

# Résumé\_R\_Dplyr\_Amine

Xueting YIN

2020/12/23

## Entête :

Nous allons analyser un document de package dplyr expliqué par Amine BENOUCIEF, le lien vers son github est <https://github.com/benben62/PSBX>.

## Synthèse :

Dans ce document, Amine a analysé le package dplyr dans le langage R avec le data frame hdv2003 en utilisant les différentes fonctions: filter, mutate, arrange. Les fonctions sont bien expliquées grâce aux exemples variables. Surtout pour la fonction filter, il a mis au total 9 exemples pour expliquer comment l'utiliser sous différents aspects. Nous pouvons utiliser un critère, deux critères, ou même trois critères dans cette fonction. Chaque fonction est accompagnée de l'affichage des résultats, ce qui permet au lecteur de comprendre plus facilement les résultats de l'utilisation de ces fonctions. ## Extrait commenté : Amine a choisi le data frame hdv2003, les codes ci-dessous sont expliqués la partie importante de la fonction filter. Il commence par filtrer sous un critère : >, <, == :

```
rest1 <- filter(rest, age >20) #nous pouvons afficher toute la liste rest à condition que l'age est supérieur à 20
rest2 <- filter(rest,age == 22) # nous pouvons afficher toute la liste rest à condition que l'age est supérieur à 22
rest3 <- filter(rest, sport=='Non') # filtrer avec les mots : afficher toute la liste rest à condition que le sport égale « non »
```

bien expliqué filtrer sur les 2 :

```
rest4 <- filter(rest, age >20 & sport == 'Non') # afficher toute la liste rest à condition que l'age est supérieur à 20 et le sport égale « non »
rest5 <- filter(rest, age >20 , sport == 'Non') # afficher toute la liste rest à condition que l'age est supérieur à 20 ou le sport égale « non »
```

bien expliqué filtrer sur 3 critères vec la virgule:

```
rest6 <- filter(rest, age >20 ,sexe =='Femme', sport == 'Non') # afficher toute la liste rest à condition que l'age est supérieur à 20 et sexe égale « femme » et le sport égale « non »
```

bien expliqué comment créer des intervalles :

```
rest7 <- filter(rest,age >= 22, age <= 30) # ici c'est pour afficher toute la liste rest à condition que l'age est supérieur ou égale à 22 et inférieure ou égale à 30
```

bien expliqué comment faire la sélection multiple mais manque de précision sur ! et %in%

```
rest8 <- filter(rest,age %in% c(20,22,60)) #afficher toute la liste rest à condition que l'age soit 20, soit 22, ou soit 60
rest9 <- filter(rest,!age %in% c(20,22,60)) # afficher toute la liste rest à condition que l'age ne soit pas 20, ni 22, ni 60
```

# Evaluation :

## 1. Le document est-il visuellement agréable ?

Ce document est relativement agréable d'un point de vue visuel, car Amine montre les résultats de chaque ligne de code, il est utile pour les lecteurs de mieux comprendre les fonctions, mais il n'est pas nécessaire de montrer le fonctionnement de certains codes, comme le chargement du package. Il y a non seulement les résultats du chargement, ainsi que des informations supplémentaires, ce qui rendrait l'affichage des résultats très encombrant et confondrait les informations efficaces que les lecteurs peuvent recevoir. Amine utilise `head()` pour afficher les résultats des six premières lignes. On voit qu'il a fait des efforts pour rendre le document moins lourd, mais pour certains résultats évidents, il pourrait exprimer seulement en phrases, ce qui serait plus lisible.

## 2. Est-il compréhensible ? ( qualité des explications)

Pour ce package, Amine explique bien l'utilisation des trois fonctions, et chaque fonction est accompagnée de l'affichage des résultats, mais il y a encore des codes qui n'ont pas été expliqués en détail, ce qui sera plus difficile à comprendre pour les débutants.

## 3. Combien de fonctions ont été utilisé ?

Il y a trois fonctions au total, bien qu'elles ne soient pas nombreuses, chacune est décrite de manière très détaillée donc il est suffisant.

## 4. Est-il intéressant d'utiliser telles fonctions ?

Dplyr est un package dans le langage R pour manipuler les données, Amine s'est basé sur les fonctions de base, il va pas à pas pour montrer les fonctions, il est donc assez pertinent d'utiliser la fonction `filter()` qui permet de filtrer les lignes, la fonction `mutate()` qui permet de créer des nouvelles colonnes, la fonction `arrange()` qui permet de trier les résultat en ordre croissant ou décroissant.

## 5. Y-a-t-il une bibliographie?

Oui, il y a une bibliographie à la fin du document qui permet de trouver facilement aux lecteurs les sources et avoir plus d'informations sur ce package.

# Conclusion :

Pour ce document, j'apprécie beaucoup la partie bien détaillée dans chaque fonction. Bien que Amine n'ait choisi que trois fonctions, mais il y a un grand nombre d'exemples avec, donc le document entier n'apparaîtra pas mince, et nous pouvons encore en apprendre beaucoup.