Science des Données
Date : 12/02/2026
Prof : M TORRES

TP1 Prise en main – Utilisation d'IA interdite, IN à volonté
---

*« Le RMS **Titanic** est un paquebot transatlantique britannique qui a fait naufrage dans l'océan Atlantique Nord en 1912 à la suite d'une collision avec un iceberg, lors de son voyage inaugural de Southampton à New York. Entre 1 490 et 1 520 personnes trouvent la mort, ce qui fait de cet événement l'une des plus grandes catastrophes maritimes survenues en temps de paix et la plus grande pour l'époque. »* (Source : https://fr.wikipedia.org/wiki/Titanic)

Dans ce premier TP de Science des données nous allons utiliser le jeu de données du Titanic

Nous avons accès à des informations sur une partie des passagers (891 passagers) du Titanic.

- Une question intéressante qu'on pourrait tenter de répondre est : *pourquoi certains passagers ont survécu et d'autres sont morts?*
- Nous utiliserons **python** comme langage de travail pour ce cours.

Commençons donc l'exploration des données

Préalables : les indispensables...

```
# Packages données intéressants pour l'exploration des données
import pandas as pd
import numpy as np
# pip install --upgrade pandas==1.3.5 # Si jamais il le faut...
```

Indispensables : les données...

```
# Chargement des données
# https://www.kaggle.com/datasets/yasserh/titanic-dataset?select=Titanic-Dataset.csv
# Crééz un répertoire au dessous appelé DONNEES pour y deposer les données du cours

titanic = pd.read_csv("../DONNEES/titanic.csv")
```

Fondamental : l'observation des données… **Dans votre rapport le code commenté avec VOS réponses fera partie du rapport**

```
# On OBSERVE d'abord tout ce qu'on a...

print(titanic)   # Qué fait cette commande?

print (titanic.head(15)) # Qué fait cette commande? Qu'est qui se passe si on met pas 15?

print(titanic.tail()) # Qué fait cette commande? Peux je mettre un argument à tail()?

print(titanic.describe()) #  Qué fait cette commande?

print(titanic.info())   # Qué fait cette commande?

# Les colonnes de titanic sont:
    #  PassengerId : Identifiant du passager
    #  Survived : True (1) / False (0)
    #  Pclass : Classe du ticket : 1, 2 ou 3.
    #  Name : Nom du passager
    #  Sex : Genre du passager (male/female)
```

```
    #  Age : Age en années
    # Cabin : Numéro de cabine
    # Embarked : Port d'embarquement ( C : Cherbourg; Q Queenstown; S Southampton)

df = pd.DataFrame(titanic)    # A quoi ça sert ceci?
print(df)

print ("Dimensions n,p=",titanic.shape)   # Combien de dimensions possède votre table?

print (titanic.describe(include='all'))   # Que fait cette commande?

# Observez bien la commande suivante et commentez un peu sa fonctionnalité:
print (titanic.isnull().sum()) # This is called method chaining using two functions
sequentially
# isnull() - gives boolean values for whether a particular value is null or not
# if null the value is true
# sum - it sums up the values in a list/array/column
# now our column is full of booleans - its sums of the True values (True = 1)

print(titanic.columns) # Que fait cette instruction? (ne répondez pas uniquement: « elle
affiche quelque chose »)

# Que montre et à quoi sert cette boucle for?
for col in titanic.columns: # iterating through all the columns
    print(col, titanic[col].nunique()) # displaying the number of unique values
```

Pasons maintenant à visualiser un peu les choses

```
#------------ Visualisation: les packages
import matplotlib.pyplot as plt

# Que pensez vous que ceci montrera ? (réponse avant d'habiliter plt.show())
plt.hist(titanic['Age'], edgecolor = 'black', color=(0.2,0.7,1))
plt.title('Distribution d'ages')
plt.xlabel('Age')
plt.ylabel('Compte')
#plt.show()

# Que pensez vous que ceci montrera ? (réponse avant d'habiliter plt.show())
plt.hist(titanic['Survived'], edgecolor = 'black', color=(0.2,0.7,1))
plt.title('Distribution de survivants')
plt.xlabel('Age')
plt.ylabel('Compte')
#plt.show()
```

Peut être toute l'info de la table n'est pas pertinente pour notre question…

```
print( df.drop(columns=['Ticket', 'Cabin','Name','SibSp']) ) # On a fait quoi?

print(titanic.groupby(['Sex','Survived']).count()['PassengerId']) # On a fait quoi?

# On a fait quoi ici?
passengers = titanic.groupby('Sex')['PassengerId'].count()
print(passengers)

# Pourquoi on utilise sum() et pas count() ?
survivors = titanic.groupby('Sex')['Survived'].sum()
#survivors = titanic.groupby('Sex')['Survived'].count()
print(survivors)

# On a fait quoi ici?
summary = pd.DataFrame({"Survivants": survivors,
                        "Passagers": passengers,
                        "%": round(100*survivors / passengers,1)})
print(summary)

# On a fait quoi ici?
print( titanic['Survived'].sum() / titanic['PassengerId'].count() )

# On représente des barplots car on a des catégories
```

```
# Qu'est-ce que vous attendez comme plot? réponse avant d'habiliter plt.show
summary[["Survivants", "Passagers"]].plot(kind='bar');
plt.xlabel('Sexe')
plt.ylabel('Total')
plt.title('Comparaison de la survie selon le sexe');
#plt.show()
```

Et maintenant traiter un peu les données manquantes

```
#----------------- Données manquantes
print("Si j'enlève toutes les lignes contenant un 'NaN': ", titanic.dropna().shape)
print("\nSi je n'enlève que les 'NaN' de la colonne Age : ",
titanic.loc[titanic['Age'].notna(),:].shape)

# On a fait quoi ici?
titanic['Adult'] = titanic['Age'] >=18
print(titanic.head())

# On a fait quoi ici?
titanic_filt_age = titanic.loc[titanic['Age'].notna(),:]
passengers = titanic_filt_age.groupby(['Adult','Sex']).count()['PassengerId']
print(passengers)

# On a fait quoi ici?
survivors = titanic_filt_age.groupby(['Adult','Sex'])['Survived'].sum()
print(survivors)

# On a fait quoi ici?
passengers = titanic_filt_age.groupby(['Adult','Sex'])['PassengerId'].count()
summary = pd.DataFrame({"Survivants": survivors,
                        "Passagers": passengers,
                        "%": round(survivors/passengers*100, 1)})
summary.index=['Girl','Boy','Woman','Man']
print(summary)

# On représente ici un barplot car on a des catégories
summary.plot(kind='bar')
plt.xlabel("Personnes classées selon l'age et le sexe")
plt.ylabel('Total')
plt.title("Comparaison de la survie selon l'age et le sexe");
#plt.show()

# Conclusions: Complétez les valeurs:
#    Observation :
#        XXX% des passengers ont survécus et plus précisément XXX% des femmes contre
seulement XXX% des hommes
#        Il y a plus d'hommes que de femmes sur le paquebot
#    Interprétation : Les femmes ont eu plus de chances de survivre que les hommes
#    Etes vous d'accord avec cette interpretation?
# Pour aller plus loin, nous allons regarder à quel âge les hommes et femmes avaient la
plus grande chance de survie.

# ----- ETUDE SELON LE PORT
# Combien de ports y a t il ?
print(f'Nombre de ports:',titanic['Embarked'].nunique())
# Liste des noms de ports
print(f'Liste des ports:',titanic.loc[:,'Embarked'].unique())

# On fait quoi ici?
print (titanic.groupby(['Embarked']).count() )

#--- CHOIX
# La colonne du port d'embarquement à des valeurs manquantes (889 disponibles/891). Comme
la plupart des passagers et passagères sont montées à Southampton, on peut supposer que
les données manquantes viennent de là
# Attention c'est un choix... Toujours garder en tête qu'il modifie vos résultats et peut
donc modifier vos interpétations! Ici, c'est vraiment à la marge

# On fait quoi ici?
titanic["Embarked"] = titanic["Embarked"].fillna('S')
```

```
print(titanic["Embarked"])

# On fait quoi ici?
survivors_per_port = titanic.groupby('Embarked')['Survived'].sum()
passengers_per_port = titanic.groupby('Embarked')['PassengerId'].count()

# On fait quoi ici?
comparaison_port_survie = pd.DataFrame({"Survivants": survivors_per_port,
                                        "Passagers": passengers_per_port,
                                        "%":
round(survivors_per_port/passengers_per_port*100, 1)})
print(comparaison_port_survie)

# Quest-ce qu'on affiche ?
comparaison_port_survie.plot(kind='bar')
plt.xlabel("Port d'Embarquement")
plt.ylabel("Nombre d'individus")
plt.title('Comparaison de survie selon le port')
#plt.show()

# 2 HYPOTHESES
# Hypothèse 1: Il y a plus de femmes à Cherbourg (?)
# Hypothèse 2: On est plus riche à Cherbourg et plus on est riche plus on a survécu (?)
# Qu'estce que vous en pensez avant d'explorer plus loin?

# Tester l'Hypothèse 1: Il y a plus de femmes à Cherbourg
female_per_port = titanic[titanic['Sex']=='female'].groupby('Embarked')
['PassengerId'].count()
male_per_port = titanic[titanic['Sex']=='male'].groupby('Embarked')
['PassengerId'].count()
pd.DataFrame({"Female": female_per_port,
              "Male" : male_per_port,
              "Total": passengers_per_port,
              "% Female": female_per_port / passengers_per_port
              })

print(female_per_port)
print(male_per_port)
# Alors? Il y a ou il n'y a pas plus d'individus féminins à Cherbourg qu'à Queenstown?


# Tester l'Hypothese 2

survivors_per_class = titanic.groupby('Pclass')['Survived'].sum()
passengers_per_class = titanic['Pclass'].value_counts()
print(
pd.DataFrame({"Survivants": survivors_per_class,
              "Passagers": passengers_per_class,
              "%": round(survivors_per_class/passengers_per_class*100, 1)})
)
#print(survivors_per_class)      # Une autre sortie pour Hypothese 2
#print(passengers_per_class)

# Alors? Il y a ou il n'y a pas une corrélation entre classe et probabilité de survie?

# Explorons encore plus...

pclass1_per_port = titanic[titanic['Pclass']==1].groupby('Embarked')
['PassengerId'].count()
pclass2_per_port = titanic[titanic['Pclass']==2].groupby('Embarked')
['PassengerId'].count()
pclass3_per_port = titanic[titanic['Pclass']==3].groupby('Embarked')
['PassengerId'].count()

print(
pd.DataFrame({'Classe 1': pclass1_per_port,
              'Classe 2': pclass2_per_port,
              'Classe 3': pclass3_per_port,
              'Passengers': passengers_per_port,
              '% Classe 1': round(pclass1_per_port/passengers_per_port*100,1)})
)
```

```
#print(pclass1_per_port)
#print(pclass2_per_port)
#print(pclass3_per_port)
```

Résultats de nos explorations (ce type de observations et de conclusions sont celles qu'on attend d'un « Scientifique de données ») :

Observations

- *Les passagers ayant embarqué à Cherbourg regroupent principalement des individus de première classe*
- *Les 77 passagers qui embarquent à Queenstown en Irlande sont principalement de la 3ème classe, cad des migrants en route vers les États-Unis*

Conclusions

- *Les passagers ayant embarqué à Cherbourg arrivent de Paris (France) et sont plutôt riches*
- *Les 77 passagers qui embarquent à Queenstown sont principalement des migrants en route vers les États-Unis*
- *Il semble que la classe plus que le port d'embarquement a une relation de causalité avec la survie (à vérifier)*

---------- **AUTRES QUESTIONS A REPONDRE**

Question 1.1 : Convertisez le format CSV de titanic.csv au format JSON titanic.json.
Question 1.2 : Convertisez le format CSV de titanic.csv au format XML titanic.xml.

Vérifiez vos sorties

Question 2.1. Compressez le répertoire contenant les 3 fichiers titanic.XYZ XYZ= csv|json|xml ainsi que votre CODE TP1.py avec vos réponses en commentaires dans un seul fichier **tar zippé** en ligne de commande (donnez la commande shell qui permet de le faire)

Question 2.2. Compressez le répertoire contenant les 3 fichiers titanic.XYZ XYZ= csv|json|xml ainsi que votre CODE TP1.py avec vos réponses en commentaires dans un seul fichier **bzippé** en ligne de commande (donnez la commande shell qui permet de le faire)

Vérifiez la taille de vos archives et commentez laquelle vous préferez et pourquoi ?