Jour6-Démarrer avec R : Fusion et jointures de données avec dplyr

Kofivi YENA || CoinDataConsulting

Table des matières

Dans cette leçon, nous allons apprendre à fusionner des datasets en R avec dplyr, en utilisant des jointures relationnelles similaires à SQL.

left_join() : jointure gauche

right_join() : jointure droite

inner_join() : jointure interne

full_join() : jointure complète

semi_join() : jointure semi (filtrage)

anti_join() : jointure anti (exclusion)

1 Chargement des données

Nous allons utiliser deux datasets fictifs :

clients : contient des informations sur les clients

achats : contient les achats effectués

```r
library(dplyr)
library(flextable)

# Créer un dataframe de clients
clients <- data.frame(
  id_client = c(1, 2, 3, 4),
  nom = c("Alice", "Bob", "Charlie", "David"),
  ville = c("Paris", "Lyon", "Marseille", "Bordeaux")
)

# Créer un dataframe d'achats
achats <- data.frame(
  id_client = c(1, 2, 2, 5),
  produit = c("Ordinateur", "Téléphone", "Casque", "Tablette"),
  montant = c(1200, 800, 150, 500)
)

# Afficher les tables
flextable(clients) %>%
  autofit() %>%
  align(j = 1, part = "all", align = "center")
```

id_client	nom	ville
1	Alice	Paris
2	Bob	Lyon
3	Charlie	Marseille
4	David	Bordeaux

```
flextable(achats) %>%
  autofit() %>%
  align(j = 1, part = "all", align = "center")
```

id_client	produit	montant
1	Ordinateur	1,200
2	Téléphone	800
2	Casque	150
5	Tablette	500

2 Jointure gauche (left_join())

Récupère toutes les lignes du premier dataset (clients) et ajoute les données correspondantes du second (achats).

```
# Jointure gauche
clients_achats_left <- left_join(clients, achats, by = "id_client")

# Afficher le tableau
clients_achats_left[is.na(clients_achats_left)]<- "NA"
flextable(clients_achats_left) %>%
  autofit() %>%
  align(j = 1, part = "all", align = "center")
```

id_client	nom	ville	produit	montant
1	Alice	Paris	Ordinateur	1200
2	Bob	Lyon	Téléphone	800

id_client	nom	ville	produit	montant
2	Bob	Lyon	Casque	150
3	Charlie	Marseille	NA	NA
4	David	Bordeaux	NA	NA

Remarque : David n'a pas d'achat, donc les colonnes produit et montant sont NA.

3 Jointure droite (right_join())

Récupère toutes les lignes du second dataset (achats) et ajoute les données correspondantes du premier (clients).

```
# Jointure droite
clients_achats_right <- right_join(clients, achats, by = "id_client")

# Afficher le tableau
clients_achats_right[is.na(clients_achats_right)] <- "NA"
flextable(clients_achats_right) %>%
  autofit() %>%
  align(j = 1, align = "center", part = "all")
```

id_client	nom	ville	produit	montant
1	Alice	Paris	Ordinateur	1,200
2	Bob	Lyon	Téléphone	800
2	Bob	Lyon	Casque	150
5	NA	NA	Tablette	500

Remarque : L'id 5 (Tablette) n'existe pas dans clients, donc nom et ville sont NA.

4 Jointure interne (inner_join())

Garde uniquement les valeurs communes entre les deux datasets.

```r
# Jointure interne
clients_achats_inner <- inner_join(clients, achats, by = "id_client")

# Afficher le tableau
clients_achats_inner[is.na(clients_achats_inner)] <- "NA"
flextable(clients_achats_inner) %>%
  autofit() %>%
  align(j = 1, part = "all", align = "center")
```

id_client	nom	ville	produit	montant
1	Alice	Paris	Ordinateur	1,200
2	Bob	Lyon	Téléphone	800
2	Bob	Lyon	Casque	150

Remarque : David (id 4) et l'achat de la Tablette (id 5) sont supprimés.

5 Jointure complète (full_join())

Récupère toutes les lignes des deux datasets et remplit avec NA si besoin.

```r
# Jointure complète
clients_achats_full <- full_join(clients, achats, by = "id_client")

# Afficher le tableau
clients_achats_full[is.na(clients_achats_full)] <- "NA"
flextable(clients_achats_full) %>%
  autofit() %>%
  align(j = 1, part = "all", align = "center")
```

id_client	nom	ville	produit	montant
1	Alice	Paris	Ordinateur	1200
2	Bob	Lyon	Téléphone	800
2	Bob	Lyon	Casque	150
3	Charlie	Marseille	NA	NA

id_client	nom	ville	produit	montant
4	David	Bordeaux	NA	NA
5	NA	NA	Tablette	500

Remarque : On retrouve tous les clients et tous les achats.

6 Jointure semi (semi_join())

Garde uniquement les clients qui ont au moins un achat, sans ajouter d'infos supplémentaires.

```
# Jointure semi
clients_semi <- semi_join(clients, achats, by = "id_client")

# Afficher le tableau
clients_semi[is.na(clients_semi)] <- "NA"
flextable(clients_semi) %>%
  autofit() %>%
  align(j = 1, part = "all", align = "center")
```

id_client	nom	ville
1	Alice	Paris
2	Bob	Lyon

Remarque : David (id 4) est exclu car il n'a pas d'achat.

7 Jointure anti (anti_join())

Garde uniquement les clients sans achat.

```
# Jointure anti
clients_anti <- anti_join(clients, achats, by = "id_client")

# Afficher le tableau
clients_anti[is.na(clients_anti)] <- "NA"
flextable(clients_anti) %>%
```

```
  autofit() %>%
  align(j =, part = "all", align = "center")
```

id_client	nom	ville
3	Charlie	Marseille
4	David	Bordeaux

Remarque : Ici, seul David (id 4) est conservé.

Prochaine leçon (Jour 7) : Visualisation de données avec ggplot2 Dans le Jour 7, nous verrons comment créer des graphiques interactifs et statiques avec ggplot2.