Exercises lecture 4 – joining reshaping tidying

Paolo Crosetto

octobre 2020

`join_...()` family of functions

first run this setup R code chunk. It will load in your workspace 3 data frames:

- **airports**: avec données sur les aéroports américains
- **flights**: qu'on connait déjà
- **planes**: avec les données pour chaque avion

```
planes <- planes
flights <- flights
airports <- airports
```

Exercice 1

est-ce que les routes plus longues sont desservies par les avions les plus modernes?

notes: utilisez `left_join()` et mergez les dataframes `flights` et `planes`

Exercice 2

combien de vols qui partent des trois aéroport de NY atterrisent dans des destinations au dessus de 1000m s.n.m.?

Exercice 3

concentrez vous sur les vols faits par des avions construits avant 2000. Choisissez les destinations qui se trouvent dans un autre fuseau horaire par rapport à NY. Combien de retard à l'arrivé ont-ils en moyenne par compagnie aérienne?

tidy data: reshape with `pivot_longer()` and `pivot_wider()`

Exercice 4

tidy world_bank_pop dataset so that 'year' is a variable and for each country and each year you have urban population and urban population growh only. Plot as a geom_line the total population for each country over the years.

Exercice 5

use us_rent_income. créez une base de données qui n'a qu'une ligne par état et montre `estimate` et `moe` pour `income` et `rent` chacun dans sa variable spécifique.

Exercice 6

utilisez `flights`. Calculez le retard à l'arrivée moyen par compagnie par mois. Puis visualisez le résultats dans un tableau large, avec une variable identifiant chaque mois.

creating tidy data: `separate()` and `unite()`

Exercice 7: `separate()`

utilisez `world_bank_pop` et créez une variable qui vous permette de distinguer entre indicateurs 'pop' and 'urban' et une autre pour distinguer enre indicateurs de 'total' and 'growth'.

Exercice 8: `unite()`

utilisez le jeu de données `table5` et mergez dans une seule variable 'année' les colones 'century' et 'year'

Exercice 9: `unite()`

utilisez le jeu de données `flights` et créez une variable unique 'date' pour jour, mois et année

récapitulons le tout: babynames

On va ici utiliser le jeu de données `babynames` (dans le package `babynames`).

1. installez le package `babynames`
2. regardez les données. Cela contient quoi?
3. on va faire quelaues exercices.

babynames ex1: 'Mary' (`filter` + `ggplot`)

plottez (geom_line) le nombre de Mary aux EE.UU. sur toute la longueur des données.

```
# install.packages("babynames")
library(babynames)
df <- babynames
```

babynames ex2: 'Mary vs. 'Anna' (`filter` + `ggplot`)

plottez (geom_line) le nombre de Mary et de Anna aux EE.UU. sur toute la longueur des données. Quand est-ce que Anna est devenue plus populaire que Mary (si jamais)? COloriez différemment les lignes pour Anna et Mary

babynames ex3: prénoms de garçons

isolez le prénom le plus populaire pour les garçons pour chaque année. Quel nom était le plus utilisé en 1890? et en 1990?

babynames ex4: dispersion des prénoms

est-ce que les prénoms étaient plus concentrés dans le passé (moins de noms, plus de gens pour chaque nom) qu'aujours'hui? Calculez le nombre de noms, séparément pour hommes et femmes, par année. Plottez les résultats comme une geom_line, colorié par sexe. Est-ce que le nombre de noms augmente ou diminue? plus pour les filles ou pour les garçons?

babynames ex5: tab prénom populaires

créez un tableau avec le prénom le plus populaire (en valeur absolue) pour chaque **décennie** du 20ème siècle, pour les filles et les garçons. Faites un tableau 'large' avec comme variable 'décennie', 'M' et 'F'

babynames ex6: ggplot

créez un plot qui montre, dans un facet (subplot) pour chaque prénom (lignes) et par sèxe (colonnes), le profil au cours du temps des prénoms suivants: "Mary" "John" "Robert" "James" "Linda" "Michael" "David" "Lisa" "Jennifer" "Jessica" "Ashley" "Emily" "Jacob" "Emma" "Isabella" "Sophia" "Noah" "Liam"