P10 - Statistiques, en utilisant numpy et pandas

Partie 1 - Avec numpy

On va reprendre l'analyse d'une série statistique vu en TP7 en utilisant cette fois les outils offerts par numpy.

Exercice 1

En utilisant numpy :

1. calculer la moyenne
2. calculer l'écart-type
3. calculer la médiane
4. calculer l'étendue

des notes du groupeA.

In [ ]:
```
groupeA = [47,78,50,73,55,39,62,41,80,50,58,54,51,57,62,77,72,68,47,46,81,69,28,63,49,44,71,66,38,48,53,49,59,6
```

In [2]:
```
import numpy as np
print('moyenne =',np.mean(groupeA))
print('écart-type =',np.std(groupeA))
print('étendue =', np.max(groupeA) - np.min(groupeA))
print('médiane =', np.median(groupeA))
```

```
moyenne = 60.98876404494382
écart-type = 15.584848866856301
étendue = 94
médiane = 61.0
```

Partie 2 - Utilisation de pandas pour étudier les données d'un tableur

Le fichier titanic.csv est un tableur contenant les données des passagers du Titanic. Nous allons l'ouvrir et l'étudier avec python en utilisant la bibliothèque pandas (voir documentation jointe).

In [3]:
```
import pandas as pd
titanic = pd.read_csv('titanic.csv')
```

In [4]:
```
titanic
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

- PassengerId (identifiant passager)
- Survived (0 : décédé, 1 : a survécu)
- Pclass (classe, de 1 à 3)
- Name (Nom, prénom et titre)
- Sex (male/female)
- Age (en années)
- SibSp (nombre de frère, soeur, beau-frère, belle soeur, mari ou femme)
- Parch (nombre de parents et d'enfants)
- Ticket (numéro du billet)
- Fare (prix du billet)
- Cabin (numéro de cabine)
- Embarked (port d'emarquement : C - Cherbourg, S - Southampton, Q = Queenstown)

Exercice 2 - manipulations de base

En utilisant la documentation pandas :

1. afficher les colonnes et leur type (avec info);
   int** -> entier, float** -> nombre décimal, object -> donnée non numérique
2. afficher (un aperçu de) la liste des passagers, uniquement leur nom (Name);
3. afficher les statistiques de base avec describe.

In [5]:
```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [6]:
```
titanic['Name']
```

Out[6]:
```
0                              Braund, Mr. Owen Harris
1    Cumings, Mrs. John Bradley (Florence Briggs Th...
2                               Heikkinen, Miss. Laina
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                             Allen, Mr. William Henry
                           ...                        
886                              Montvila, Rev. Juozas
887                       Graham, Miss. Margaret Edith
888           Johnston, Miss. Catherine Helen "Carrie"
889                              Behr, Mr. Karl Howell
890                                Dooley, Mr. Patrick
Name: Name, Length: 891, dtype: object
```

In [7]:
```
titanic.describe()
```

Out[7]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Exercice 3 - étude des survivants

1. Que fait titanic['Name'].count() ?

In [8]:
```
titanic['Name'].count()
```

Out[8]:
```
891
```

2. Calculer la proportion des passagers qui a survécu au naufrage, de deux façons différentes.

In [9]:
```
titanic['Survived'].mean()
```

Out[9]:
```
0.3838383838383838
```

In [10]:
```
titanic['Survived'].sum()/titanic['Name'].count()
```

Out[10]:
```
0.3838383838383838
```

3. Calculer cette même proportion mais uniquement pour les passagers de première classe.

In [11]:
```
# Prenons la table filtrée qui ne contient que les données de la première classe
titanic1 = titanic[ titanic['Pclass'] == 1 ]
titanic1
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S
...	...	...	...	...	...	...	...	...	...	...	...	...
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

216 rows × 12 columns

In [12]:
```
titanic1['Survived'].mean()
```

Out[12]:
```
0.6296296296296297
```

4. Faire de même pour la deuxième et troisième classe.

In [13]:
```
titanic2 = titanic[ titanic['Pclass'] == 2 ]
print('2e classe :', titanic2['Survived'].mean())

titanic3 = titanic[ titanic['Pclass'] == 3 ]
print('3e classe :', titanic3['Survived'].mean())
```

```
2e classe : 0.47282608695652173
3e classe : 0.24236252545824846
```

5. Comparer maintenant le taux de survie chez les hommes et les femmes.

In [14]:
```
hommes = titanic[ titanic['Sex'] == 'male' ]
print('hommes :', hommes['Survived'].mean())

femmes = titanic[ titanic['Sex'] == 'female' ]
print('femmes :', femmes['Survived'].mean())
```

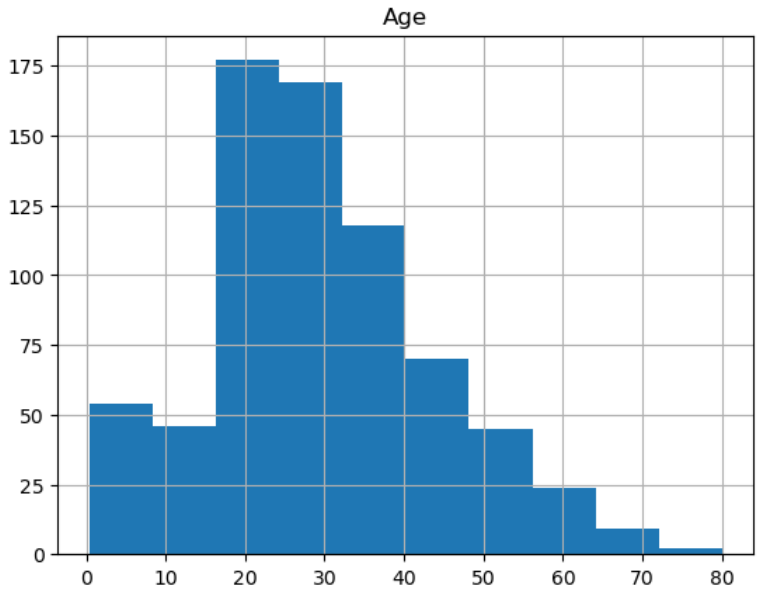
```
hommes : 0.18890814558058924
femmes : 0.7420382165605095
```

Exercice 4 - représentations graphiques

1. Tracer l'histogramme de répartition des âges dans le navire.

In [15]:
```
titanic.hist(column = 'Age')
```

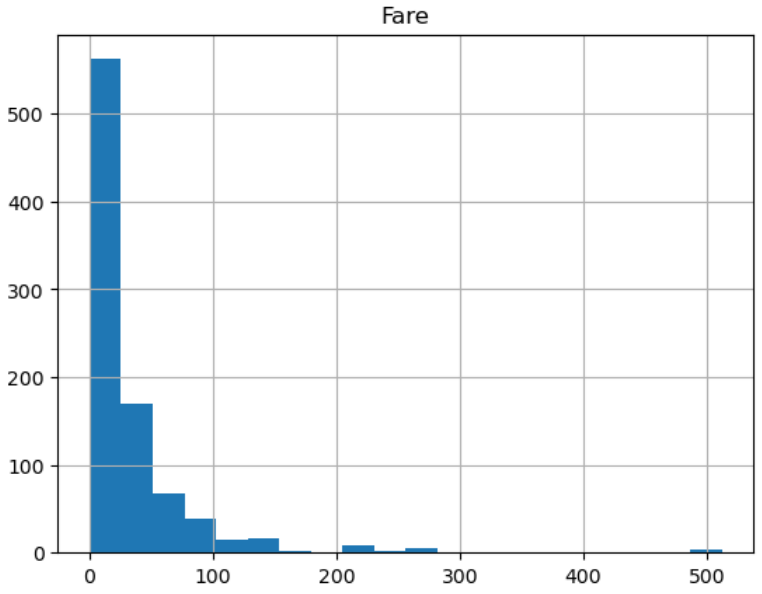
Out[15]:
```
array([[<AxesSubplot: title={'center': 'Age'}>]], dtype=object)
```



2. Faire de même avec le prix des billets,avec 20 intervalles. (bins).
3. On remarque une valeur isolée, on va l'ignorer pour la suite. Tracer les histogrammes de prix des billets sur [0,300] découpé en 20 intervalles, pour chaque classe du navire séparemment.

In [16]:
```
titanic.hist(column = 'Fare', bins = 20)
```

Out[16]:
```
array([[<AxesSubplot: title={'center': 'Fare'}>]], dtype=object)
```

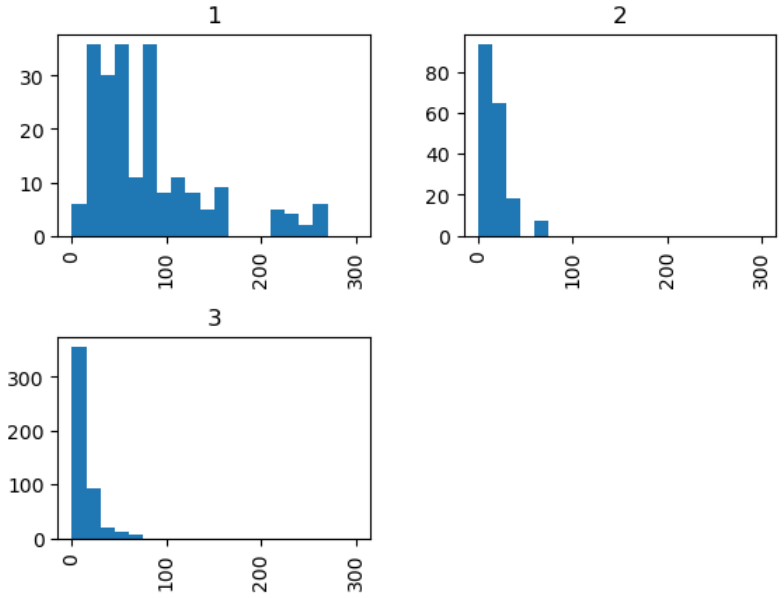


In [17]:
```
titanic['Fare'].max()
```

Out[17]: 512.3292

In [18]:
```
titanic.hist(column = 'Fare', by = 'Pclass', range=(0,300), bins = 20)
```

Out[18]:
```
array([[<AxesSubplot: title={'center': '1'}>,
        <AxesSubplot: title={'center': '2'}>],
       [<AxesSubplot: title={'center': '3'}>, <AxesSubplot: >]],
      dtype=object)
```



Exercice 5 - compléments d'étude

1. Déterminer le prix moyen du billet par classe de voyage ainsi que l'écart-type.

In [19]:
```
for classe in [1,2,3]:
    #filtre
    titanic_classe = titanic[ titanic['Pclass'] == classe ]
    moyenne = titanic_classe['Fare'].mean()
    print('Prix moyen en classe', classe, ':', moyenne)
```

```
Prix moyen en classe 1 : 84.1546875
Prix moyen en classe 2 : 20.662183152173913
Prix moyen en classe 3 : 13.675550101832993
```

In [20]:
```
for classe in [1,2,3]:
    #filtre
    titanic_classe = titanic[ titanic['Pclass'] == classe ]
    ecart = titanic_classe['Fare'].std()
    print('Écart-type en classe', classe, ':', ecart)
```

```
Écart-type en classe 1 : 78.38037264672882
Écart-type en classe 2 : 13.417398756149343
Écart-type en classe 3 : 11.778141704387307
```

2. Mettre en avant d'autres critères discriminants (ou pas) concernant la probabilité de survie d'un passager.

In [21]:
```
# On a déjà vu la classe et le genre. Étudions l'âge.
max_age = int(titanic['Age'].max())
for age in range(0,max_age+10,10): # age par dizaine
    data = titanic[(age <= titanic['Age']) & (titanic['Age'] < age+10)]
    taux = data['Survived'].mean()
    print('Age', age, 'à', age+10, ':', taux)
```

```
Age 0 à 10 : 0.6129032258064516
Age 10 à 20 : 0.4019607843137255
Age 20 à 30 : 0.35
Age 30 à 40 : 0.437125748502994
Age 40 à 50 : 0.38202247191011235
Age 50 à 60 : 0.4166666666666667
Age 60 à 70 : 0.3157894736842105
Age 70 à 80 : 0.0
Age 80 à 90 : 1.0
```

In [ ]: