Jour4-Démarrer avec R : Manipulation des données avec dplyr

Kofivi YENA || CoinDataConsulting

Table des matières

Dans cette leçon, nous allons explorer dplyr, un package puissant pour manipuler et transformer des données en R. Nous verrons comment utiliser les principales fonctions :

select() : pour choisir des colonnes

filter() : pour filtrer les lignes

mutate() : pour créer/modifier des variables

arrange() : pour trier les données

summarise() et **group_by()** : pour les statistiques résumées

rename() et **relocate()** : pour renommer et réorganiser les colonnes

1 Chargement du package et du dataset

Utilisons un dataset disponible dans R pour les pratiques

```
# Charger le package dplyr
library(dplyr)

# Charger un dataset exemple (mtcars)
data("mtcars")

# Afficher les 5 premières lignes
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

2 Sélection de colonnes avec select()

select() permet de choisir uniquement les colonnes pertinentes.

```
# Sélectionner uniquement les colonnes mpg, cyl et hp
mtcars_selected <- mtcars %>%
  select(mpg, cyl, hp) %>%
  head()

# Afficher le tableau avec flextable
library(flextable)
flextable(mtcars_selected) %>% autofit()
```

mpg	cyl	hp
21.0	6	110
21.0	6	110
22.8	4	93

mpg	cyl	hp
21.4	6	110
18.7	8	175
18.1	6	105

3 Filtrage des lignes avec filter()

filter() permet de garder uniquement certaines lignes selon des conditions.

```
# Filtrer les voitures ayant plus de 20 mpg et 6 cylindres
mtcars_filtered <- mtcars %>%
  filter(mpg > 20, cyl == 6) %>%
  head()

# Afficher le tableau
flextable(mtcars_filtered) %>% autofit()
```

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
21.4	6	258	110	3.08	3.215	19.44	1	0	3	1

4 Création de nouvelles variables avec mutate()

mutate() permet d'ajouter ou de modifier des colonnes.

```
# Ajouter une colonne puissance/kilo
mtcars_mutated <- mtcars %>%
  mutate(hp_per_kg = hp / wt)

# Afficher le tableau
flextable(mtcars_mutated) %>% autofit()
```

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	hp_per_kg
21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4	41.98473
21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4	38.26087
22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	40.08621
21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	34.21462
18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2	50.87209
18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1	30.34682
14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4	68.62745
24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2	19.43574
22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2	30.15873
19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4	35.75581
17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4	35.75581
16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3	44.22604
17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3	48.25737
15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3	47.61905
10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4	39.04762
10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4	39.63864
14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4	43.03087
32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1	30.00000
30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2	32.19814
33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1	35.42234
21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1	39.35091
15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2	42.61364
15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2	43.66812
13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4	63.80208
19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2	45.51365
27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1	34.10853
26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2	42.52336

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	hp_per_kg
30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2	74.68605
15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4	83.28076
19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6	63.17690
15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8	93.83754
21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2	39.20863

5 Trier les données avec arrange()

arrange() permet de trier les lignes selon une colonne.

```
# Trier par mpg (croissant)
mtcars_sorted <- mtcars %>%
  arrange(mpg) %>%
  head()

# Afficher le tableau
flextable(mtcars_sorted) %>% autofit()
```

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
10.4	8	472	205	2.93	5.250	17.98	0	0	3	4
10.4	8	460	215	3.00	5.424	17.82	0	0	3	4
13.3	8	350	245	3.73	3.840	15.41	0	0	3	4
14.3	8	360	245	3.21	3.570	15.84	0	0	3	4
14.7	8	440	230	3.23	5.345	17.42	0	0	3	4
15.0	8	301	335	3.54	3.570	14.60	0	1	5	8

6 Regrouper et résumer avec group_by() et summarise()

group_by() permet de regrouper les données et summarise() d'obtenir des statistiques.

```r
# Calculer la moyenne du mpg par nombre de cylindres
mtcars_summary <- mtcars %>%
  group_by(cyl) %>%
  summarise(avg_mpg = mean(mpg))

# Afficher le tableau
mtcars_summary %>%
  flextable() %>%
  autofit()
```

cyl	avg_mpg
4	26.66364
6	19.74286
8	15.10000

7 Renommer et réorganiser les colonnes

rename() pour modifier les noms et relocate() pour réorganiser les colonnes.

Prochaine leçon (Jour 6) : Fusion et jointures de données avec dplyr.

On continue demain avec la fusion de datasets grâce aux fonctions comme left_join(), inner_join(), etc.