

# Méthodes de reshaping avec pandas

Francis Wolinski

Master 2 TIDE – mars 2020

# Agenda

1. `crosstab()`, `pivot_table()`
2. `groupby()`
3. `set_index()`, `reset_index()`, `reindex()`
4. `stack()`, `unstack()`
5. `droplevel()`, `swaplevel()`
6. `melt()`
7. `concat()`, `append()`
8. `merge()`, `join()`

# Fonction crosstab()

	A	B	C
0	A1	B1	C1
1	A1	B2	C1
2	A2	B1	C1
3	A2	B1	C1
4	A2	B1	C2
5	A3	B2	C2

`pandas.crosstab(A, B)`

A/B	B1	B2
A1	1	1
A2	3	0
A3	0	1

`pandas.crosstab(B, C)`

B/C	C1	C2
B1	3	1
B2	1	1

# Méthode pivot\_table()

	A	B	C
0	A1	B1	1
1	A1	B2	2
2	A2	B1	3
3	A2	B1	4
4	A2	B1	5
5	A3	B2	6

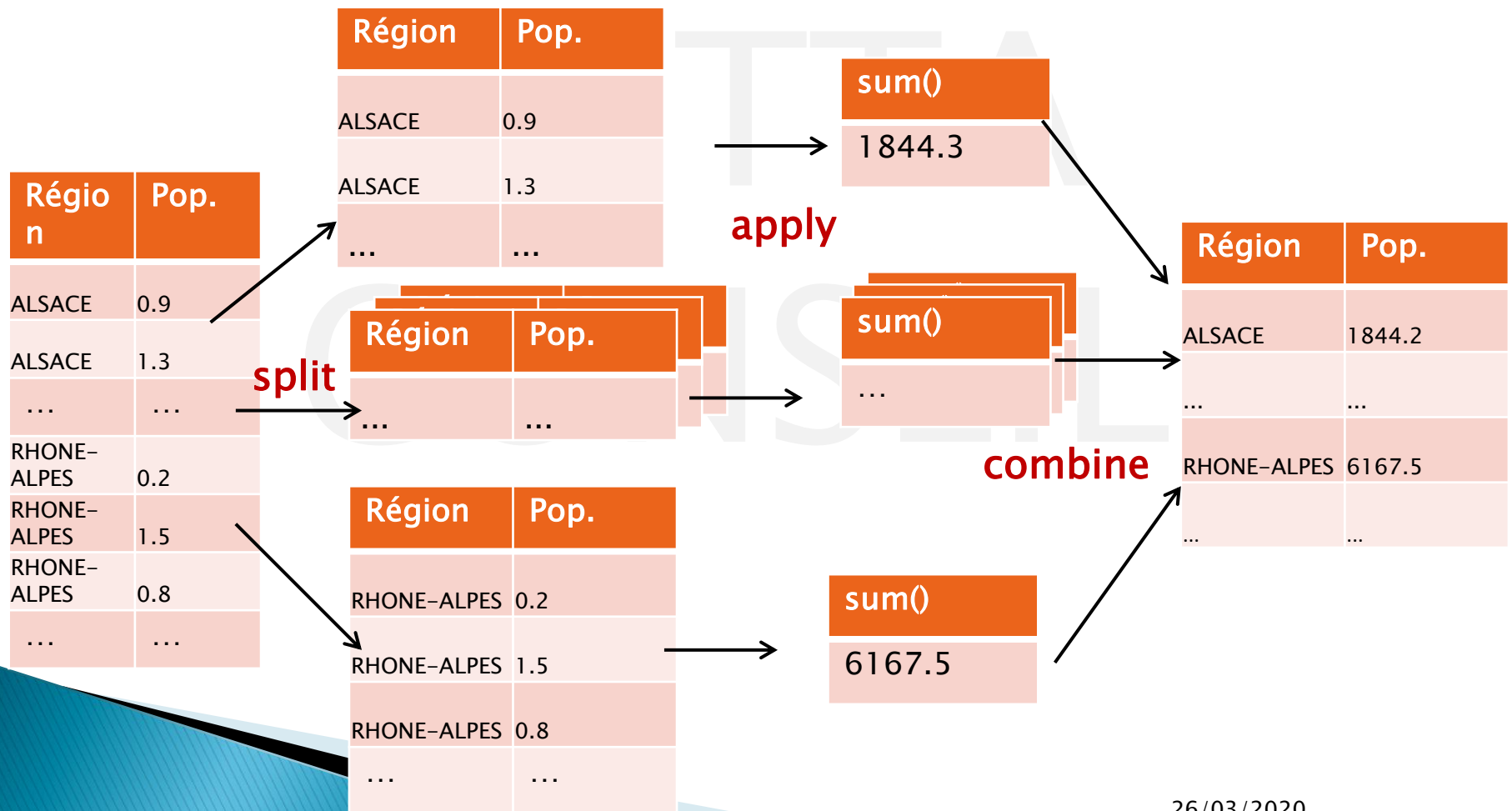
`df.pivot_table(values=C,  
index=A,  
columns=B  
aggfunc=sum)`

A/B	B1	B2
A1	1	2
A2	12	NaN
A3	NaN	6

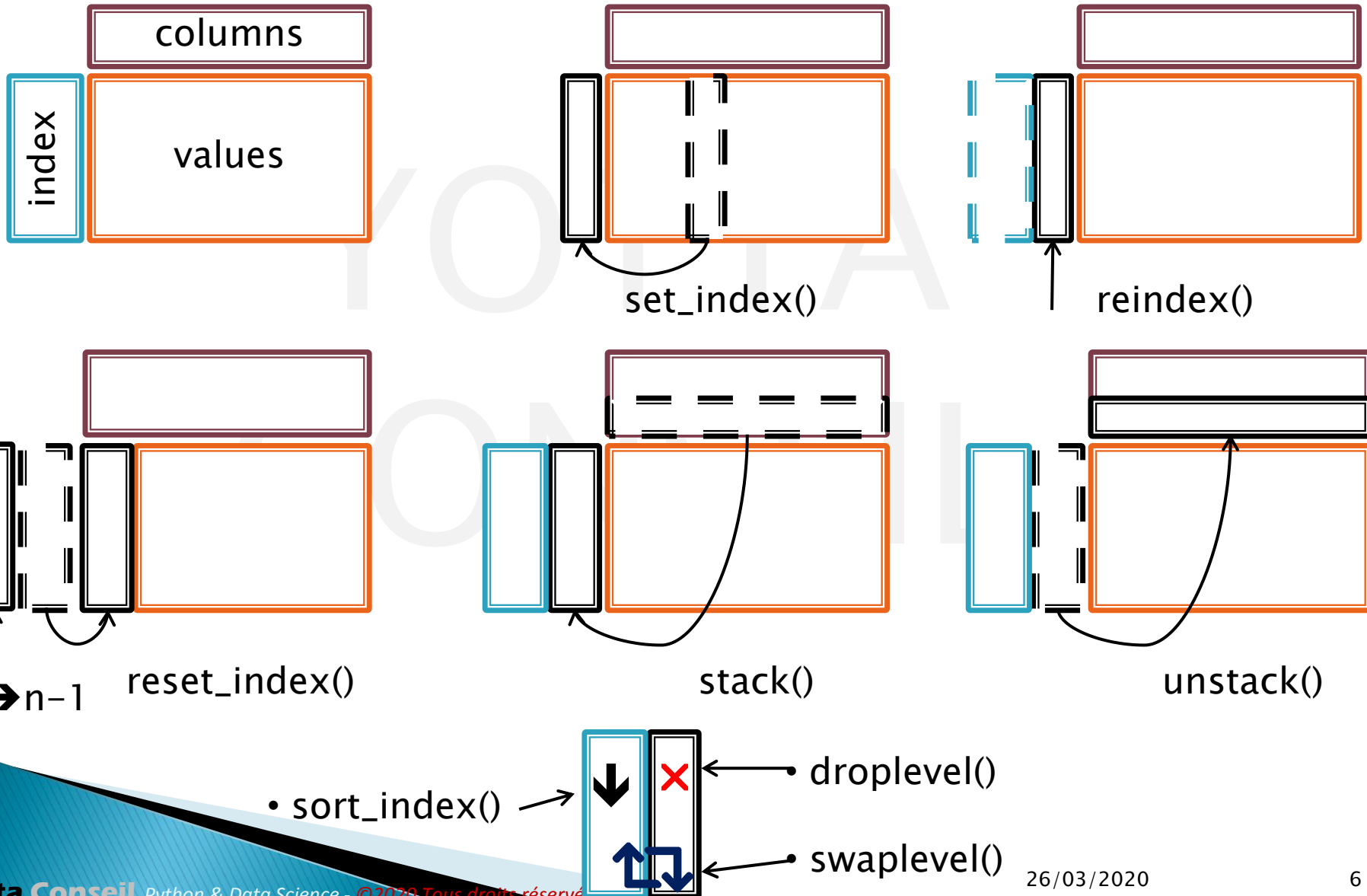
# Méthode groupby()

► **Stratégie *split-apply-combine***

- Éclatement d'un jeu de données en groupes : `groupby()` => `DataFrameGroupBy`
- Application d'une fonction à chacun des groupes
- Regroupement des données traitées



# Vision d'ensemble des méthodes sur les index



# Fonction melt() : wide to long

```
pandas.melt(df,  
            id_vars=[A, B],  
            value_vars=[C1, C2, C3])
```

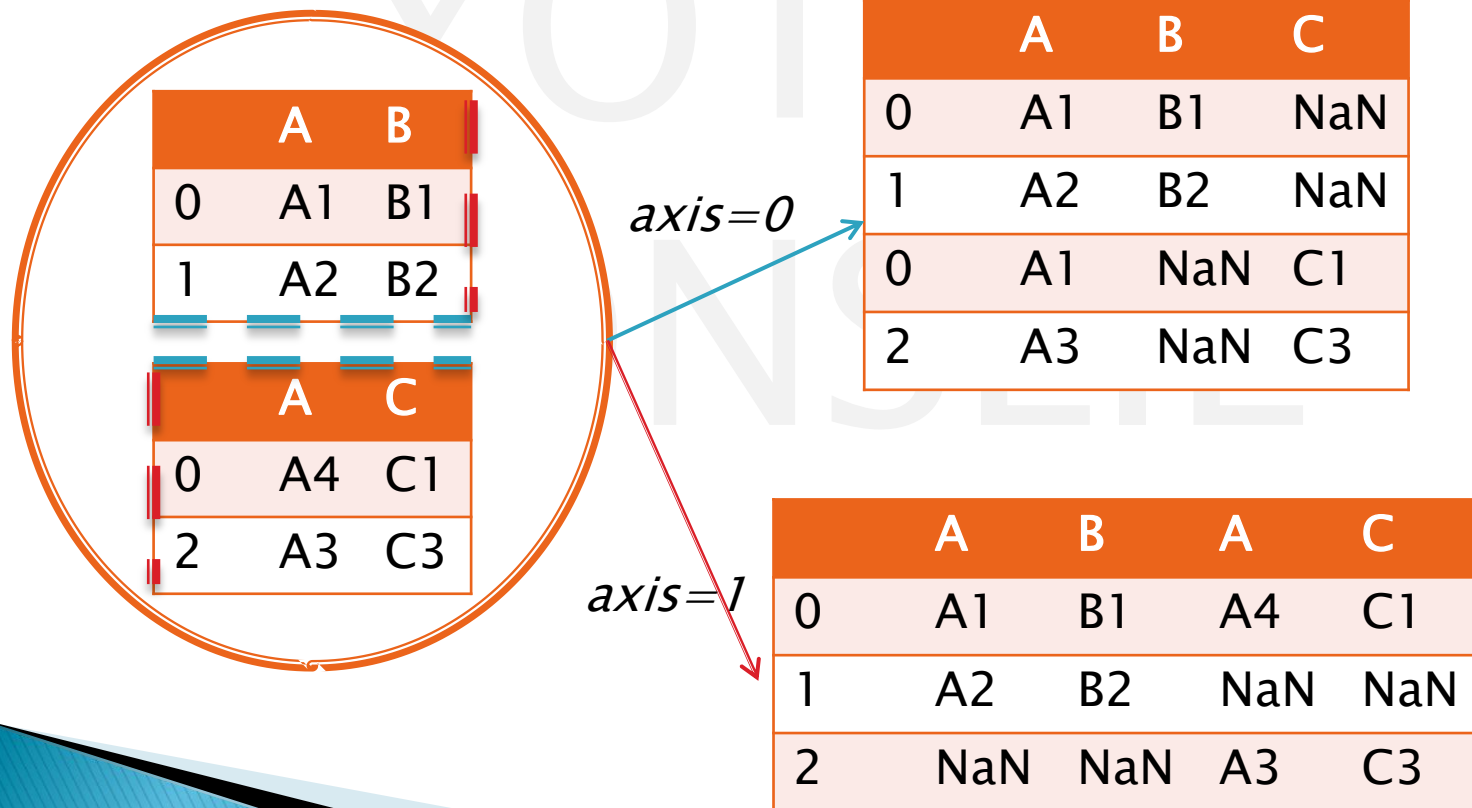
	A	B	C1	C2	C3
0	A1	B1	1	7	13
1	A1	B2	2	8	14
2	A1	B3	3	9	15
3	A2	B1	4	10	16
4	A2	B2	5	11	17
5	A2	B3	6	12	18

	A	B	Variable	Value
0	A1	B1	C1	1
1	A1	B1	C2	7
2	A1	B1	C3	13
...	...	...	...	...
15	A2	B3	C1	6
16	A2	B3	C2	12
17	A2	B3	C3	18

# Fonction concat() – Méthode append()

## Concaténation

- concat() : fonction qui concatène plusieurs *DataFrames* à partir d'une liste
- append() : méthode qui retourne un nouveau *DataFrame* avec l'ajout d'un second





# Fonction merge() – Méthode join()

- ▶ **Jointure (cf. SQL)**
- ▶ `merge()` : fonction qui effectue la jointure de 2 *DataFrames* selon une ou plusieurs colonnes
- ▶ `join()` : méthode qui retourne un nouveau *DataFrame* en effectuant la jointure avec un second *DataFrame*

