Premiers pas avec data.table

Datastorm - B. Thieurmel

19 septembre 2017

```
library(data.table)
```

1. Importer le fichier **flights14** avec `fread`. Un **data.table** est créé (la méthode pour `print` est différente)

```
flights <- fread("flights14.csv")
flights
dim(flights)
tables()
```

2. Sélectionner les 10 premières lignes.
3. Sélectionner les lignes qui ont pour `origin` l'aéroport **JFK**
4. Sélectionner les lignes qui ont pour `origin` l'aéroport **JFK** et comme date de vol le mois (`month`) de juillet.
5. Ordonner par mois, jour, `dep_time`, et cela avec les fonctions `order`, `setorder`, et en utlisant des clés.
6. Ordonner par mois, jour, `dep_time` (ordre décroissant)
7. Sélectionner la première colonne de **flights** et renvoyer un **vecteur** (tester le resultat avec `is.vector`).
8. Sélectionner la première colonne de **flights** et renvoyer **un data.table** (tester le resultat avec `is.data.table`).
9. Afficher le nuage de points `air_time` fonction de `distance`.
10. Afficher le nuage de points `air_time` fonction de `distance` avec une couleur qui dépend du mois, puis du `carrier` (convertir ce dernier en **factor** puis en **as.numeric**)
11. Afficher le nuage de points `air_time` fonction de `distance` par mois (sur une fenêtre graphique contenant 10 graphiques)
12. Effectuer une régression linéaire simple `air_time` en fonction de `distance`.
13. Calculer le nombre de vols qui démarrent de **JFK** par mois
14. Calculer le nombre de vols qui démarrent de **JFK** par mois et par jour, en renommant la nouvelle variable `nbvols`
15. Calculer le nombre moyen de vol par jour (sur tous les mois), grâce à un **chainage**
16. Calculer le nombre de vols qui démarrent de **JFK** et qui arrive (`dest`) à **LAX**.
17. Faire la moyenne des retards au départ (`dep_delay`) et à l'arrivée (`arr_delay`) par compagnie (`carrier`)
18. Utiliser l'opérateur de `.SD` pour faire les quantiles des retards au départ et à l'arrivée par compagnie (`carrier`) via `.SDcols`
19. Créer un **data.table** `ff` qui est la concaténation en colonne de **flights** et la distance au carré contenue dans flights (`cbind`)
20. Refaire la même chose en utilisant l'opérateur `:=`
21. Remplacer la distance par la distance au carré en utilisant `:=`
22. Créer une colonne `speed` égale à `distance` /(`air_time`/60)

23. Créer deux nouvelles colonnes : `trip`, concaténation de `origin` et `dest`, et `delay` somme de `arr_delay` et `dep_delay`

24. Que fait le code suivant :

```
flights[, max_speed := max(speed), by=.(trip)]
```

25. Remplacer, pour la colonne `origin`, le code **JFK** par **JFKennedy**

26. Créons le data.table miniature suivant:

```
DT <- data.table(ID = c("b","b","b","a","a","c"), x = 1:6, y = 7:12, z = 13:18)
```

27. que font :

```
DT[, .(val = c(x, y)), by = ID]
DT[, .(val = list(c(x, y))), by = ID]
DT[, .(val = list(paste(x, y ,sep = ":"))), by = ID]
DT[, .(val = list(paste(x, y, sep = ":")))]
DT[, .(val = paste(x, y, sep = ":"))]
```

28. Reprendre les questions du TD en utilisant un système de clés et en comparant les performances (quand cela est pertinent...)

29. Reprendre les questions du TD et effectuer les mêmes opérations avec un **data.frame**, en comparant les performances (quand cela est pertinent...)