

# 2-Datamangement

Thibaut FABACHER

## A rajouter/ Modifier

Mettre des exemples de jointures en sql Modifier la fin sur R

## SQL

### Le langage SQL

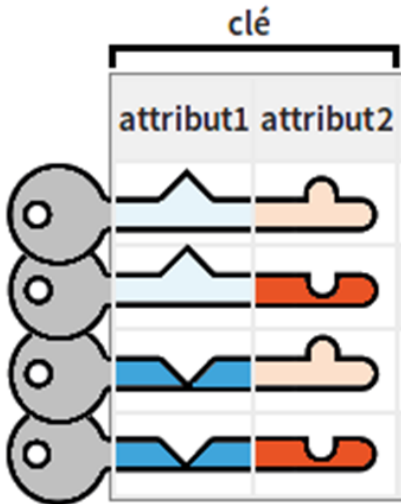
- Perme d'interroger un logiciel sgbdr (système de gestion de base de données)
- SGBR : MySQL, Oracle Database, SQLite...
- ! Différent SQL pour différente SGBR

### Base de données relationnelle

- La relation, chaque ligne est unique
- Possède une clef unique

identifiant	masse	diamètre	couleur
1	151 g	8.3 cm	rouge
3	169 g	9.1 cm	jaune
3	134g	8.0 cm	jaune

## Clefs



clé		nom	prenom	bureau	departement
attribut1	attribut2				
light blue	orange	Dom	Malika	27	ressources humaines
light blue	red	Dirichlet	John	01	marketing
blue	orange	Hati	Hassia	12	marketing
blue	red	Bernard	George	51	maintenance

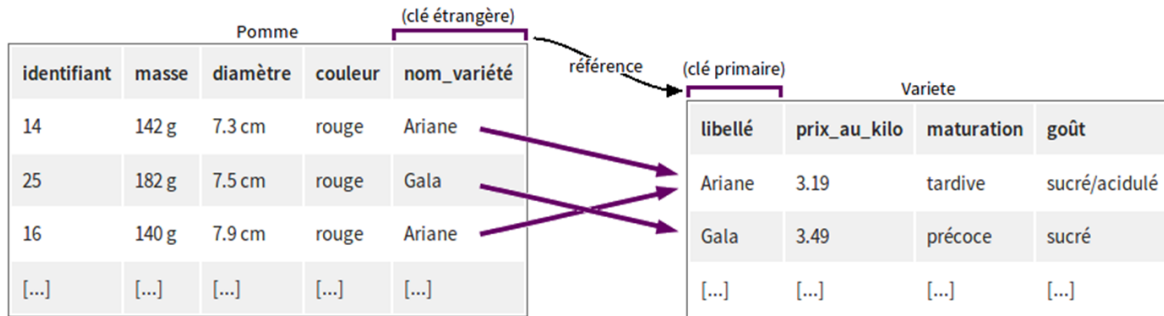
## Intérêt des clefs

libellé	prix_au_kilo	maturation	goût
Ariane	3.19	tardive	sucré/acidulé
Gala	3.49	précoce	sucré
Reinette	3.19	mi-saison	sucré
Boskoop	2.99	mi-saison	acidulé
[...]	[...]	[...]	[...]

## Intérêt des clefs

identifiant	masse	diamètre	couleur	nom_variété
14	142 g	7.3 cm	rouge	Ariane
25	182 g	7.5 cm	rouge	Gala
16	140 g	7.9 cm	rouge	Ariane
[...]	[...]	[...]	[...]	[...]

## Intérêt des clefs

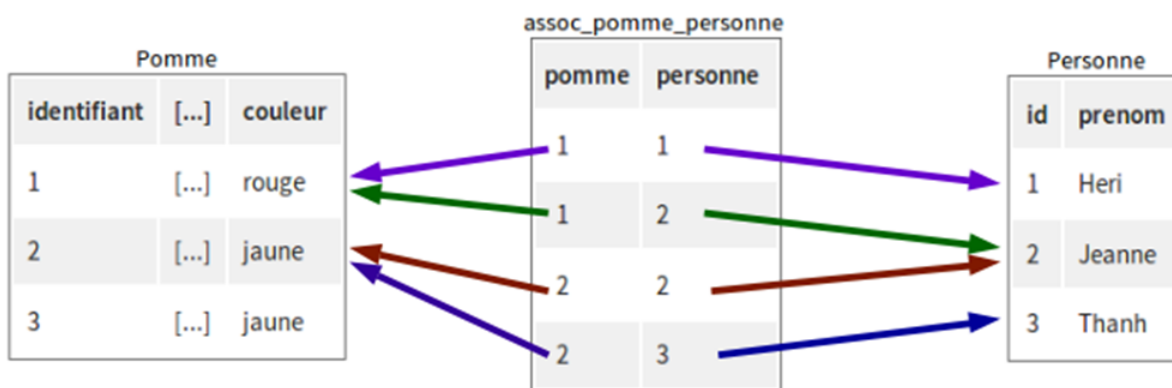


## Intérêt des clefs

Pas de redondance pour le stockage

identifiant	masse	diamètre	couleur	nom_variété	prix_au_kilo	maturation	goût
1	151 g	8.3 cm	rouge	Ariane	3.19	tardive	sucré/acidulé
2	169 g	9.1 cm	jaune	Gala	3.49	précoce	sucré
3	134 g	8.0 cm	jaune	Gala	3.49	précoce	sucré

## Table d'association



## LE SQL

### Langage pour interroger ces bases

```
SELECT *  
FROM pommes ;
```

identifiant	masse	diamètre	couleur
1	151 g	8.3 cm	rouge
2	169 g	9.1 cm	jaune
3	134 g	8.0 cm	jaune

## LE SQL

### Projection

```
SELECT identifiant, masse  
FROM  
Pommes ;
```

identifiant	masse	diamètre	couleur
1	151 g	8.3 cm	rouge
2	169 g	9.1 cm	jaune
3	134 g	8.0 cm	jaune

## SQL

### Restriction

```
SELECT * from  
Pommes  
Where identifiant =1 ;
```

identifiant	masse	diamètre	couleur
1	151 g	8.3 cm	rouge
2	169 g	9.1 cm	jaune

identifiant	masse	diamètre	couleur
3	134 g	8.0 cm	jaune

## SQL

Opérateur	Teste si ...
A = B	A égal à B
A <> B	A différent de B
A > B et A < B	A supérieur à B / A inférieur à B
A >= B et A <= B	A supérieur ou égal à B / A inférieur ou égal à B
A BETWEEN B AND C	A est compris entre B et C
A LIKE 'chaîne de caractères'	(nous verrons cet opérateur dans un prochain chapitre)
A IN (B1, B2, B3, etc.)	A est présent dans la liste (B1, B2, etc.)
A IS NULL	A n'a pas de valeur

- Restriction :  
Opérateur disponible
- Opérateur logique :  
OR, AND, NOT

## Exercice

id_livre	titre	isbn_10	auteur	prix
1	Forteresse digitale	2709626306	Dan Brown	20.5
2	La jeune fille et la nuit	2253237620	Guillaume Musso	21.9
3	T'choupi se brosse les dents	2092589547	Thierry Courtin	5.7
4	La Dernière Chasse	2226439412	Jean-Christophe Grangé	22.9
5	Le Signal	2226319484	Maxime Chattam	23.9

Table 8: Table: Livre

### Exercice

id_livre	titre	isbn_10	auteur	prix
1	Forteresse digitale	2709626306	Dan Brown	20.5
2	La jeune fille et la nuit	2253237620	Guillaume Musso	21.9
3	T'choupi se brosse les dents	2092589547	Thierry Courtin	5.7
4	La Dernière Chasse	2226439412	Jean-Christophe Grangé	22.9
5	Le Signal	2226319484	Maxime Chattam	23.9

Table 9: Table: Livre

Quelle requête utiliser pour afficher l'ensemble des enregistrements de la table ?

...

```
1 SELECT *
2 FROM livres;
```

### Exercice

id_livre	titre	isbn_10	auteur	prix
1	Forteresse digitale	2709626306	Dan Brown	20.5
2	La jeune fille et la nuit	2253237620	Guillaume Musso	21.9
3	T'choupi se brosse les dents	2092589547	Thierry Courtin	5.7
4	La Dernière Chasse	2226439412	Jean-Christophe Grangé	22.9
5	Le Signal	2226319484	Maxime Chattam	23.9

Table 10: Table: Livre

Quelle requête utiliser pour sélectionner uniquement les livres qui ont un **prix strictement supérieur à 20** ?

...

```
1 SELECT *
2 FROM livres
3 WHERE prix > 20;
```

### Exercice

id_livre	titre	isbn_10	auteur	prix
1	Forteresse digitale	2709626306	Dan Brown	20.5
2	La jeune fille et la nuit	2253237620	Guillaume Musso	21.9
3	T'choupi se brosse les dents	2092589547	Thierry Courtin	5.7
4	La Dernière Chasse	2226439412	Jean-Christophe Grangé	22.9
5	Le Signal	2226319484	Maxime Chattam	23.9

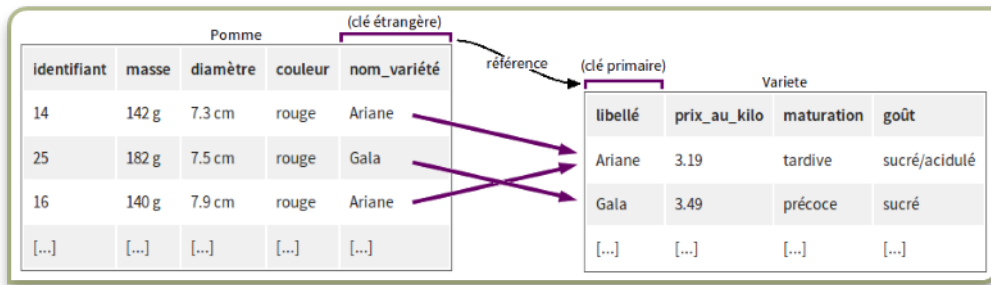
Table 11: Table: Livre

Quelle requête utiliser pour récupérer les livres de la table qui ont **un prix compris entre 20 et 22** ?

...

```
1 SELECT *
2 FROM livres
3 WHERE prix BETWEEN 20 AND 22;
```

## Le SQL



Jointure entre les tables :

```
1 #
2 SELECT *
3 FROM pommes,
4 variete
5 WHERE pommes.nom_variété =variete.libellé ;
```

```
1 #
2 SELECT *
3 FROM pommes
4 JOIN variete ON pommes.nom_variété =variete.libellé ;
```

## Les jointures en SQL

## Les jointures en SQL

## Les jointures en SQL

### Exercice

id_etudiant	prenom	nom
30	Joseph	Biblo
31	Paul	Bismuth
32	Jean	Michel
33	Ted	Bundy



id_etudiant	prenom	nom
34	Caroline	Martinez
35	Joséphine	Henry

Table 12: Table : Etudiant

id	id_examen	id_etudiant	matiere	note
788	45	30	Histoire-Geographie	10.5
789	87	33	Mathématiques	14
790	87	34	Mathématiques	4
791	45	31	Histoire-Geographie	15.5
792	45	32	Histoire-Geographie	8
793	87	31	Mathématiques	14

Table 13: Table : Examen

## Exercice

id_etudiant	prenom	nom
30	Joseph	Biblo
31	Paul	Bismuth
32	Jean	Michel
33	Ted	Bundy
34	Caroline	Martinez
35	Joséphine	Henry

Table 14: Table : Etudiant

id	id_examen	id_etudiant	matiere	note
788	45	30	Histoire-Geographie	10.5
789	87	33	Mathématiques	14
790	87	34	Mathématiques	4
791	45	31	Histoire-Geographie	15.5
792	45	32	Histoire-Geographie	8

id	id_examen	id_etudiant	matiere	note
793	87	31	Mathématiques	14

Table 15: Table : Examen

Quelle requête utiliser pour afficher tous les enregistrement de la table examens avec en plus, si c'est possible, le prenom et le nom de l'étudiant ?

...

```

1  SELECT tbl_ex.*,
2  et.prenom,
3  et.nom
4  FROM examens tbl_ex
5  LEFT JOIN etudiants tbl_et ON tbl_ex.id_etudiant = tbl_et.id_etudiant;
```

## Exercice

id_etudiant	prenom	nom
30	Joseph	Biblo
31	Paul	Bismuth
32	Jean	Michel
33	Ted	Bundy
34	Caroline	Martinez
35	Joséphine	Henry

Table 16: Table : Etudiant

id	id_examen	id_etudiant	matiere	note
788	45	30	Histoire-Geographie	10.5
789	87	33	Mathématiques	14
790	87	34	Mathématiques	4
791	45	31	Histoire-Geographie	15.5
792	45	32	Histoire-Geographie	8
793	87	31	Mathématiques	14

Table 17: Table : Examen

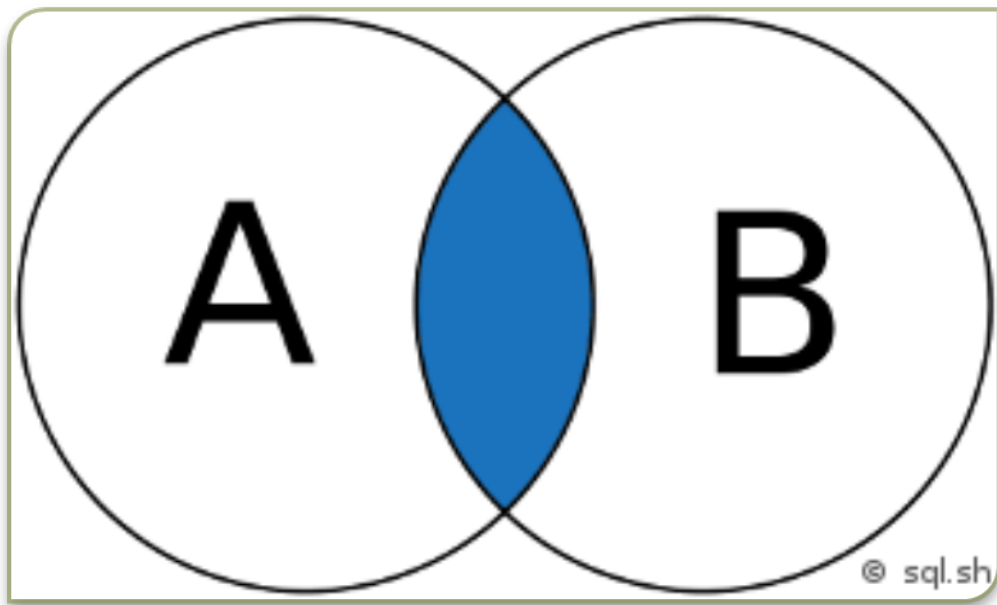


Figure 1: Inner join

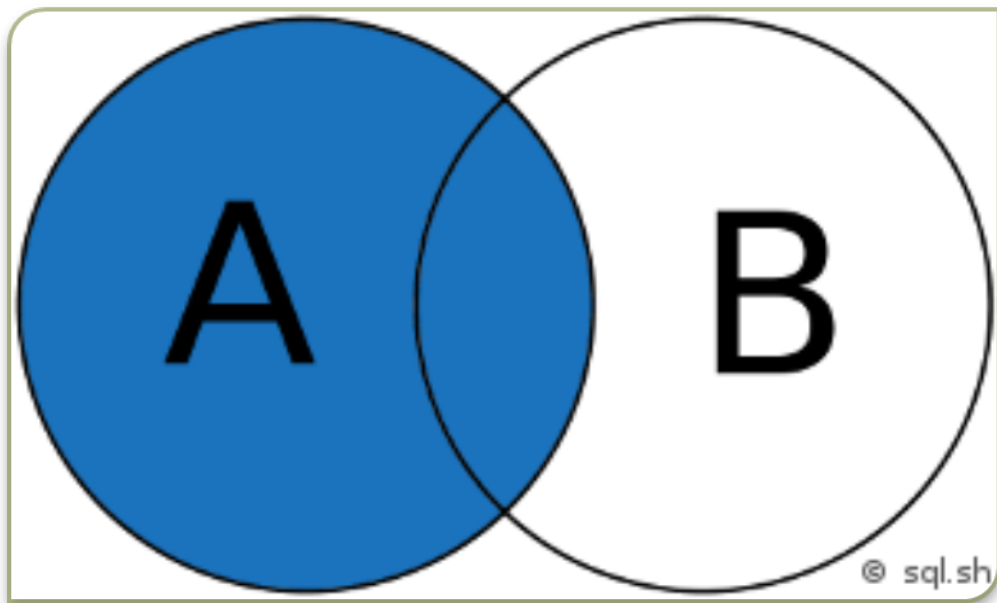


Figure 2: Left Join

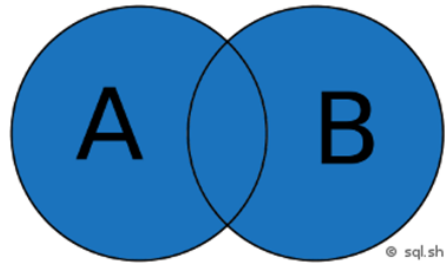


Figure 3: Full join

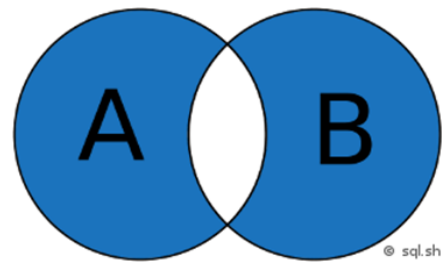


Figure 4: Outer Join

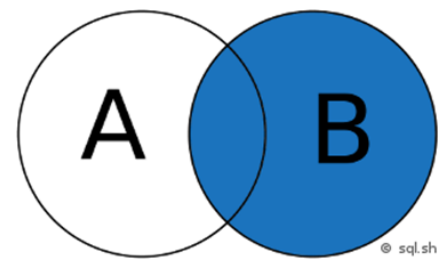


Figure 5: Right Join

Quelle requête utiliser pour afficher les résultats d’histoire des étudiants qui ont un resultats ?

. . .

```
1 SELECT et.prenom,  
2    et.nom,  
3    ex.note  
4 FROM etudiants tbl_et  
5 INNER JOIN examens tbl_ex ON tbl_ex.id_etudiant = tbl_et.id_etudiant  
6 WHERE ex.matiere = "Histoire-Geographie" ;
```

## Data management

### Les données

Les données sont des valeurs de variables quantitatives ou qualitatives appartenant à un ensemble de sujet.

### Données Brutes

- Données disponible dans la base de données d’origine
- Preprocessing nécessaire pour les analyser
- Souvent dans des bases de données relationnelles

### Comment mettre en forme des données ?

#### Notion de tidy data :

- 1 variable par colonne
- 1 une information par ligne
- si tables multiples, clefs de lien présente dans les tables
- 1 ligne avec des noms de colonnes, noms des variables
- 1 table par fichier

## **Données pour l'analyse**

### **Souvent dans un fichier plat :**

- 1 Individu par ligne
- Redondance d'information
- Nécessité de croiser des tables d'origine

### **Noms des variables**

- En minuscule
- Sans accent
- Pas de doublons
- Débutent pas une lettre

### **Données**

- Brutes : pas d'unité
- Descriptive: Vrai/faux, oui/non , 1/0
- Une donnée par variables
- Homogène : attention à la casse

### **Fichier descriptif**

- Information précises sur les variables (unités de mesure)

### **Liste d'instruction :**

#### **Données brutes tidy**

- L'idéal : un script R/python
- En entrée les données brutes
- En sortie les données propre
- Préciser les étapes supplémentaires dans ce script

## Importer un fichier

```
1 # Le plus simple
2 read.csv2(...)
3 BDD<- read.csv2(...)
4
5
6 ## D'autres solutions
7 library(xlsx)
8 read.xlsx(...)
```

## Regarder la structure d'un fichier

```
str(BDD)
```

```
'data.frame':  32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

## Analyser les types de variables

Différents types de variables : - Quantitatives

- Qualitatives
- Dates
- Texte Libre

## Variables Quantitatives

- Stockées sous un format numérique
- Discrètes ou continues
- Possibilité de convertir de «character» à numérique :

. . .

```
1 #  
2 as.numeric(var)
```

## Variables qualitatives

- Variables à plusieurs modalités :
  - Nominale
  - Ordinale
- Représenter sous forme de facteur dans r

. . .

```
1 #  
2 as.factor(var)
```

- Eviter les variables factoriel à plus de 5 modalités
- Cas spécifique si deux modalités : variables binomiales

## Variables qualitatives

Questions à choix multiples dans un questionnaire

Maladies du patient
Diabète; Infarctus
Infarctus; Covid



## Variables qualitatives

Peuvent toujours être séparées en n variables binaires (n = nombre de modalités)

Maladies du patient	Diabète	Infarctus	Covid
Diabète; Infarctus	Oui	Oui	Non
Infarctus; Covid	Non	Oui	Oui

## Dates

- Format anglais : mois/jour/année
- Format Français : jour/mois/année
- Stockées sous forme de nombre par rapport à une date 1er janvier 1900 dans excel

## R

### Installation de package

```
install.package(...)  
  
# aide pour les fonctions  
?install.package  
  
install.packages("readxl")
```

### Chargement du package + de la base de donnée

```
library("readxl")  
  
# Chemin du fichier, remplacer "\" par "/" ou "\\ "  
  
read_excel("~chemin du fichier~/data1.xlsx")
```

```
data1 <- read_excel(".data1.xlsx",
                    1,na =c(" ","","N/A","NA"))

data2 <- read_excel(path = "~chemin du fichier~/data1.xlsx",
                    sheet = 2)
```

## Chargement de la base de donnée

```
data1 <- read_excel("./data1.xlsx",
                    1,na =c(" ","","N/A","NA"))
```

New names:

```
* `` -> ...8
* `Pathologie lié au travail ? [Commentaire]` -> `Pathologie lié au travail ? [Commentaire].
* `Pathologie lié au travail ? [Commentaire]` -> `Pathologie lié au travail ? [Commentaire].
* `Pathologie lié au travail ? [Commentaire]` -> `Pathologie lié au travail ? [Commentaire].
* `Type examen complementaire : [Commentaire]` -> `Type examen complementaire : [Commentaire].
* ...
```

```
data2 <- read_excel(path = "./data1.xlsx",
                    sheet = 2)
```

## Structure de la base de donnée

```
dim(data1)
```

```
[1] 36 45
```

## Structure de la base de donnée

```
str(data1[,1:10])
```

```
tibble [36 x 10] (S3: tbl_df/tbl/data.frame)
 $ ID de la réponse      : num [1:36] 45 46 47 48 49 50 51 52 53 54 ...
 $ Date de soumission    : chr [1:36] "2021-07-19 21:16:11" "2021-07-19 21:31:35" "2021-07-19 21:31:35" "2021-07-19 21:31:35" ...
```

```

$ Dernière page      : num [1:36] 2 2 2 2 NA 2 2 2 2 2 ...
$ Langue de départ   : chr [1:36] "fr" "fr" "fr" "fr" ...
$ Tête de série      : num [1:36] 4.88e+08 1.86e+09 2.10e+09 1.67e+09 1.97e+09 ...
$ Date de lancement  : chr [1:36] "2021-07-19 20:57:49" "2021-07-19 21:18:42" "2021-07-19 21:16:11" "2021-07-19 21:31:35" "2021-07-19 21:47:58" ...
$ Date de la dernière action: chr [1:36] "2021-07-19 21:16:11" "2021-07-19 21:31:35" "2021-07-19 21:47:58" "2021-07-19 22:04:06" "2021-07-19 22:04:06" ...
$ ...8              : logi [1:36] NA NA NA NA NA NA ...
$ ID :              : num [1:36] 30 NA 32 33 NA 34 35 36 37 39 ...
$ Sexe du medcin traitant : chr [1:36] "Féminin" "Féminin" "Masculin" "Féminin" ...

```

## Structure de la base de donnée

```
head(data1)
```

```

# A tibble: 6 x 45
  `ID de la réponse` `Date de soumission` `Dernière page` `Langue de départ`
      <dbl> <chr>                <dbl> <chr>
1         45 2021-07-19 21:16:11             2 fr
2         46 2021-07-19 21:31:35             2 fr
3         47 2021-07-19 21:47:58             2 fr
4         48 2021-07-19 22:04:06             2 fr
5         49 <NA>                      NA fr
6         50 2021-07-27 18:57:00             2 fr
# ... with 41 more variables: `Tête de série` <dbl>, `Date de lancement` <chr>,
#   `Date de la dernière action` <chr>, ...8 <lgl>, `ID :` <dbl>,
#   `Sexe du medcin traitant` <chr>, `Date de la première consultation` <chr>,
#   `Délais de prise en charge (mois)` <chr>, `Délais de RDV (mois)` <dbl>,
#   `Medecin adresseur` <chr>, `Pathologie lié au travail ? [AT]` <chr>,
#   `Pathologie lié au travail ? [Commentaire]...16` <dbl>,
#   `Pathologie lié au travail ? [AT non reconnu]` <chr>, ...

```

## Noms présents dans la base de données

```
names(data1)[1:10]
```

```

[1] "ID de la réponse"      "Date de soumission"
[3] "Dernière page"        "Langue de départ"
[5] "Tête de série"        "Date de lancement"
[7] "Date de la dernière action" "...8"
[9] "ID :"                 "Sexe du medcin traitant"

```

## notion de vecteur

```
c(1,2,3,4)
```

```
[1] 1 2 3 4
```

```
c("a","b","c")
```

```
[1] "a" "b" "c"
```

```
c("a",1,"c")
```

```
[1] "a" "1" "c"
```

## notion de vecteur

```
vecteur<- c("a","b","c")  
vecteur2<- vecteur  
vecteur3<- c(vecteur, vecteur2)
```

## Variable d'une base de données

```
data1$`Pathologie lié au travail ? [AT non reconnu]`
```

```
[1] "Non" "Non" "Non" "Non" NA      "Non" "Non" "Non" "Non" "Non" "Non" "Non"  
[13] "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non"  
[25] "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non" "Non"
```

```
data1[,1]
```

```
# A tibble: 36 x 1
  `ID de la réponse`
      <dbl>
1             45
2             46
3             47
4             48
5             49
6             50
7             51
8             52
9             53
10            54
# ... with 26 more rows
```

## Sélection des variables

### base R

```
data1$`ID de la réponse`
data1[,1:3]
data1[,1:3]
data1[, -1]
data1[, -c(1,3,4)]

data1[, c("ID de la réponse")]
```

## Sélection des variables

### base R

```
names(data1)=="ID.de.la.r?ponse"
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
data1[,!(names(data1)=="ID.de.la.r?ponse")]

data1[,!(names(data1)%in%c(var1, var2 ...)]
```

## Dplyr

- Version classique :

...

```
allerauboulot(preparer(dejeuner(jemeleve(moi))))
var1<- jemeleve(moi)
var2<- déjeuner(var1)
```

...

- Version Dplyr :

```
moi%>%jemeleve%>%dejeuner%>%preparer%>%allerauboulot
data1%>%dim
data1%>%names%>%dput
```

## Selection de variable, Dplyr

```
library(dplyr)

data1 %>% select(var1)
data1 %>% select(c(var1, var2, var3))
data1%>%select(-var1)
```

Attachement du package : 'dplyr'

Les objets suivants sont masqués depuis 'package:stats':

filter, lag

Les objets suivants sont masqués depuis 'package:base':

intersect, setdiff, setequal, union

## Changer le type de données :

### base R

```
as.numeric(c('1','2','3'))
```

```
[1] 1 2 3
```

```
as.character(c(1,2,3))
```

```
[1] "1" "2" "3"
```

```
as.factor(c(2,3,4))
```

```
[1] 2 3 4
```

```
Levels: 2 3 4
```

```
# Attention pas de as.numeric directement sur un as.factor  
as.factor(c(4,3,2))>%as.character()%>%as.numeric()
```

```
[1] 4 3 2
```

## Changer le type de données :

### base R

```
data1$Sexe.du.medcin.traitant<-as.factor(data1$Sexe.du.medcin.traitant)  
data1$Medecin.adresseur<- as.factor(data1$Medecin.adresseur)
```

## Changer le type de données :

### dplyr

```
data1<-data1%>%
  mutate(Sexe.du.medcin.traitant=as.factor(Sexe.du.medcin.traitant),
         Medecin.adresseur = as.factor(Medecin.adresseur))

data1<-data1%>%
  mutate_at(c("Sexe.du.medcin.traitant", "Medecin.adresseur"),
            as.factor)
```

## Modification des variables

### Base R

```
data1$Sexe.du.medcin.traitant[which(data1$Sexe.du.medcin.traitant==45)]<-
  NA
data1$Sexe.du.medcin.traitant%>%droplevels()
data1$Sexe.du.medcin.traitant[which(data1$Sexe.du.medcin.traitant=="f")]<-
  "Féminin"
```

## Modification des variables

### dplyr

```
data1$Sexe.du.medcin.traitant<- as.character(data1$Sexe.du.medcin.traitant)
data1<-data1%>%
  mutate(Sexe.du.medcin.traitant =
         case_when(Sexe.du.medcin.traitant==45~NA_character_,
                   Sexe.du.medcin.traitant=="f"~"Féminin",
                   Sexe.du.medcin.traitant=="m"~"Masculin",
                   TRUE ~Sexe.du.medcin.traitant)%>%as.factor)
```

## Remplacement des données manquantes

### base R

```
data[,indice][is.na(data[,indice])<-0
```



## dplyr

```
data2%>%mutate_at("Age",  
                  function(x) ifelse(is.na(x),0,x))
```

## extraction de termes

```
library(stringr)  
data1$fibromyalgie <-str_detect(data1$var1,"fibromyalgie")
```

## Joindre base de données

```
basefinal<-base1%>%left_join(base2,by =c("id" = "id"))  
basefinal<-base1%>%inner_join(base2,by =c("id" = "id"))  
basefinal<-base1%>%right_join(base2,by =c("id" = "id"))  
basefinal<-base1%>%outer_join(base2,by =c("id" = "id"))
```

## Ecrire la base de données finales

```
write.csv2(basefinal,"C:/Users/enseignant/Desktop/basefinal.csv")
```

