

td2a_cenonce_session_5

April 14, 2021

1 2A.i - Modèle relationnel, analyse d'incidents dans le transport aérien

Base de données relationnelles, logique SQL.

```
[1]: from jupyterhelper import add_notebook_menu
      add_notebook_menu()
```

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

1.0.1 Données

Le code suivant télécharge les données nécessaires [tp_2a_5_compagnies.zip](#).

```
[2]: import pyensae.datasource
      pyensae.datasource.download_data("tp_2a_5_compagnies.zip")
```

```
      downloading of
      http://www.xavierdupre.fr/enseignement/complements/tp_2a_5_compagnies.zip to
      tp_2a_5_compagnies.zip
      unzipped Cities.csv to .\Cities.csv
      unzipped Compagnies.csv to .\Compagnies.csv
      unzipped Crews.csv to .\Crews.csv
      unzipped Crews_planes_habilitation.csv to .\Crews_planes_habilitation.csv
      unzipped Data_exemple.csv to .\Data_exemple.csv
      unzipped Data_exemple_2.csv to .\Data_exemple_2.csv
      unzipped Data_exemple_3.csv to .\Data_exemple_3.csv
      unzipped Data_exemple_4.csv to .\Data_exemple_4.csv
      unzipped data_model.txt to .\data_model.txt
      unzipped data_model_to_csv.py to .\data_model_to_csv.py
      unzipped enonce.txt to .\enonce.txt
      unzipped Exemple_arborescent.xml to .\Exemple_arborescent.xml
      unzipped Flights.csv to .\Flights.csv
      unzipped Incident.csv to .\Incident.csv
      unzipped Motors.csv to .\Motors.csv
      unzipped Motor_models.csv to .\Motor_models.csv
      unzipped Planes.csv to .\Planes.csv
      unzipped Plane_models.csv to .\Plane_models.csv
```

```
[2]: ['.\\Cities.csv',
      '.\\Compagnies.csv',
```

```
'..\Crews.csv',
'..\Crews_planes_habilitation.csv',
'..\Data_exemple.csv',
'..\Data_exemple_2.csv',
'..\Data_exemple_3.csv',
'..\Data_exemple_4.csv',
'..\data_model.txt',
'..\data_model_to_csv.py',
'..\enonce.txt',
'..\Exemple_arborescent.xml',
'..\Flights.csv',
'..\Incident.csv',
'..\Motors.csv',
'..\Motor_models.csv',
'..\Planes.csv',
'..\Plane_models.csv']
```

```
[3]: import os
import pandas
df_Incident = pandas.read_csv('Incident.csv', sep=';')
df_Flights = pandas.read_csv('Flights.csv', sep=';')
df_Crews = pandas.read_csv('Crews.csv', sep=';')
df_Crews_planes_habilitation = pandas.read_csv('Crews_planes_habilitation.csv', sep=';
↵')
df_Planes = pandas.read_csv('Planes.csv', sep=';')
df_Plane_models = pandas.read_csv('Plane_models.csv', sep=';')
df_Motors = pandas.read_csv('Motors.csv', sep=';')
df_Motor_models = pandas.read_csv('Motor_models.csv', sep=';')
df_Compagnies = pandas.read_csv('Compagnies.csv', sep=';')
df_Cities = pandas.read_csv('Cities.csv', sep=';')
```

```
[4]: df_Incident.head(5)
```

```
[4]: Empty DataFrame
Columns: []
Index: [2, 11, 21, 30, 33]
```

```
[5]: df_Flights.head(5)
```

```
[5]:
```

	Plane_id	Crew_id	Departure_id	Arrival_id	Date
Id					
0	40	85	7	0	datetime.date(2013, 2, 20)
1	67	6	16	4	datetime.date(2013, 1, 9)
2	57	67	1	18	datetime.date(2013, 8, 7)
3	41	69	16	14	datetime.date(2013, 1, 10)
4	60	24	4	7	datetime.date(2013, 9, 26)

```
[6]: df_Crews.head(5)
```

```
[6]:
```

	Captain_age
Id	
0	54
1	53

```
2          56
3          43
4          45
```

```
[7]: df_Crews_planes_habilitation.head(5)
```

```
[7]:      Plane_model_id
Crew_id
85          40
6          67
67         57
69         41
24         60
```

```
[8]: df_Planes.head(5)
```

```
[8]:  Immatriculation  Plane_model_id  Motor_id  Compagny_id  Nb_hours
Id
0      'Q-KGNK'          3          0          0          1213
1      'Q-HTQG'          3          1          0          7932
2      'Q-XRMC'          1          2          0          2444
3      'Q-GUOA'          1          3          0          1595
4      'Q-WGIW'          0          4          0          2270
```

```
[9]: df_Plane_models.head(5)
```

```
[9]:  Constructor  Model_name
Id
0      AIRBUS      A380
1      AIRBUS      A350
2      BOEING      787
3      BOEING      747
```

```
[10]: df_Motors.head(5)
```

```
[10]:  Motor_model_id  Nb_hours
Id
0          4          2334
1          2          3609
2          2          7867
3          3          2673
4          2          3871
```

```
[11]: df_Motor_models.head(5)
```

```
[11]:      Name  Manufacturer
Id
0      SNECMA-123      SNECMA
1      SNECMA-246      SNECMA
2          GE-247          GE
3          GE-656          GE
4  ROLLS ROYCE-345  ROLLS ROYCE
```

```
[12]: df_Compagnies.head(5)
```

```
[12]:      Name      Country
      Id
0  Qatar Airways      Qatar
1    Air France      France
2      Iberia      Espagne
3    Lufthansa  Allemagne
4    Ryan Air      Irlande
```

```
[13]: df_Cities.head(5)
```

```
[13]:      Name      Country
      Id
0    Bombay      Inde
1  New York      USA
2   Tokyo      Japon
3   Paris      France
4 Francfort  Allemagne
```

Bien sûr, toutes les informations ne sont pas dans les tables telles quelles, il faudra faire principalement des jointures et des groupby pour obtenir les informations que l'on souhaite. Pour obtenir une table contenant les vols avec une colonne "est_incident", il faut faire :

```
[14]: import numpy as np
      try:
          df_Flights.reset_index( inplace = True )
          df_Incident.reset_index( inplace = True )
      except Exception:
          pass
      ## On suppose que cela vient du fait que les index ont déjà été remis à zéros
      df_Flight_Incident = pandas.merge( df_Flights, df_Incident, left_on = "Id", right_on = "Id",
      ↪ "Flight_id", how="outer" )
      df_Flight_Incident["Is_incident"] = np.isnan( df_Flight_Incident["Flight_id"] ) == False
      ↪ False
      df_Flight_Incident.head(5)
```

```
[14]:      Id  Plane_id  Crew_id  Departure_id  Arrival_id  \
0     0         40        85             7           0
1     1         67         6            16           4
2     2         57        67             1          18
3     3         41        69            16          14
4     4         60        24             4           7
```

```
      Date  Flight_id  Is_incident
0  datetime.date(2013, 2, 20)      NaN      False
1  datetime.date(2013, 1, 9)      NaN      False
2  datetime.date(2013, 8, 7)         2       True
3  datetime.date(2013, 1, 10)      NaN      False
4  datetime.date(2013, 9, 26)      NaN      False
```

On peut aussi faire des statistiques par ville de départ ...

```
[15]: df_Flight_Incident.groupby( "Departure_id" )["Is_incident"].mean()
```

```
[15]: Departure_id
0      0.146789
1      0.192308
2      0.220930
3      0.222222
4      0.206107
5      0.218391
6      0.180180
7      0.261682
8      0.120879
9      0.131868
10     0.141176
11     0.184466
12     0.097087
13     0.204082
14     0.174312
15     0.186916
16     0.206897
17     0.272727
18     0.189474
19     0.211538
Name: Is_incident, dtype: float64
```

1.0.2 Question 1 : Question préliminaires, taux d'incident

A vous de jouer. Commencer par calculer le taux d'incident global.

[16]:

1.0.3 Question 2 : Calcul de taux

Calculer le taux d'incidents par :

- => Compagnies
- => Destination
- => Equipage
- => Constructeur d'appareil (Boeing, Airbus ou Bombardier)
- => Modèle d'appareil (A320, A380, B737, etc ...)
- => Moteurs

[17]:

1.0.4 Question 3 : Régression linéaire

Pouvez-vous raisonnablement faire une régression linéaire sur le nombre d'heure de l'appareil ? Du moteur ? Pourquoi ?

[18]:

1.0.5 Question 4 : Iberia VS Lufthansa

Pouvez-vous dire si, toute chose égale par ailleurs, un avion d'Iberia a plus ou moins de chance d'avoir un incident qu'un avion de la Lufthansa ?

[19]:

1.0.6 Question 5 : Facteurs explicatifs

Essayer de déterminer quels sont les facteurs explicatifs. Lesquels vous paraissent statistiquement significatifs ? Y-a-t-il des corrélations indirectes ?

[20] :