jour == 'Lu' & df$temp > 20]`

{ Analysons code par code :

- `df[df$jour = 'Lu' & df$temp > 20, 'lieu']` contient une erreur. En effet, l'opérateur d'égalité dans une condition est `==`

- `df[df$jour == 'Lu' & df$temp >= 20, 'lieu']` ce code est quasiment juste, sauf qu'il sélectionne les villes pour lesquelles les températures sont supérieures ou égales à 20. Hors nous voulons juste celles supérieures. L'écriture correcte serait : `df[df$jour == 'Lu' & df$temp > 20, 'lieu']`

- `df$lieu[df$jour == 'Lu' and df$temp > 20]` contient une erreur : le mot clé `and` n'existe pas en R, l'opérateur correspondant est `&`

`df$lieu[df$jour == 'Lu' & df$temp > 20]` est correcte et fait bien ce que nous recherchons ici. }

Question 10

Reprenons le dataframe `df` présenté **Question** précédente. Que fait le code suivant :

```
df[df$jour != 'Ma' & df$temp == max(df$temp[df$jour == 'Lu'], 'lieu']
```

1. Il permet de sélectionner la température maximale observée sur le mardi
2. Il permet de sélectionner le lieu de la température maximale observée sur un autre jour que mardi
3. Il permet de sélectionner un lieu qui a eu une température égale à la maximale observé le lundi
4. Il permet de sélectionner tous les lieux qui ont eu une température égale à la maximale observé le lundi, sur un autre jour que le mardi

{Décomposons le code pour bien comprendre. Il y a ici deux conditions :

la première (`df$jour != 'Ma'`) va sélectionner toutes les lignes dont le jours est différent de Mardi

la seconde (`df$temp == max(df$temp[df$jour == 'Lu'])`) va sélectionner toutes lignes qui ont l'exacte même température que la maximale enregistrée le Lundi

Étant donné qu'il y a l'opérateur `&` qui correspond au ET logique, la sélection doit valider la première et la seconde condition. Ainsi, on prend toutes les lignes différentes de Mardi, qui ont l'exacte même température que la maximale enregistrée le Lundi

La fin de la sélection indique qu'on ne prend que la colonne Lieu.

Finalement, ce code permet donc de sélectionner tous les lieux qui ont eu une température égale à la maximale observé le lundi, sur un autre jour que le mardi. }

LearnerWorld