SAÉ : Data analysis, reporting and data visualization

Mme Canonne, M. Bureau, M. Garnier

XXX XXX Lola Dixneuf XXX XXX XXX XXX XXX XXX

Table of contents

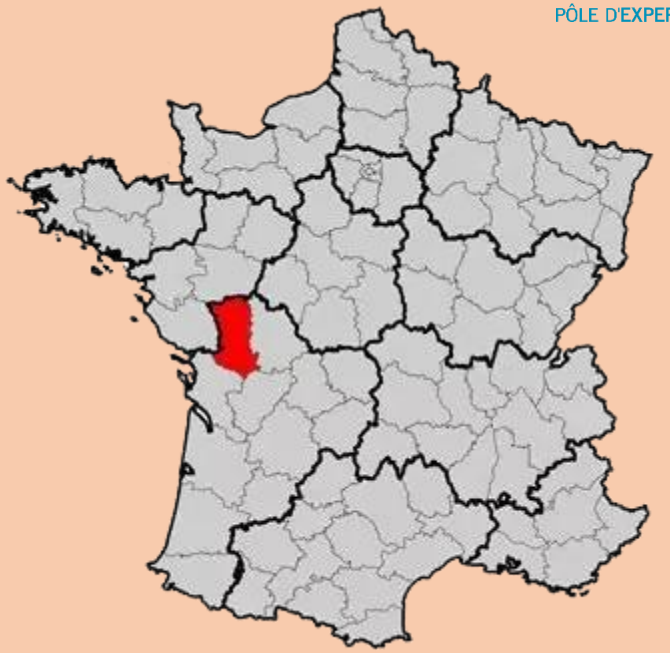
Presentation :

I. **Presentation of Calyxis and Mavie**
II. Our Goal
III. A closer look at the Dataset

Our Work:

I. Filtering and analyses of the Data
II. Demonstrations

Calyxis
PÔLE D'EXPERTISE DU RISQUE



Deux Sèvres, 1997

Goal :

Reduce the vulnerability of people

- Everyday life accidents
- Health and nutrition
- Risks of flooding

MAVIE observatory

Everyday life accidents

Goal of the observatory

- ✓ Collect
- ✓ Identify
- ✓ Suggest

+ BECOME A VOLUNTEER

Presentation :

I. Presentation of Calyxis and Ma**vie**
II. <u>Our Goal</u>
III. A closer look at the Dataset

Our Work:

I. Filtering and analyses of the Data
II. Demonstrations

Calyxis' needs

<u>A better understanding of :</u>

- The volunteers

- The victims of accidents

Global guideline

Filter and analyse the Data bases

Develop a dynamic and computing tool

Compare our data with external data

A closer look at the Dataset

707 volunteers

- Personal information
- Household Information

Id_volo	Annee_na	Sexe	Age_actuel	Date_remp
1	1958	Homme	66	11/04/2023 17:24:20
2	1963	Femme	61	15/03/2023 10:30:18
3	1948	Homme	76	14/03/2023 10:32:14
4	1950	Homme	74	09/04/2023 10:48:41
5	1969	Femme	55	21/03/2023 15:57:07
6	1997	Homme	27	21/05/2023 12:18:41
7	1960	Homme	64	24/05/2023 16:31:43
8	1958	Homme	66	25/05/2023 10:35:11
9	1949	Homme	75	
10	1954	Homme	70	
11	1937	Homme	87	
12	1973	Femme	51	
13	1981	Femme	43	14/03/2023 11:31:35
14	1943	Femme	81	14/03/2023 11:10:28
15	1949	Homme	75	14/03/2023 11:11:04
16	1955	Homme	69	14/03/2023 11:11:18
17	1945	Homme	79	14/03/2023 12:04:03
18	1962	Homme	62	14/03/2023 12:45:01
19	1947	Homme	77	25/05/2023 11:59:38
20		Homme	1	24/05/2023 11:06:01

Sample of the data set

A closer look at the Dataset

Every 2 months

Occurrence of accident

Cause of accident

Injuries or inconfort

Id_volontaire	Date	Acc	Date_acc	Heure	Tiers
282	01/06/	Oui	09/05/2023	13:45	Oui
472	05/10/	Oui	01/08/2023	16:00	Non
110	12/04/	Oui	08/03/2023	15:00	Non
87	12/04/	Oui	05/04/2023	14:00	Non
344	12/04/	Oui	10/04/2023	13:30	Non
344	12/04/	Oui	10/04/2023	16:00	Non
155	12/04/	Oui	27/03/2023	13h	Non
413	12/04/	Oui	05/04/2023	18:00	Non
192	12/04/	Oui	13/03/2023	19:10	Non
282	12/04/	Oui	06/04/2023	07:25	Non
249	13/04/	Oui	24/03/2023	23:00	Non
126	14/06/	Oui	24/05/2023	16:00	Non
461	16/05/	Oui	09/03/2023	18:50	Non
257	16/05/	Oui	15/04/2023	18:00	Non
300	16/05/	Oui	29/03/2023	13:50	Non
270	16/10/	Oui	03/09/2023	11:00	Non

Filtering the data

Delete useless data

Group & Organize

```python
def extraire_apres_deux_points(cellule):
    try:
        data = json.loads(cellule)
        valeur_apres_deux_points = cellule.split('"Type d\'accident":"')[1].strip('"')
        valeur_apres_deux_points = valeur_apres_deux_points[:-3]
        valeur_apres_deux_points = valeur_apres_deux_points.encode().decode('unicode_escape')
        return valeur_apres_deux_points
    except (json.JSONDecodeError, IndexError):
        return None

df_accident_oui['Type_acc'] = df_accident_oui['Type_acc'].apply(extraire_apres_deux_points)

Liste_typeacc = ["Chute", "Écrasement, coupure, perforation",
                 "Choc (coup, heurt par contact avec un objet, une personne ou un animal)",
                 "Corps étranger dans un orifice naturel",
                 "Noyade, suffocation, asphyxie",
                 "Intoxication ou autre effet chimique",
                 "Brûlure, refroidissement ou autre effet thermique",
                 "Électricité/rayonnement et effet d'autres ondes d'énergie",
                 "Surmenage physique (sur-sollicitation du corps, faux mouvement)"]
```

→ Read correctly the «Type d’accident» colomn having encoding issues

```python
def supprimer_apartir_deuxieme(caracteres):
    return caracteres[:2]

df_accident_oui['Heure_acc'] = df_accident_oui['Heure_acc'].apply(supprimer_apartir_deuxieme)
df_accident_oui = df_accident_oui.rename(columns={df_accident_oui.columns[44]: 'Type_acc'})
df_accident_non = df_accident_non.rename(columns={df_accident_non.columns[44]: 'Type_acc'})
```

→ Separate in two different spreadsheets the people who had an accident and those who didn’t

Presentation :

I. Presentation of Calyxis and Mavie
II. Our Goal
III. A closer look at the Dataset

Our Work:

I. **Filtering and analyses of the Data**
II. Demonstrations

Analyses of the data

```
df_accident_subset_2 = df_final[['Acc', 'Age_actuel']]
contingency_table_2 = pd.crosstab(df_accident_subset_2['Acc'], df_accident_subset_2['Age_actuel'])
chi2, p, dof, expected = chi2_contingency(contingency_table_2)
if p <= 0.05:
    p8 = 1
elif p <= 0.1:
    p8 = 2
else:
    p8 = 0
```

Example of one of the chi-square tests

People who had an accident depending on their age

Chi-square tests : Corelation between the variables

```
Statistique Chi-2: 89.67248961648401
P-value: 0.0012292695186889172
Degrés de liberté: 53
Fréquences attendues :
[[ 0.94490818  0.94490818  0.94490818  1.88981636  0.94490818  2.83472454
   4.7245409   0.94490818  0.94490818  1.88981636  4.7245409   4.7245409
   4.7245409   7.55926544  4.7245409   4.7245409   4.7245409  15.11853088
   6.61435726 10.39398998  3.77963272  3.77963272 16.06343907  4.7245409
  13.22871452  6.61435726  8.50417362 11.33889816 12.28380634 10.39398998
  17.95325543  5.66944908 16.06343907 14.1736227  15.11853088  9.4490818
  17.00834725 12.28380634 23.62270451 27.40233723 23.62270451 34.96160267
  30.23706177 27.40233723 21.73288815  9.4490818  18.89816361  8.50417362
   9.4490818   3.77963272  0.94490818  2.83472454  1.88981636 37.79632721]
```

List_descision	list	9	[0, 1, 0, 1, 1, 1, 2, 1, 0]

Presentation :

I. Presentation of Calyxis and Mavie
II. Our Goal
III. A closer look at the Dataset

Our Work:

I. Filtering and analyses of the Data
II. Demonstrations

Time for the demonstrations!