Tidyverse

Hugo Harari-Kermadec

EOS - Econometrie

2020

Des données bien rangées

Tidyverse est une suite de packages qui servent à gérer et visualiser des données.
Il faut l'installer (une seule fois) et l'appeler (à chaque redémarrage de R)

```
install.packages("tidyverse")
library(tidyverse)
```

Les 3 commandements de tidyverse :

- chaque variable est une colonne,
- chaque observation est une ligne,
- une table différente pour chaque situation.

Des données bien rangées

Une base de données de tidyverse s'appelle un tibble.
On importe des données directement en tibble avec

```
pop<-read_delim("population.csv",delim=";")
```

On peut aussi convertir un dataframe en tibble

```
pop2<-as_tibble(pop1)
```

Verbes

`arrange`	permet de classer les lignes
`slice`	permet de choisir des lignes.
`filter`	permet de choisir des lignes suivant un test.
`select`	permet de choisir des variables.
`rename`	permet de renommer une variable.
`mutate`	permet de créer des variables et faire des opérations ligne à ligne

```
arrange(pop,age)
slice(pop,1:5) ; filter(pop,age>25)
select(pop,age) ; select(pop,-age) ;
rename(pop,Revenu=Income)
mutate(pop,Revenu_Annuel=Revenu*12)
```

Fonctions

`start_with` permet de choisir un ensemble de variables similaires, par ex. `Revenu_janvier`, `Revenu_fevrier...`

`end_with` idem suivant la fin.

`case_when` permet de définir une variable suivant les cas.

```
select(pop,start_with{Revenu_})
mutate(pop,Génération=case_when(age<25~jeune,
           age>=25 & age<65~ actif, age>=65~vieux))
```

Le pipe

Le pipe permet d'enchaîner des opérations sur une même base. On n'a pas besoin de rappeler à chaque verbe sur quelle base on travaille. On passe d'une opération à l'autre avec `%>%`

```
pop2<-population %>% select(-Gender)%>%
arrange(age)%>% slice(1:5)
```

produira une base pop2 avec 5 lignes.

group_by

Le verbe group_by permet de constituer des groupes dans l'échantillon et de faire une opération dans chaque groupe, comme calculer une moyenne sur les individus du groupe.

```
pop<-pop %>% group_by(Génération) %>%
mutate(Revenu_par_gen=mean(Revenu)) %>% ungroup()
```

ggplot

ggplot permet de faire de sublimes graphiques On appelle ggplot avec les options principales du graphique, puis on ajoute des éléments au graphique avec `+`

`geom_point` permet de dessiner des points

`geom_line` permet de dessiner des lignes.

`geom_smooth` permet de lisser des données.

Chacune des ces fonctions peut prendre comme option la base (data=), les variables (x= , y=), la couleur (color=" "), la taille (size=)... Il faut mettre les variables dans une fonction `aes` pour *aesthetics*