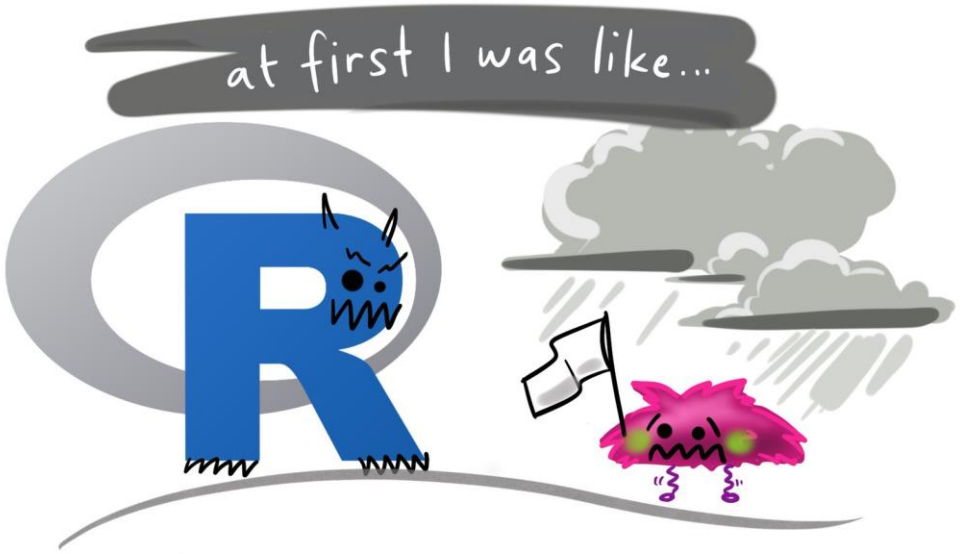
Manipuler un dataset en R avec la suite tidyverse: quelques astuces

Emeline Flajollet



Doctorante en écologie microbienne au LISBP à l'INSA sur la dégradation de la biomasse lignocellulosique par des microorganismes issus de rumen de vache pour la production d'acides gras volatils

Utilisatrice R depuis 3 ans sur des jeux de données (~15000 lignes x ~100 colonnes)

Tous les dessins sont d'Allison Horst
@allison_horst (Twitter)

Doctorante en écologie microbienne au LISBP à l'INSA sur la dégradation de la biomasse lignocellulosique par des microorganismes issus de rumen de vache pour la production d'acides gras volatils

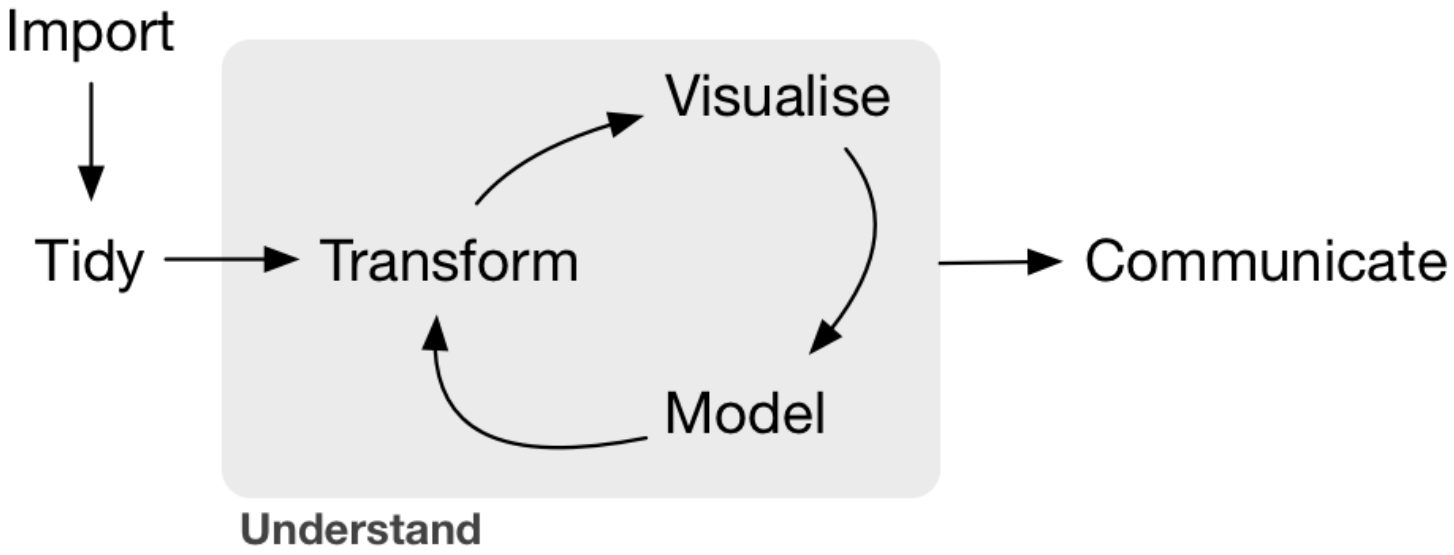
Utilisatrice R depuis 3 ans sur des jeux de données (~15000 lignes x ~100 colonnes)
Mais depuis moins d'1 an, j'utilise la suite Tidyverse.
Fonctions que j'utilise le plus lorsque je manipule mes données :

Tidyr
- gather
- spread
- separate

Dplyr
- %>%
- select
- filter
- group_by
- mutate
- summarise
- arrange

Tous les dessins sont d'Allison Horst
@allison_horst (Twitter)

{dplyr} et {tidyr}



{dplyr} et {tidyr}

~80% du travail des data scientists est consacré au nettoyage et à l'ordonnancement des données brutes

Packages dédiés à la manipulation, l'exploration et les calculs de données, créés par Hadley Wickham (Chief Scientist chez Rstudio)

{dplyr} et {tidyr}

~80% du travail des data scientists est consacré au nettoyage et à l'ordonnancement des données brutes

Packages dédiés à la manipulation, l'exploration et les calculs de données, créés par Hadley Wickham (Chief Scientist chez Rstudio)

row	a	b	c
A	1	4	7
B	2	5	8
C	3	6	9

{dplyr} et {tidyr}

~80% du travail des data scientists est consacré au nettoyage et à l'ordonnancement des données brutes

Packages dédiés à la manipulation, l'exploration et les calculs de données, créés par Hadley Wickham (Chief Scientist chez Rstudio)

Selon Wickham, un jeu de données est propre quand :
– **chaque variable se trouve dans une colonne**
– **chaque observation compose une ligne**
– **les éléments sont contenus dans le même dataset**

row	a	b	c
A	1	4	7
B	2	5	8
C	3	6	9

Happy families are all alike, every unhappy family is unhappy in its own way. **Tolstoy**
Tidy datasets are all alike but every messy dataset is messy in its own way. **Wickham**

{dplyr} et {tidyr}

~80% du travail des data scientists est consacré au nettoyage et à l'ordonnancement des données brutes

Packages dédiés à la manipulation, l'exploration et les calculs de données, créés par Hadley Wickham (Chief Scientist chez Rstudio)

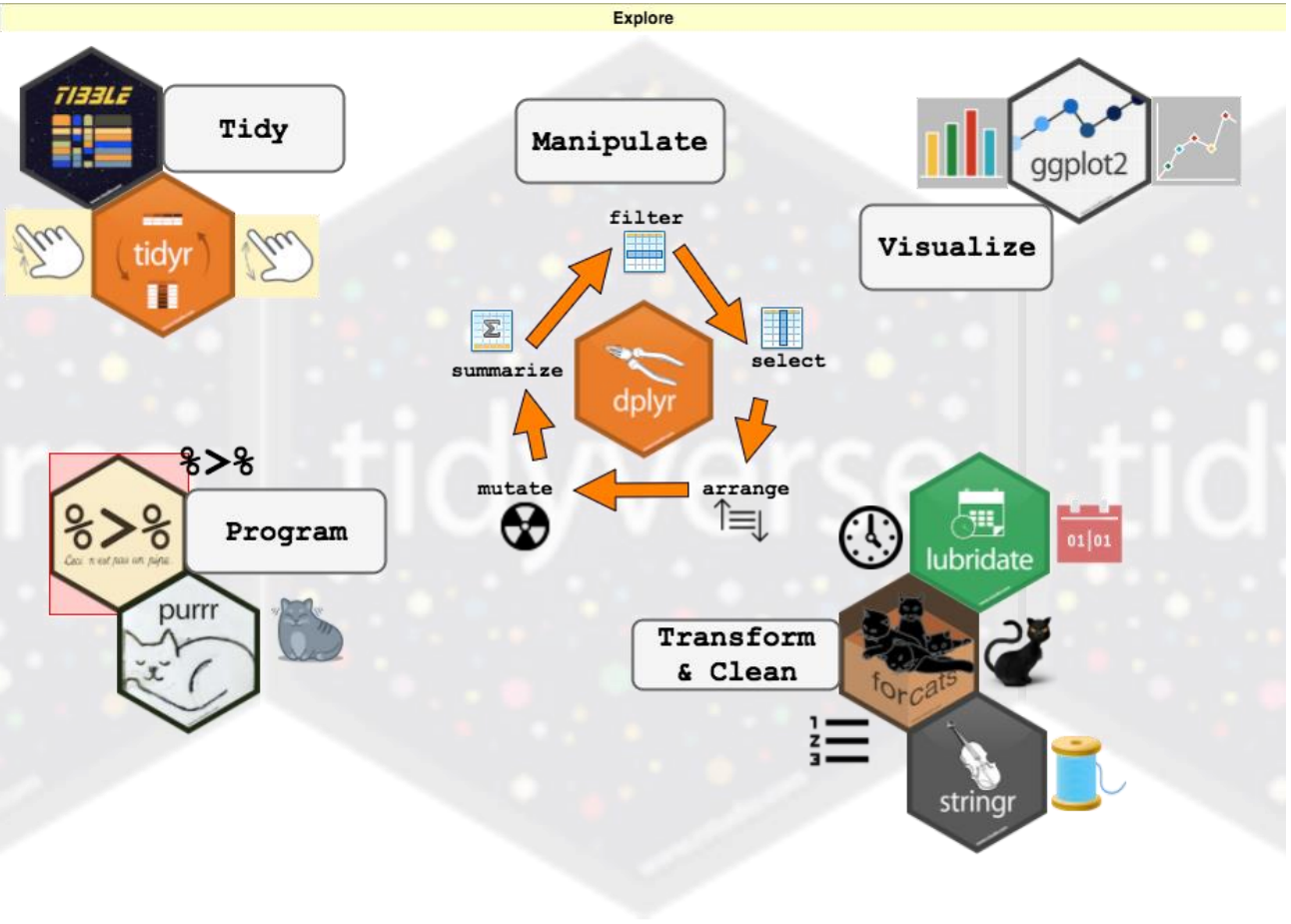
Selon Wickham, un jeu de données est propre quand :
– **chaque variable se trouve dans une colonne**
– **chaque observation compose une ligne**
– **les éléments sont contenus dans le même dataset**

row	a	b	c
A	1	4	7
B	2	5	8
C	3	6	9

row	column	value
A	a	1
B	a	2
C	a	3
A	b	4
B	b	5
C	b	6
A	c	7
B	c	8
C	c	9

Happy families are all alike, every unhappy family is unhappy in its own way. **Tolstoy**
Tidy datasets are all alike but every messy dataset is messy in its own way. **Wickham**

{dplyr} et {tidyr}



Pourquoi les utiliser ?

Grammaire de manipulation commune : un verbe qui prend le data frame comme 1er argument.

verbe (dataframe , …)

☺ Couvre l'essentiel des opérations usuelles à faire sur un data frame

☺ Très intuitif une fois qu'on a compris les bases : on code comme on pense

☺ Rapidité d'exécution

☺ Ecriture claire dans l'enchainement de fonctions

☺ Script simple à comprendre et à transmettre à des débutants ou utilisateurs R occasionnels

Dogs dataset

```
> as.tbl(Dogsdata) #Remplace str et head
# A tibble: 30 x 10
   Numero Nom                 Groupe Race                      Sexe  Presentation Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2
    <int> <chr>                <int> <chr>                     <chr> <chr>                <dbl>          <dbl>          <dbl>          <dbl>
 1      1 Madness                  1 Chien de berger des She~ F     EXC                   36.0           34.8              7              8
 2      2 Lady Stella              8 Cocker spaniel anglais   F     TB                    62.8           65.9             NA             NA
 3      3 Java                     1 Chien de berger des She~ M     TB                    67.3           61.9              5              9
 4      4 Ioup-la-boom             3 West highland white ter~ M     B                     58.4           63.6             NA             NA
 5      5 Estive                   1 Chien de berger des Pyr~ F     TB                    44.0           44.3              5              8
 6      6 Garonne                  1 Chien de berger des Pyr~ F     TB                    59.6           52.4              7              8
 7      7 Chance                   2 Schnauzer nain           F     EXC                   59.4           75.4             NA             NA
 8      8 Maïka                    1 Chien de berger des Pyr~ F     EXC                   34.4           35.4              8.5            9
 9      9 Hunte                    1 Chien de berger des Pyr~ M     EXC                   46.2           46.1              7.5            8
10     10 I'm too fast for y~      1 Chien de berger des She~ M     EXC                   40.6           36.2              4              6
# ... with 20 more rows
```

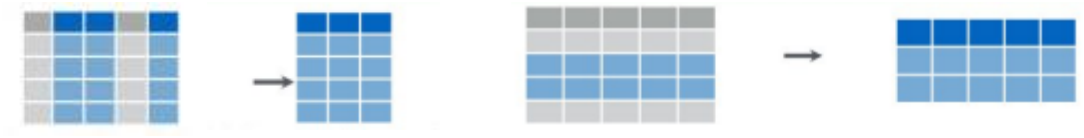
Caractéristiques 30 individus (nom, groupe, race, sexe)

Epreuves (qualificatif, temps, note)

Les principaux verbes {dplyr}

select(df, …) = sélectionner des colonnes

filter(df, …) = filtrer des lignes

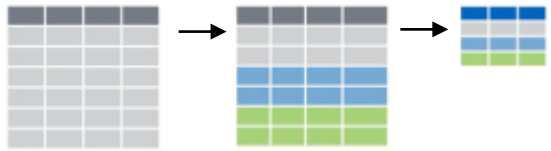


mutate(df, …) = créer une nouvelle variable

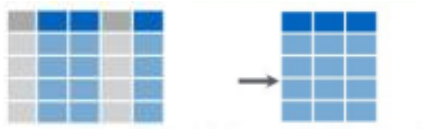
arrange(df, …) = trie les données

summarise(df, …) = résume l'information en une seule ligne

group_by(df, …) = regroupe les observations



select()



select() permet de sélectionner des variables : select(df, colonne)

```
> select(Dogsdata,Race) #permet de sélectionner une variable
                                Race
1     Chien de berger des Shetland
2           Cocker spaniel anglais
3     Chien de berger des Shetland
4      West highland white terrier
5     Chien de berger des Pyrénées
6     Chien de berger des Pyrénées
7                  Schnauzer nain
8     Chien de berger des Pyrénées
9     Chien de berger des Pyrénées
10    Chien de berger des Shetland
...
```

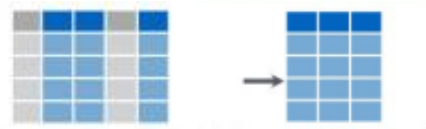
select()

Sélectionner plusieurs variables : select(df, var1:var5) [si contiguës] OU select(df, var1, var3)

```
> select(Dogsdata,Sexe:Moutons_Round1)
   Sexe Presentation Agility_Round1 Agility_Round2 Moutons_Round1
1     F          EXC          36.03          34.78            7.0
2     F           TB          62.84          65.94             NA
3     M           TB          67.29          61.91            5.0
4     M            B          58.41          63.59             NA
5     F           TB          43.96          44.34            5.0
6     F           TB          59.58          52.36            7.0
7     F          EXC          59.42          75.42             NA
8     F          EXC          34.42          35.38            8.5
9     M          EXC          46.20          46.06            7.5
10    M          EXC          40.65          36.23            4.0
...
```

```
> select(Dogsdata,Nom,Agility_Round1)
                  Nom Agility_Round1
1            Madness          36.03
2        Lady Stella          62.84
3               Java          67.29
4       Ioup-la-boom          58.41
5             Estive          43.96
6            Garonne          59.58
7             Chance          59.42
8              Maïka          34.42
9              Hunte          46.20
10 I'm too fast for you          40.65
...
```

select()



Ne pas sélectionner des variables : select(df, -var1, -var8)

```
> select(Dogsdata,-Sexe,-Race)
   Numero                  Nom Groupe Presentation Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2
1       1              Madness      1          EXC          36.03          34.78           7.0           8.0
2       2         Lady Stella       8           TB          62.84          65.94            NA            NA
3       3                Java       1           TB          67.29          61.91           5.0           9.0
4       4        Ioup-la-boom       3            B          58.41          63.59            NA            NA
5       5              Estive       1           TB          43.96          44.34           5.0           8.0
6       6             Garonne       1           TB          59.58          52.36           7.0           8.0
7       7              Chance       2          EXC          59.42          75.42            NA            NA
8       8               Maïka       1          EXC          34.42          35.38           8.5           9.0
9       9               Hunte       1          EXC          46.20          46.06           7.5           8.0
10     10 I'm too fast for you      1          EXC          40.65          36.23           4.0           6.0
...
```

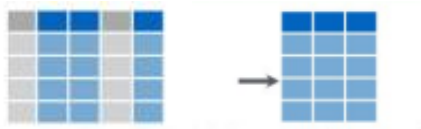
select()

Fonctions assistantes :

Nom de variable commence par : select(df, starts_with(‘Mot’))
Nom de variable finit par : select(df, ends_with(‘Mot’))
Nom de variable contient : select(df, contains(‘ot’))
Expression régulière : select(df, matches(‘._.’))
Liste de noms spécifiée : select(df, one_of(c(‘Mot1’,’Mot2’)))
Variables nommées Mot1, Mot2, Mot3, Mot4 : select(num_range(‘Mot’, 1:4))

Pour ne pas sélectionner :
select(df, -starts_with(‘Mot’))
select(df, -ends_with(‘Mot’))
select(df, -contains(‘ot’))
select(df, -matches(‘._.’))
select(df, -one_of(c(‘Mot1’,’Mot2’)))
select(-num_range(‘Mot’, 1:4))

select()



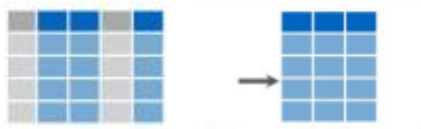
Fonctions assistantes :

Nom de variable commence par : select(df, starts_with(‘Mot’))
Nom de variable finit par : select(df, ends_with(‘Mot’))
Nom de variable contient : select(df, contains(‘ot’))
Expression régulière : select(df, matches(‘._.’))
Liste de noms spécifiée : select(df, one_of(c(‘Mot1’,’Mot2’)))
Variables nommées Mot1, Mot2, Mot3, Mot4 : select(num_range(‘Mot’, 1:4))

Pour anti-sélectionner :
select(df, -starts_with(‘Mot’))
select(df, -ends_with(‘Mot’))
select(df, -contains(‘ot’))
select(df, -matches(‘._.’))
select(df, -one_of(c(‘Mot1’,’Mot2’)))
select(-num_range(‘Mot’, 1:4))

```
> select(Dogsdata, Nom:Sexe, starts_with("Agility"))
                Nom Groupe                          Race Sexe Agility_Round1 Agility_Round2
1          Madness      1 Chien de berger des Shetland    F          36.03          34.78
2       Lady Stella      8       Cocker spaniel anglais    F          62.84          65.94
3              Java      1 Chien de berger des Shetland    M          67.29          61.91
4      Ioup-la-boom      3  West highland white terrier    M          58.41          63.59
5            Estive      1 Chien de berger des Pyrénées    F          43.96          44.34
6           Garonne      1 Chien de berger des Pyrénées    F          59.58          52.36
7           Chance       2                Schnauzer nain    F          59.42          75.42
8            Maïka      1 Chien de berger des Pyrénées    F          34.42          35.38
...
```

select()



Fonctions assistantes :

Nom de variable commence par : select(df, starts_with(‘Mot’))
Nom de variable finit par : select(df, ends_with(‘Mot’))
Nom de variable contient : select(df, contains(‘ot’))
Expression régulière : select(df, matches(‘._.’))
Liste de noms spécifiée : select(df, one_of(c(‘Mot1’,’Mot2’)))
Variables nommées Mot1, Mot2, Mot3, Mot4 : select(num_range(‘Mot’, 1:4))

Pour anti-sélectionner :
select(df, -starts_with(‘Mot’))
select(df, -ends_with(‘Mot’))
select(df, -contains(‘ot’))
select(df, -matches(‘._.’))
select(df, -one_of(c(‘Mot1’,’Mot2’)))
select(-num_range(‘Mot’, 1:4))

```
> select(Dogsdata, Nom:Sexe, starts_with("Agility"))
                Nom Groupe                                Race Sexe Agility_Round1 Agility_Round2
1          Madness      1    Chien de berger des Shetland    F          36.03          34.78
2       Lady Stella      8          Cocker spaniel anglais    F          62.84          65.94
3              Java      1    Chien de berger des Shetland    M          67.29          61.91
4      Ioup-la-boom      3     West highland white terrier    M          58.41          63.59
5            Estive      1    Chien de berger des Pyrénées    F          43.96          44.34
6           Garonne      1    Chien de berger des Pyrénées    F          59.58          52.36
7           Chance       2                  Schnauzer nain    F          59.42          75.42
8            Maïka       1    Chien de berger des Pyrénées    F          34.42          35.38
...
> select(Dogsdata, -matches("_"))
   Numero                Nom Groupe                                Race Sexe Presentation
1       1          Madness      1    Chien de berger des Shetland    F          EXC
2       2       Lady Stella      8          Cocker spaniel anglais    F           TB
3       3              Java      1    Chien de berger des Shetland    M           TB
4       4      Ioup-la-boom      3     West highland white terrier    M            B
5       5            Estive      1    Chien de berger des Pyrénées    F           TB
...
```

filter()

filter() permet de filtrer les lignes du jeu de données qui correspondent à un critère :

filter(df, variable operateur_logique critère)

Opérateurs logiques dans R - ?Comparison et ?base::Logic			
<	Inférieur strictement à	!=	Différent de
>	Supérieur strictement à	%in%	Appartient à
==	Egal à	is.na	Est manquant
<=	Inférieur ou égal à	!is.na	N'est pas manquant
>=	Supérieur ou égal à	&,|,!,xor,any,all	Opérateurs booléens

filter()

filter() permet de filtrer les lignes du jeu de données qui correspondent à un critère :

filter(df, variable operateur_logique critère)

Opérateurs logiques dans R - ?Comparison et ?base::Logic			
<	Inférieur strictement à	!=	Différent de
>	Supérieur strictement à	%in%	Appartient à
==	Egal à	is.na	Est manquant
<=	Inférieur ou égal à	!is.na	N'est pas manquant
>=	Supérieur ou égal à	&,|,!,xor,any,all	Opérateurs booléens

```
> filter(Dogsdata, Presentation %in% c("EXC","TB"))
   Numero                  Nom Groupe                                Race Sexe Presentation Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2
1       1                Madness      1   Chien de berger des Shetland    F          EXC          36.03          34.78           7.0           8.0
2       2           Lady Stella      8         Cocker spaniel anglais    F           TB          62.84          65.94            NA            NA
3       3                  Java      1   Chien de berger des Shetland    M           TB          67.29          61.91           5.0           9.0
4       5                Estive      1   Chien de berger des Pyrénées    F           TB          43.96          44.34           5.0           8.0
5       6               Garonne      1   Chien de berger des Pyrénées    F           TB          59.58          52.36           7.0           8.0
6       7                Chance      2                Schnauzer nain    F          EXC          59.42          75.42            NA            NA
7       8                 Maïka      1   Chien de berger des Pyrénées    F          EXC          34.42          35.38           8.5           9.0
8       9                 Hunte      1   Chien de berger des Pyrénées    M          EXC          46.20          46.06           7.5           8.0
9      10 I'm too fast for you      1   Chien de berger des Shetland    M          EXC          40.65          36.23           4.0           6.0
...
```

filter()

filter() permet de filtrer les lignes du jeu de données qui correspondent à un critère :

filter(df, variable operateur_logique critere)

Opérateurs logiques dans R - ?Comparison et ?base::Logic			
<	Inférieur strictement à	!=	Différent de
>	Supérieur strictement à	%in%	Appartient à
==	Egal à	is.na	Est manquant
<=	Inférieur ou égal à	!is.na	N'est pas manquant
>=	Supérieur ou égal à	&,|,!,xor,any,all	Opérateurs booléens

Filtrer sur plusieurs critères est simple :

filter(df, var1 > 15, var2 < 30) identique à filter(df, var1 > 15 & var2 < 30)

```
> filter(Dogsdata, Agility_Round1 < 50 & Race %in% c("Chien de berger des Pyrénées","Border Collie") )
  Numero          Nom Groupe                          Race Sexe Presentation Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2
1      5       Estive      1 Chien de berger des Pyrénées    F           TB          43.96          44.34            5.0            8
2      8        Maïka      1 Chien de berger des Pyrénées    F          EXC          34.42          35.38            8.5            9
3      9        Hunte      1 Chien de berger des Pyrénées    M          EXC          46.20          46.06            7.5            8
4     12     Flanelle      1 Chien de berger des Pyrénées    F          EXC          38.14          42.84            5.0            6
5     13 Extase Noire      1 Chien de berger des Pyrénées    F          EXC          47.90             NA            6.0            8
6     26         Gaia      1                Border Collie    F          EXC          40.81          41.34            8.0            8
7     30      Guiness      1                Border Collie    M           TB          49.92             NA            7.0            9
> |
```

filter()

filter() permet de filtrer les lignes du jeu de données qui correspondent à un critère :

filter(df, variable operateur_logique critere)

Opérateurs logiques dans R - ?Comparison et ?base::Logic			
<	Inférieur strictement à	!=	Différent de
>	Supérieur strictement à	%in%	Appartient à
==	Egal à	is.na	Est manquant
<=	Inférieur ou égal à	!is.na	N'est pas manquant
>=	Supérieur ou égal à	&,|,!,xor,any,all	Opérateurs booléens

Filtrer sur un critère OU un autre :

filter(df, var1 > critère | var2 < critère)

```
> filter(Dogsdata, Agility_Round1 < 50 | Race %in% c("Chien de berger des Pyrénées","Border Collie") )
  Numero                 Nom Groupe                          Race Sexe Presentation Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2
1      1              Madness      1 Chien de berger des Shetland    F          EXC          36.03          34.78            7.0            8.0
2      5               Estive      1 Chien de berger des Pyrénées    F           TB          43.96          44.34            5.0            8.0
3      6              Garonne      1 Chien de berger des Pyrénées    F           TB          59.58          52.36            7.0            8.0
4      8                Maïka      1 Chien de berger des Pyrénées    F          EXC          34.42          35.38            8.5            9.0
5      9                Hunte      1 Chien de berger des Pyrénées    M          EXC          46.20          46.06            7.5            8.0
6     10 I'm too fast for you      1 Chien de berger des Shetland    M          EXC          40.65          36.23            4.0            6.0
7     11                Ivizy      1             Berger Australien    M          EXC          40.99          41.41            3.0            5.0
...
```

mutate()

mutate() permet de créer une nouvelle variable en gardant les anciennes existantes

mutate(df, nom_nvelle_variable = fonction(ancienne_variable))

Somme

```
> mutate(Dogsdata,Moutons = Moutons_Round1 + Moutons_Round2)
  Numero                 Nom Groupe                          Race Sexe Presentation Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2 Moutons
1      1            Madness      1  Chien de berger des Shetland    F          EXC          36.03          34.78            7.0            8.0    15.0
2      2        Lady Stella      8        Cocker spaniel anglais    F           TB          62.84          65.94             NA             NA      NA
3      3               Java      1  Chien de berger des Shetland    M           TB          67.29          61.91            5.0            9.0    14.0
4      4       Ioup-la-boom      3   West highland white terrier    M            B          58.41          63.59             NA             NA      NA
5      5             Estive      1  Chien de berger des Pyrénées    F           TB          43.96          44.34            5.0            8.0    13.0
...
```

mutate()

mutate() permet de créer une nouvelle variable en gardant les anciennes existantes

mutate(df, nom_nvelle_variable = fonction(ancienne_variable))

Somme

```
> mutate(Dogsdata,Moutons = Moutons_Round1 + Moutons_Round2)
  Numero                 Nom Groupe                              Race Sexe Presentation Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2 Moutons
1      1                Madness      1 Chien de berger des Shetland    F          EXC          36.03          34.78            7.0            8.0    15.0
2      2            Lady Stella      8       Cocker spaniel anglais    F           TB          62.84          65.94             NA             NA      NA
3      3                   Java      1 Chien de berger des Shetland    M           TB          67.29          61.91            5.0            9.0    14.0
4      4           Ioup-la-boom      3  West highland white terrier    M            B          58.41          63.59             NA             NA      NA
5      5                 Estive      1 Chien de berger des Pyrénées    F           TB          43.96          44.34            5.0            8.0    13.0
...
```

Garder le temps le plus rapide

```
> mutate(Dogsdata, Agility = pmin(Agility_Round1, Agility_Round2, na.rm = TRUE))
  Numero                 Nom Groupe                              Race Sexe Presentation Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2 Agility
1      1                Madness      1 Chien de berger des Shetland    F          EXC          36.03          34.78            7.0            8.0   34.78
2      2            Lady Stella      8       Cocker spaniel anglais    F           TB          62.84          65.94             NA             NA   62.84
3      3                   Java      1 Chien de berger des Shetland    M           TB          67.29          61.91            5.0            9.0   61.91
4      4           Ioup-la-boom      3  West highland white terrier    M            B          58.41          63.59             NA             NA   58.41
5      5                 Estive      1 Chien de berger des Pyrénées    F           TB          43.96          44.34            5.0            8.0   43.96
...
```

arrange()

arrange() permet de trier le jeu de données dans l'ordre croissant ou décroissant selon un ou plusieurs critères

arrange(df, variable_to_sort_by) [!] Par défaut : ordre croissant

```
> arrange(Dogsdata, Agility)
```

	Numero	Nom	Groupe	Race	Sexe	Presentation	Agility_Round1	Agility_Round2	Moutons_Round1	Moutons_Round2	Moutons	Agility
1	8	Maïka	1	Chien de berger des Pyrénées	F	EXC	34.42	35.38	8.5	9.0	17.5	34.42
2	1	Madness	1	Chien de berger des Shetland	F	EXC	36.03	34.78	7.0	8.0	15.0	34.78
3	16	Gallway	1	Chien de berger des Shetland	F	EXC	35.95	36.98	4.0	5.0	9.0	35.95
4	10	I'm too fast for you	1	Chien de berger des Shetland	M	EXC	40.65	36.23	4.0	6.0	10.0	36.23
5	15	Minta	1	Mudi (Chien de berger hongrois)	F	EXC	36.97	NA	5.0	7.0	12.0	36.97

...

NB : Les NA sont placées en fin de tableau, peu importe l'ordre du tri

arrange()

arrange() permet de trier le jeu de données dans l'ordre croissant ou décroissant selon un ou plusieurs critères

arrange(df, variable_to_sort_by) [!] Par défaut : ordre croissant

```
> arrange(Dogsdata, Agility)
```

	Numero	Nom	Groupe	Race	Sexe	Presentation	Agility_Round1	Agility_Round2	Moutons_Round1	Moutons_Round2	Moutons	Agility
1	8	Maïka	1	Chien de berger des Pyrénées	F	EXC	34.42	35.38	8.5	9.0	17.5	34.42
2	1	Madness	1	Chien de berger des Shetland	F	EXC	36.03	34.78	7.0	8.0	15.0	34.78
3	16	Gallway	1	Chien de berger des Shetland	F	EXC	35.95	36.98	4.0	5.0	9.0	35.95
4	10	I'm too fast for you	1	Chien de berger des Shetland	M	EXC	40.65	36.23	4.0	6.0	10.0	36.23
5	15	Minta	1	Mudi (Chien de berger hongrois)	F	EXC	36.97	NA	5.0	7.0	12.0	36.97

...

Ordre décroissant : arrange(df, desc(variable_to_sort_by))

```
> arrange(Dogsdata, desc(Agility))
```

	Numero	Nom	Groupe	Race	Sexe	Presentation	Agility_Round1	Agility_Round2	Moutons_Round1	Moutons_Round2	Moutons	Agility
1	2	Lady Stella	8	Cocker spaniel anglais	F	TB	62.84	65.94	NA	NA	NA	62.84
2	3	Java	1	Chien de berger des Shetland	M	TB	67.29	61.91	5.0	9.0	14.0	61.91
3	7	Chance	2	Schnauzer nain	F	EXC	59.42	75.42	NA	NA	NA	59.42
4	4	Ioup-la-boom	3	West highland white terrier	M	B	58.41	63.59	NA	NA	NA	58.41
5	19	Guiness	7	Epagneul breton	M	B	56.04	56.08	NA	NA	NA	56.04

...

NB : Les NA sont placées en fin de tableau, peu importe l'ordre du tri

summarise()

summarise() permet de « résumer » son jeu de données en une seule information, comme la moyenne, l'écart-type, l'effectif…

summarise(df, nouvelle_var = mean(var))

```
> summarise (Dogsdata,Moyenne_Moutons=mean(Moutons,na.rm = TRUE))
  Moyenne_Moutons
1        12.73684
```

summarise()

summarise() permet de « résumer » son jeu de données en une seule information, comme la moyenne, l'écart-type, l'effectif…

summarise(df, nouvelle_var = mean(var))

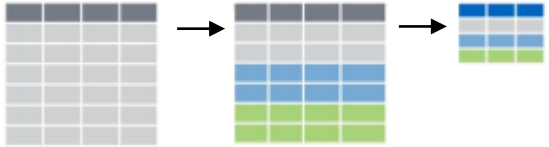
```
> summarise (Dogsdata,Moyenne_Moutons=mean(Moutons,na.rm = TRUE))
  Moyenne_Moutons
1        12.73684
```

Pour calculer plusieurs informations en même temps :

summarise(df, nouvelle_var1 = mean(var1), nouvelle_var2 = sd(var1), nouvelle_var3=mean(var2))

```
> summarise (Dogsdata,Moyenne_Moutons=mean(Moutons,na.rm = TRUE), EType_Moutons=sd(Moutons,na.rm=TRUE))
  Moyenne_Moutons EType_Moutons
1        12.73684       2.84492
```

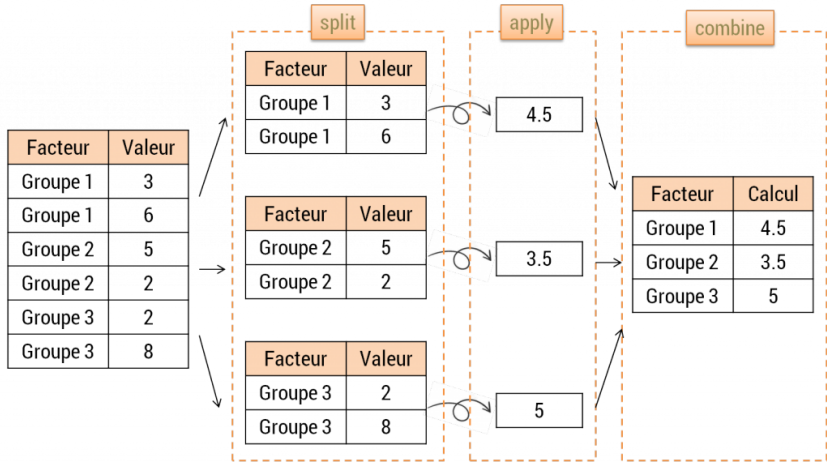
group_by()



group_by(df, variable_à_grouper)

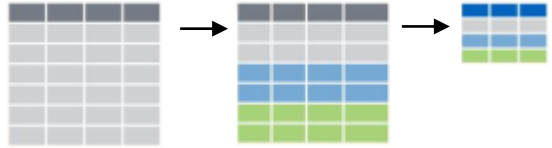
Permet de découper un jeu de données pour réaliser des opérations sur chacun des sous-ensembles afin de les restituer ensuite de façon organisée

Permet de résumer pour différents groupes la moyenne, l'écart type, l'effectif …



Source illustration : www.thinkr.fr

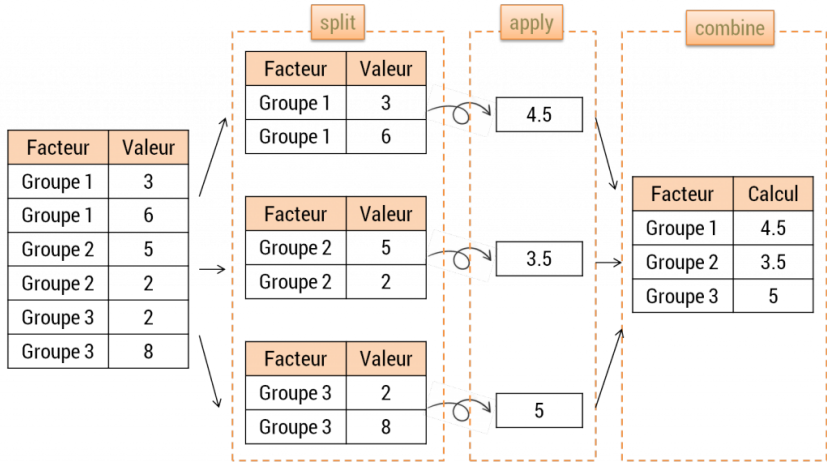
group_by()



group_by(df, variable_à_grouper)

Permet de découper un jeu de données pour réaliser des opérations sur chacun des sous-ensembles afin de les restituer ensuite de façon organisée

Permet de résumer pour différents groupes la moyenne, l'écart type, l'effectif …

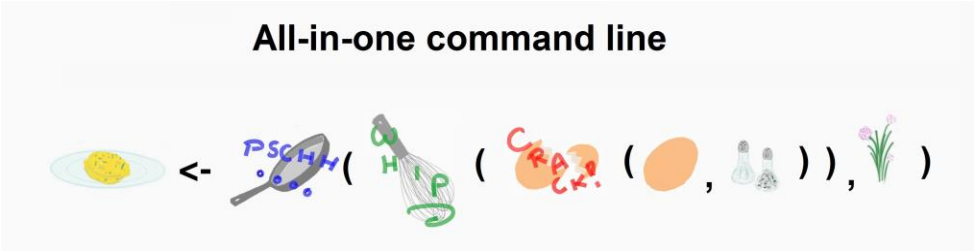



```
> summarise(group_by(Dogsdata, Groupe , Sexe),Moyenne_agility=mean(Agility,na.rm = TRUE), EType_agility=sd(Agility,na.rm = TRUE), Moyenne_moutons=mean(Moutons,na.rm = TRUE),  EType_moutons=sd(Moutons,na.rm = TRUE))
# A tibble: 10 x 6
# Groups:   Groupe [?]
   Groupe Sexe  Moyenne_agility EType_agility Moyenne_moutons EType_moutons
    <int> <chr>           <dbl>         <dbl>           <dbl>         <dbl>
 1      1 F                41.7          6.33            13            2.91
 2      1 M                47.0          7.97            12.4          2.91
 3      2 F                59.4        NaN             NaN           NaN
 4      2 M                42.4        NaN             NaN           NaN
 5      3 M                58.4        NaN             NaN           NaN
 6      7 M                56.0        NaN             NaN           NaN
 7      8 F                62.8        NaN             NaN           NaN
 8      8 M                44.5        NaN             NaN           NaN
 9      9 F                46.0        NaN             NaN           NaN
10      9 M                40.7        NaN             NaN           NaN
>
```

Source illustration : www.thinkr.fr

Maintenant… enchainons les !

%>% {magrittr} permet d'enchainer les opérations, et peut être lu comme « ensuite » ou « puis » comme une recette de cuisine. Attention à l'ordre des opérations

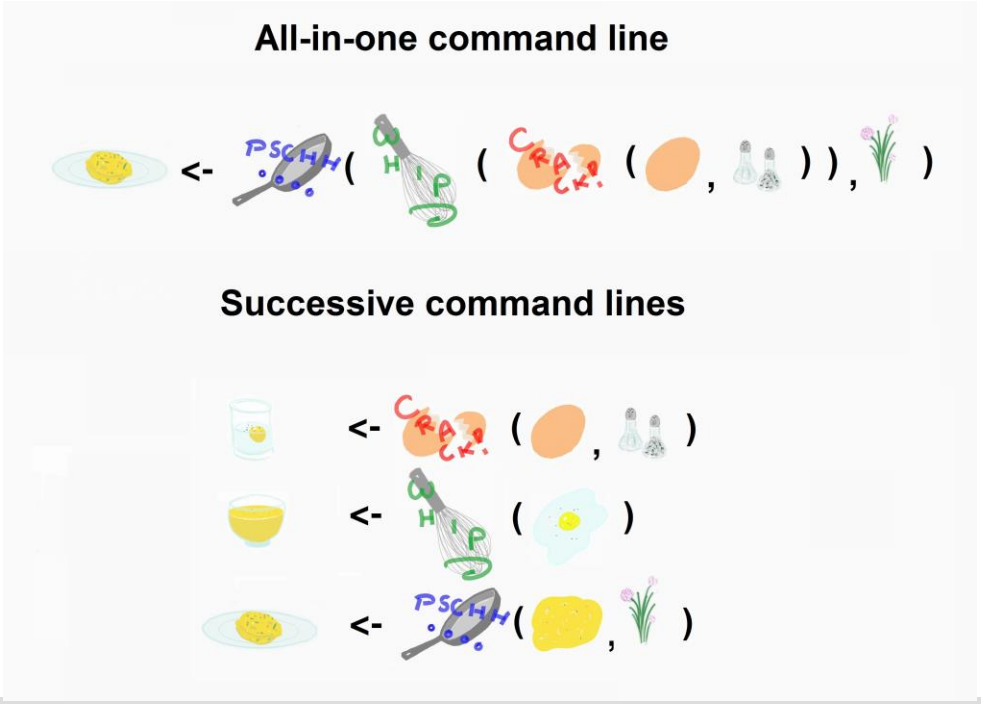


Astuce : CTRL+SHIFT+M fait le symbole %>%

Recette de cuisine par R :
http://perso.ens-lyon.fr/lise.vaudor/utiliser-des-pipes-pour-enchainer-des-instructions/

Maintenant… enchainons les !

%>% {magrittr} permet d'enchainer les opérations, et peut être lu comme « ensuite » ou « puis » comme une recette de cuisine. Attention à l'ordre des opérations

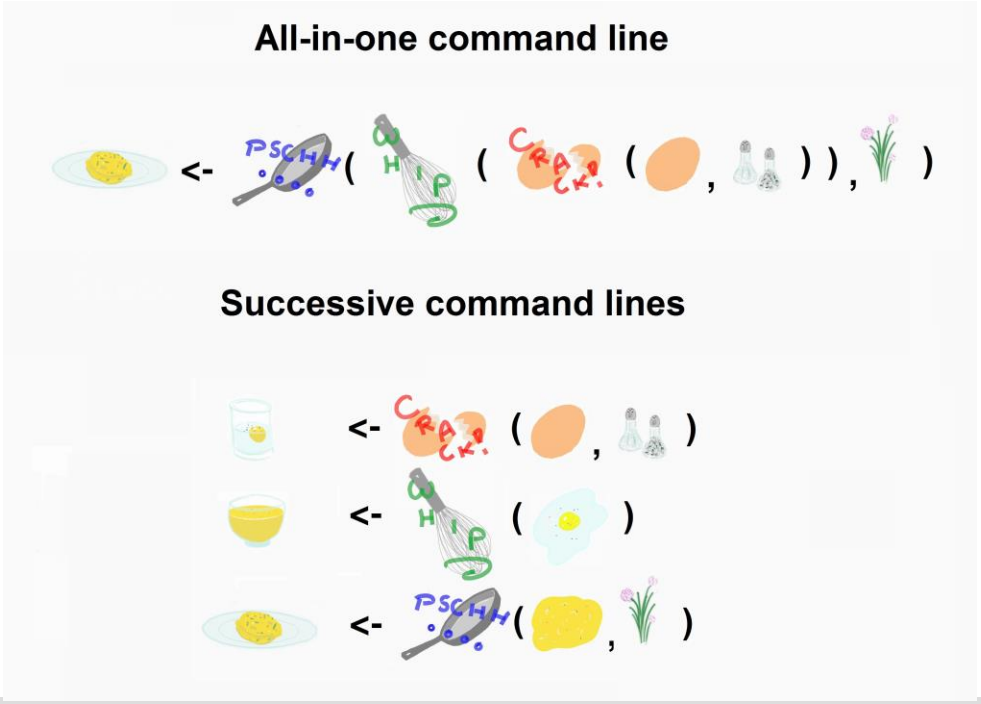


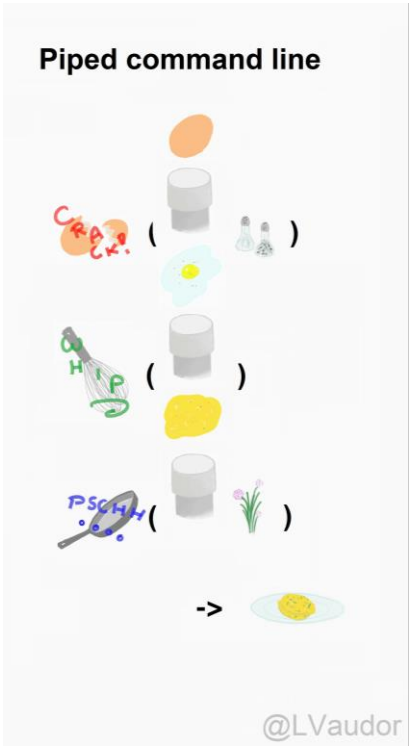
Astuce : CTRL+SHIFT+M fait le symbole %>%

Recette de cuisine par R : http://perso.ens-lyon.fr/lise.vaudor/utiliser-des-pipes-pour-enchainer-des-instructions/

Maintenant… enchainons les !

%>% {magrittr} permet d'enchainer les opérations, et peut être lu comme « ensuite » ou « puis » comme une recette de cuisine. Attention à l'ordre des opérations





Astuce : CTRL+SHIFT+M fait le symbole %>%

Recette de cuisine par R : http://perso.ens-lyon.fr/lise.vaudor/utiliser-des-pipes-pour-enchainer-des-instructions/

Maintenant… enchainons les !

%>% {magrittr} permet d'enchainer les opérations, et peut être lu comme « ensuite » ou « puis » comme une recette de cuisine. Attention à l'ordre des opérations

L'opérateur %>% passe ce qui se trouve à sa gauche, comme premier argument à la fonction qui se trouve à sa droite. df %>% filter (var1 %in% c('A','B','C'))

« En partant de Dogsdata brut, quels sont la moyenne et l'écart type de la note totale à l'épreuve de moutons pour les femelles berger de Pyrénées ayant eu EXC ou TB en fonction de leur présentation?»

Attention à l'ordre des opérations !

Astuce : CTRL+SHIFT+M fait le symbole %>%

Maintenant… enchainons les !

%>% {magrittr} permet d'enchainer les opérations, et peut être lu comme « ensuite » ou « puis » comme une recette de cuisine. Attention à l'ordre des opérations

L'opérateur %>% passe ce qui se trouve à sa gauche, comme premier argument à la fonction qui se trouve à sa droite. df %>% filter (var1 %in% c('A','B','C'))

« En partant de Dogsdata brut, quels sont la moyenne et l'écart type de la note totale à l'épreuve de moutons pour les femelles berger de Pyrénées ayant eu EXC ou TB en fonction de leur présentation?»

```
> Dogsdata1 <- Dogsdata %>%
+     filter(Race == "Chien de berger des Pyrénées" & Sexe == "F" & Presentation %in% c("EXC","TB")) %>%
+   mutate(Somme_moutons = Moutons_Round1+Moutons_Round2) %>%
+   group_by(Presentation) %>%
+   summarise(Mean_moutons=mean(Somme_moutons, na.rm=TRUE),EType_moutons=sd(Somme_moutons, na.rm=TRUE))
> tbl_df(Dogsdata1)
# A tibble: 2 x 3
  Presentation Mean_moutons EType_moutons
  <chr>              <dbl>         <dbl>
1 EXC                 14.2          3.25
2 TB                  14            1.41
> 
```

Attention à l'ordre des opérations !

Astuce : CTRL+SHIFT+M fait le symbole %>%

Mais avec tout ça, on n’a toujours pas de tidy data !

{tidyr}

Qu'est qu'un jeu de données « propre » ? selon Hadley Wickham:

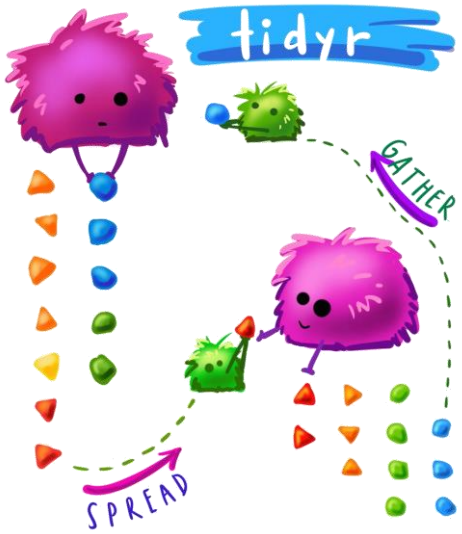
– **chaque variable se trouve dans une colonne**
– **chaque observation compose une ligne**
– **les éléments sont contenus dans le même dataset**

row	a	b	c
A	1	4	7
B	2	5	8
C	3	6	9

Large « *wide* »

row	column	value
A	a	1
B	a	2
C	a	3
A	b	4
B	b	5
C	b	6
A	c	7
B	c	8
C	c	9

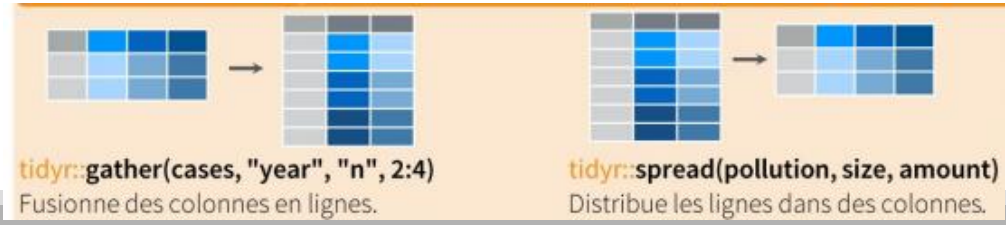
Long



2 verbes principaux :

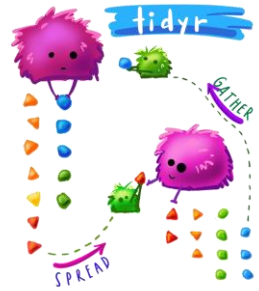
gather(df, …) = fusionner les colonnes en une seule et transforme les données « larges » en format « long »

spread(df, …) = distribue les lignes en colonne et transforme les données en format « long » en larges



Tous les dessins sont d'Allison Horst
@allison_horst (Twitter)

gather() wide to long



Exemple d'un jeu de données non ordonné

	Numero	Nom	Groupe	Race	Sexe
	<int>	<chr>	<int>	<chr>	<chr>
1	1	Madness	1	Chien de berger des Shetland	F
2	2	Lady Stella	8	Cocker spaniel anglais	F
3	3	Java	1	Chien de berger des Shetland	M
4	4	Ioup-la-boom	3	West highland white terrier	M
5	5	Estive	1	Chien de berger des Pyrénées	F
6	6	Garonne	1	Chien de berger des Pyrénées	F
7	7	Chance	2	Schnauzer nain	F
8	8	Maïka	1	Chien de berger des Pyrénées	F
9	9	Hunte	1	Chien de berger des Pyrénées	M
10	10	I'm too fast for you	1	Chien de berger des Shetland	M

Colonnes à indiquer

Presentation	Agility_Round1	Agility_Round2	Moutons_Round1	Moutons_Round2
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
EXC	36.0	34.8	7	8
TB	62.8	65.9	NA	NA
TB	67.3	61.9	5	9
B	58.4	63.6	NA	NA
TB	44.0	44.3	5	8
TB	59.6	52.4	7	8
EXC	59.4	75.4	NA	NA
EXC	34.4	35.4	8.5	9
EXC	46.2	46.1	7.5	8
EXC	40.6	36.2	4	6

'Epreuve'

'Resultat'

```
> Dogsdata_tidy <- Dogsdata %>%
+                  gather (Epreuve,Resultat,Presentation:Moutons_Round2)
> as.tbl(Dogsdata_tidy)
# A tibble: 150 x 7
```

	Numero	Nom	Groupe	Race	Sexe	Epreuve	Resultat
	<int>	<chr>	<int>	<chr>	<chr>	<chr>	<chr>
1	1	Madness	1	Chien de berger des Shetland	F	Presentation	EXC
2	2	Lady Stella	8	Cocker spaniel anglais	F	Presentation	TB
3	3	Java	1	Chien de berger des Shetland	M	Presentation	TB
4	4	Ioup-la-boom	3	West highland white terrier	M	Presentation	B
5	5	Estive	1	Chien de berger des Pyrénées	F	Presentation	TB
6	6	Garonne	1	Chien de berger des Pyrénées	F	Presentation	TB
7	7	Chance	2	Schnauzer nain	F	Presentation	EXC
8	8	Maïka	1	Chien de berger des Pyrénées	F	Presentation	EXC
9	9	Hunte	1	Chien de berger des Pyrénées	M	Presentation	EXC
10	10	I'm too fast for you	1	Chien de berger des Shetland	M	Presentation	EXC

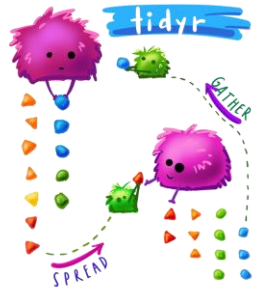
```
# ... with 140 more rows
```

```
as.tbl(tail(Dogsdata_tidy))
A tibble: 6 x 7
```

Numero	Nom	Groupe	Race	Sexe	Epreuve	Resultat
<int>	<chr>	<int>	<chr>	<chr>	<chr>	<chr>
25	Ginger spice	2	Pinscher nain	F	Moutons_Round2	NA
26	Gaia	1	Border Collie	F	Moutons_Round2	8
27	Dali	1	Border Collie	M	Moutons_Round2	NA
28	Jiggle	1	Berger de Beauce	M	Moutons_Round2	5
29	Iellow	8	Golden retriever	M	Moutons_Round2	NA
30	Guiness	1	Border Collie	M	Moutons_Round2	9

Tous les dessins sont d'Allison Horst
@allison_horst (Twitter)

spread() long to wide




```
> as.tbl(Dogsdata_tidy)
# A tibble: 150 x 7
   Numero Nom                 Groupe Race                        Sexe  Epreuve      Resultat
    <int> <chr>                <int> <chr>                       <chr> <chr>        <chr>
 1      1 Madness                  1 Chien de berger des Shetland F     Presentation EXC
 2      2 Lady Stella              8 Cocker spaniel anglais      F     Presentation TB
 3      3 Java                     1 Chien de berger des Shetland M     Presentation TB
 4      4 Ioup-la-boom             3 West highland white terrier M     Presentation B
 5      5 Estive                   1 Chien de berger des Pyrénées F     Presentation TB
 6      6 Garonne                  1 Chien de berger des Pyrénées F     Presentation TB
 7      7 Chance                   2 Schnauzer nain              F     Presentation EXC
 8      8 Maïka                    1 Chien de berger des Pyrénées F     Presentation EXC
 9      9 Hunte                    1 Chien de berger des Pyrénées M     Presentation EXC
10     10 I'm too fast for you     1 Chien de berger des Shetland M     Presentation EXC
# ... with 140 more rows
```

On veut que ‘Epreuve’ deviennent des variables

On veut que les valeurs de chaque épreuve soit le Résultat

```
> Dogsdata_large <- Dogsdata_tidy %>%
+   spread (Epreuve, Resultat)
> tbl_df(Dogsdata_large)
# A tibble: 30 x 10
   Numero Nom                 Groupe Race                        Sexe  Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2 Presentation
    <int> <chr>                <int> <chr>                       <chr> <chr>          <chr>          <chr>          <chr>          <chr>
 1      1 Madness                  1 Chien de berger des Shetland F     36.03          34.78          7              8              EXC
 2      2 Lady Stella              8 Cocker spaniel anglais      F     62.84          65.94          NA             NA             TB
 3      3 Java                     1 Chien de berger des Shetland M     67.29          61.91          5              9              TB
 4      4 Ioup-la-boom             3 West highland white terrier M     58.41          63.59          NA             NA             B
 5      5 Estive                   1 Chien de berger des Pyrénées F     43.96          44.34          5              8              TB
 6      6 Garonne                  1 Chien de berger des Pyrénées F     59.58          52.36          7              8              TB
 7      7 Chance                   2 Schnauzer nain              F     59.42          75.42          NA             NA             EXC
 8      8 Maïka                    1 Chien de berger des Pyrénées F     34.42          35.38          8.5            9              EXC
 9      9 Hunte                    1 Chien de berger des Pyrénées M     46.2           46.06          7.5            8              EXC
10     10 I'm too fast for you     1 Chien de berger des Shetland M     40.65          36.23          4              6              EXC
# ... with 20 more rows
> |
```

Tous les dessins sont d’Allison Horst
@allison_horst (Twitter)

Jusqu'au plot !


```
> Dogsdata1 <- Dogsdata %>%
+     filter(Sexe == "F" & Presentation %in% c("EXC","TB") & Groupe=="1") %>%
+   group_by(Presentation,Race) %>%
+   mutate(Moutons = Moutons_Round1+Moutons_Round2)
> tbl_df(Dogsdata1)
# A tibble: 10 x 11
```

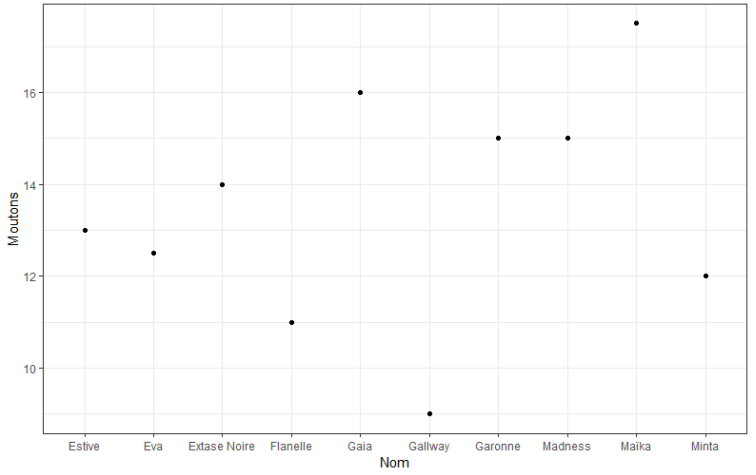
	Numero	Nom	Groupe	Race	Sexe	Presentation	Agility_Round1	Agility_Round2	Moutons_Round1	Moutons_Round2	Moutons
	<int>	<chr>	<int>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1	Madness	1	Chien de berger des Shetland	F	EXC	36.0	34.8	7	8	15
2	5	Estive	1	Chien de berger des Pyrénées	F	TB	44.0	44.3	5	8	13
3	6	Garonne	1	Chien de berger des Pyrénées	F	TB	59.6	52.4	7	8	15
4	8	Maïka	1	Chien de berger des Pyrénées	F	EXC	34.4	35.4	8.5	9	17.5
5	12	Flanelle	1	Chien de berger des Pyrénées	F	EXC	38.1	42.8	5	6	11
6	13	Extase Noire	1	Chien de berger des Pyrénées	F	EXC	47.9	NA	6	8	14
7	15	Minta	1	Mudi (Chien de berger hongrois)	F	EXC	37.0	NA	5	7	12
8	16	Gallway	1	Chien de berger des Shetland	F	EXC	36.0	37.0	4	5	9
9	18	Eva	1	Berger Australien	F	EXC	42.7	45.6	6	6.5	12.5
10	26	Gaia	1	Border Collie	F	EXC	40.8	41.3	8	8	16

```
>
```

Jusqu'au plot !

```
> Dogsdata1 <- Dogsdata %>%
+     filter(Sexe == "F" & Presentation %in% c("EXC","TB") & Groupe=="1") %>%
+   group_by(Presentation,Race) %>%
+   mutate(Moutons = Moutons_Round1+Moutons_Round2)
> tbl_df(Dogsdata1)
# A tibble: 10 x 11
   Numero Nom          Groupe Race                          Sexe  Presentation Agility_Round1 Agility_Round2 Moutons_Round1 Moutons_Round2 Moutons
    <int> <chr>         <int> <chr>                         <chr> <chr>                <dbl>           <dbl>          <dbl>          <dbl>   <dbl>
 1      1 Madness           1 Chien de berger des Shetland  F     EXC                   36.0            34.8              7              8      15
 2      5 Estive            1 Chien de berger des Pyrénées  F     TB                    44.0            44.3              5              8      13
 3      6 Garonne           1 Chien de berger des Pyrénées  F     TB                    59.6            52.4              7              8      15
 4      8 Maïka             1 Chien de berger des Pyrénées  F     EXC                   34.4            35.4              8.5            9      17.5
 5     12 Flanelle          1 Chien de berger des Pyrénées  F     EXC                   38.1            42.8              5              6      11
 6     13 Extase Noire      1 Chien de berger des Pyrénées  F     EXC                   47.9            NA                6              8      14
 7     15 Minta             1 Mudi (Chien de berger hongrois) F   EXC                   37.0            NA                5              7      12
 8     16 Gallway           1 Chien de berger des Shetland  F     EXC                   36.0            37.0              4              5       9
 9     18 Eva               1 Berger Australien             F     EXC                   42.7            45.6              6              6.5    12.5
10     26 Gaia              1 Border Collie                 F     EXC                   40.8            41.3              8              8      16
>
```

```
> (Dogsdata1 <- Dogsdata %>%
+     filter(Sexe == "F" & Presentation %in% c("EXC","TB") & Groupe=="1") %>%
+   group_by(Presentation,Race) %>%
+   mutate(Moutons = Moutons_Round1+Moutons_Round2) %>%
+   ggplot(aes(x=Nom, y= Moutons))+geom_point() + theme_bw())
```



Jusqu'au plot !

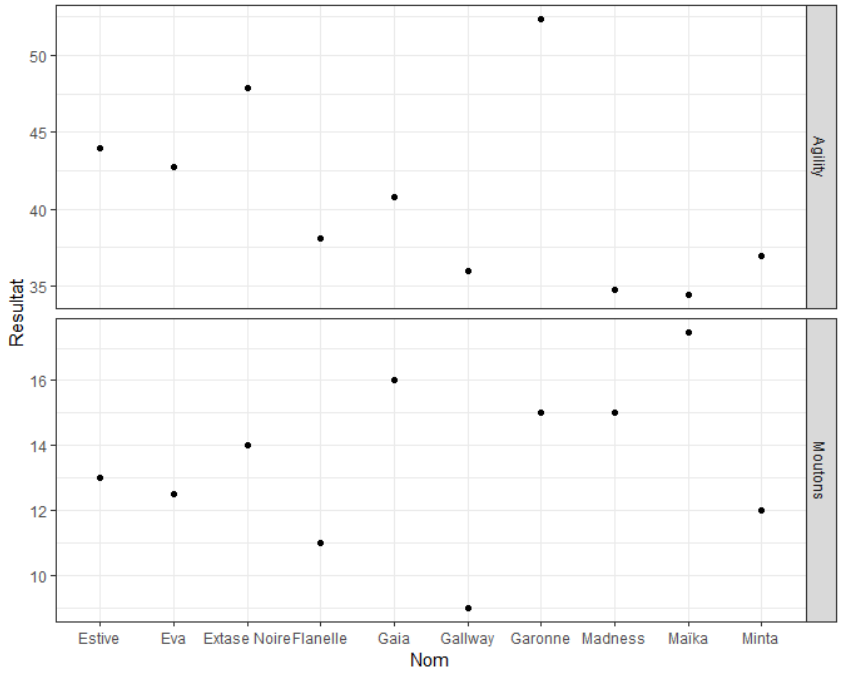
Sans plot :

```
> Dogsdata %>%
+     filter(Sexe == "F" & Presentation %in% c("EXC","TB") & Groupe=="1") %>%
+     group_by(Presentation,Race) %>%
+     mutate(Moutons = Moutons_Round1+Moutons_Round2, Agility = pmin(Agility_Round1, Agility_Round2, na.rm = TRUE)) %>%
+     ungroup() %>%
+     select(Nom:Sexe, Moutons, Agility) %>%
+     gather (Epreuve , Resultat , Moutons:Agility)
# A tibble: 20 x 6
   Nom          Groupe Race                            Sexe  Epreuve Resultat
   <chr>         <int> <chr>                           <chr> <chr>      <dbl>
 1 Madness           1 Chien de berger des Shetland    F     Moutons     15
 2 Estive            1 Chien de berger des Pyrénées    F     Moutons     13
 3 Garonne           1 Chien de berger des Pyrénées    F     Moutons     15
 4 Maïka             1 Chien de berger des Pyrénées    F     Moutons     17.5
 5 Flanelle          1 Chien de berger des Pyrénées    F     Moutons     11
 6 Extase Noire      1 Chien de berger des Pyrénées    F     Moutons     14
 7 Minta             1 Mudi (Chien de berger hongrois) F     Moutons     12
 8 Gallway           1 Chien de berger des Shetland    F     Moutons      9
 9 Eva               1 Berger Australien               F     Moutons     12.5
10 Gaia              1 Border Collie                   F     Moutons     16
11 Madness           1 Chien de berger des Shetland    F     Agility     34.8
12 Estive            1 Chien de berger des Pyrénées    F     Agility     44.0
13 Garonne           1 Chien de berger des Pyrénées    F     Agility     52.4
14 Maïka             1 Chien de berger des Pyrénées    F     Agility     34.4
15 Flanelle          1 Chien de berger des Pyrénées    F     Agility     38.1
16 Extase Noire      1 Chien de berger des Pyrénées    F     Agility     47.9
17 Minta             1 Mudi (Chien de berger hongrois) F     Agility     37.0
18 Gallway           1 Chien de berger des Shetland    F     Agility     36.0
19 Eva               1 Berger Australien               F     Agility     42.7
20 Gaia              1 Border Collie                   F     Agility     40.8
```

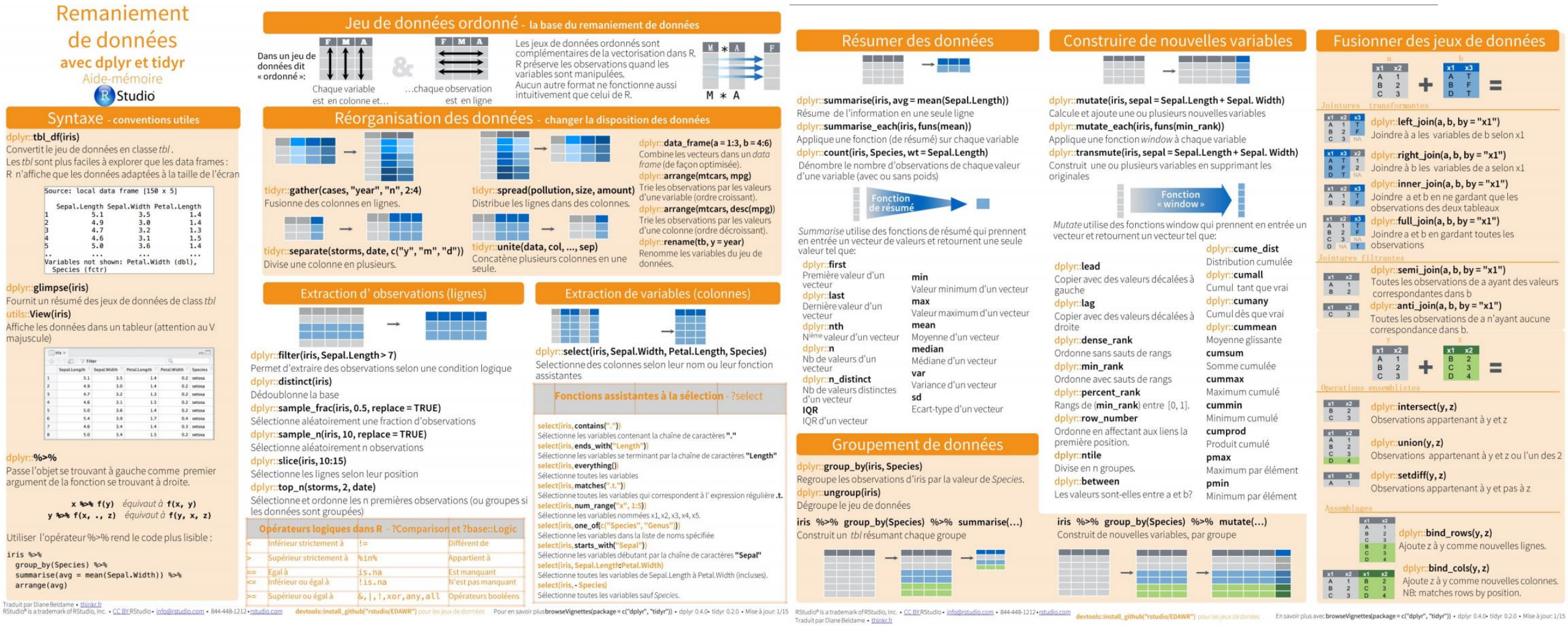
Jusqu'au plot !

Avec plot :

```
> Dogsdata %>%
+     filter(Sexe == "F" & Presentation %in% c("EXC","TB") & Groupe=="1") %>%
+     group_by(Presentation,Race) %>%
+     mutate(Moutons = Moutons_Round1+Moutons_Round2, Agility = pmin(Agility_Round1, Agility_Round2, na.rm = TRUE)) %>%
+     ungroup() %>%
+     select(Nom:Sexe, Moutons, Agility) %>%
+     gather (Epreuve , Resultat , Moutons:Agility) %>%
+     ggplot(aes(x=Nom, y= Resultat)) + facet_grid(Epreuve ~ ., scales = "free_y") + geom_point() + theme_bw()
> |
```



Aide-mémoires – Dplyr et Tidyr



Remaniement de données avec dplyr et tidyr

Aide-mémoire

RStudio

Syntaxe - conventions utiles

dplyr::tbl_df(iris)
Convertit le jeu de données en classe *tbl*.
Les *tbl* sont plus faciles à explorer que les data frames : R n'affiche que les données adaptées à la taille de l'écran

```
Source: local data frame [150 x 5]

   Sepal.Length Sepal.Width Petal.Length
1           5.1         3.5          1.4
2           4.9         3.0          1.4
3           4.7         3.2          1.3
4           4.6         3.1          1.5
5           5.0         3.6          1.4
..          ...         ...          ...
Variables not shown: Petal.Width (dbl),
  Species (fctr)
```

dplyr::glimpse(iris)
Fournit un résumé des jeux de données de class *tbl*

utils::View(iris)
Affiche les données dans un tableau (attention au V majuscule)

dplyr::%>%
Passe l'objet se trouvant à gauche comme premier argument de la fonction se trouvant à droite.

`x %>% f(y)` *équivaut à* `f(x, y)`
`y %>% f(x, ., z)` *équivaut à* `f(y, x, z)`

Utiliser l'opérateur %>% rend le code plus lisible :

```
iris %>%
  group_by(Species) %>%
  summarise(avg = mean(Sepal.Width)) %>%
  arrange(avg)
```

Traduit par Diane Beldame • thinkr.fr
RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com

Jeu de données ordonné - la base du remaniement de données

Dans un jeu de données dit « ordonné » :
Chaque variable est en colonne et… …chaque observation est en ligne

Les jeux de données ordonnés sont complémentaires de la vectorisation dans R. R préserve les observations quand les variables sont manipulées. Aucun autre format ne fonctionne aussi intuitivement que celui de R.

Réorganisation des données - changer la disposition des données

tidyr::gather(cases, "year", "n", 2:4)
Fusionne des colonnes en lignes.

tidyr::spread(pollution, size, amount)
Distribue les lignes dans des colonnes.

tidyr::separate(storms, date, c("y", "m", "d"))
Divise une colonne en plusieurs.

tidyr::unite(data, col, ..., sep)
Concatène plusieurs colonnes en une seule.

dplyr::data_frame(a = 1:3, b = 4:6)
Combine les vecteurs dans un *data frame* (de façon optimisée).
dplyr::arrange(mtcars, mpg)
Trie les observations par les valeurs d'une variable (ordre croissant).
dplyr::arrange(mtcars, desc(mpg))
Trie les observations par les valeurs d'une colonne (ordre décroissant).
dplyr::rename(tb, y = year)
Renomme les variables du jeu de données.

Extraction d' observations (lignes)

dplyr::filter(iris, Sepal.Length > 7)
Permet d'extraire des observations selon une condition logique
dplyr::distinct(iris)
Dédoublonne la base
dplyr::sample_frac(iris, 0.5, replace = TRUE)
Sélectionne aléatoirement une fraction d'observations
dplyr::sample_n(iris, 10, replace = TRUE)
Sélectionne aléatoirement n observations
dplyr::slice(iris, 10:15)
Sélectionne les lignes selon leur position
dplyr::top_n(storms, 2, date)
Sélectionne et ordonne les n premières observations (ou groupes si les données sont groupées)

Opérateurs logiques dans R - ?Comparison et ?base::Logic

<	Inférieur strictement à	!=	Différent de
>	Supérieur strictement à	%in%	Appartient à
==	Egal à	is.na	Est manquant
<=	Inférieur ou égal à	!is.na	N'est pas manquant
>=	Supérieur ou égal à	&, |, !, xor, any, all	Opérateurs booléens

Extraction de variables (colonnes)

dplyr::select(iris, Sepal.Width, Petal.Length, Species)
Selectionne des colonnes selon leur nom ou leur fonction assistantes

Fonctions assistantes à la sélection - ?select

select(iris, contains("."))
Sélectionne les variables contenant la chaîne de caractères "."
select(iris, ends_with("Length"))
Sélectionne les variables se terminant par la chaîne de caractères "Length"
select(iris, everything())
Sélectionne toutes les variables
select(iris, matches("t.t"))
Sélectionne toutes les variables qui correspondent à l'expression régulière .t.
select(iris, num_range("x", 1:5))
Sélectionne les variables nommées x1, x2, x3, x4, x5.
select(iris, one_of(c("Species", "Genus")))
Sélectionne les variables dans la liste de noms spécifiée
select(iris, starts_with("Sepal"))
Sélectionne les variables débutant par la chaîne de caractères "Sepal"
select(iris, Sepal.Length:Petal.Width)
Sélectionne toutes les variables de Sepal.Length à Petal.Width (incluses).
select(iris, - Species)
Sélectionne toutes les variables sauf *Species*.

Pour en savoir plus browseVignettes(package = c("dplyr", "tidyr")) • dplyr 0.4.0 • tidyr 0.2.0 • Mise à jour: 1/15

Résumer des données

dplyr::summarise(iris, avg = mean(Sepal.Length))
Résume de l'information en une seule ligne
dplyr::summarise_each(iris, funs(mean))
Applique une fonction (de résumé) sur chaque variable
dplyr::count(iris, Species, wt = Sepal.Length)
Dénombre le nombre d'observations de chaque valeur d'une variable (avec ou sans poids)

Fonction de résumé

Summarise utilise des fonctions de résumé qui prennent en entrée un vecteur de valeurs et retournent une seule valeur tel que:

dplyr::first
Première valeur d'un vecteur
dplyr::last
Dernière valeur d'un vecteur
dplyr::nth
N^ième valeur d'un vecteur
dplyr::n
Nb de valeurs d'un vecteur
dplyr::n_distinct
Nb de valeurs distinctes d'un vecteur
IQR
IQR d'un vecteur
min
Valeur minimum d'un vecteur
max
Valeur maximum d'un vecteur
mean
Moyenne d'un vecteur
median
Médiane d'un vecteur
var
Variance d'un vecteur
sd
Ecart-type d'un vecteur

Groupement de données

dplyr::group_by(iris, Species)
Regroupe les observations d'iris par la valeur de *Species*.
dplyr::ungroup(iris)
Dégroupe le jeu de données

iris %>% group_by(Species) %>% summarise(…)
Construit un *tbl* résumant chaque groupe

Construire de nouvelles variables

dplyr::mutate(iris, sepal = Sepal.Length + Sepal. Width)
Calcule et ajoute une ou plusieurs nouvelles variables
dplyr::mutate_each(iris, funs(min_rank))
Applique une fonction *window* à chaque variable
dplyr::transmute(iris, sepal = Sepal.Length + Sepal. Width)
Construit une ou plusieurs variables en supprimant les originales

Fonction « window »

Mutate utilise des fonctions window qui prennent en entrée un vecteur et retournent un vecteur tel que:

dplyr::lead
Copier avec des valeurs décalées à gauche
dplyr::lag
Copier avec des valeurs décalées à droite
dplyr::dense_rank
Ordonne sans sauts de rangs
dplyr::min_rank
Ordonne avec sauts de rangs
dplyr::percent_rank
Rangs de (**min_rank**) entre [0, 1].
dplyr::row_number
Ordonne en affectant aux liens la première position.
dplyr::ntile
Divise en n groupes.
dplyr::between
Les valeurs sont-elles entre a et b?
dplyr::cume_dist
Distribution cumulée
dplyr::cumall
Cumul tant que vrai
dplyr::cumany
Cumul dès que vrai
dplyr::cummean
Moyenne glissante
cumsum
Somme cumulée
cummax
Maximum cumulé
cummin
Minimum cumulé
cumprod
Produit cumulé
pmax
Maximum par élément
pmin
Minimum par élément

iris %>% group_by(Species) %>% mutate(…)
Construit de nouvelles variables, par groupe

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com
Traduit par Diane Beldame • thinkr.fr
devtools::install_github("rstudio/EDAWR") pour les jeux de données
En savoir plus avec browseVignettes(package = c("dplyr", "tidyr")) • dplyr 0.4.0 • tidyr 0.2.0 • Mise à jour: 1/15

Fusionner des jeux de données

Jointures transformantes

dplyr::left_join(a, b, by = "x1")
Joindre à a les variables de b selon x1

dplyr::right_join(a, b, by = "x1")
Joindre à b les variables de a selon x1

dplyr::inner_join(a, b, by = "x1")
Joindre a et b en ne gardant que les observations des deux tableaux

dplyr::full_join(a, b, by = "x1")
Joindre a et b en gardant toutes les observations

Jointures filtrantes

dplyr::semi_join(a, b, by = "x1")
Toutes les observations de a ayant des valeurs correspondantes dans b

dplyr::anti_join(a, b, by = "x1")
Toutes les observations de a n'ayant aucune correspondance dans b.

Operations ensemblistes

dplyr::intersect(y, z)
Observations appartenant à y et z

dplyr::union(y, z)
Observations appartenant à y et z ou l'un des 2

dplyr::setdiff(y, z)
Observations appartenant à y et pas à z

Assemblages

dplyr::bind_rows(y, z)
Ajoute z à y comme nouvelles lignes.

dplyr::bind_cols(y, z)
Ajoute z à y comme nouvelles colonnes.
NB: matches rows by position.

Existe (en FR et EN) pour d'autres packages : https://www.rstudio.com/resources/cheatsheets/

Merci de votre attention

Toulouse

dplyr
ggplot2
tidyr

Tous les dessins sont d'Allison Horst
@allison_horst (Twitter)

Tibble

Les tibble sont des data frame

Subset un tibble donne un tibble (pas un vecteur)

Visualisation rapide des 10ères lignes du tibble (contrairement au data frame)

Ne change jamais le nom des variables

Ne crée jamais des row names

Maintenant… enchainons les !

Pour piper avec des opérations dont l'argument 1er n'est pas l'objet de gauche. Utilisez le .

```
df %>% filter (var1 %in% c('A','B','C')) %>%
qplot (x=var1 , data = . , geom='histogram')
```

Astuce : CTRL+SHIFT+M fait le symbole %>%