CR TP n°1 POO

TP1 : INITIATION ET PRISE EN MAIN DU LANGAGE PYTHON (MODULES : NUMPY, MATPLOTLIB, PANDAS, FICHIERS CSV, ETC)

ATOUSSI Amina BOUBAKEUR Wafa

1 Environnement de travail :

Définition du Python : Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels.

Pour programmer en Python, on a besoin des ressources suivantes :

• un interpréteur Python installé sur sa machine

• un éditeur de texte ou un EDI pour produire des programmes Python (VS Code ou jupyter)

2 Prise en main du langage Python :

Les bibliothèques :
Pandas : est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données.
Seaborn : une bibliothèque permettant de créer des graphiques statistiques en Python. Elle s'intègre avec les structures Pandas.
matplotlib.pyplot :La bibliothèque matplotlib.pyplot, ou simplement plt, est un outil de visualisation de données en Python.

3 Base des données :

Dans ce tp la base des données est un fishier CSV (Comma-Separated Values) est un format texte ouvert représentant des données tabulaires sous forme de valeurs séparées par le délimiteur (;).

4 Phase de prétraitement :

Introduction : le but de cette partie est de prétraiter les données en

4.1 Visualiser notre Data :

Le fichier(titanic-passengers.csv) est visualisé avec la fonction read_csv() avec le paramètre delimiter (ou sep) pour spécifier le caractère de délimiteur .

```
In [2]: import pandas as pd
        df = pd.read_csv(r"C:\Users\am\Downloads\tp1.csv", delimiter=';')
```

```
In [1]: import pandas as pd
        df = pd.read_csv(r"C:\Users\am\Downloads\tp1.csv", sep=';')
```

4.2 Vérification des informations manquantes :

Les valeurs nulles (ou manquantes) sont des cellules dans un ensemble de données qui ne contiennent aucune information. Elles sont représentées par "NaN" (Not a Number) dans les DataFrame pandas en Python. Figure

Le code suivant nous permet de visualiser les valeurs nulles on obtient un tableau de valeurs booléennes de la même forme que votre DataFrame d'origine, indiquant quelles cellules contiennent des valeurs nulles avec :

Cabin
NaN
F G73
NaN
NaN
B71
...
NaN
NaN
NaN
NaN
NaN

Cabin
True
False
True
True
False
...
True
True
True
True
True

4.2.1 "False" :

signifie que la cellule dans le DataFrame d'origine ne contient pas de valeur nulle, c'est-à-dire que la donnée est présente.

4.2.2 "True" :

signifie que la cellule dans le DataFrame d'origine contient une valeur nulle, c'est-à-dire qu'il manque des données à cet emplacement.

4.3 La gestion des valeurs nulles :

Est essentielle dans le prétraitement pour garantir la précision des analyses et améliorer la qualité des résultats. Comme méthodes Il existe plusieurs, notamment : Élimination Remplissage Imputation.

4.3.1 Méthode1 (Remplissage) :

Remplacer les valeurs nulles par la valeur la plus fréquente dans la colonne dans l'exemple c'est 'G6' pour la colonne 'Cabine' et 'S'pour 'Embarked'.

```
Embarked
S      644
C      168
Q       77
..
```

```
Cabin
G6                  4
B96 B98             4
C23 C25 C27         4
F33                 3
D                   3
                   ..
C91                 1
D45                 1
F G63               1
A34                 1
E63                 1
```

4.3.2 Méthode 2 Imputation :

Imputer les valeurs nulles dans une colonne par la moyenne des valeurs non nulles de cette colonne cet méthode est utilisé pour la colonne 'Age' :

```
In [20]: df['Age'].fillna(df['Age'].mean(),inplace=True)
         df
```

Out[20]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Ticket	Fare	Embarked
0	343	No	2	28.000000	0	0	248740	13.0000	S
1	76	No	3	25.000000	0	0	348123	7.6500	S
2	641	No	3	20.000000	0	0	350050	7.8542	S
3	568	No	3	29.000000	0	4	349909	21.0750	S
4	672	No	1	31.000000	1	0	F.C. 12750	52.0000	S
...	...	...	...	...	...	...	...	...	...
886	10	Yes	2	14.000000	1	0	237736	30.0708	C
887	61	No	3	22.000000	0	0	2669	7.2292	C
888	535	No	3	30.000000	0	0	315084	8.6625	S
889	102	No	3	29.699118	0	0	349215	7.8958	S
890	428	Yes	2	19.000000	0	0	250655	26.0000	S

4.4 La vérification que les données sont prêtes à être traitées :

```
In [5]: df.isnull().sum()

Out[5]: PassengerId      0
        Survived         0
        Pclass           0
        Name             0
        Sex              0
        Age            177
        SibSp            0
        Parch            0
        Ticket           0
        Fare             0
        Cabin          687
        Embarked         2
        dtype: int64
```

Avant

```
In [23]: df.isnull().sum()

Out[23]: PassengerId    0
         Survived       0
         Pclass         0
         Age            0
         SibSp          0
         Parch          0
         Ticket         0
         Fare           0
         Embarked       0
         dtype: int64
```

Apres

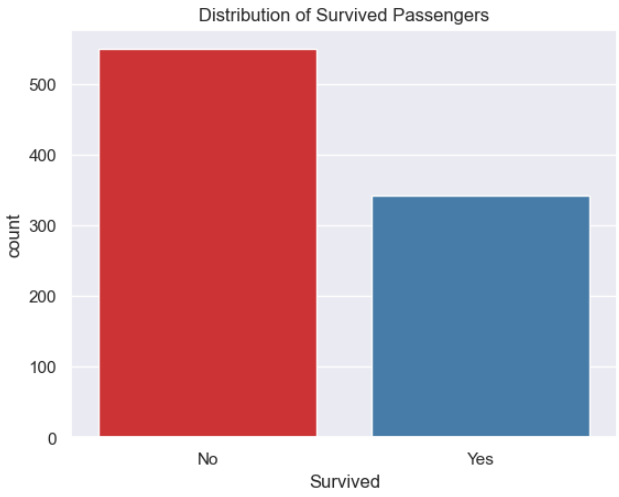
5 Conclusion 1 :

Le prétraitement des données dans les fichiers CSV est essentiel. Il permet d'assurer la qualité des données, d'éliminer les valeurs manquantes, et de rendre les données exploitables pour la phase de visualisation.

5 Phase de visualisation : (this part i will write it all in English)

5.1 Visualising the colonne 'SURVIVED':

after importing matplotlib.pyplot and Seaborn we are able to analyse the number of survivals using the code sns.countplot('here I witre the name of the colonne',data=df) This code will create a countplot showing the distribution of passengers who survived (Yes) and those who did not survive (No) in the Titanic dataset .



COMMENT: we can tell from this countplot that the number of people who died is way more then the survivals

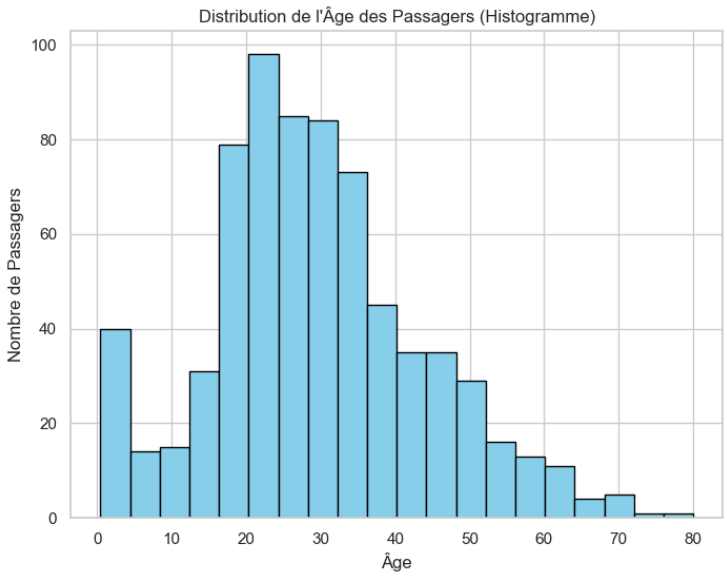
THE PROBLEM: based on the previous study the problem is to be able to predict if the passenger survived or not.

the most important features: the most important features are 'AGE', 'PCLASS', 'SEX'

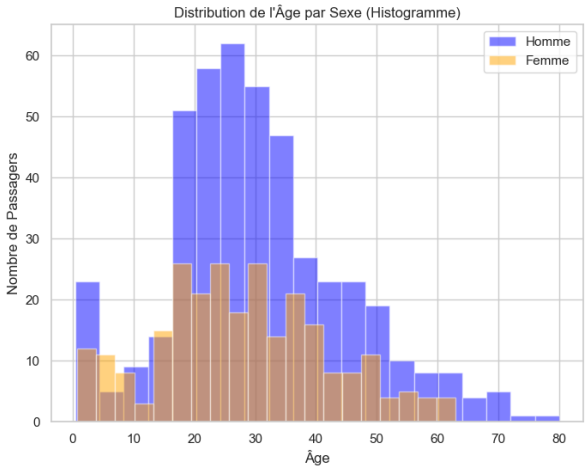
5.2 Analyzing the different colons using python:

5.3 Histogramme:

5.3.1 Distribution of age using histogram: we used the instruction plt.hist() the results show that young adults age around 20 to30 are more present .



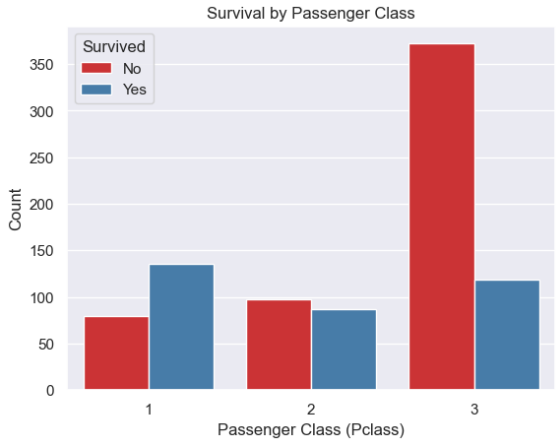
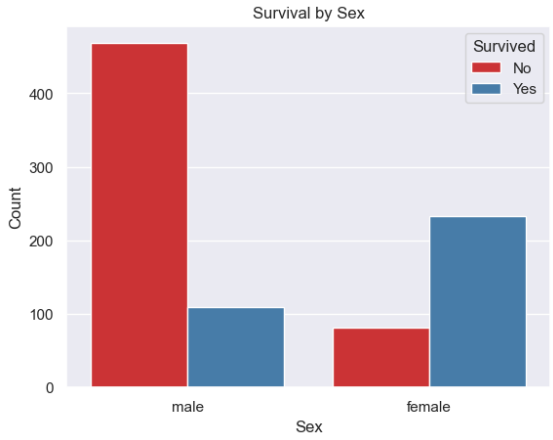
5.3.2 The correlation between age and sex we used histogram it allows us to visualize in form of a chart the results this figure showes that the number of young male adults are more than wemen and the number of kids is quite remarkable. based on this we can tell what category of people



5.4 COUNTPLOT:

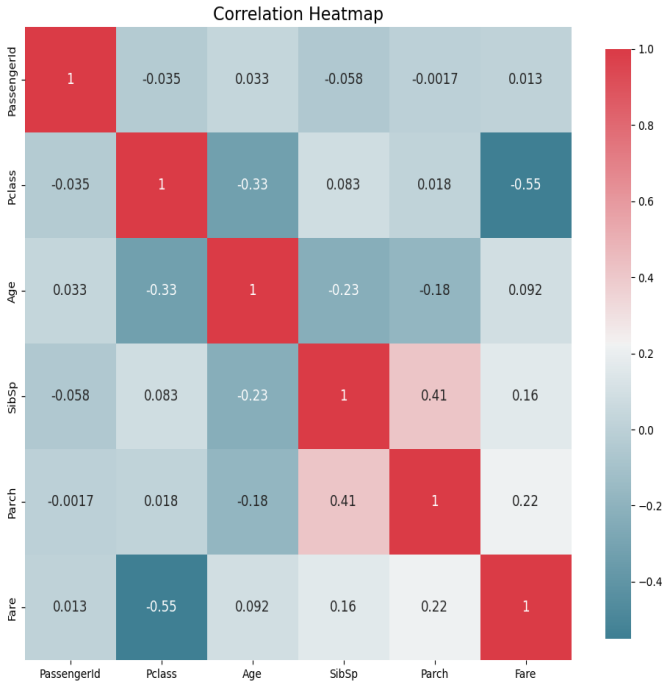
For question 4 the code used is: sns.countplot(x="Sex", hue="Survived", data=df ") x is the first parameter and I introduced the second parameter in the code using 'hue' .

I was able to use countplot for sex and passenger class (1,2,3) only. I tried it for age countplot is not right to visualize a colonne with a various number of variables.

Survival and passenger class	Survival and sex
	
• The first class the people that lived are more then the people that died • The second class the amount of people that died is more then the survivals • The third class the majority of passengers died	• Male the majority died around 500 deaths and 100 survivals • Females the majority of females lived
Conclusion: Based on these observations females and 1st class passengers are more likely to survive	

5.5 Correlation heatmap:

The paragraphe explaining the figure:



before using this function clones that includes alphabets must be dropped. the code used to drop the colons at ones is the following:

- columns_to_drop = ['Ticket','Embarked','Survived','Sex', 'Name'].
- df.drop(columns=columns_to_drop, axis=1, inplace=True).

the correlation heat map is a function that calculates the correlation between numirique colonnes only it uses Seaborn and matplotlib.pyplot library only. It a function that ranges from -1 to +1

After running the function:

I observe the various degrees of numbers and bunch of numbers with 1 is the color red and -1 is dark blue

this function gives the linear relationship between two colons in other words we realize the degree of each colon on the other with the following interpretation:

- negative correlation: [-1,0] if one varible goes down the other goes down
  as example a passengers Fare goes down the Pclass goes down
- no correlation: the closer it is to 0 there is no mutual impact like age and Fare and passengerId and other variables
- positive correlation: the closer it is to 1 for example if someone is traveling with family parch (parents and children) fare goes up we can see from the heatmap it at 0.22 between the two variables and we have sibsp (sibling and spouse) and parch is at 0.4

conclusion 2: based on the observations female and a first class rich passenger were more likely to survive they were priority.

As a conclusion:

Preprocessing data is very important in further analysis like countplot and histogram we used different techniques like replacing deleting colons to achieve a data that's ready for examining and visualizing

Using python instructions from matplot and seaborn

- We were able to visualize various variables like sex(countplot) and age(hist)
- We were able to see the correlation between survived and pclass and sex
- We visualized the correlation heatmap of different variables

It helped us understand furthermore the data frame that we had and helped us solve our problem that we had.