lab01(Pandas)

October 25, 2023

0.0.1 Utilisation de la classe DataFrame (Pendas)

Module “Machine Learning”, MST IASD/S1 2023-2024 (M. AIT KBIR - FST de Tanger)

[1]:
```
import seaborn as sns
```

[2]:
```
# Chargement des données
df = sns.load_dataset("titanic")

# L'objet retourné est un DataFrame
type(df)
```

[2]:
```
pandas.core.frame.DataFrame
```

[3]:
```
df.tail(5) # Les 5 derniers exemples
```

[3]:
```
     survived  pclass     sex   age  sibsp  parch   fare embarked   class  \
886         0       2    male  27.0      0      0  13.00        S  Second
887         1       1  female  19.0      0      0  30.00        S   First
888         0       3  female   NaN      1      2  23.45        S   Third
889         1       1    male  26.0      0      0  30.00        C   First
890         0       3    male  32.0      0      0   7.75        Q   Third

       who  adult_male deck  embark_town alive  alone
886    man        True  NaN  Southampton    no   True
887  woman       False    B  Southampton   yes   True
888  woman       False  NaN  Southampton    no  False
889    man        True    C    Cherbourg   yes   True
890    man        True  NaN   Queenstown    no   True
```

[4]:
```
df.info() # Informations générales
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
```

```
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

[5]:
```
df.columns # Les Champs ou attributs
# df.values # Récupérer sousforme d'un objet numpy.array,
```

[5]:
```
Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
       'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
       'alive', 'alone'],
      dtype='object')
```

[6]:
```
df.shape # Dimensions
```

[6]:
```
(891, 15)
```

[7]:
```
df.describe() # Résumé sur chaque colonne, avec un count < 891(ici Age), on␣
↪peut déduire qu'il y a des exemples sans valeurs pour cet axe
```

[7]:

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

[8]:
```
df.sort_values("age").head(20) ### Tri par ordre croissant du nom, puis␣
↪affchage des 10 premiers exemples
```

[8]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	\
803	1	3	male	0.42	0	1	8.5167	C	Third	
755	1	2	male	0.67	1	1	14.5000	S	Second	
644	1	3	female	0.75	2	1	19.2583	C	Third	

469	1	3	female	0.75	2	1	19.2583	C	Third
78	1	2	male	0.83	0	2	29.0000	S	Second
831	1	2	male	0.83	1	1	18.7500	S	Second
305	1	1	male	0.92	1	2	151.5500	S	First
827	1	2	male	1.00	0	2	37.0042	C	Second
381	1	3	female	1.00	0	2	15.7417	C	Third
164	0	3	male	1.00	4	1	39.6875	S	Third
183	1	2	male	1.00	2	1	39.0000	S	Second
386	0	3	male	1.00	5	2	46.9000	S	Third
172	1	3	female	1.00	1	1	11.1333	S	Third
788	1	3	male	1.00	1	2	20.5750	S	Third
642	0	3	female	2.00	3	2	27.9000	S	Third
7	0	3	male	2.00	3	1	21.0750	S	Third
530	1	2	female	2.00	1	1	26.0000	S	Second
297	0	1	female	2.00	1	2	151.5500	S	First
824	0	3	male	2.00	4	1	39.6875	S	Third
205	0	3	female	2.00	0	1	10.4625	S	Third

	who	adult_male	deck	embark_town	alive	alone
803	child	False	NaN	Cherbourg	yes	False
755	child	False	NaN	Southampton	yes	False
644	child	False	NaN	Cherbourg	yes	False
469	child	False	NaN	Cherbourg	yes	False
78	child	False	NaN	Southampton	yes	False
831	child	False	NaN	Southampton	yes	False
305	child	False	C	Southampton	yes	False
827	child	False	NaN	Cherbourg	yes	False
381	child	False	NaN	Cherbourg	yes	False
164	child	False	NaN	Southampton	no	False
183	child	False	F	Southampton	yes	False
386	child	False	NaN	Southampton	no	False
172	child	False	NaN	Southampton	yes	False
788	child	False	NaN	Southampton	yes	False
642	child	False	NaN	Southampton	no	False
7	child	False	NaN	Southampton	no	False
530	child	False	NaN	Southampton	yes	False
297	child	False	C	Southampton	no	False
824	child	False	NaN	Southampton	no	False
205	child	False	G	Southampton	no	False

Calculs

[9]:
```
df['age'].mean() # Age moyen des passagers
```

[9]: 29.69911764705882

[10]:
```
df['age'].min() # Age minimum
```

[10]:
```
0.42
```

[11]:
```
df['embarked'].value_counts() # Nombre de survivants 342
```

[11]:
```
S    644
C    168
Q     77
Name: embarked, dtype: int64
```

[12]:
```
type(df["pclass"].unique()) # Nombre de classes différentes
```

[12]:
```
numpy.ndarray
```

[13]:
```
age=60
df[df['age']==age] # Les passagers agés de "age" ans
```

[13]:
```
     survived  pclass     sex   age  sibsp  parch   fare embarked   class  \
366         1       1  female  60.0      1      0  75.25        C   First
587         1       1    male  60.0      1      1  79.20        C   First
684         0       2    male  60.0      1      1  39.00        S  Second
694         0       1    male  60.0      0      0  26.55        S   First

       who  adult_male deck  embark_town alive  alone
366  woman       False    D    Cherbourg   yes  False
587    man        True    B    Cherbourg   yes  False
684    man        True  NaN  Southampton    no  False
694    man        True  NaN  Southampton    no   True
```

[14]:
```
df[(df['age']==age) & (df['survived']==1)] # Les passagers survivants, agés de␣
↪"age" ans
```

[14]:
```
     survived  pclass     sex   age  sibsp  parch   fare embarked  class  \
366         1       1  female  60.0      1      0  75.25        C  First
587         1       1    male  60.0      1      1  79.20        C  First

       who  adult_male deck embark_town alive  alone
366  woman       False    D   Cherbourg   yes  False
587    man        True    B   Cherbourg   yes  False
```

[15]:
```
df.isnull().sum() # df.isnull(): chaque cellule prend True ou False
```

[15]:
```
survived      0
pclass        0
sex           0
age         177
sibsp         0
parch         0
fare          0
```

```
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

[16]:
```
df[df['age'].isnull()==True] # Exemples avec age non disponible
```

[16]:
```
     survived  pclass     sex  age  sibsp  parch     fare embarked   class  \
5           0       3    male  NaN      0      0   8.4583        Q   Third
17          1       2    male  NaN      0      0  13.0000        S  Second
19          1       3  female  NaN      0      0   7.2250        C   Third
26          0       3    male  NaN      0      0   7.2250        C   Third
28          1       3  female  NaN      0      0   7.8792        Q   Third
..        …     …      …   …    …    …       …      …     …
859         0       3    male  NaN      0      0   7.2292        C   Third
863         0       3  female  NaN      8      2  69.5500        S   Third
868         0       3    male  NaN      0      0   9.5000        S   Third
878         0       3    male  NaN      0      0   7.8958        S   Third
888         0       3  female  NaN      1      2  23.4500        S   Third

       who  adult_male deck  embark_town alive  alone
5      man        True  NaN   Queenstown    no   True
17     man        True  NaN  Southampton   yes   True
19   woman       False  NaN    Cherbourg   yes   True
26     man        True  NaN    Cherbourg    no   True
28   woman       False  NaN   Queenstown   yes   True
..     …           …  …          …    …    …
859    man        True  NaN    Cherbourg    no   True
863  woman       False  NaN  Southampton    no  False
868    man        True  NaN  Southampton    no   True
878    man        True  NaN  Southampton    no   True
888  woman       False  NaN  Southampton    no  False

[177 rows x 15 columns]
```

[18]:
```
# L'age moyen des passagers par sexe
df.groupby('sex').mean(numeric_only=True)['age']
```

[18]:
```
sex
female    27.915709
male      30.726645
Name: age, dtype: float64
```

Voir les autres méthodes : https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html