

# **Méthodologie Projet 8**

[1 - Téléchargement et prise de connaissance des fichiers fournis pour l'exécution du projet.](#)

[2 - Analyse des dataframes.](#)

[3 - Anomalies et modifications apportées.](#)

[3.1 - management.](#)

[3.1.1 - Anomalies.](#)

[3.1.2 - Modifications liées aux anomalies.](#)

[3.1.3 - Modifications \(hors anomalie\).](#)

[3.2 - mortality.](#)

[3.2.1 - Anomalies.](#)

[3.2.2 - Modifications liées aux anomalies.](#)

[3.2.3 - Modifications \(hors anomalie\).](#)

[3.3 - politic.](#)

[3.3.1 - Anomalies.](#)

[3.3.2 - Modifications liées aux anomalies](#)

[3.3.3 - Modifications \(hors anomalie\) :](#)

[3.4 - population.](#)

[3.4.1 - Anomalies.](#)

[3.4.2 - Modifications liées aux anomalies.](#)

[3.4.3 - Modifications \(hors anomalies\).](#)

[3.5 - regions.](#)

[3.5.1 - Anomalies.](#)

[3.5.2 - Modifications liées aux anomalies.](#)

[3.5.3 - Modifications \(hors anomalies\).](#)

[4 - Création de dataframes par granularités.](#)

[4.1 - population](#)

[4.1.1 - population\\_male.](#)

[4.1.2 - population\\_female.](#)

[4.1.3 - population\\_rurale.](#)

[4.1.4 - population\\_urbaine.](#)

[4.1.5 - population\\_totale.](#)

[4.2 - mortality.](#)

[4.2.1 - mortality\\_female.](#)

[4.2.2 - mortality\\_male.](#)

[4.2.3 - mortality\\_total.](#)

[4.3 - management.](#)

[4.3.1 - management\\_rural.](#)

[4.3.2 - management\\_urban.](#)

[4.3.3 - management\\_total.](#)

[5 - Ajout de données externes.](#)

[6 - Tableau Prep Builder.](#)

[6.1 - jointure des tables "population"](#)  
    [6.1.1 - jointures 1.](#)  
    [6.1.2 - jointures 3.](#)  
    [6.1.3 - jointures 5.](#)  
    [6.1.4 - jointures 7.](#)  
[6.2 - jointure des tables "management"](#)  
    [6.2.1 - jointures 9.](#)  
    [6.2.3 - jointures 11.](#)  
[6.3 - jointure de la table "mortality"](#)  
    [6.3.1 - jointures 17.](#)  
    [6.3.2 - jointures 19.](#)  
[6.4 - jointure 13.](#)  
[6.5 - jointure de la table "politic"](#)  
    [6.5.1 - jointures 15.](#)  
[6.6 - jointure 21.](#)  
[6.7 - jointure de la table "regions"](#)  
[6.8 - jointures 23.](#)  
[6.9 - jointures 25.](#)  
[6.10 - jointures 27.](#)  
[6.11 - jointures 29.](#)  
[6.12 - jointures 31.](#)  
[6.13 - jointures 33.](#)  
[6.14 - Sortie.](#)

# **1 - Téléchargement et prise de connaissance des fichiers fournis pour l'exécution du projet.**

Utilisation de Python et de sa librairie Pandas via Jupyter notebook et Spyder (en parallèle).

- BasicAndSafelyManagedDrinkingWaterServices.csv	→	<b><i>management</i></b>
- MortalityRateAttributedToWater.csv	→	<b><i>mortality</i></b>
- PoliticalStability.csv	→	<b><i>politic</i></b>
- Population.csv	→	<b><i>population</i></b>
- RegionCountry.csv	→	<b><i>regions</i></b>

## **2 - Analyse des dataframes.**

Après avoir parcouru et analysé les données des dataframes, j'ai pu me rendre compte que ces derniers comportent plusieurs anomalies que j'ai rectifiées en même temps que les changements faits dans une optique de confort d'utilisation des données (renommage de séries, etc.)

## 3 - Anomalies et modifications apportées.

### 3.1 - management.

- Aperçu avant modifications :

Indice	Year	Country	Granularity	Population using at least basic drinking-water services (%)	Population using safely managed drinking-water services (%)
0	2000	Afghanistan	Rural	21.61913	nan
1	2000	Afghanistan	Total	27.7719	nan
2	2000	Afghanistan	Urban	49.48745	nan
3	2000	Albania	Rural	81.78472	nan
4	2000	Albania	Total	87.86662	49.29324
5	2000	Albania	Urban	96.35529	nan
6	2000	Algeria	Rural	83.32201	nan
7	2000	Algeria	Total	89.83726	nan
8	2000	Algeria	Urban	94.19543	nan
9	2000	Andorra	Rural	100	nan
10	2000	Andorra	Total	100	90.64
11	2000	Andorra	Urban	100	nan
12	2000	Angola	Rural	21.15264	nan
13	2000	Angola	Total	41.14431	nan

#### 3.1.1 - Anomalies.

- Aucune anomalie particulière n'a été relevée.

#### 3.1.2 - Modifications liées aux anomalies.

- Néant.

#### 3.1.3 - Modifications (hors anomalie).

- Renommage des colonnes :

```
management.rename(columns={
    "Year": "annee",
    "Country": "pays",
    "Granularity": "granularite",
    "Population using at least basic drinking-water services (%)": "ratio acces eau de base",
    "Population using safely managed drinking-water services (%)": "ratio acces eau qualite"
},
inplace = True
)
```

- Passage de l'arrondi des valeurs à 2 décimales et remplacement des valeurs nulles par "0" :

```
management["ratio acces eau de base"] = round(management["ratio acces eau de base"],2)
management["ratio acces eau qualite"] = round(management["ratio acces eau qualite"],2)
management["ratio acces eau de base"] = management["ratio acces eau de base"].fillna(0)
management["ratio acces eau qualite"] = management["ratio acces eau qualite"].fillna(0)
```

- Aperçu après modifications :

Indice	annee	pays	granularite	ratio acces eau de base	ratio acces eau qualite
0	2000	Afghanistan	Rural	21.62	0
1	2000	Afghanistan	Total	27.77	0
2	2000	Afghanistan	Urban	49.49	0
3	2000	Albania	Rural	81.78	0
4	2000	Albania	Total	87.87	49.29
5	2000	Albania	Urban	96.36	0

## 3.2 - mortality.

- Aperçu avant modifications :

Indice	Year	Country	Granularity	Mortality rate attributed to exposure to unsafe WASH services	WASH deaths
0	2016	Afghanistan	Female	15.31193	nan
1	2016	Afghanistan	Male	12.61297	nan
2	2016	Afghanistan	Total	13.92067	4824.353
3	2016	Albania	Female	0.12552	nan
4	2016	Albania	Male	0.2065	nan
5	2016	Albania	Total	0.16641	4.86975
6	2016	Algeria	Female	2.1989	nan
7	2016	Algeria	Male	1.72837	nan
8	2016	Algeria	Total	1.86723	758.21

### 3.2.1 - Anomalies.

- Aucune anomalie particulière n'a été relevée.

### 3.2.2 - Modifications liées aux anomalies.

- Néant.

### 3.2.3 - Modifications (hors anomalie).

- Renommage des colonnes :

```
mortality.rename(columns=
{
    "Year": "annee",
    "Country": "pays",
    "Granularity": "granularite",
    "Mortality rate attributed to exposure to unsafe WASH services": "décès pour 100000 habitants",
    "WASH deaths": "nb morts eau insalubre"
},
inplace = True
)
```

- Passage de l'arrondi des la valeur du taux de mortalité à 2 décimales.
- Passage de l'arrondi des la valeur du nombre de décès à 0 décimale,remplacement des valeurs nulles par "0" puis sa conversion en type entier (int) :

```
mortality["décès pour 100000 habitants"] = round(mortality["décès pour 100000 habitants"],2)
mortality["nb morts eau insalubre"] = round(mortality["nb morts eau insalubre"]).fillna(0).astype(int)
```

- Confirmation de l'unicité de l'année 2016 dans le dataframe:

```
In [17]: mortality["annee"].unique()
Out[17]: array([2016], dtype=int64)
```

- Aperçu après modifications :

	Indice	annee	pays	granularite	décès pour 100000 habitants	nb morts eau insalubre
	0	2016	Afghanistan	Female	15.31	0
	1	2016	Afghanistan	Male	12.61	0
	2	2016	Afghanistan	Total	13.92	4824
	3	2016	Albania	Female	0.13	0
	4	2016	Albania	Male	0.21	0
	5	2016	Albania	Total	0.17	5
	6	2016	Algeria	Female	2.2	0

### 3.3 - politic.

- Aperçu avant modifications :

	Indice	Country	Year	Political_Stability	granularit
	0	Afghanistan	2000	-2.44	Total
	1	Afghanistan	2002	-2.04	Total
	2	Afghanistan	2003	-2.2	Total
	3	Afghanistan	2004	-2.3	Total
	4	Afghanistan	2005	-2.07	Total
	5	Afghanistan	2006	-2.22	Total
	6	Afghanistan	2007	-2.41	Total
	7	Afghanistan	2008	-2.69	Total

### 3.3.1 - Anomalies.

- Après analyse et recoupements des différents dataframes, j'ai pu me rendre compte qu'il y avait un problème de représentativité concernant la **Chine**:  
Suivant quelle source de donnée est consultée, la chine est représentée soit :
  - Dans sa manière globale, en un seul bloc.
  - Scindée en 4 provinces distinctes :
    - Chine continentale.
    - Taïwan.
    - Macao.
    - Hong Kong.
  - Les 2 cumulées.
- Concernant le dataframe "politic", j'ai pu remarquer que la Chine n'était représentée que sous la forme de ses 4 provinces et non en un bloc. Ceci pourrait ne pas constituer un problème, puisque les données sont plus précises, mais au regard de la répartition de ce pays dans les autres fichiers de données, on se rend compte que celui-ci est réparti de la manière suivante :

- management	→	Chine totale.
- mortality	→	Chine totale.
- politic	→	Chine par provinces.
- regions	→	Chine totale.
- population	→	Chine totale + Chine par provinces.
- La chine, dans sa totalité, étant la plus représentée, mais aussi le fait que l'on puisse noter l'absence de ses régions dans d'autres dataframes permettant d'effectuer des analyses en fonction de ces régions, je fais donc le choix de privilégier l'utilisation de la Chine dans sa représentation totale.
- **La Macédoine du Nord** est nommée de 2 manières différentes, suivant le dataframe dans lequel se trouve ce pays:

- management	→	Republic of North Macedonia.
- mortality	→	Republic of North Macedonia.
- politic	→	North Macedonia.
- regions	→	Republic of North Macedonia.
- population	→	North Macedonia.

### 3.3.2 - Modifications liées aux anomalies

- Ayant une absence de valeur pour la stabilité politique pour la Chine dans sa totalité, contrairement à son format par régions, je fais la moyenne du cumul de la stabilité politique de ces dernières, par année et crée un dataframe "politic\_chine" avec les valeurs calculées. J'ajoute ensuite ce df à "politic" avec la méthode "concat":

```
politic_chine = pd.DataFrame({
    "pays":["China","China","China","China","China","China","China","China","China","China",
    "China","China","China","China","China","China","China","China","China"],
    "annee":[2000,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018],
    "stabilite politique":[0.44,0.46,0.54,0.68,0.70,0.58,0.40,0.50,0.43,0.43,0.48,0.50,0.50,0.65,0.66,0.65,0.72,0.66]
})
#utilisation de la méthode "concat()".
politic = pd.concat([politic,politic_chine], ignore_index = True)
```

	Indice	pays	annee	stabilite politique
	0	China	2000	0.44
	1	China	2002	0.46
	2	China	2003	0.54
	3	China	2004	0.68
	4	China	2005	0.7
	5	China	2006	0.58
	6	China	2007	0.4

- Renommage de la Macédoine du Nord :

```
politic["pays"] = politic["pays"].replace({"North Macedonia": "Republic of North Macedonia"})
```

### 3.3.3 - Modifications (hors anomalie) :

- Suppression de la colonne "granularité", inutile car ne comportant qu'une valeur unique.

```
del politic["granularite"]
```

- Renommage des colonnes :

```
politic.rename(columns={
    "Country": "pays",
    "Year": "annee",
    "Political_Stability": "stabilite politique",
    "Granularity": "granularite"
},
inplace = True
)
```

- Aperçu après modifications :

	Indice	pays	annee	stabilite politique
	3524	Zimbabwe	2017	-0.71
	3525	Zimbabwe	2018	-0.71
	3526	China	2000	0.44
	3527	China	2002	0.46
	3528	China	2003	0.54
	3529	China	2004	0.68
	3530	China	2005	0.7



### 3.4 - population.

- Aperçu avant modifications :

	Indice	Country	granularité	Year	Population
	0	Afghanistan	Total	2000	20779.953
	1	Afghanistan	Male	2000	10689.508
	2	Afghanistan	Female	2000	10090.449
	3	Afghanistan	Rural	2000	15657.474
	4	Afghanistan	Urban	2000	4436.282
	5	Afghanistan	Total	2001	21606.988
	6	Afghanistan	Male	2001	11117.754

#### 3.4.1 - Anomalies.

- Comme cité précédemment (anomalies du df "politic", en 3.3.1), ce dataframe comporte le pays "Chine" en 2 styles de représentativité... ce qui fait que celui-ci est comptabilisé 2 fois.

Je fais le choix de conserver ce doublon, en cas de besoin ultérieur, en fonction d'une demande quelconque à venir. Je n'oublierai pas, cependant, de supprimer un des 2 à l'aide de tableau prep. Mais j'aurai, ainsi, une base de données comportant les 2 modèles, si besoin.

- Le soudan est nommé "sudan (former)" de 2000 à 2011, avant de devenir "sudan" de 2012 à 2018, avant que le soudan du sud ne fasse sécession de la république du soudan le 9 Juillet 2011 (suite au référendum d'autodétermination du 9 au 15 Janvier 2011). J'ai donc renommé "sudan (former)" en "sudan", gardant ainsi la correspondance avec les autres fichiers.

- La Macédoine du Nord est nommée "North Macedonia".

#### 3.4.2 - Modifications liées aux anomalies.

- Renommage de la Macédoine du Nord et du Soudan:

```
population["pays"] = population["pays"].replace({"North Macedonia":"Republic of North Macedonia"})
population["pays"] = population["pays"].replace({"Sudan (former)":"Sudan"})
```

#### 3.4.3 - Modifications (hors anomalies).

- Renommage des colonnes :

```
population.rename(columns={
    "Country":"pays",
    "Granularity":"granularite",
    "Year":"annee",
    "Population":"population"
},
inplace = True
)
```

- Passage de la valeur de la population en unité et au type entier (int) :

```
population["population"]=(population["population"]*1000).astype(int)
```

- Aperçu après modifications :

Indice	pays	granularite	annee	population
0	Afghanistan	Total	2000	20779953
1	Afghanistan	Male	2000	10689508
2	Afghanistan	Female	2000	10090449
3	Afghanistan	Rural	2000	15657474
4	Afghanistan	Urban	2000	4436282
5	Afghanistan	Total	2001	21606988
6	Afghanistan	Male	2001	11117754

### 3.5 - regions.

- Aperçu avant modifications :

Indice	REGION (DISPLAY)	COUNTRY (DISPLAY)
0	Europe	Albania
1	Europe	Andorra
2	Europe	Armenia
3	Western Pacific	Australia
4	Europe	Austria
5	Europe	Azerbaijan

#### 3.5.1 - Anomalies.

On peut noter que le dataframe "regions" comporte un grand nombre d'anomalies qu'il me faudra corriger, afin d'avoir la possibilité d'obtenir des analyses correctes, par la suite :

- 40 pays et territoires ne sont pas présents, ce qui constitue un problème, en terme de représentativité lors d'analyses faisant appel à des valeurs régionales (palestine, porto rico, îles féroé, gibraltar, sahara occidental, polynésie française, guadeloupe, etc.
- 11 pays ne sont pas dans leurs régions respectives (Maroc et Soudan en Méditerranée orientale, Israël et Azerbaïdjan en Europe, etc.).

### 3.5.2 - Modifications liées aux anomalies.

- Ajout des provinces Chinoises par la création du dataframe "regions\_chine", puis sa concaténation au df "regions".

```
# Création d'un dictionnaire "regions_chines", où j'ajoute les régions composant la chine.
regions_chine = pd.DataFrame({
    "pays":["China, Hong Kong SAR","China, Macao SAR","China, mainland","China, Taiwan Province of"],
    "region":["Western Pacific","Western Pacific","Western Pacific","Western Pacific"]
})

# Utilisation de la méthode "concat()".
regions = pd.concat([regions,regions_chine], ignore_index = True)
```

- Ajout des 40 pays et territoires n'ayant pas de correspondance avec les régions.

```
territoires = pd.DataFrame({
    "pays" : ["Aruba","Channel Islands","Curaçao","French Guyana","French Polynesia","Guadeloupe","Guam","Martinique","Mayotte","Netherlands Antilles (former)",
    "New Caledonia","Palestine","Puerto Rico","Réunion","Serbia and Montenegro","United States Virgin Islands","Western Sahara","American Samoa",
    "Anguilla","Bermuda","Bonaire, Sint Eustatius and Saba","British Virgin Islands","Cayman Islands","Falkland Islands (Malvinas)","Faroe Islands","Gibraltar",
    "Isle of Man","Liechtenstein","Montserrat","Northern Mariana Islands","Saint Barthélemy","Saint Helena, Ascension and Tristan da Cunha","Saint Pierre and",
    "Wallis and Futuna Islands","Sint Maarten (Dutch part)","Montenegro"],
    "region" : ["Americas","Europe","Americas","Americas","Western Pacific","Americas","Western Pacific","Americas","Africa","Americas",
    "Western Pacific","Eastern Mediterranean","Americas","Africa","Europe","Americas","Africa","Western Pacific",
    "Americas","Americas","Americas","Americas","Americas","Americas","Europe","Europe","Europe","Europe",
    "Europe","Europe","Americas","Western Pacific","Americas","Europe","Americas","Americas","Western Pacific","Americas",
    "Western Pacific","Americas","Europe"]
})

regions = pd.concat([regions,territoires], ignore_index = True)
```

- Réaffectation des 11 pays dans leurs régions respectives.

```
regions.loc[regions["pays"] == "Sudan", "region"] = "Africa"
regions.loc[regions["pays"] == "Morocco", "region"] = "Africa"
regions.loc[regions["pays"] == "Libya", "region"] = "Africa"
regions.loc[regions["pays"] == "Egypt", "region"] = "Africa"
regions.loc[regions["pays"] == "Tunisia", "region"] = "Africa"
regions.loc[regions["pays"] == "Somalia", "region"] = "Africa "
regions.loc[regions["pays"] == "Djibouti", "region"] = "Africa "
regions.loc[regions["pays"] == "Turkey", "region"] = "Eastern Mediterranean"
regions.loc[regions["pays"] == "Israel", "region"] = "Eastern Mediterranean"
regions.loc[regions["pays"] == "Azerbaijan", "region"] = "Eastern Mediterranean"
regions.loc[regions["pays"] == "Armenia", "region"] = "Eastern Mediterranean"
```

### 3.5.3 - Modifications (hors anomalies).

- Renommage des colonnes.

```
regions.rename(columns=
{
    "REGION (DISPLAY)": "region",
    "COUNTRY (DISPLAY)": "pays"
},
inplace=True
)
```

## 4 - Création de dataframes par granularités.

Chaque dataframe est ensuite exporté au format .csv pour importation dans tableau prep en vue de leur fusion.

### 4.1 - population

#### 4.1.1 - population\_male.

```
population_male = population.loc[population["granularite"]=="Male",:].copy()
population_male.head()

population_male.rename(columns=
    {
        "population": "population hommes"
    },
    inplace=True
)

del population_male["granularite"]
population_male.to_csv("population_male.csv")
```

	Indice	pays	annee	population hommes
	1	Afghanistan	2000	10689508
	6	Afghanistan	2001	11117754
	11	Afghanistan	2002	11642106
	16	Afghanistan	2003	12214634
	21	Afghanistan	2004	12763726
	26	Afghanistan	2005	13239684

#### 4.1.2 - population\_female.

```
population_female = population.loc[population["granularite"]=="Female",:].copy()
population_female.head()

population_female.rename(columns=
    {
        "population": "population femmes"
    },
    inplace=True
)

del population_female["granularite"]
population_female.to_csv("population_female.csv")
```

Indice	pays	annee	population femmes
2	Afghanistan	2000	10090449
7	Afghanistan	2001	10489238
12	Afghanistan	2002	10958668
17	Afghanistan	2003	11466237
22	Afghanistan	2004	11962963
27	Afghanistan	2005	12414590
32	Afghanistan	2006	12809162

### 4.1.3 - population\_rurale.

```
population_rurale = population.loc[population["granularite"]=="Rural",:].copy()
population_rurale.head()

population_rurale.rename(columns=
    {
        "population": "population rurale"
    },
    inplace=True
)

del population_rurale["granularite"]
population_rurale.to_csv("population_rurale.csv")
```

Indice	pays	annee	population rurale
3	Afghanistan	2000	15657474
8	Afghanistan	2001	16318324
13	Afghanistan	2002	17086910
18	Afghanistan	2003	17909063
23	Afghanistan	2004	18692107
28	Afghanistan	2005	19378962
33	Afghanistan	2006	19961972

### 4.1.4 - population\_urbaine.

```
population_urbaine = population.loc[population["granularite"]=="Urban",:].copy()
population_urbaine.head()

population_urbaine.rename(columns=
    {
        "population": "population urbaine"
    },
    inplace=True
)

del population_urbaine["granularite"]
population_urbaine.to_csv("population_urbaine.csv")
```

Indice	pays	annee	population urbaine
4	Afghanistan	2000	4436282
9	Afghanistan	2001	4648139
14	Afghanistan	2002	4893013
19	Afghanistan	2003	5155788
24	Afghanistan	2004	5426872
29	Afghanistan	2005	5691836
34	Afghanistan	2006	5931478

#### 4.1.5 - population\_totale.

```
population_totale = population.loc[population["granularite"]=="Total",:].copy()
population_totale.head()

population_totale.rename(columns=
    {
        "population": "population totale"
    },
    inplace=True
)

del population_totale["granularite"]
population_totale.to_csv("population_totale.csv")
```

Indice	pays	annee	population totale
0	Afghanistan	2000	20779953
5	Afghanistan	2001	21606988
10	Afghanistan	2002	22600770
15	Afghanistan	2003	23680871
20	Afghanistan	2004	24726684
25	Afghanistan	2005	25654277
30	Afghanistan	2006	26433049

## 4.2 - mortality.

### 4.2.1 - mortality\_female.

```
mortality_female = mortality.loc[mortality["granularite"]=="Female",:].copy()
mortality_female.head()

mortality_female.rename(columns=
    {
        "décès pour 100000 habitants":"décès femmes/100.000 habitants",
        "nb morts eau insalubre":"nb morts femmes"
    },
    inplace=True
)

del mortality_female["granularite"]
mortality_female.to_csv("mortality_female.csv")
```

	Indice	annee	pays	écès femmes/100.000 habitar	nb morts femmes
0		2016	Afghanistan	15.31	0
3		2016	Albania	0.13	0
6		2016	Algeria	2.2	0
9		2016	Angola	45.15	0
12		2016	Antigua and Barbuda	0.1	0
15		2016	Argentina	0.4	0
18		2016	Armenia	0.16	0
21		2016	Australia	0.12	0

### 4.2.2 - mortality\_male.

```
mortality_male = mortality.loc[mortality["granularite"]=="Male",:].copy()
mortality_male.head()

mortality_male.rename(columns=
    {
        "décès pour 100000 habitants":"décès hommes/100.000 habitants",
        "nb morts eau insalubre":"nb morts hommes"
    },
    inplace=True
)

del mortality_male["granularite"]
mortality_male.to_csv("mortality_male.csv")
```

	Indice	annee	pays	décès hommes/100.000 habitants	nb morts hommes
	1	2016	Afghanistan	12.61	0
	4	2016	Albania	0.21	0
	7	2016	Algeria	1.73	0
	10	2016	Angola	52.63	0
	13	2016	Antigua and Barbuda	0.12	0
	16	2016	Argentina	0.32	0
	19	2016	Armenia	0.19	0
	22	2016	Australia	0.07	0

### 4.2.3 - mortality\_total.

```
mortality_total = mortality.loc[mortality["granularite"]=="Total",:].copy()
mortality_total.head()

mortality_total.rename(columns=
    {
        "décès pour 100000 habitants":"décès total/100.000 habitants",
        "nb morts eau insalubre":"nb morts total"
    },
    inplace=True
)

del mortality_total["granularite"]
mortality_total.to_csv("mortality_total.csv")
```

	Indice	annee	pays	décès total/100.000 habitants	nb morts total
	2	2016	Afghanistan	13.92	4824
	5	2016	Albania	0.17	5
	8	2016	Algeria	1.87	758
	11	2016	Angola	48.81	14065
	14	2016	Antigua and Barbuda	0.11	0
	17	2016	Argentina	0.36	159
	20	2016	Armenia	0.18	5
	23	2016	Australia	0.1	23



## 4.3 - management.

### 4.3.1 - management\_rural.

```
management_rural = management.loc[management["granularite"]=="Rural",:].copy()
management_rural.head()

management_rural.rename(columns=
    {
        "ratio acces eau de base":"ratio eau de base rural",
        "ratio acces eau qualite":"ratio eau qualite rural"
    },
    inplace=True
)

del management_rural["granularite"]
management_rural.to_csv("management_rural.csv")
```

	Indice	annee	pays	ratio eau de base rural	ratio eau qualite rur. ▼
	5055	2008	Paraguay	74.95	38.17
	7092	2012	Colombia	81.94	37.95
	5472	2009	India	82.76	37.85
	6510	2011	Colombia	81.04	37.53
	5481	2009	Iraq	73.05	37.35
	8493	2014	Morocco	64.71	37.31
	5928	2010	Colombia	80.13	37.11
	4473	2007	Paraguay	72.18	36.75
	5346	2009	Colombia	79.23	36.69

### 4.3.2 - management\_urban.

```
management_urban = management.loc[management["granularite"]=="Urban",:].copy()
management_urban.head()

management_urban.rename(columns=
    {
        "ratio acces eau de base":"ratio eau de base urbain",
        "ratio acces eau qualite":"ratio eau qualite urbain"
    },
    inplace=True
)

del management_urban["granularite"]
management_urban.to_csv("management_urban.csv")
```

Indice	annee	pays	ratio eau de base urbain	ratio eau qualite urbain
10241	2017	Morocco	96.65	88.93
7478	2012	Sri Lanka	96.46	88.92
5141	2008	South Africa	98.62	88.78
9488	2016	Eswatini	96.78	88.75
10070	2017	Eswatini	96.78	88.75
9077	2015	Morocco	96.35	88.75
6680	2011	Kyrgyzstan	96.84	88.63
8495	2014	Morocco	96.05	88.58
6896	2011	Sri Lanka	96.36	88.53

### 4.3.3 - management total.

```
management_total = management.loc[management["granularite"]=="Total",:].copy()
management_total.head()

management_total.rename(columns=
    {
        "ratio acces eau de base":"ratio eau de base total",
        "ratio acces eau qualite":"ratio eau qualite total"
    },
    inplace=True
)

del management_total["granularite"]
management_total.to_csv("management_total.csv")
```

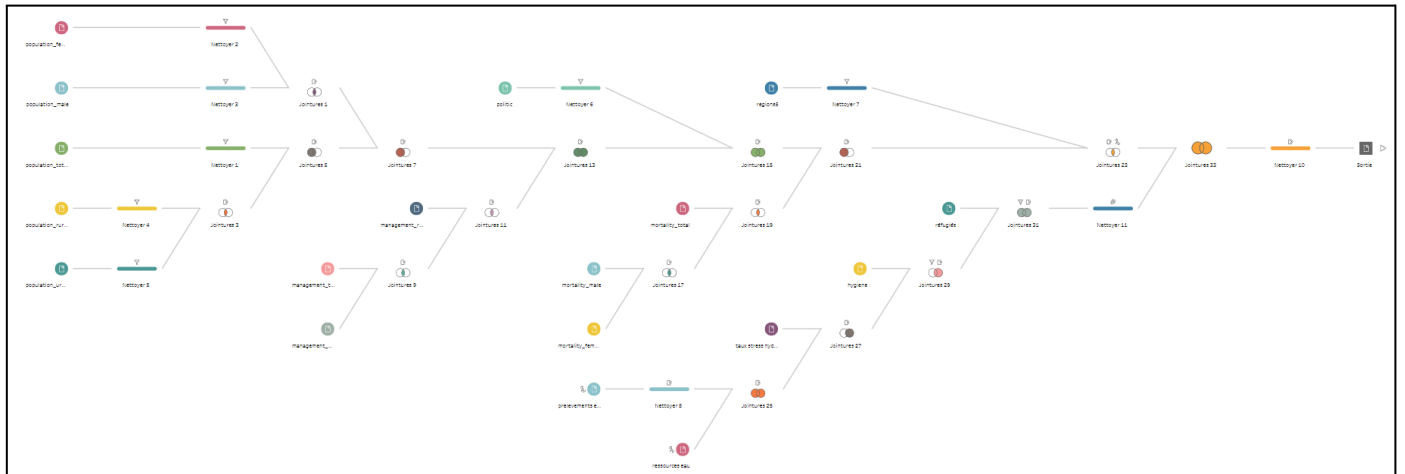
Indice	annee	pays	ratio eau de base total	ratio eau qualite total
6988	2012	Albania	87.86	65.9
6406	2011	Albania	87.76	63.93
5824	2010	Albania	87.81	61.98
5242	2009	Albania	87.85	60.05
4660	2008	Albania	87.89	58.14
4078	2007	Albania	87.91	56.25
3496	2006	Albania	87.93	54.39
2914	2005	Albania	87.94	52.55
2332	2004	Albania	87.94	50.73

## 5 - Ajout de données externes.

Dans un soucis d'une meilleure compréhension de l'environnement concernant l'impact de l'accès à l'eau, j'ai choisi d'ajouter des données que j'ajoute à celles, déjà disponibles :

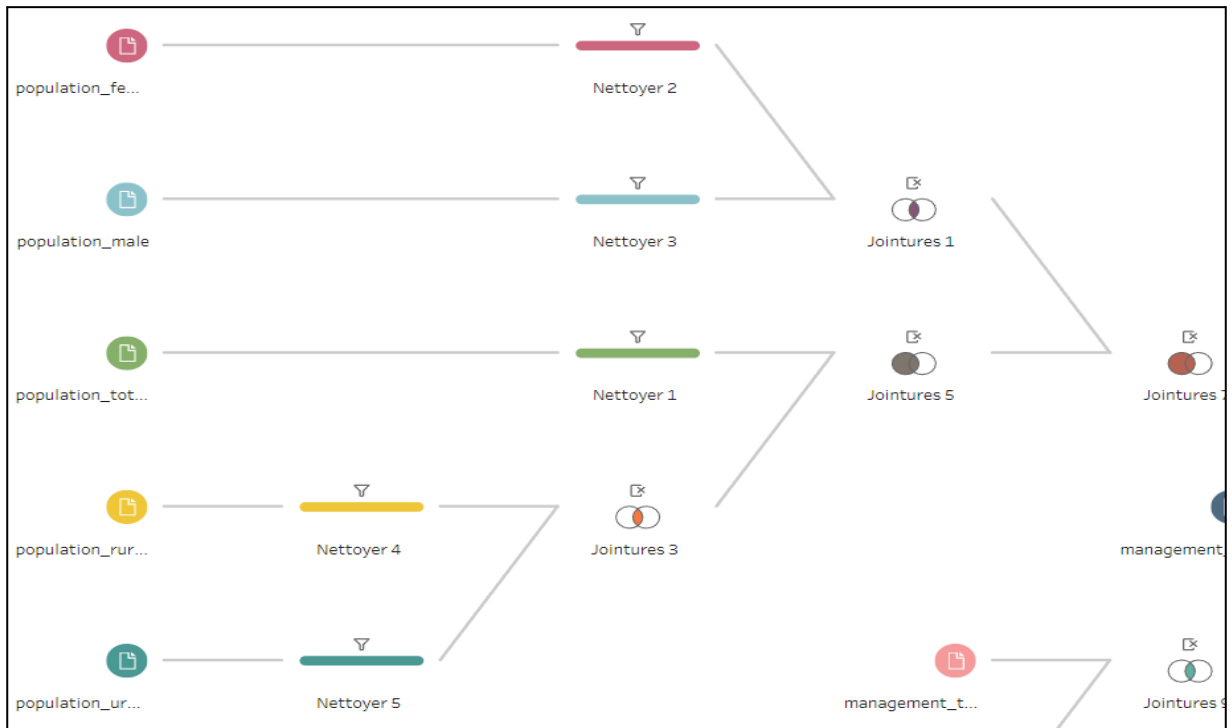
- Ressources en eau douce renouvelables
- Prélèvements dans les ressources d'eau douce
- Les populations contraintes de quitter leur pays d'origine
- Le taux d'accès à l'hygiène des mains
- Le taux de stress hydrique après prélèvement dans les ressources en eau douce.
- 

## 6 - Tableau Prep Builder.



## 6.1 - jointure des tables “population”

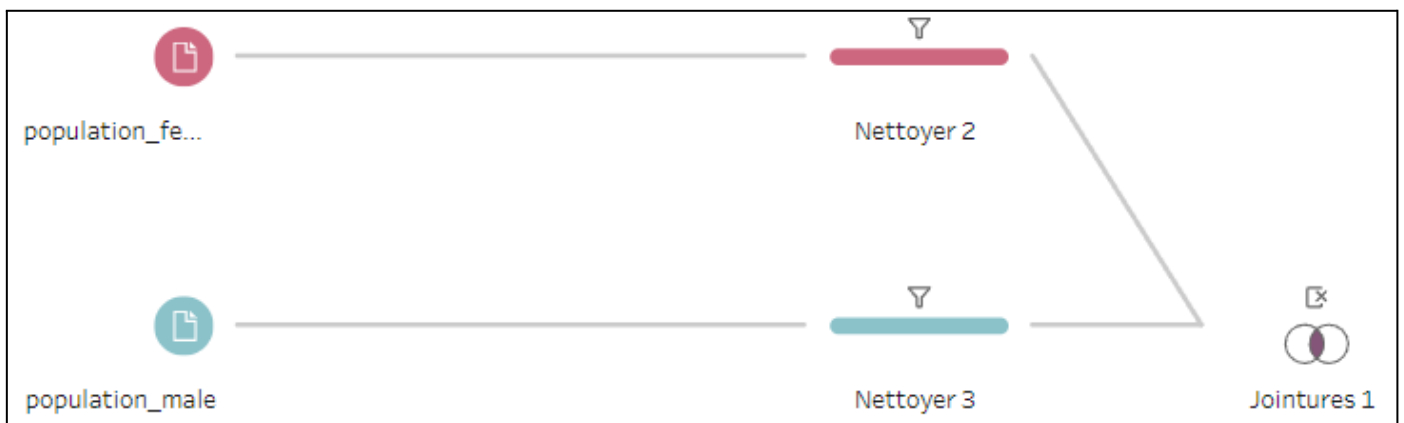
Utilisation du nettoyage pour supprimer les 4 provinces composant la Chine, afin d’éviter les doublons.



Calcul      Doit renvoyer une valeur booléenne

```
NOT ((([pays] == "China, Hong Kong SAR" OR [pays] ==  
"China, Macao SAR" OR [pays] == "China, mainland" OR  
[pays] == "China, Taiwan Province of") AND NOT  
(ISNULL([pays]))))
```

### 6.1.1 - jointures 1.



Jointures 15 champs4 k lignes

Filtrer les valeurs

ParamètresModifications (2)

Clauses de jointure appliquées

Nettoyer 2

pays

=

Nettoyer 3

pays

annee


=

annee

Type de jointure : Jointure interne

Cliquez sur le graphique pour modifier le type de jointure.

Nettoyer 2



Nettoyer 3

Résumé des résultats de la jointure

Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.

Valeurs discordantes

	Incluse(s)
Nettoyer 2	3752
Nettoyer 3	3752
Résultat ...	3752

Jointures 15 champs4 k lignes

Filtrer les valeurs

ParamètresModifications (2)

Jointures

[pays] == [pays],[annee] == [annee]

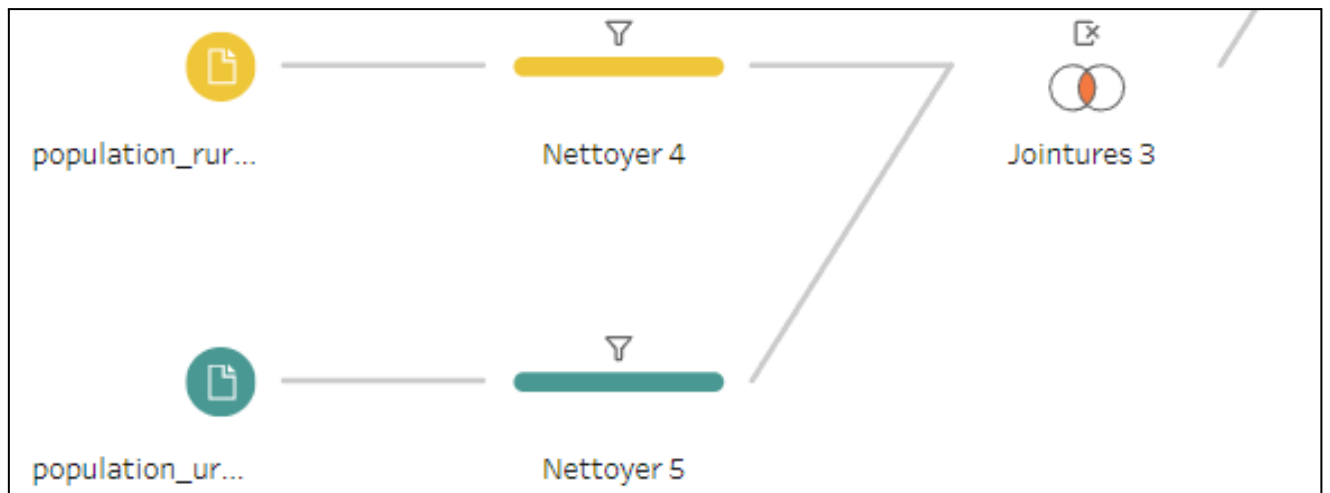
Supprimer le champ

[pays-1]

Supprimer le champ

[annee-1]

### 6.1.2 - jointures 3.



Jointures 3 4 champs 4 k lignes

Filtrer les valeurs

Paramètres

Modifications (2)

Clauses de jointure appliquées

Nettoyer 4

pays

=

Nettoyer 5

pays

annee

=

annee

Type de jointure : Jointure interne

Cliquez sur le graphique pour modifier le type de jointure.

Nettoyer 4

Nettoyer 5

Résumé des résultats de la jointure

Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.

Valeurs discordantes

	Incluse(s)
Nettoyer 4	4 338
Nettoyer 5	4 338
Résultat ...	4 338

Jointures 3 4 champs 4 k lignes

Filtrer les valeurs

Paramètres

Modifications (2)

Jointures

[pays] == [pays],[annee] == [annee]

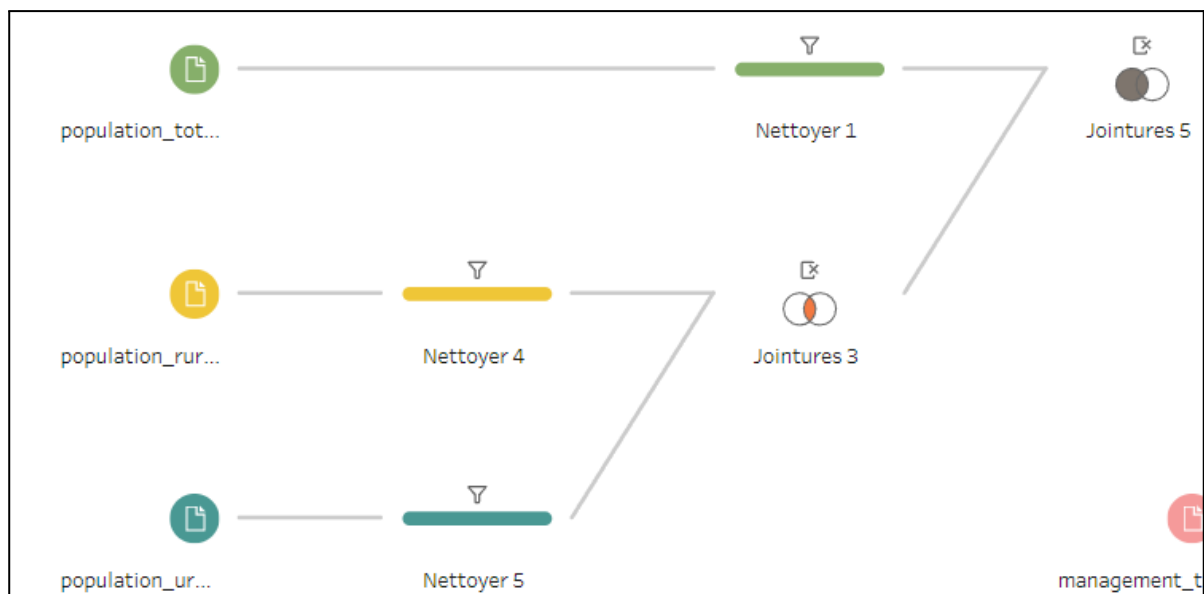
Supprimer le champ

[pays-1]

Supprimer le champ

[annee-1]

### 6.1.3 - jointures 5.



Paramètres
Modifications (2)

**Clauses de jointure appliquées**

Nettoyer 1

Jointures 3

pays = pays

annee = annee

**Type de jointure : Jointure gauche**

Cliquez sur le graphique pour modifier le type de jointure.

Nettoyer 1

Jointures 3

**Résumé des résultats de la jointure**

Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.

Valeurs discordantes

	Incluse(s)
Nettoyer 1	4 354
Jointures 3	4 338
Résultat ...	4 354

Paramètres
Modifications (2)

Jointures

[pays] == [pays],[annee] == [annee]

Supprimer le champ

[pays-1]

Supprimer le champ

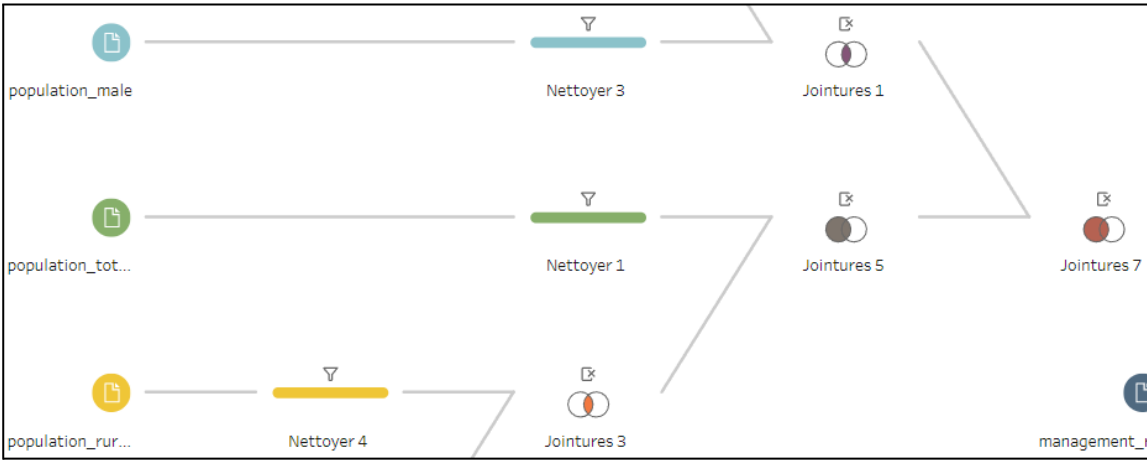
[annee-1]

Valeurs discordantes:

	Inclure(s)	Exclure(s)
Nettoyer 1	4 338	16
Jointures 3	4 338	0
Résultat...	4 338	

Abc	#
<b>pays</b> 2	<b>annee</b> 8
Saint Barthélemy	01/01/2011
Saint-Martin (French part)	01/01/2012
	01/01/2013
	01/01/2014
	01/01/2015
	01/01/2016
	01/01/2017
	01/01/2018
	<b>population totale</b> 16
	7 500
	17 500
	27 500
	37 500

### 6.1.4 - jointures 7.



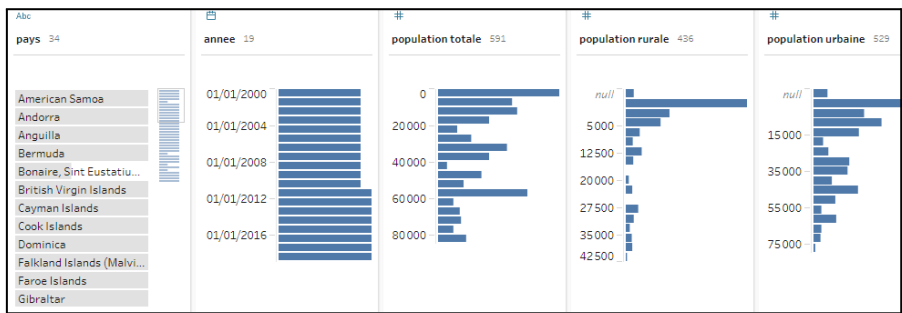
Paramètres	Modifications (2)
<b>Clauses de jointure appliquées</b>	
Jointures 5	Jointures 1
pays	= pays
annee	= annee
<b>Type de jointure : Jointure gauche</b>	
Cliquez sur le graphique pour modifier le type de jointure.	
<b>Résumé des résultats de la jointure</b>	
Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.	
<b>Valeurs discordantes</b>	
	Inclure(s)
Jointures 5	4 354
Jointures 1	3 752
Résultat...	4 354

Paramètres	Modifications (2)
<b>Jointures</b>	
[pays] == [pays],[annee] == [annee]	
<b>Supprimer le champ</b>	
[pays-1]	
<b>Supprimer le champ</b>	
[annee-1]	



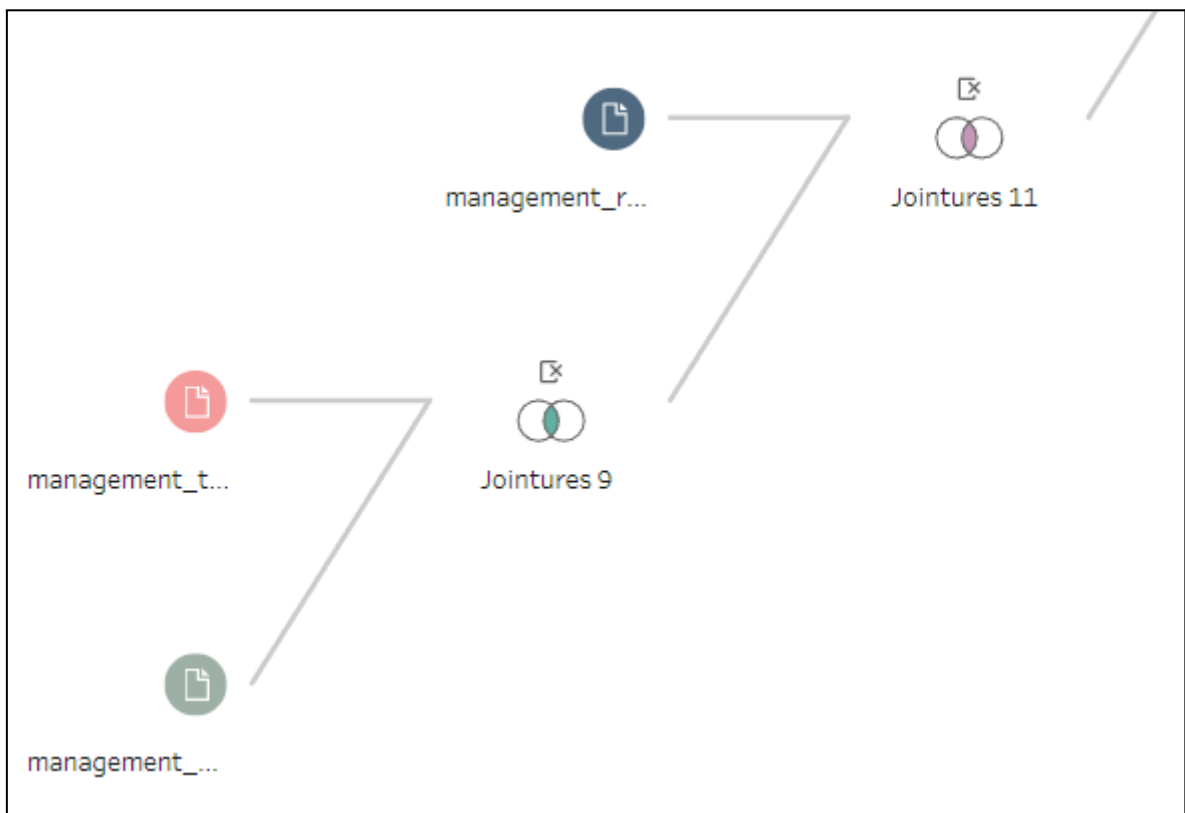
Valeurs discordantes :

	Include(s)	Exclude(s)
Jointures 5	3752	602
Jointures 1	3752	0
Résultat ...	3752	



## 6.2 - jointure des tables “management”.

Aucun nettoyage des tables n'est nécessaire.



6.2.1 - jointures 9.

Paramètres

Modifications (2)

Clauses de jointure appliquées

management\_total

pays

annee

=

management\_urban

pays

annee

Type de jointure : Jointure interne

Cliquez sur le graphique pour modifier le type de jointure.

management\_total

management\_urban

Résumé des résultats de la jointure

Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.

Valeurs discordantes

Incluse(s)

managem...

3492

managem...

3492

Résultat ...

3492

Paramètres

Modifications (2)

Jointures

[pays] == [pays],[annee] == [annee]



Supprimer le champ

[annee-1]

Supprimer le champ

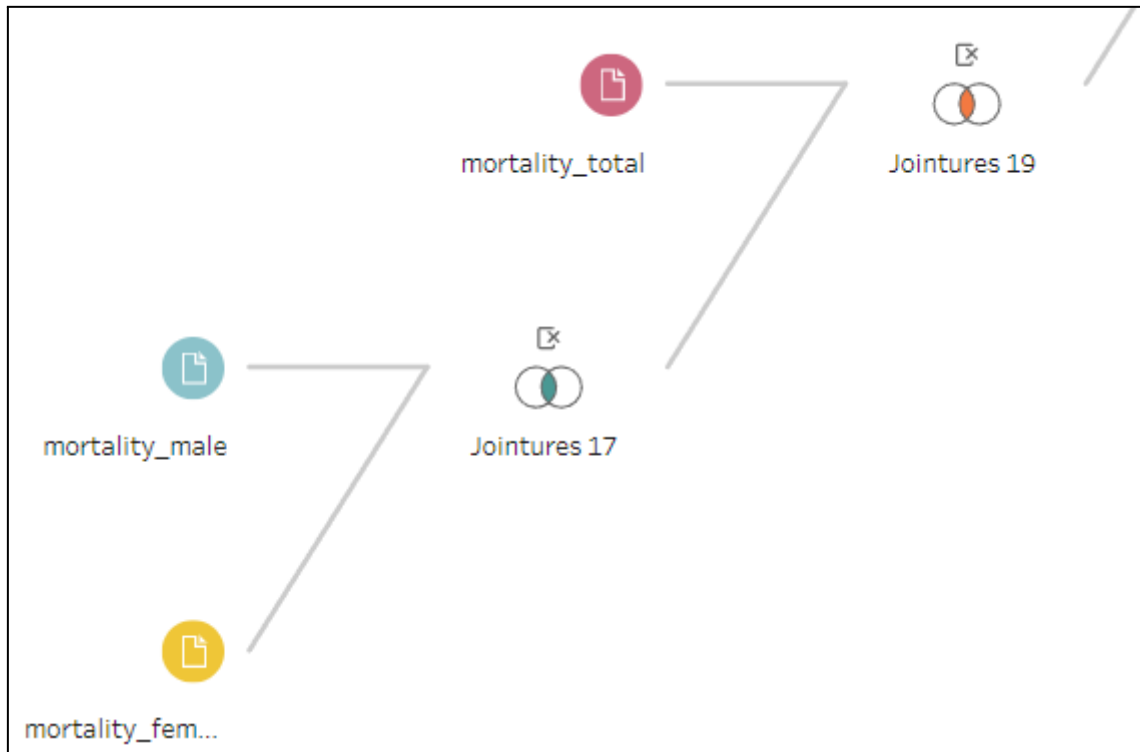
[pays-1]

6.2.3 - jointures 11.

Paramètres	Modifications (2)	Paramètres	Modifications (2)
<div>Clauses de jointure appliquées</div> <div><div>management_rural</div><div>Jointures 9</div></div> <div><div>pays</div><div>=</div><div>pays</div></div> <div><div>annee</div><div>=</div><div>annee</div></div>		<div>① Jointures</div> <div>  [pays] == [pays],[annee] == [annee]</div> <div><div>✕</div> Supprimer le champ</div> <div>  [annee-1]</div> <div><div>✕</div> Supprimer le champ</div> <div>  [pays-1]</div>	
<div>Type de jointure : Jointure interne</div> <div>Cliquez sur le graphique pour modifier le type de jointure.</div> <div><div>management_rural</div><div></div><div>Jointures 9</div></div>			
<div>Résumé des résultats de la jointure</div> <div>Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.</div> <div><div> Valeurs discordantes</div><div>Incluse(s)</div></div> <div><div>managem...</div><div>3 492</div></div> <div><div>Jointures 9</div><div>3 492</div></div> <div><div>Résultat ...</div><div>3 492</div></div>			

### **6.3 - jointure de la table “mortality”.**

Aucun nettoyage des tables n'est nécessaire.



6.3.1 - jointures 17.

Paramètres

Modifications (4)

Clauses de jointure appliquées

mortality\_male mortality\_female

pays = pays

Type de jointure : Jointure interne

Cliquez sur le graphique pour modifier le type de jointure.

mortality\_male mortality\_female

Résumé des résultats de la jointure

Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.

Valeurs discordantes

Incluse(s)

mortality... 183

mortality... 183

Résultat... 183

Paramètres

Modifications (4)

Supprimer le champ

[annee]

Supprimer le champ

[nb morts hommes]

Supprimer le champ

[nb morts femmes]

Jointures


[pays] == [pays]

Supprimer le champ

[pays-1]

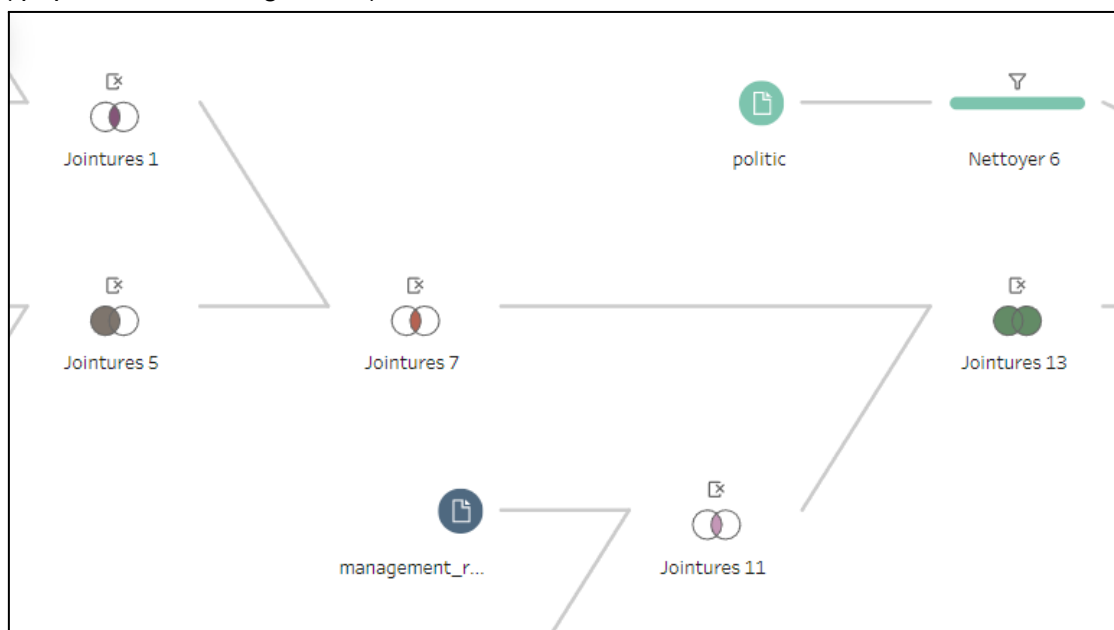
- Suppression des colonnes “nb morts hommes” et “nb morts femmes”, ces dernières n’ayant que des valeurs égales à 0. Elles sont donc inutiles aux analyses à venir.

### 6.3.2 - jointures 19.

Paramètres	Modifications (2)								
<b>Clauses de jointure appliquées</b> mortality_total      Jointures 17 <b>pays</b> = <b>pays</b>	<input checked="" type="checkbox"/> Supprimer le champ   [annee] ●								
<b>Type de jointure : Jointure interne</b> Cliquez sur le graphique pour modifier le type de jointure. mortality_total  Jointures 17	<input checked="" type="checkbox"/> Jointures   [pays] == [pays]								
<b>Résumé des résultats de la jointure</b> Cliquez sur les segments de barre pour afficher les valeurs incl et exclues. /// Valeurs discordantes	<input checked="" type="checkbox"/> Supprimer le champ   [pays-1]								
<table border="1"> <thead> <tr> <th></th> <th>Incluse(s)</th> </tr> </thead> <tbody> <tr> <td>mortality...</td> <td>183</td> </tr> <tr> <td>Jointures ...</td> <td>183</td> </tr> <tr> <td>Résultat ...</td> <td>183</td> </tr> </tbody> </table>		Incluse(s)	mortality...	183	Jointures ...	183	Résultat ...	183	
	Incluse(s)								
mortality...	183								
Jointures ...	183								
Résultat ...	183								

### 6.4 - jointure 13.

( population + management )



Paramètres	Modifications (2)	Paramètres	Modifications (2)
<b>Clauses de jointure appliquées</b> <div> <div>Jointures 7</div> <div>Jointures 11</div> <div>pays = pays</div> <div>annee = annee</div> </div>		<b>Jointures</b> <div>[pays] == [pays],[annee] == [annee]</div> <div>Supprimer le champ [pays-1]</div> <div>Supprimer le champ [annee-1]</div>	
<b>Type de jointure : Jointure externe entière</b> Cliquez sur le graphique pour modifier le type de jointure. <div> <div>Jointures 7</div> <div>Jointures 11</div> </div>			
<b>Résumé des résultats de la jointure</b> Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues. <div> <div>Valeurs discordantes</div> <div>Inclure(s)</div> <div> <div>Jointures 7</div> <div>3752</div> </div> <div> <div>Jointures ...</div> <div>3492</div> </div> <div> <div>Résultat ...</div> <div>3974</div> </div> </div>			

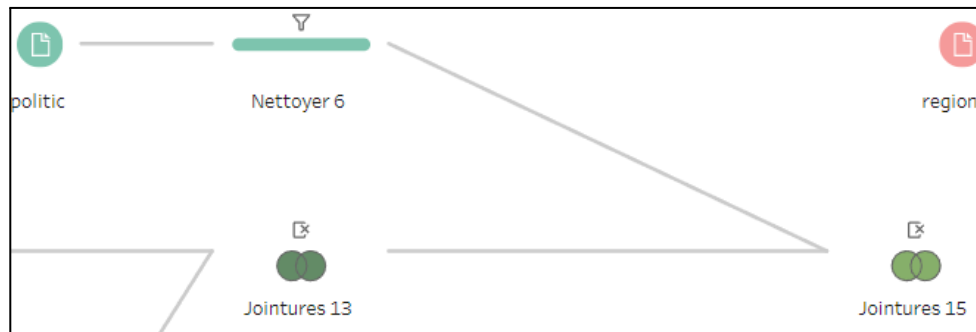
## 6.5 - jointure de la table "politic".

Utilisation du nettoyage pour supprimer les 4 provinces composant la Chine, afin d'éviter les doublons.

```
NOT ((([pays] == "China, Hong Kong SAR" OR [pays] ==
"China, Macao SAR" OR [pays] == "China, Taiwan Province
of" OR [pays] == "China, mainland") AND NOT
(ISNULL([pays]))))
```

### 6.5.1 - jointures 15.

( population + management + politic )

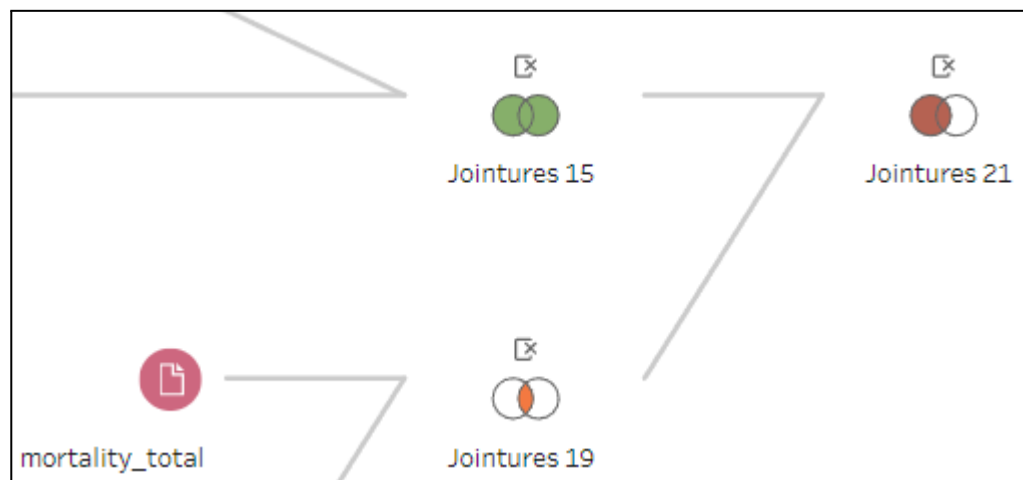


Paramètres	Modifications (2)
<b>Clauses de jointure appliquées</b> <div>Jointures 13      Nettoyer 6</div> <div>pays = pays</div> <div>annee = annee</div>	<div>Jointures</div> <div>[pays] == [pays],[annee] == [annee]</div> <div>Supprimer le champ</div> <div>[pays-1]</div> <div>Supprimer le champ</div> <div>[annee-1]</div>
<b>Type de jointure : Jointure externe entière</b> Cliquez sur le graphique pour modifier le type de jointure. <div>Jointures 13      Nettoyer 6</div>	
<b>Résumé des résultats de la jointure</b> Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues. <div>Valeurs discordantes</div> <div>Incluse(s)</div> <div>Jointures ... 3974</div> <div>Nettoyer 6 3472</div> <div>Résultat ... 4143</div>	



## 6.6 - jointure 21.

( population + management + politic + mortality )



Paramètres	Modifications (2)
<b>Clauses de jointure appliquées</b> <div> <div>Jointures 15</div> <div>pays = annee</div> </div> <div> <div>Jointures 19</div> <div>pays = annee</div> </div>	
<b>Type de jointure : Jointure gauche</b> Cliquez sur le graphique pour modifier le type de jointure. <div> <div>Jointures 15</div> <div>Jointures 19</div> </div>	
<b>Résumé des résultats de la jointure</b> Cliquez sur les segments de barre pour afficher les valeurs incl et exclues. <div> <div>Valeurs discordantes</div> </div>	
	Incluse(s)
Jointures ...	4143
Jointures ...	183
Résultat ...	4143

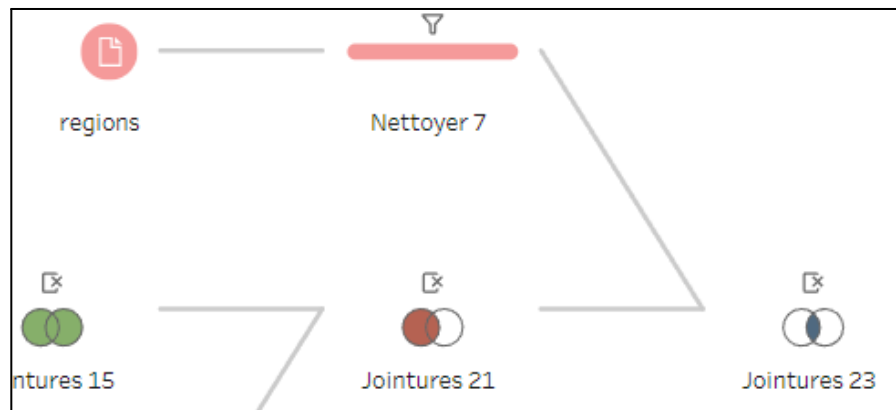
☒ Jointures  
 | [pays] == [pays],[annee] == [annee]  
☒ Supprimer le champ  
 | [pays-1]  
☒ Supprimer le champ  
 | [annee-1]



## 6.7 - jointure de la table "regions".

Utilisation du nettoyage pour supprimer les 4 provinces composant la Chine, ces dernières n'apparaissant plus dans les autres tables.

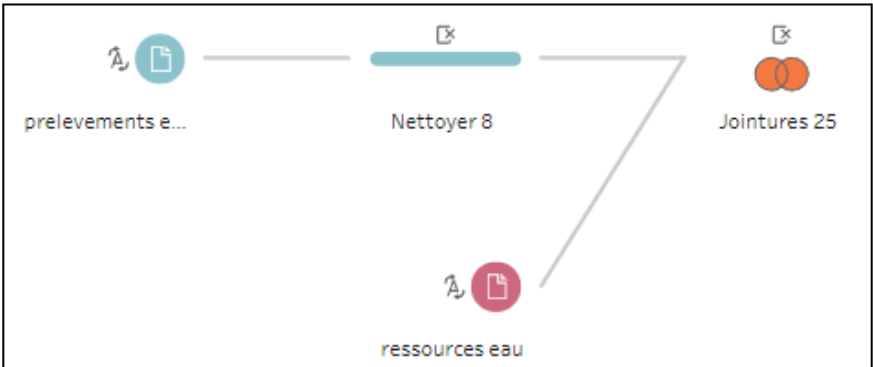
```
NOT ((([pays] == "China, Taiwan Province of" OR [pays] == "China, mainland" OR [pays] == "China, Macao SAR" OR [pays] == "China, Hong Kong SAR") AND NOT (ISNULL([pays]))))
```

## 6.8 - jointures 23.



Paramètres	Modifications (1)	Paramètres	Modifications (1)
<b>Clauses de jointure appliquées</b>		<b>Jointures</b>	
Jointures 21	Nettoyer 7	[pays] == [pays]	
pays	=	Supprimer le champ	
		[pays-1]	
<b>Type de jointure : Jointure interne</b>			
Cliquez sur le graphique pour modifier le type de jointure.			
Jointures 21  Nettoyer 7			
<b>Résumé des résultats de la jointure</b>			
Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.			
 Valeurs discordantes			
	Inclure(s)		Exclure(s)
Jointures ...	4 354		30
Nettoyer 7	235		0
Résultat ...	4 367		

6.9 - jointures 25.



Paramètres

Modifications (2)

Clauses de jointure appliquées

Nettoyer 8

ressources eau

pays = pays

annee = année

Type de jointure : Jointure externe entière

Cliquez sur le graphique pour modifier le type de jointure.

Nettoyer 8

ressources eau

Résumé des résultats de la jointure

Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.

Valeurs discordantes

	Incluse(s)
Nettoyer 8	3439
ressource...	3477
Résultat ...	3534

Paramètres

Modifications (2)

Jointures

[pays] == [pays],[annee] == [année]

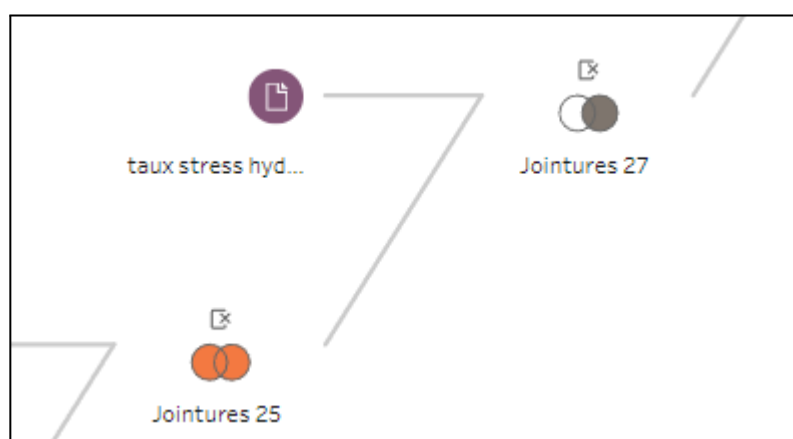
Supprimer le champ

[annee]

Supprimer le champ

[pays]

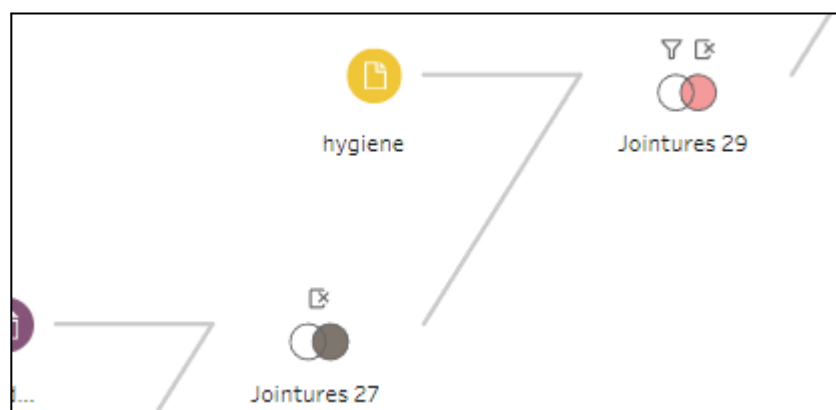
## 6.10 - jointures 27.



Paramètres	Modifications (2)
<b>Clauses de jointure appliquées</b>	
taux stress hydrique	Jointures 25
annee	= année
pays	= pays-1
<b>Type de jointure : Jointure droite</b>	
Cliquez sur le graphique pour modifier le type de jointure.	
taux stress hydrique	Jointures 25
<b>Résumé des résultats de la jointure</b>	
Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues.	
Valeurs discordantes	
	Incluse(s)
taux stre...	3 382
Jointures ...	3 534
Résultat ...	3 534

Paramètres	Modifications (2)
<b>Jointures</b>	
	[annee] == [année],[pays] == [pays-1]
	Supprimer le champ
	[pays]
	Supprimer le champ
	[annee]

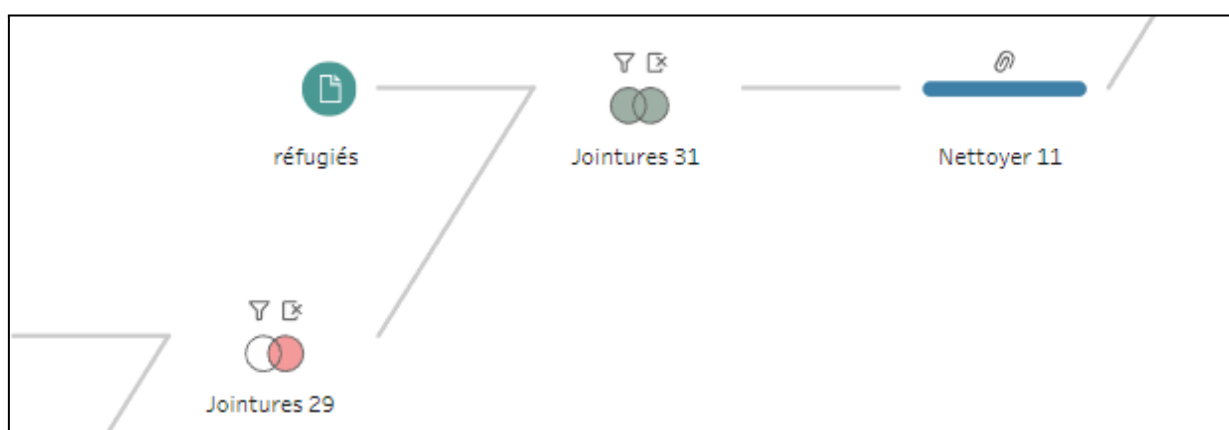
## 6.11 - jointures 29.



Paramètres	Modifications (3)
<b>Clauses de jointure appliquées</b>	
hygiene	Jointures 27
<u>année</u>	= <u>année</u>
<u>pays</u>	= <u>pays-1</u>
<b>Type de jointure : Jointure droite</b>	
Cliquez sur le graphique pour modifier le type de jointure.	
hygiene	Jointures 27
<b>Résumé des résultats de la jointure</b>	
Cliquez sur les segments de barre pour afficher les valeurs inclues et exclues.	
<div> <div></div> Valeurs discordantes         </div>	
	Incluse(s)
hygiene	1900
Jointures ...	3477
Résultat ...	3477

Paramètres	Modifications (3)
<b>Jointures</b>	
	[année] == [année],[pays] == [pays-1]
<b>Filtrer</b>	
	[pays-1]
	Exclure : null
<b>Supprimer le champ</b>	
	[pays]
<b>Supprimer le champ</b>	
	[année]

## 6.12 - jointures 31.

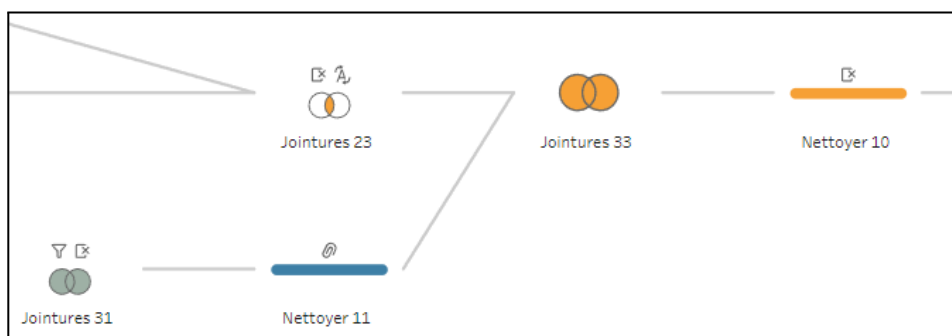


Paramètres	Modifications (3)
<b>Clauses de jointure appliquées</b>	
réfugiés	Jointures 29
pays	= pays-1
année	= année-1
<b>Type de jointure : Jointure externe entière</b>	
Cliquez sur le graphique pour modifier le type de jointure.	
réfugiés	Jointures 29
<b>Résumé des résultats de la jointure</b>	
Cliquez sur les segments de barre pour afficher les valeurs inclues et exclues.	
Valeurs discordantes	
	Incluse(s)
réfugiés	3439
Jointures ...	3477
Résultat ...	3477

Paramètres	Modifications (3)
<b>Jointures</b>	
	[pays] == [pays-1],[année] == [année-1]
<b>Supprimer le champ</b>	
	[année]
<b>Supprimer le champ</b>	
	[pays]
<b>Filtrer</b>	
	[pays-1]
	Exclure : null

Modifications (1)
Regrouper les valeurs
[pays-1]
Remplacement de "Turkiye" par "Turkey"

## 6.13 - jointures 33.



Paramètres	Modifications (0)
<b>Clauses de jointure appliquées</b>	
Jointures 23 pays = annee =	Nettoyer 11 pays-1 année-1
<b>Type de jointure : Jointure externe entière</b>	
Cliquez sur le graphique pour modifier le type de jointure. 	
<b>Résumé des résultats de la jointure</b>	
Cliquez sur les segments de barre pour afficher les valeurs incluses et exclues. 	
Jointures ...	4 354
Nettoyer ...	3 477
Résultat ...	4 391

**Modifications (2)**

- Supprimer le champ [pays-1]
- Supprimer le champ [année-1]

## **6.14 - Sortie.**

sortie effectuée sous 3 formats : .xlsx, .csv et .hyper

**Sortie** 29 champs

**Enregistrer la sortie sur**  

Fichier ▼

Parcourir

Nom  
P8\_5.3

Localisation  
C:\Users\motar\Desktop\1-  
openclassrooms\1-projets\projet8\4-  
python\feuilles tableau

Type de résultat  
Extrait de données Tableau (.hyper) ▼

**Options d'écriture**  
Sélectionnez une option pour créer ou mettre  
à jour votre table de sortie.

Actualisation complète  
Créer une table ▼