TP Introduction à la statistique spatiale

QUELQUES RAPPELS SUR LE PIPE ET DPLYR

Parmi les packages les plus utilisés au monde, on trouve les packages de la famille “tidyverse”. Le tidyverse est un ensemble de packages R compatibles entre eux, destinés à faciliter la manipulation de données, leur visualisation et la programmation. Pour plus d’informations sur le tidyverse, vous pouvez lire cet excellent article. Au sein du tidyverse, on trouve un package particulièrement connu : `dplyr`.

Le package `dplyr` est un des packages les plus utilisés au monde pour la manipulation de données avec R. Les packages de manipulation de données spatiales, tel que le package `sf`, se sont ainsi inspirés de ce package pour la construction de leurs fonctions.

Par ailleurs, le package `dplyr` importe lors de son chargement un nouvel opérateur : l’opérateur `%>%` (dit “pipe” en anglais, raccourci clavier `Ctrl+Maj+m`). Cet opérateur, issu du package `magrittr` (package faisant aussi partie de la famille `tidyverse`) permet d’enchainer les fonctions de manière plus lisible :

```
# Exemple culinaire : on veut créer un objet "omelette", notre objet R initial est "oeufs".
# 3 étapes (3 fonctions) pour faire une omelette :
# casser les oeufs, les battre, les faire cuire.
# En R, cela donnerait :
oeufs_casses<-casser(oeufs,ajout_sel=TRUE) # paramètre ajout_sel = TRUE ou FALSE
oeufs_battus<-battre(oeufs_casses)
omelette<-cuire(oeufs_battus, ajout_huile=TRUE) #paramètre ajout_huile = TRUE ou FALSE

# Si l'on faisait ça en une ligne, cela donnerait un code peu lisible.
# L'ordre des opérations est inversé :
omelette <- cuire(battre(casser(oeufs, ajout_sel=TRUE)),ajout_huile=FALSE)

# Le pipe, permet de faire passer le premier argument d'une fonction à gauche du pipe :
casser(oeufs,ajout_sel=TRUE)
# avec le pipe, peut ainsi s'écrire :
oeufs %>%
  casser(ajout_sel=TRUE)

# Avec le pipe, les opérations s'enchainent dans l'ordre souhaité, de manière lisible :
omelette <- oeufs %>%
  casser(ajout_sel=TRUE) %>%
  battre() %>%
  cuire(ajout_huile=TRUE)
```

Le code étant facile à lire, il est facilement reprenable. C’est donc une syntaxe extrêmement utilisée dans le monde.

Les fonctions du package dplyr rappellent parfois celles du langage SQL. Pour avoir de la documentation sur les fonctions de ce package, vous pouvez consulter le cheatsheet associé.

Les fonctions principales du package dplyr sont :

- arrange() : trier la table selon une (ou des) variable(s)
- filter() : sélectionner des observations (des lignes) selon une condition
- slice() : sélectionne les n premières lignes
- select() : sélectionner des variables (des colonnes)
- distinct() : enlever les doublons
- mutate() : créer ou modifier une variable
- group_by() : équivalent du group by en SQL
- summarise() : afficher un résumé sur une ligne
- left_join()/right_join()/inner_join()/... : comme en SQL