

Qusetion 3

Entrée [1]: *#changement des librairies*

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

SQ.1

Entrée [2]: *#Lecture de la base smp*

```
smp=pd.read_csv("smp2.csv", sep=";")
```

Entrée [3]: *print(smp)# tabulation de la base*

```

      age      prof  duree  discip  n.enfant  n.fratie
ecole \
0    31.0      autre    4.0    0.0        2.0        4
1.0
1    49.0        NaN    NaN    0.0        7.0        3
2.0
2    50.0  prof.intermediaire    5.0    0.0        2.0        2
2.0
3    47.0      ouvrier    NaN    0.0        0.0        6
1.0
4    23.0    sans emploi    4.0    1.0        1.0        6
1.0
..     ...      ...      ...      ...      ...      ...
...
794  28.0    sans emploi    5.0    0.0        1.0        4
1.0
795  44.0      ouvrier    4.0    1.0        1.0       12
2.0
796  31.0      cadre    4.0    0.0        3.0        6
4.0
797  38.0      employe    5.0    0.0        0.0        1
3.0
798  71.0      autre    4.0    0.0        2.0        4
1.0

      separation  juge.enfant  place  ...  subst.cons  scz.cons  ch
ar  rs \
0      0.0      0.0    0.0  ...      0      0    1
.0  2.0
1      1.0      0.0    0.0  ...      0      0    1
.0  2.0
2      0.0      0.0    0.0  ...      0      0    1
.0  2.0
```

```

3      1.0      0.0      1.0      ...      0      0      1
.0  2.0
4      1.0      NaN      1.0      ...      0      0      1
.0  2.0
..      ...      ...      ...      ...      ...      ...
..      ...
794      NaN      1.0      0.0      ...      0      0      N
aN  3.0
795      0.0      0.0      0.0      ...      0      0      1
.0  1.0
796      1.0      1.0      1.0      ...      1      0      2
.0  1.0
797      0.0      0.0      0.0      ...      0      0      1
.0  1.0
798      1.0      0.0      0.0      ...      0      0      1
.0  1.0

      ed      dr      suicide.s      suicide.hr      suicide.past      dur.interv
0      1.0      1.0      0.0      0.0      0.0      NaN
1      2.0      1.0      0.0      0.0      0.0      70.0
2      3.0      2.0      0.0      0.0      0.0      NaN
3      2.0      2.0      1.0      0.0      0.0      105.0
4      2.0      2.0      0.0      0.0      1.0      NaN
..      ...      ...      ...      ...      ...      ...
794      1.0      3.0      1.0      0.0      0.0      60.0
795      1.0      1.0      0.0      0.0      1.0      105.0
796      1.0      1.0      1.0      0.0      0.0      105.0
797      1.0      1.0      0.0      0.0      0.0      75.0
798      1.0      1.0      0.0      0.0      0.0      45.0

```

[799 rows x 26 columns]

Entrée [4]: `smp.columns`

Out[4]: `Index(['age', 'prof', 'duree', 'discip', 'n.enfant', 'n.fratricie', 'ecole', 'separation', 'juge.enfant', 'place', 'abus', 'grav.cons', 'dep.cons', 'ago.cons', 'ptsd.cons', 'alc.cons', 'subst.cons', 'scz.cons', 'char', 'rs', 'ed', 'dr', 'suicide.s', 'suicide.hr', 'suicide.past', 'dur.interv'], dtype='object')`

Entrée [5]: `#vérification du nombre d'observation et du nombre de variables`
`smp.shape`

Out[5]: `(799, 26)`

SQ.2

Entrée [6]: `smp.dtypes# les types des variables dans la base`

```
Out[6]: age          float64
prof          object
duree         float64
discip        float64
n.enfant      float64
n.fratie      int64
ecole         float64
separation    float64
juge.enfant   float64
place         float64
abus          float64
grav.cons     float64
dep.cons      int64
ago.cons      int64
ptsd.cons     int64
alc.cons      int64
subst.cons    int64
scz.cons      int64
char          float64
rs            float64
ed            float64
dr            float64
suicide.s     float64
suicide.hr    float64
suicide.past  float64
dur.interv    float64
dtype: object
```

Entrée [7]: `# recodation des variables qualitatives`

```
def duree(series):
    if series==1.0:
        return 'Moins de 1 mois'
    if series==2.0:
        return '1 à 6 mois'
    if series==3.0:
        return '6 mois à 1 an'
    if series==4.0:
        return '1 à 5 ans'
    if series==5.0:
        return '5 ans ou plus'
    else:
        return series

smp['duree']=smp['duree'].apply(duree)

def discipl(series):
    if series==0.0:
        .
        .
        .
```

```
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['discip']=smp['discip'].apply(discip1)

def ecole(series):
    if series==1.0:
        return 'sans diplome'
    if series==2.0:
        return 'college'
    if series==3.0:
        return 'CAP,BEP'
    if series==4.0:
        return 'Lycee'
    if series==5.0:
        return 'universite'
    else:
        return series

smp['ecole']=smp['ecole'].apply(ecole)

def separation(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['separation']=smp['separation'].apply(separation)

def jugeenfant(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['juge.enfant']=smp['juge.enfant'].apply(jugeenfant)

def place(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['place']=smp['place'].apply(place)
```

```
def abus(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['abus']=smp['abus'].apply(abus)

def gravcons(series):
    if series==1.0:
        return 'normal'
    if series==2.0:
        return 'limite'
    if series==3.0:
        return 'légèrement'
    if series==4.0:
        return 'modérément'
    if series==5.0:
        return 'manifestement'
    if series==6.0:
        return 'gravement'
    if series==7.0:
        return 'parmi les patients les plus malades'
    else:
        return series

smp['grav.cons']=smp['grav.cons'].apply(gravcons)

def depcons(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['dep.cons']=smp['dep.cons'].apply(depcons)

def agocons(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['ago.cons']=smp['ago.cons'].apply(agocons)

def ptsdcons(series):
    if series==0.0:
        return 'non'
    if series==1.0:
```

```
        return 'oui'
    else:
        return series

smp['ptsd.cons']=smp['ptsd.cons'].apply(ptsdcons)

def alccons(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['alc.cons']=smp['alc.cons'].apply(alccons)

def substcons(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['subst.cons']=smp['subst.cons'].apply(substcons)

def sczcons(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['scz.cons']=smp['scz.cons'].apply(sczcons)

def char(series):
    if series==1.0:
        return 'absent'
    if series==2.0:
        return 'léger'
    if series==3.0:
        return 'moyen'
    if series==4.0:
        return 'sévere'
    else:
        return series

smp['char']=smp['char'].apply(char)

def rs(series):
    if series==1.0:
        return 'faible'
    if series==2.0:
```

```
        return 'modéré'
    if series==3.0:
        return 'élevé'
    else:
        return series

smp['rs']=smp['rs'].apply(rs)

def ed(series):
    if series==1.0:
        return 'faible'
    if series==2.0:
        return 'modéré'
    if series==3.0:
        return 'élevé'
    else:
        return series

smp['ed']=smp['ed'].apply(ed)

def dr(series):
    if series==1.0:
        return 'faible'
    if series==2.0:
        return 'modéré'
    if series==3.0:
        return 'élevé'
    else:
        return series

smp['dr']=smp['dr'].apply(dr)

def suicidehr(series):# Je suis pas sure de cette recodage
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['suicide.hr']=smp['suicide.hr'].apply(suicidehr)

def suicidepast(series):
    if series==0.0:
        return 'non'
    if series==1.0:
        return 'oui'
    else:
        return series

smp['suicide.past']=smp['suicide.past'].apply(suicidepast)
```

SQ.3

Entrée [8]: `, 'alc.cons', 'subst.cons', 'scz.cons', 'char','rs', 'ed', 'dr', 'su`

Entrée [9]: `print(list1)`

	age	n.enfant	n.fratie	dur.interv
0	31.0	2.0	4	NaN
1	49.0	7.0	3	70.0
2	50.0	2.0	2	NaN
3	47.0	0.0	6	105.0
4	23.0	1.0	6	NaN
...
794	28.0	1.0	4	60.0
795	44.0	1.0	12	105.0
796	31.0	3.0	6	105.0
797	38.0	0.0	1	75.0
798	71.0	2.0	4	45.0

[799 rows x 4 columns]

Entrée [10]: `list1.describe()`

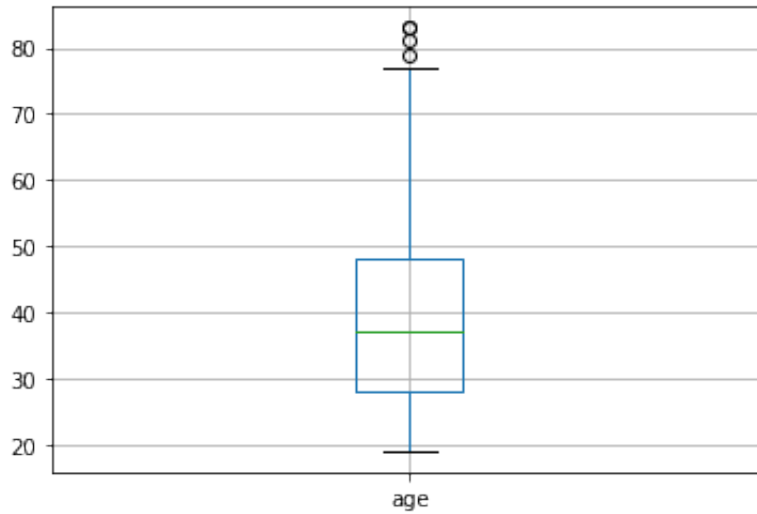
Out[10]:

	age	n.enfant	n.fratie	dur.interv
count	797.000000	773.000000	799.000000	749.000000
mean	38.899624	1.755498	4.286608	61.891856
std	13.280978	1.834044	3.441485	19.669605
min	19.000000	0.000000	0.000000	0.000000
25%	28.000000	0.000000	2.000000	48.000000
50%	37.000000	1.000000	3.000000	60.000000
75%	48.000000	3.000000	6.000000	75.000000
max	83.000000	13.000000	21.000000	120.000000

SQ.4

Entrée [11]: `smp.boxplot(column='age')`

Out [11]: `<matplotlib.axes._subplots.AxesSubplot at 0x1175065d0>`



SQ.5

Entrée [12]: `smp.loc[(smp['prof']=="agriculteur")&(smp['n.enfant']>2.0),:]`

Out [12]:

	age	prof	duree	discip	n.enfant	n.fratie	ecole	separation	juge.enfant	p
14	64.0	agriculteur	NaN	non	3.0	2	sans diplome	non	non	
311	42.0	agriculteur	1 à 5 ans	non	3.0	6	sans diplome	non	non	
390	36.0	agriculteur	1 à 5 ans	oui	3.0	4	CAP,BEP	oui	oui	
441	79.0	agriculteur	5 ans ou plus	non	5.0	6	college	non	non	

4 rows × 26 columns

SQ.6

Entrée [13]: `liste2=smp['prof'].value_counts()`

Entrée [14]: `print(liste2)`

```
ouvrier          227
sans emploi     222
employe         135
artisan          90
prof.intermediaire  58
autre           31
cadre           24
agriculteur       6
Name: prof, dtype: int64
```

Entrée [15]: `liste2.sum()`

Out[15]: 793

Entrée [16]: `liste2=(liste2/793)*100`

Entrée [17]: `print(liste2)`

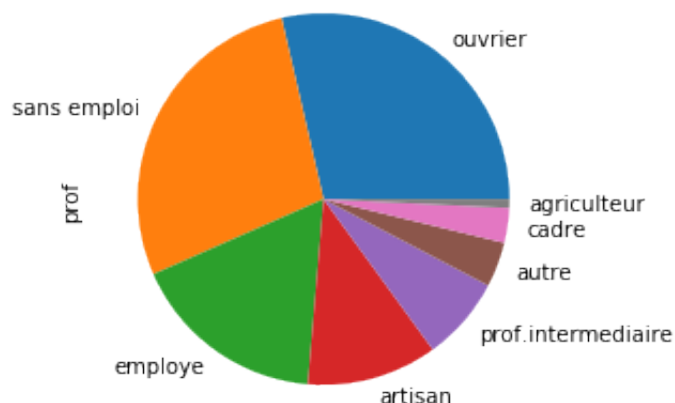
```
ouvrier          28.625473
sans emploi     27.994956
employe         17.023960
artisan          11.349306
prof.intermediaire  7.313997
autre           3.909206
cadre           3.026482
agriculteur      0.756620
Name: prof, dtype: float64
```

Entrée [18]: *# la catégorie ouvrier a le plus grand pourcentage, ainsi c'est la*

SQ.7

Entrée [19]: `liste2.plot.pie()`

Out[19]: `<matplotlib.axes._subplots.AxesSubplot at 0x1a1a8c2fd0>`



SQ.8

Entrée [20]: `pd.crosstab(smp['prof'], smp['age'], values=smp['age'], aggfunc=pd.Series)`

Out[20]:

	age	19.0	20.0	21.0	22.0	23.0	24.0	25.0	26.0	27.0	28.0	...	69.0	70.0	
prof															
agriculteur		NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	
artisan		NaN	20.0	NaN	NaN	NaN	24.0	NaN	26.0	27.0	28.0	...	NaN	NaN	
autre		19.0	20.0	21.0	NaN	23.0	24.0	25.0	26.0	27.0	NaN	...	NaN	70.0	
cadre		NaN	NaN	NaN	NaN	NaN	NaN	25.0	NaN	NaN	NaN	...	NaN	NaN	
employe		19.0	20.0	21.0	22.0	23.0	24.0	25.0	26.0	27.0	28.0	...	NaN	NaN	
ouvrier		19.0	20.0	21.0	22.0	23.0	24.0	25.0	26.0	27.0	28.0	...	69.0	NaN	
prof.intermediaire		19.0	NaN	NaN	22.0	NaN	NaN	NaN	26.0	27.0	28.0	...	NaN	NaN	
sans emploi		19.0	20.0	21.0	22.0	23.0	24.0	25.0	26.0	27.0	28.0	...	NaN	NaN	

8 rows × 60 columns

SQ.9

Entrée []:

SQ.10

Entrée []:

SQ.11Entrée [21]: `smp.dropna(inplace=True)`Entrée [22]: `print(smp)`

	age	prof	duree	discip	n.enfant	n.f
7	52.0	artisan	5 ans ou plus	non	2.0	
9						
8	42.0	ouvrier	1 à 5 ans	oui	1.0	
12						
12	21.0	employe	1 à 5 ans	non	0.0	
3						
13	40.0	artisan	1 à 5 ans	non	3.0	
5						
16	60.0	prof.intermediaire	5 ans ou plus	non	2.0	
4						
..	
...						
793	27.0	prof.intermediaire	1 à 5 ans	non	2.0	
0						
795	44.0	ouvrier	1 à 5 ans	oui	1.0	
12						
796	31.0	cadre	1 à 5 ans	non	3.0	
6						
797	38.0	employe	5 ans ou plus	non	0.0	
1						
798	71.0	autre	1 à 5 ans	non	2.0	
4						

	ecole	separation	juge.enfant	place	... subst.cons	scz
.cons \						
7	college	non	non	non	...	non
non						
8	sans diplome	oui	oui	non	...	oui
non						
12	college	oui	oui	oui	...	oui
non						
13	sans diplome	non	oui	non	...	non
non						
16	college	oui	non	oui	...	non
non						
..
...						
793	college	oui	non	oui	...	non
non						
795	college	non	non	non	...	non

```

non
796          Lycee          oui          oui    oui    ...          oui
non
797          CAP,BEP          non          non    non    ...          non
non
798 sans diplome          oui          non    non    ...          non
non

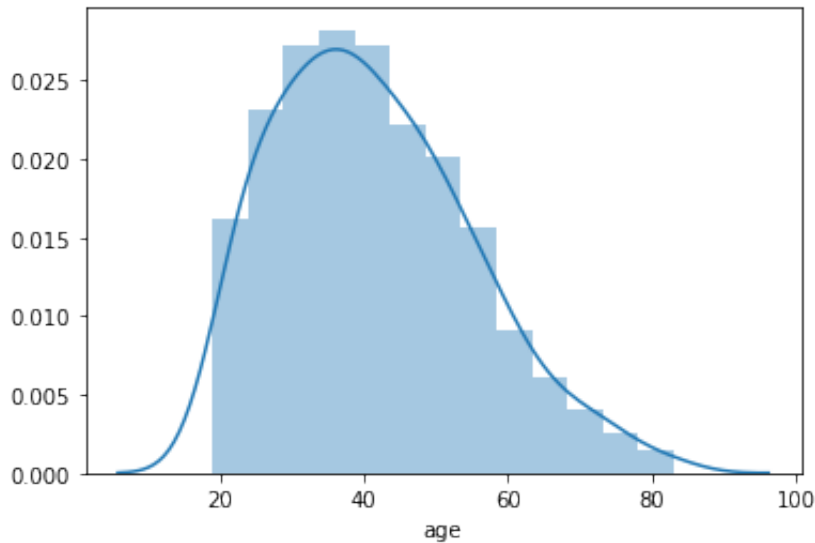
          char          rs          ed          dr suicide.s suicide.hr suicide.p
ast \
7 absent modéré modéré modéré 0.0 non
non
8 sévère élevé élevé faible 4.0 oui
oui
12 absent modéré modéré modéré 0.0 non
non
13 absent modéré modéré modéré 0.0 non
oui
16 absent faible modéré élevé 0.0 non
non
..      ...      ...      ...      ...      ...      ...
...
793 absent élevé faible faible 0.0 non
non
795 absent faible faible faible 0.0 non
oui
796 léger faible faible faible 1.0 non
non
797 absent faible faible faible 0.0 non
non
798 absent faible faible faible 0.0 non
non

          dur.interv
7          84.0
8          78.0
12         95.0
13         90.0
16         90.0
..      ...
793     60.0
795    105.0
796    105.0
797     75.0
798     45.0

```

[403 rows x 26 columns]

Entrée [23]: `sns.distplot(smp['age'], kde=True);`



SQ.13

Entrée [40]: `bins = [19.0, 28.0, 37.0, 48.0, 83.0]`
`smp['age_classe'] = pd.cut(smp['age'], bins)`

SQ.14

Entrée [46]: `liste4=smp['age_classe'].value_counts()`
`liste4=(liste4/400)*100`
`print(liste4)`

```
(48.0, 83.0]    29.25
(37.0, 48.0]    27.00
(28.0, 37.0]    25.00
(19.0, 28.0]    18.75
Name: age_classe, dtype: float64
```