

# seance5\_cube\_multidimensionnel\_correction

February 19, 2021

## 1 Cube multidimensionnel - correction

Manipulation de tables de mortalités façon OLAP, correction des exercices.

```
[1]: %matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')
import pyensae
from pyquickhelper.helpgen import NbImage
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

Populating the interactive namespace from numpy and matplotlib

```
[1]: <IPython.core.display.HTML object>
```

On lit les données puis on recrée un `DataSet` :

```
[2]: from actuariat_python.data import table_mortalite_euro_stat
table_mortalite_euro_stat()
import pandas
df = pandas.read_csv("mortalite.txt", sep="\t", encoding="utf8", low_memory=False)
df2 = df[["annee", "age_num", "indicateur", "pays", "genre", "valeur"]].dropna().
    ↪reset_index(drop=True)
piv = df2.pivot_table(index=["annee", "age_num", "pays", "genre"],
                      columns=["indicateur"],
                      values="valeur")
import xarray
ds = xarray.Dataset.from_dataframe(piv)
ds
```

```
[2]: <xarray.Dataset>
Dimensions:      (age_num: 84, annee: 54, genre: 3, pays: 54)
Coordinates:
  * annee        (annee) int64 1960 1961 1962 1963 1964 1965 1966 1967 1968 ...
  * age_num      (age_num) float64 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 ...
  * pays         (pays) object 'AM' 'AT' 'AZ' 'BE' 'BG' 'BY' 'CH' 'CY' 'CZ' ...
  * genre        (genre) object 'F' 'M' 'T'
Data variables:
  DEATHRATE     (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
  LIFEXP        (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
  PROBDEATH     (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
```

```

PROBSURV      (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
PYLIVED       (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
SURVIVORS     (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
TOTPYLIVED    (annee, age_num, pays, genre) float64 nan nan nan nan nan ...

```

### 1.0.1 Exercice 1 : que font les lignes suivantes ?

Le programme suivant utilise les fonctions `align` `nad` `reindex` pour faire une moyenne sur une des dimensions du DataSet (le pays) puis à ajouter une variable `meanp` contenant le résultat.

```
[3]: ds.assign(LIFEEXP_add = ds.LIFEEXP-1)
```

```
[3]: <xarray.Dataset>
Dimensions:      (age_num: 84, annee: 54, genre: 3, pays: 54)
Coordinates:
  * annee        (annee) int64 1960 1961 1962 1963 1964 1965 1966 1967 1968 ...
  * age_num      (age_num) float64 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 ...
  * pays        (pays) object 'AM' 'AT' 'AZ' 'BE' 'BG' 'BY' 'CH' 'CY' 'CZ' ...
  * genre       (genre) object 'F' 'M' 'T'
Data variables:
  DEATHRATE     (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
  LIFEEXP       (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
  PROBDEATH     (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
  PROBSURV     (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
  PYLIVED       (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
  SURVIVORS     (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
  TOTPYLIVED    (annee, age_num, pays, genre) float64 nan nan nan nan nan ...
  LIFEEXP_add   (annee, age_num, pays, genre) float64 nan nan nan nan nan ...

```

```
[4]: meanp = ds.mean(dim="pays")
ds1, ds2 = xarray.align(ds, meanp, join='outer')
```

```
[5]: joined = ds1.assign(meanp = ds2["LIFEEXP"])
```

```
[6]: joined.to_dataframe().head()
```

```
[6]:
```

				DEATHRATE	LIFEEXP	PROBDEATH	PROBSURV	PYLIVED	\
age_num	annee	genre	pays						
1	1960	F	AM	NaN	NaN	NaN	NaN	NaN	
			AT	NaN	NaN	NaN	NaN	NaN	
			AZ	NaN	NaN	NaN	NaN	NaN	
			BE	0.00159	73.7	0.00159	0.99841	97316	
			BG	0.00652	73.2	0.00650	0.99350	95502	
				SURVIVORS	TOTPYLIVED	meanp			
age_num	annee	genre	pays						
1	1960	F	AM	NaN	NaN	73.52			
			AT	NaN	NaN	73.52			
			AZ	NaN	NaN	73.52			
			BE	97393	7179465	73.52			
			BG	95813	7017023	73.52			

Les valeurs `meanp` sont constantes quelque soient le pays à `annee`, `age_num`, `genre` fixés.

```
[7]: joined.sel(annee=2000, age_num=59, genre='F')['meanp']
```

```
[7]: <xarray.DataArray 'meanp' ()>  
array(23.83243243243243)  
Coordinates:  
  annee    int64 2000  
  genre    object 'F'  
  age_num  float64 59.0
```

```
[8]:
```