

# TP #6- Correction

## Manipulations de tableaux avec Pandas

### I. Création d'un nouveau projet

Ouvrez l'IDE de votre choix (VSCode ou PyCharm), et initiez un projet vierge (tp6).

Créez un fichier de type python (tp6.py).

Installez la librairie Pandas (pip install pandas).

### II. Chargement d'un jeu de données depuis un fichier .csv

#### Exercice 1

Récupérez les données du tableau dans le fichier titanic.csv grâce à la librairie pandas, et affichez le dans la console.

```
import pandas as pd

df = pd.read_csv("titanic.csv")
print(df)
```

	PassengerId	Survived	Pclass	Name	Sex	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	0	113803
4	5	0	3	Allen, Mr. William Henry	male	0	373450
...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	0	211536
887	888	1	1	Graham, Miss. Margaret Edith	female	0	112053
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	2	W./C. 6607
889	890	1	1	Behr, Mr. Karl Howell	male	0	111369
890	891	0	3	Dooley, Mr. Patrick	male	0	370376

### III. Opérations basiques

#### Exercice 2

Affichez l'âge moyen de tous les passagers, puis l'âge minimum et l'âge maximum.

Indice : Utilisez les fonctions `mean()`, `min()`, et `max()`.

```
print('Âge moyen :', df['Age'].mean())
print('Âge minimum :', df['Age'].min())
print('Âge maximum :', df['Age'].max())
```

```
Âge moyen : 29.69911764705882
Âge minimum : 0.42
Âge maximum : 80.0
```

### Exercice 3

Triez les passagers par Age croissant.

Indice : Utilisez `df[df['Age'].notna()]` pour supprimer les valeurs nulles.

```
df = df[df['Age'].notna()]
df = df.sort_values(by=['Age'])
print(df)
```

	PassengerId	Survived	Pclass	Name	Sex	Age
803	804	1	3	Thomas, Master. Assad Alexander	male	0.42
755	756	1	2	Hamalainen, Master. Viljo	male	0.67
644	645	1	3	Baclini, Miss. Eugenie	female	0.75
469	470	1	3	Baclini, Miss. Helene Barbara	female	0.75
78	79	1	2	Caldwell, Master. Alden Gates	male	0.83
..	...	...	...	...	...	...
116	117	0	3	Connors, Mr. Patrick	male	70.50
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.00
96	97	0	1	Goldschmidt, Mr. George B	male	71.00
851	852	0	3	Svensson, Mr. Johan	male	74.00
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.00

### Exercice 4

Créez un nouveau tableau qui ne contient que l'identifiant et le nom des hommes qui ont payé plus de 10\$.

```
# En une seule ligne
print(df[(df['Sex'] == "male") & (df['Fare'] > 10.0)][['PassengerId', 'Name', 'Sex']])

# Etape par étape
df = df[(df['Sex'] == "male") & (df['Fare'] > 10.0)] # Filtre lignes
df = df[['PassengerId', 'Name', 'Sex']] # Filtre colonnes
print(df)
```

	PassengerId	Name	Sex
6	7	McCarthy, Mr. Timothy J	male
7	8	Palsson, Master. Gosta Leonard	male
13	14	Andersson, Mr. Anders Johan	male
16	17	Rice, Master. Eugene	male
17	18	Williams, Mr. Charles Eugene	male
..	...	...	...
867	868	Roebeling, Mr. Washington Augustus II	male
869	870	Johnson, Master. Harold Theodor	male
883	884	Banfield, Mr. Frederick James	male
886	887	Montvila, Rev. Juozas	male
889	890	Behr, Mr. Karl Howell	male

### Exercice 5

Supprimez tous les passagers de la 3<sup>ème</sup> classe.

```
df = df.drop(df[df['Pclass'] == 3].index)
print(df)
```

	PassengerId	Survived	Pclass	Name
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)
6	7	0	1	McCarthy, Mr. Timothy J
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)
11	12	1	1	Bonnell, Miss. Elizabeth
..	...	...	...	...
880	881	1	2	Shelley, Mrs. William (Imanita Parrish Hall)
883	884	0	2	Banfield, Mr. Frederick James
886	887	0	2	Montvila, Rev. Juozas
887	888	1	1	Graham, Miss. Margaret Edith
889	890	1	1	Behr, Mr. Karl Howell

## IV. Partitionnement (Group By)

### Exercice 6

Pour chaque sexe, afficher l'âge moyen, et le prix de billet moyen.

```
print(df.groupby(['Sex'])[['Age', 'Fare']].mean())
```

	Age	Fare
Sex		
female	27.915709	44.479818
male	30.726645	25.523893

### Exercice 7

Pour chaque sexe et chaque classe de billet, afficher l'âge moyen, et le prix de billet moyen.

```
print(df.groupby(['Sex', 'Pclass'])[['Age', 'Fare']].mean())
```

		Age	Fare
Sex	Pclass		
female	1	34.611765	106.125798
	2	28.722973	21.970121
	3	21.750000	16.118810
male	1	41.281386	67.226127
	2	30.740707	19.741782
	3	26.507589	12.661633

## V. Substituer des valeurs par d'autres

### Exercice 8

Remplacez tous les noms des passagers par leur nom en majuscule.

```
df['Name'] = df['Name'].map(lambda name: name.upper())
# OU :
df['Name'] = df.apply(lambda row: row['Name'].upper(), axis=1)

print(df)
```

	PassengerId	Survived	Pclass	Name
0	1	0	3	BRAUND, MR. OWEN HARRIS
1	2	1	1	CUMINGS, MRS. JOHN BRADLEY (FLORENCE BRIGGS TH...
2	3	1	3	HEIKKINEN, MISS. LAINA
3	4	1	1	FUTRELLE, MRS. JACQUES HEATH (LILY MAY PEEL)
4	5	0	3	ALLEN, MR. WILLIAM HENRY
..	...	...	...	...
886	887	0	2	MONTVILA, REV. JUOZAS
887	888	1	1	GRAHAM, MISS. MARGARET EDITH
888	889	0	3	JOHNSTON, MISS. CATHERINE HELEN "CARRIE"
889	890	1	1	BEHR, MR. KARL HOWELL
890	891	0	3	DOOLEY, MR. PATRICK

### Exercice 9

Ajoutez une nouvelle colonne dans le dataframe, qui contient la valeur « Retraité » si le passager a plus de 65 ans, « En activité » sinon.

```
df['Activité'] = df.apply(lambda row: ('Retraité' if row['Age'] > 65 else 'En
activité'), axis=1)
df = df[df['Age'].notna()]
df = df.sort_values(by=['Age'])
print(df[['Name', 'Age', 'Activité']])
```

	Name	Age	Activité
803	Thomas, Master. Assad Alexander	0.42	En activité
755	Hamalainen, Master. Viljo	0.67	En activité
644	Baclini, Miss. Eugenie	0.75	En activité
469	Baclini, Miss. Helene Barbara	0.75	En activité
78	Caldwell, Master. Alden Gates	0.83	En activité
..	...	...	...
116	Connors, Mr. Patrick	70.50	Retraité
493	Artagaveytia, Mr. Ramon	71.00	Retraité
96	Goldschmidt, Mr. George B	71.00	Retraité
851	Svensson, Mr. Johan	74.00	Retraité
630	Barkworth, Mr. Algernon Henry Wilson	80.00	Retraité