Travaux dirigés pour l'introduction au logiciel R

Marco Pascucci

25/10/2018

data.frame

C'est une liste de vecteurs de même longueur... un tableau! Avec des proprietés pratiques pour selectionner, ordonner, visualiser, manipuler les données.

Créer un data.frame

Les données sont entrées par colonne

```r
enseignants <- data.frame(
  # nom des lignes (optionnel)
  row.names = c("Marie", "Yani", "Jildaz", "Jasmina"),
  # colonnes: <nom> = <liste>
  title = c("PostDoc", "Professor", "Doctorant", "Doctorant"),
  enseignement = c("TD","cours","TD","TP"),
  genre = factor(c("F","M","M","F")),
  cours = c("bio", "bio", "bio", "math"),
  hours_week = c(1,6,2,3),
  # option
  stringsAsFactors = FALSE
)
```

Ajouter une observation (ligne) au data.frame

on utilise la fonction rbind()

```
enseignants %<>% rbind("Marco" = c(title="PostDoc",
                                   enseignement="TD",
                                   genre="M",
                                   hours_week=1.5,
                                   cours="stat")
                       )

enseignants
```

```
##              title enseignement genre cours hours_week
## Marie      PostDoc           TD     F   bio          1
## Yani     Professor        cours     M   bio          6
## Jildaz   Doctorant           TD     M   bio          2
## Jasmina  Doctorant           TP     F  math          3
## Marco      PostDoc           TD     M   1.5       stat
```

data.frame slicing (“trancher”)

```
enseignants[5]   # colonne 5
```

```
##         hours_week
## Marie            1
## Yani             6
## Jildaz           2
## Jasmina          3
## Marco         stat
```

```
enseignants[[5]]   # colonne 5 comme vecteur
```

```
## [1] "1"    "6"    "2"    "3"    "stat"
```

```
enseignants[1,]  # observation (ligne) 1
```

```
##         title enseignement genre cours hours_week
## Marie PostDoc           TD     F   bio          1
```

Nom des observations

il est préférable avoir le nom de chaque observation dans une colonne comme les autres. la fonction `rownames_to_column()` en génère une automatiquement.

```
enseignants %<>% rownames_to_column(var="Nom")
enseignants
```

```
##       Nom     title enseignement genre cours hours_week
## 1   Marie   PostDoc           TD     F   bio          1
## 2    Yani Professor         cours     M   bio          6
## 3  Jildaz Doctorant           TD     M   bio          2
## 4 Jasmina Doctorant           TP     F  math          3
## 5   Marco   PostDoc           TD     M   1.5       stat
```

Manipulation de données with dyplr

`dplyr` est une grammaire de la manipulation des données, fournissant un ensemble cohérent de verbes (fonctions) qui vous aident à résoudre les défis de manipulation de données les plus courants:

- `select()` sélectionne les variables en fonction de leur nom.
- `filter()` sélectionne les observations en fonction de leurs valeurs.
- `arrange()` modifie l'ordre des lignes.
- `mutate()` ajoute de nouvelles variables qui sont des fonctions de variables existantes
- `summarize()` réduit plusieurs valeurs à un seul résumé.

Exemple: mutate()

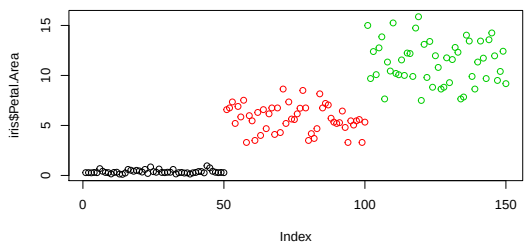
La fonction `mutate()` permet d'ajouter une colonne à un data.frame en faisant des operation sur les données. ATTENTION : les fonctions dyplr retournent une nouvelle structure de données, ne modifient pas le data.frame de départ

```
data("iris")  # observation sur des fleurs
iris

mutate(iris, Petal.Area=Petal.Length*Petal.Width)

iris %<>% mutate(Petal.Area=Petal.Length*Petal.Width)
iris

plot(iris$Petal.Area, col=iris$Species)
```



Exercice 1

0. charger les 25 premières entrées du dataset mtcars et ajouter une colonne avec le nom de chaque ligne
1. afficher un resumé des moyennes de poids “wt” des voitures et puissance “hp”
2. selectionner seulement les modèles des voitures et les colonnes de “cyl” à “wt”
3. ordonner par nombre décroissant de cylindres et puis croissant en poids
4. filtrer les voitures qui n’ont que 4 cylindres et pèsent plus de 2 tonnes
5. créer une colonne de poids en Kg (wt est en tonnes)
6. enchainer avec un PIPE les points 2 et 5

Solution ex. 1

```
data("mtcars")
mtcars %<>% head(25)
mtcars %<>% rownames_to_column(var="Model")
mtcars %>% summarize(poids_m=mean(wt), puissance_m=mean(hp))
mtcars %>% select(c(Model,cyl:wt))
mtcars %>% arrange(desc(cyl), wt)
mtcars %>% filter(cyl==4, wt>2)
mtcars %>% mutate(wt_kg=wt*1000)
mtcars %>% select(c(Model,cyl : wt)) %>%
  mutate(wt_kg=wt*1000)
```