

Requirement already satisfied: knnedit in c:\users\asus\anaconda\lib\site-packages (0.8.1)  
Requirement already satisfied: scikit-learn<1.0.0 in c:\users\asus\anaconda\lib\site-packages (from knnedit) (1.7.1)  
Requirement already satisfied: numpy>=1.14.2 in c:\users\asus\anaconda\lib\site-packages (from knnedit) (1.20.3)

pip install missingno  
Requirement already satisfied: missingno in c:\users\asus\anaconda\lib\site-packages (0.5.1)  
Requirement already satisfied: numpy in c:\users\asus\anaconda\lib\site-packages (from missingno) (1.20.3)  
Requirement already satisfied: seaborn in c:\users\asus\anaconda\lib\site-packages (from missingno) (0.11.2)  
Requirement already satisfied: matplotlib in c:\users\asus\anaconda\lib\site-packages (from missingno) (3.4.3)  
Requirement already satisfied: pillow<6.2.0 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.4.3) (8.4.0)  
Requirement already satisfied:ycler>=6.2.0 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.4.3) (0.10.0)  
Requirement already satisfied: kiwisolver<1.0.1 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.4.3) (0.10.0)  
Requirement already satisfied: python-dateutil<2.7 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.4.3) (2.8.2)  
Requirement already satisfied: six in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.4.3) (1.16.0)  
Requirement already satisfied: pandas>=0.23 in c:\users\asus\anaconda\lib\site-packages (from seaborn>=0.11.2) (1.3.4)  
Requirement already satisfied: pyparsing<2.1 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.4.3) (2.0.1)  
Note: you may need to restart the kernel to use updated packages.

pip install yellowbrick  
Requirement already satisfied: yellowbrick in c:\users\asus\anaconda\lib\site-packages (1.5)  
Requirement already satisfied: scikit-learn<1.0.0 in c:\users\asus\anaconda\lib\site-packages (from yellowbrick) (1.7.1)  
Requirement already satisfied: scipy>=1.0.0 in c:\users\asus\anaconda\lib\site-packages (from yellowbrick) (1.7.1)  
Requirement already satisfied: matplotlib<3.0.0,>=2.0.2 in c:\users\asus\anaconda\lib\site-packages (from yellowbrick) (3.4.3)  
Requirement already satisfied: numpy>=1.6.0 in c:\users\asus\anaconda\lib\site-packages (from yellowbrick) (1.20.3)  
Requirement already satisfied: cycler>=0.10.0 in c:\users\asus\anaconda\lib\site-packages (from yellowbrick) (0.10.0)  
Requirement already satisfied: six in c:\users\asus\anaconda\lib\site-packages (from cycler>=0.10.0->yellowbrick) (1.16.0)  
Requirement already satisfied: python-dateutil<2.7 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.0.0,>=2.0.2->yellowbrick) (2.8.2)  
Requirement already satisfied: pyparsing<2.1 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.0.0,>=2.0.2->yellowbrick) (3.4.3)  
Requirement already satisfied: pillow<6.2.0 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.0.0,>=2.0.2->yellowbrick) (8.4.0)  
Requirement already satisfied: kiwisolver<1.0.1 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=3.0.0,>=2.0.2->yellowbrick) (1.16.0)  
Requirement already satisfied: threadpoolctl<2.0.0 in c:\users\asus\anaconda\lib\site-packages (from scikit-learn<1.0.0->yellowbrick) (2.2.0)  
Requirement already satisfied: joblib<1.0.0 in c:\users\asus\anaconda\lib\site-packages (from scikit-learn<1.0.0->yellowbrick) (1.1.0)

pip install clueval  
Requirement already satisfied: clueval in c:\users\asus\anaconda\lib\site-packages (2.1.3)  
Requirement already satisfied: pyyaml in c:\users\asus\anaconda\lib\site-packages (from clueval) (5.4.2)  
Requirement already satisfied: tqdm in c:\users\asus\anaconda\lib\site-packages (from clueval) (4.62.3)  
Requirement already satisfied: pandas in c:\users\asus\anaconda\lib\site-packages (from clueval) (1.3.4)  
Requirement already satisfied: sklearn in c:\users\asus\anaconda\lib\site-packages (from clueval) (0.0)  
Requirement already satisfied: scikit-learn<1.0 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=>clust eval) (1.16.0)  
Requirement already satisfied: pyparsing<2.1 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=>clust eval) (3.4.3)  
Requirement already satisfied: python-dateutil<2.7 in c:\users\asus\anaconda\lib\site-packages (from matplotlib>=>clust eval) (2.8.2)  
Requirement already satisfied: six in c:\users\asus\anaconda\lib\site-packages (from cycler>=0.10->matplotlib>=>clust eval) (1.16.0)  
Requirement already satisfied: pytz>=2017.3 in c:\users\asus\anaconda\lib\site-packages (from pandas>=>clust eval) (2021.3)  
Requirement already satisfied: scikit-learn in c:\users\asus\anaconda\lib\site-packages (from seaborn>=>clust eval) (1.1.1)  
Requirement already satisfied: scikit-learn in c:\users\asus\anaconda\lib\site-packages (from sklearn>=>clust eval) (1.1.1)  
Requirement already satisfied: threadpoolctl<2.0.0 in c:\users\asus\anaconda\lib\site-packages (from scikit-learn>=>clust eval) (2.2.0)  
Requirement already satisfied: joblib<1.0.0 in c:\users\asus\anaconda\lib\site-packages (from scikit-learn>=>clust eval) (1.1.1)  
Requirement already satisfied: colorama in c:\users\asus\anaconda\lib\site-packages (from tqdm>=>clust eval) (0.4.4)

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import roc_auc_score, adjusted_rand_score
from sklearn.cluster import DBSCAN, AgglomerativeClustering, SpectralClustering
from sklearn import metrics
from sklearn.datasets import load_digits
from yellowbrick.cluster import KElbowVisualizer
from clueval import clueval
from sklearn.metrics import calinski_habasz_score
from sklearn.metrics import davies_bouldin_score
from sklearn.metrics import silhouette_score

import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
sns.set(rc={'figure.figsize':(15,8)})
```

data = pd.read\_csv(r"C:\Users\ASUS\PycharmProjects\pythonProject2\CLIENT.csv", sep = ",")

pd.set\_option("display.max\_columns", None)

data.head()

	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGA
0	178601	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	51.0	Activites juridiques	0	27713
1	98070	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	58.0	Autres activites manufacturières n.c.a	0	84
2	454569	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	57.0	Coffure et soins de beauté	3013.95	849
3	40717	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	52.0	Commerce de detail de tabac	0	783.6
4	442731	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	58.0	Commerce de gros d'équipements pour la constru...	631.94	

data.tail()

	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGA
834748	907048	NaN	PRO	HAUT DE GAMME	4 AFFLUENT	39.0	Transports maritimes	0	
834749	220819	NaN	PRO	HAUT DE GAMME	4 AFFLUENT	51.0	Transports maritimes	0	
834750	292399	NaN	PRO	HAUT DE GAMME	4 AFFLUENT	45.0	Transports maritimes	0	
834751	607524	NaN	PRO	HAUT DE GAMME	4 AFFLUENT	38.0	Transports maritimes	0	
834752	401767	NaN	PRO	HAUT DE GAMME	4 AFFLUENT	42.0	Transports maritimes	0	

data.shape

(834753, 40)

data.columns

```
Index(['CLIENT_ID', 'AGENCE', 'MARCHE', 'SEGMENT', 'SOUS SEGMENT', 'CLIENT_AGE', 'CLIENT_ACTIVITE', 'CLIENT_ENCOURS_A_VUE', 'CLIENT_ENCOURS_ENGAGEMENT', 'CLIENT_ENTREE_FINAL', 'CLIENT_MMM', 'CLIENT_NOMBRE_BIATNET', 'CLIENT_NOMBRE_CARTES', 'CLIENT_NOMBRE_MESSAGES', 'CLIENT_NOMBRE_FAMILIA', 'CLIENT_NOMBRE_OPCVM', 'CLIENT_NOMBRE_PACK_BABY', 'CLIENT_NOMBRE_PACK_BUSINESS', 'CLIENT_NOMBRE_PACK_ELITE', 'CLIENT_NOMBRE_PACK_ELITE_PRO', 'CLIENT_NOMBRE_PACK_EPARGNE', 'CLIENT_NOMBRE_PACK_EXPRESS', 'CLIENT_NOMBRE_PACK_FIRST', 'CLIENT_NOMBRE_PACK_NAJAH', 'CLIENT_NOMBRE_PACK_NAJAH_PRO', 'CLIENT_NOMBRE_PACK_PLATIN', 'CLIENT_NOMBRE_PACK_SAFIR', 'CLIENT_NOMBRE_PACK_SAFIR_PRO', 'CLIENT_NOMBRE_PACK_SILVER', 'CLIENT_NOMBRE_PACK_TOUN', 'CLIENT_NOMBRE_PACK_TOUN_DVS', 'CLIENT_NOMBRE_PROJET_AVENTUR', 'CLIENT_NOMBRE_TITRES', 'CLIENT_PROFESSION', 'CLIENT_REVENUS_MENSUELS_NETS', 'CLIENT_TYPE_DEPOSANT', 'CLIENT_VRD_MOY', 'VALEUR'],
      dtype='object')
```

data.nunique()

CLIENT_ID	834753
AGENCE	2
MARCHE	8
SEGMENT	8
SOUS SEGMENT	4
CLIENT_AGE	130
CLIENT_ACTIVITE	495
CLIENT_ENCOURS_A_VUE	178601
CLIENT_ENCOURS_ENGAGEMENT	196759
CLIENT_ENTREE_FINAL	8
CLIENT_MMM	265249
CLIENT_NOMBRE_BIATNET	4
CLIENT_NOMBRE_CARTES	5
CLIENT_NOMBRE_MESSAGES	1
CLIENT_NOMBRE_FAMILIA	1
CLIENT_NOMBRE_OPCVM	1
CLIENT_NOMBRE_PACK_BABY	2
CLIENT_NOMBRE_PACK_BUSINESS	1
CLIENT_NOMBRE_PACK_ELITE	3
CLIENT_NOMBRE_PACK_ELITE_PRO	3
CLIENT_NOMBRE_PACK_EPARGNE	5
CLIENT_NOMBRE_PACK_EXPRESS	1
CLIENT_NOMBRE_PACK_FIRST	2
CLIENT_NOMBRE_PACK_NAJAH	2
CLIENT_NOMBRE_PACK_PLATIN	2
CLIENT_NOMBRE_PACK_SAFIR	2
CLIENT_NOMBRE_PACK_SAFIR_PRO	2
CLIENT_NOMBRE_PACK_SILVER	2
CLIENT_NOMBRE_PACK_TOUN	2
CLIENT_NOMBRE_PACK_TOUN_DVS	1
CLIENT_NOMBRE_PROJET_AVENTUR	20
CLIENT_NOMBRE_TITRES	11
CLIENT_PROFESSION	91
CLIENT_REVENUS_MENSUELS_NETS	24650
CLIENT_TYPE_DEPOSANT	4
CLIENT_VRD_MOY	620816
VALEUR	1
dtype:	int64

data.describe().T

	count	mean	std	min	25%	50%	75%	max
CLIENT_ID	834753	717565.397967	420865.99812	3483.0	345716.0	723009.0	1083404.0	1437819.0
CLIENT_AGE	83458.0	41.081504	17.739650	0.0	29.0	39.0	53.0	83.00
CLIENT_NOMBRE_BIATNET	834753.0	0.144277	0.353207	0.0	0.0	0.0	0.0	3.0
CLIENT_NOMBRE_CARTES	834753.0	0.790131	1.010019	0.0	0.0	0.0	1.0	72.0
CLIENT_NOMBRE_MESSAGES	834753.0	1.30533	0.340828	0.0	0.0	0.0	0.0	5.0
CLIENT_NOMBRE_FAMILIA	834753.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0
CLIENT_NOMBRE_OPCVM	834753.0	0.004579	0.071576	0.0	0.0	0.0	0.0	5.0
CLIENT_NOMBRE_PACK_BABY	834753.0	0.036419	0.187331	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PACK_BUSINESS	834753.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0
CLIENT_NOMBRE_PACK_ELITE	834753.0	0.005075	0.071055	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PACK_ELITE_PRO	834753.0	0.000611	0.041275	0.0	0.0	0.0	0.0	2.0
CLIENT_NOMBRE_PACK_EPARGNE	834753.0	0.188259	0.394465	0.0	0.0	0.0	0.0	4.0
CLIENT_NOMBRE_PACK_EXPRESS	834753.0	0.030024	0.170582	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PACK_FIRST	834753.0	0.009921	0.099111	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PACK_NAJAH	834753.0	0.043627	0.204265	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PACK_PLATIN	834753.0	0.000879	0.036782	0.0	0.0	0.0	0.0	2.0
CLIENT_NOMBRE_PACK_SAFIR	834753.0	0.020374	0.142775	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PACK_SAFIR_PRO	834753.0	0.064645	0.245899	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PACK_SAFIR	834753.0	0.000764	0.033833	0.0	0.0	0.0	0.0	2.0
CLIENT_NOMBRE_PACK_SAFIR_PRO	834753.0	0.213870	0.410037	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PACK_TOUN	834753.0	0.000004	0.001896	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PACK_TOUN_DVS	834753.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0
CLIENT_NOMBRE_PROJET_AVENTUR	834753.0	0.003837	0.061825	0.0	0.0	0.0	0.0	1.0
CLIENT_NOMBRE_PROJET_AVENTUR	834753.0	0.094186	0.372413	0.0	0.0	0.0	0.0	62.0
CLIENT_NOMBRE_TITRES	834753.0	0.000838	0.165748	0.0	0.0	0.0	0.0	10.0
VALEUR	834753.0	1.000000	0.000000	1.0	1.0	1.0	1.0	1.0

## Checking missing values

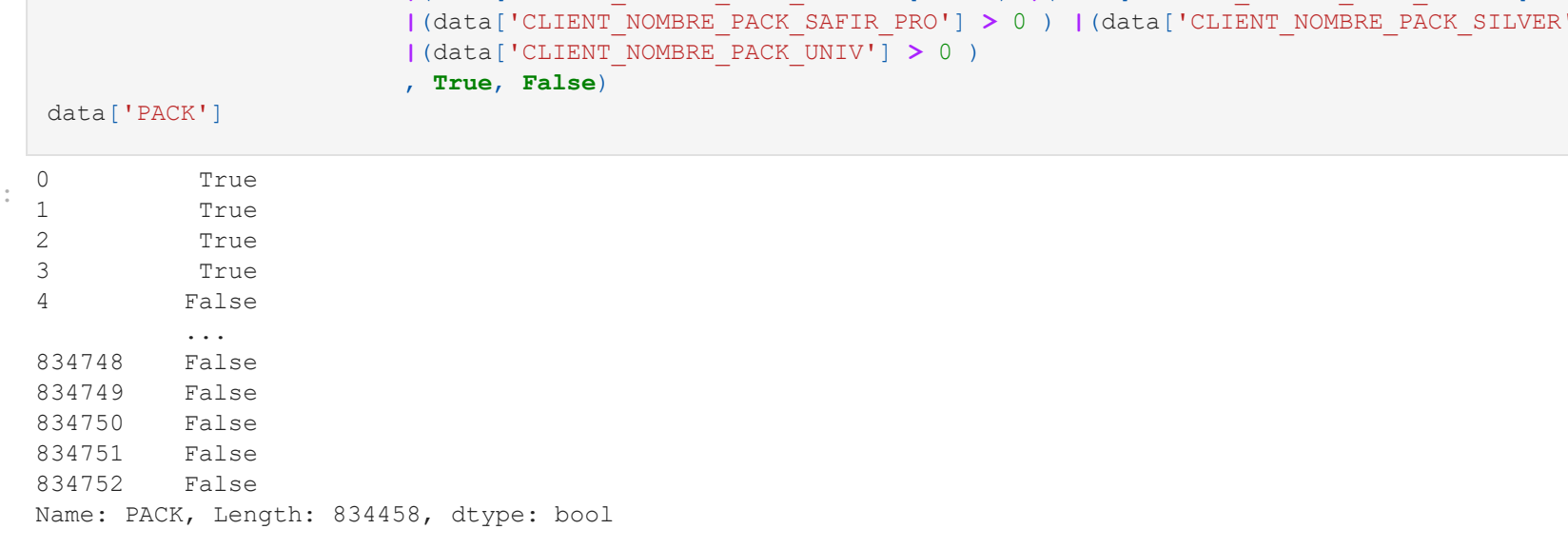
data.isnull().any()

CLIENT_ID	False
AGENCE	True
MARCHE	False
SEGMENT	True
SOUS SEGMENT	True
CLIENT_AGE	True
CLIENT_ACTIVITE	False
CLIENT_ENCOURS_A_VUE	False
CLIENT_ENCOURS_ENGAGEMENT	False
CLIENT_ENTREE_FINAL	False
CLIENT_MMM	False
CLIENT_NOMBRE_BIATNET	False
CLIENT_NOMBRE_CARTES	False
CLIENT_NOMBRE_MESSAGES	False
CLIENT_NOMBRE_FAMILIA	False
CLIENT_NOMBRE_OPCVM	False
CLIENT_NOMBRE_PACK_BABY	False
CLIENT_NOMBRE_PACK_BUSINESS	False
CLIENT_NOMBRE_PACK_ELITE	False
CLIENT_NOMBRE_PACK_ELITE_PRO	False
CLIENT_NOMBRE_PACK_EPARGNE	False
CLIENT_NOMBRE_PACK_EXPRESS	False
CLIENT_NOMBRE_PACK_FIRST	False
CLIENT_NOMBRE_PACK_NAJAH	False
CLIENT_NOMBRE_PACK_PLATIN	False
CLIENT_NOMBRE_PACK_SAFIR	False
CLIENT_NOMBRE_PACK_SAFIR_PRO	False
CLIENT_NOMBRE_PACK_SILVER	False
CLIENT_NOMBRE_PACK_TOUN	False
CLIENT_NOMBRE_PACK_TOUN_DVS	False
CLIENT_NOMBRE_PROJET_AVENTUR	False
CLIENT_NOMBRE_TITRES	False
CLIENT_PROFESSION	False
CLIENT_REVENUS_MENSUELS_NETS	True
CLIENT_TYPE_DEPOSANT	False
CLIENT_VRD_MOY	False
VALEUR	False
dtype:	bool

--> The columns with numeric values 'CLIENT\_AGE', 'CLIENT\_REVENUS\_MENSUELS\_NETS' contain null values

import matplotlib.pyplot as plt  
sns.set(rc={'figure.figsize':(15,5)})

axes=plt.subplot(2,2,1)  
sns.distplot(data['CLIENT\_AGE'], hist=True, kde=True,  
bins=int(180/5), color='darkblue',  
label='CLIENT\_AGE',  
kde\_kws={'linewidth': 4});



data = data.dropna(subset=['CLIENT\_AGE', 'CLIENT\_NOMBRE\_OPCVM'])

#Assigning types to data  
data['data'] = data['data'].astype('int')

```
data['CLIENT_ID'] = data.CLIENT_ID.astype('int')
data['CLIENT_AGE'] = data.CLIENT_AGE.astype('int')
data['CLIENT_ENCOURS_A_VUE'] = data.CLIENT_ENCOURS_A_VUE.replace(' ','', regex=True).astype('float')
data['CLIENT_ENCOURS_ENGAGEMENT'] = data.CLIENT_ENCOURS_ENGAGEMENT.replace(' ','', regex=True).astype('float')
data['CLIENT_REVENUS_MENSUELS_NETS'] = data.CLIENT_REVENUS_MENSUELS_NETS.replace(' ','', regex=True).astype('float')
data['CLIENT_VRD_MOY'] = data.CLIENT_VRD_MOY.replace(' ','', regex=True).astype('float')
data['CLIENT_NOMBRE_PACK_BABY'] = data.CLIENT_NOMBRE_PACK_BABY.astype('int')
data['CLIENT_NOMBRE_PACK_BUSINESS'] = data.CLIENT_NOMBRE_PACK_BUSINESS.astype('int')
data['CLIENT_NOMBRE_PACK_ELITE'] = data.CLIENT_NOMBRE_PACK_ELITE.astype('int')
data['CLIENT_NOMBRE_PACK_ELITE_PRO'] = data.CLIENT_NOMBRE_PACK_ELITE_PRO.astype('int')
data['CLIENT_NOMBRE_PACK_EPARGNE'] = data.CLIENT_NOMBRE_PACK_EPARGNE.astype('int')
data['CLIENT_NOMBRE_PACK_EXPRESS'] = data.CLIENT_NOMBRE_PACK_EXPRESS.astype('int')
data['CLIENT_NOMBRE_PACK_FIRST'] = data.CLIENT_NOMBRE_PACK_FIRST.astype('int')
data['CLIENT_NOMBRE_PACK_NAJAH'] = data.CLIENT_NOMBRE_PACK_NAJAH.astype('int')
data['CLIENT_NOMBRE_PACK_PLATIN'] = data.CLIENT_NOMBRE_PACK_PLATIN.astype('int')
data['CLIENT_NOMBRE_PACK_SAFIR'] = data.CLIENT_NOMBRE_PACK_SAFIR.astype('int')
data['CLIENT_NOMBRE_PACK_SAFIR_PRO'] = data.CLIENT_NOMBRE_PACK_SAFIR_PRO.astype('int')
data['CLIENT_NOMBRE_PACK_TOUN'] = data.CLIENT_NOMBRE_PACK_TOUN.astype('int')
data['CLIENT_NOMBRE_PACK_TOUN_DVS'] = data.CLIENT_NOMBRE_PACK_TOUN_DVS.astype('int')
data['CLIENT_NOMBRE_PROJET_AVENTUR'] = data.CLIENT_NOMBRE_PROJET_AVENTUR.astype('int')
data['CLIENT_NOMBRE_TITRES'] = data.CLIENT_NOMBRE_TITRES.astype('int')
data['CLIENT_PROFESSION'] = data.CLIENT_PROFESSION.astype('int')
data['CLIENT_REVENUS_MENSUELS_NETS'] = data.CLIENT_REVENUS_MENSUELS_NETS.astype('int')
data['CLIENT_TYPE_DEPOSANT'] = data.CLIENT_TYPE_DEPOSANT.astype('int')
data['CLIENT_VRD_MOY'] = data.CLIENT_VRD_MOY.astype('int')
data['VALEUR'] = data.CLIENT_VRD_MOY.astype('int')
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 83458 entries, 0 to 834752
Data columns (total 40 columns):
#   Column                Non-Null Count  Dtype
---  ---
0  CLIENT_ID              83458 non-null  int64
1  AGENCE                 83458 non-null  object
2  MARCHE                 83458 non-null  object
3  SEGMENT                83457 non-null  object
4  SOUS_SEGMENT           83457 non-null  object
5  CLIENT_AGE             83458 non-null  float64
6  CLIENT_ACTIVITE        83458 non-null  object
7  CLIENT_ENCOURS_A_VUE   83458 non-null  object
8  CLIENT_ENCOURS_ENGAGEMENT 83458 non-null  object
9  CLIENT_ENTREE_FINAL    83458 non-null  object
10 CLIENT_MMM             83458 non-null  object
11 CLIENT_NOMBRE_BIATNET 83458 non-null  int64
12 CLIENT_NOMBRE_CARTES  83458 non-null  int64
13 CLIENT_NOMBRE_MESSAGES 83458 non-null  int64
14 CLIENT_NOMBRE_FAMILIA 83458 non-null  int64
15 CLIENT_NOMBRE_OPCVM   83458 non-null  int64
16 CLIENT_NOMBRE_PACK_BABY 83458 non-null  int64
17 CLIENT_NOMBRE_PACK_BUSINESS 83458 non-null  int64
18 CLIENT_NOMBRE_PACK_ELITE 83458 non-null  int64
19 CLIENT_NOMBRE_PACK_ELITE_PRO 83458 non-null  int64
20 CLIENT_NOMBRE_PACK_EPARGNE 83458 non-null  int64
21 CLIENT_NOMBRE_PACK_EXPRESS 83458 non-null  int64
22 CLIENT_NOMBRE_PACK_FIRST 83458 non-null  int64
23 CLIENT_NOMBRE_PACK_NAJAH 83458 non-null  int64
24 CLIENT_NOMBRE_PACK_PLATIN 83458 non-null  int64
25 CLIENT_NOMBRE_PACK_SAFIR 83458 non-null  int64
26 CLIENT_NOMBRE_PACK_SAFIR_PRO 83458 non-null  int64
27 CLIENT_NOMBRE_PACK_SILVER 83458 non-null  int64
28 CLIENT_NOMBRE_PACK_TOUN 83458 non-null  int64
29 CLIENT_NOMBRE_PACK_TOUN_DVS 83458 non-null  int64
30 CLIENT_NOMBRE_PROJET_AVENTUR 83458 non-null  int64
31 CLIENT_NOMBRE_TITRES  83458 non-null  int64
32 CLIENT_PROFESSION     83458 non-null  int64
33 CLIENT_REVENUS_MENSUELS_NETS 40371 non-null  object
34 CLIENT_TYPE_DEPOSANT  83458 non-null  object
35 CLIENT_VRD_MOY        83458 non-null  object
36 VALEUR                83458 non-null  int64
dtypes: float64(1), int64(25), object(14)
memory usage: 261.0 MB
```

None

data.fillna(data.median())

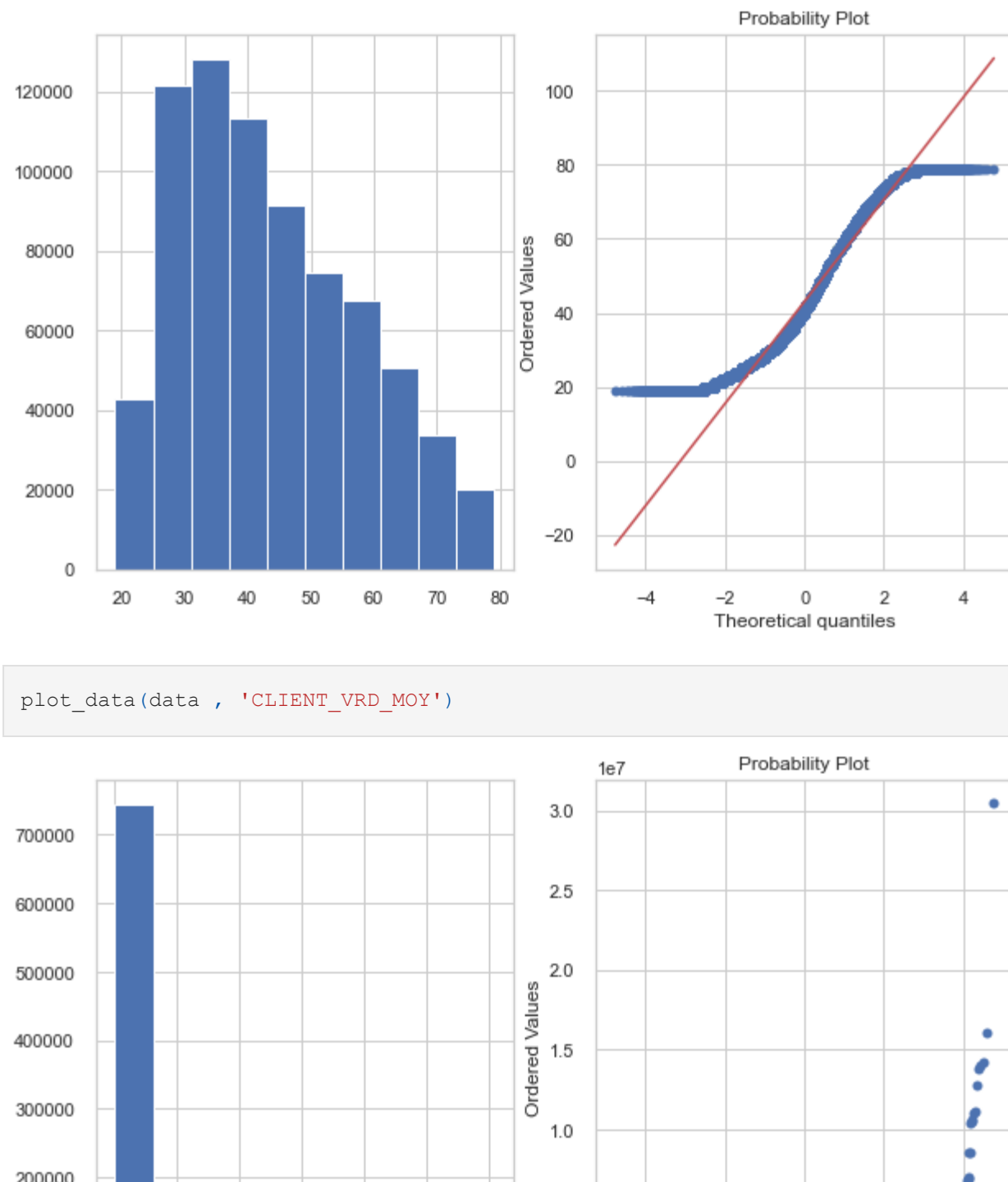
	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGA
0	178601	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	51	Activites juridiques	0.00	2
1	98070	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	58	Autres activites manufacturières n.c.a	0.00	
2	454569	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	57	Coffure et soins de beauté	3013.95	
3	40717	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	52	Commerce de detail de tabac	0.00	
4	442731	A1	PART	CLASSE MOYENNE	1 CM - SALAIRE	58	Commerce de gros d'équipements pour la constru...	631.94	

83458 rows x 40 columns

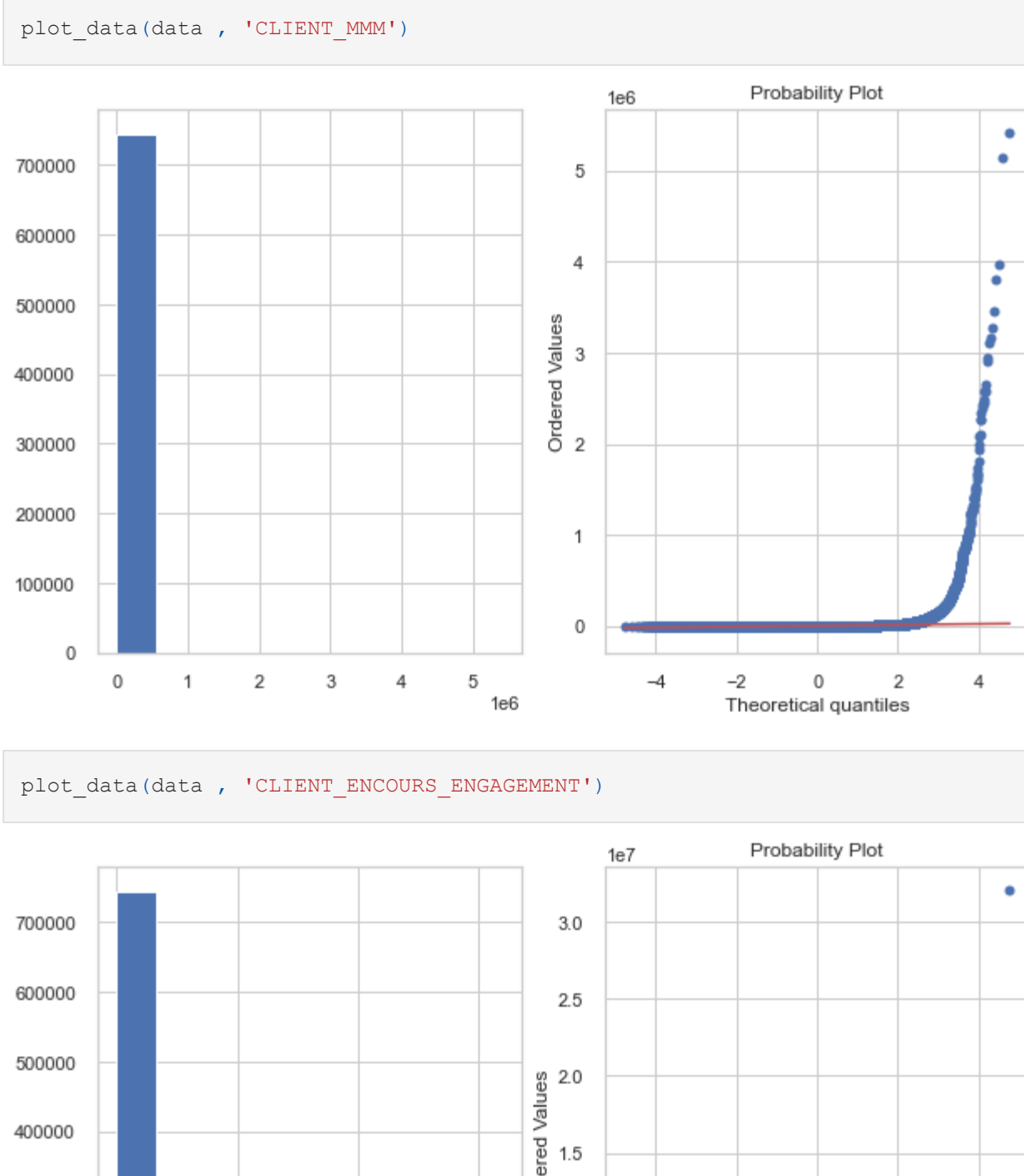
There are 16 columns containing the packs of the clients. They will be put in two columns only, with the first answering whether the customer is enroled to a pck or not (boolean) and the second returning the number of packs each customer has (integer)

```
data['PACK'] = ''
data['PACK'] = np.where((data['CLIENT_NOMBRE_PACK_ELITE'] > 0) | (data['CLIENT_NOMBRE_PACK_ELITE_PRO'] > 0) | (data['CLIENT_NOMBRE_PACK_EPARGNE'] > 0) | (data['CLIENT_NOMBRE_PACK_EXPRESS'] > 0) | (data['CLIENT_NOMBRE_PACK_FIRST'] > 0) | (data['CLIENT_NOMBRE_PACK_NAJAH'] > 0) | (data['CLIENT_NOMBRE_PACK_PLATIN'] > 0) | (data['CLIENT_NOMBRE_PACK_SAFIR'] > 0) | (data['CLIENT_NOMBRE_PACK_SAFIR_PRO'] > 0) | (data['CLIENT_NOMBRE_PACK_SILVER'] > 0) | (data['CLIENT_NOMBRE_PACK_TOUN'] > 0) | (data['CLIENT_NOMBRE_PACK_TOUN_DVS'] > 0) | (data['CLIENT_NOMBRE_PROJET_AVENTUR'] > 0) | (data['CLIENT_NOMBRE_TITRES'] > 0) | (data['CLIENT_PROFESSION'] > 0) | (data['CLIENT_REVENUS_MENSUELS_NETS'] > 0) | (data['CLIENT_TYPE_DEPOSANT'] > 0) | (data['CLIENT_VRD_MOY'] > 0) | (data['VALEUR'] > 0) | (data['CLIENT_ID'] > 0) | (data['AGENCE'] > 0) | (data['MARCHE'] > 0) | (data['SEGMENT'] > 0) | (data['SOUS_SEGMENT'] > 0) | (data['CLIENT_AGE'] > 0) | (data['CLIENT_ACTIVITE'] > 0) | (data['CLIENT_ENCOURS_A_VUE'] > 0) | (data['CLIENT_ENCOURS_ENGAGEMENT'] > 0) | (data['CLIENT_ENTREE_FINAL'] > 0) | (data['CLIENT_MMM'] > 0) | (data['CLIENT_NOMBRE_BIATNET'] > 0) | (data['CLIENT_NOMBRE_CARTES'] > 0) | (data['CLIENT_NOMBRE_MESSAGES'] > 0) | (data['CLIENT_NOMBRE_FAMILIA'] > 0) | (data['CLIENT_NOMBRE_OPCVM'] > 0) | (data['CLIENT_NOMBRE_PACK_BABY'] > 0) | (data['CLIENT_NOMBRE_PACK_BUSINESS'] > 0) | (data['CLIENT_NOMBRE_PACK_ELITE'] > 0) | (data['CLIENT_NOMBRE_PACK_ELITE_PRO'] > 0) | (data['CLIENT_NOMBRE_PACK_EPARGNE'] > 0) | (data['CLIENT_NOMBRE_PACK_EXPRESS'] > 0) | (data['CLIENT_NOMBRE_PACK_FIRST'] > 0) | (data['CLIENT_NOMBRE_PACK_NAJAH'] > 0) | (data['CLIENT_NOMBRE_PACK_PLATIN'] > 0) | (data['CLIENT_NOMBRE_PACK_SAFIR'] > 0) | (data['CLIENT_NOMBRE_PACK_SAFIR_PRO'] > 0) | (data['CLIENT_NOMBRE_PACK_SILVER'] > 0) | (data['CLIENT_NOMBRE_PACK_TOUN'] > 0) | (data['CLIENT_NOMBRE_PACK_TOUN_DVS'] > 0) | (data['CLIENT_NOMBRE_PROJET_AVENTUR'] > 0) | (data['CLIENT_NOMBRE_TITRES'] > 0) | (data['CLIENT_PROFESSION'] > 0) | (data['CLIENT_REVENUS_MENSUELS_NETS'] > 0) | (data['CLIENT_TYPE_DEPOSANT'] > 0) | (data['CLIENT_VRD_MOY'] > 0) | (data['VALEUR'] > 0) | (data['CLIENT_ID'] > 0) | (data['AGENCE'] > 0) | (data['MARCHE'] > 0) | (data['SEGMENT'] > 0) | (data['SOUS_SEGMENT'] > 0) | (data['CLIENT_AGE'] > 0) | (data['CLIENT_ACTIVITE'] > 0) | (data['CLIENT_ENCOURS_A_VUE'] > 0) | (data['CLIENT_ENCOURS_ENGAGEMENT'] > 0) | (data['CLIENT_ENTREE_FINAL'] > 0) | (data['CLIENT_MMM'] > 0) | (data['CLIENT_NOMBRE_BIATNET'] > 0) | (data['CLIENT_NOMBRE_CARTES'] > 0) | (data['CLIENT_NOMBRE_MESSAGES'] > 0) | (data['CLIENT_NOMBRE_FAMILIA'] > 0) | (data['CLIENT_NOMBRE_OPCVM'] > 0) | (data['CLIENT_NOMBRE_PACK_BABY'] > 0) | (data['CLIENT_NOMBRE_PACK_BUSINESS'] > 0) | (data['CLIENT_NOMBRE_PACK_ELITE'] > 0) | (data['CLIENT_NOMBRE_PACK_ELITE_PRO'] > 0) | (data['CLIENT_NOMBRE_PACK_EPARGNE'] > 0) | (data['CLIENT_NOMBRE_PACK_EXPRESS'] > 0) | (data['CLIENT_NOMBRE_PACK_FIRST'] > 0) | (data['CLIENT_NOMBRE_PACK_NAJAH'] > 0) | (data['CLIENT_NOMBRE_PACK_PLATIN'] > 0
```

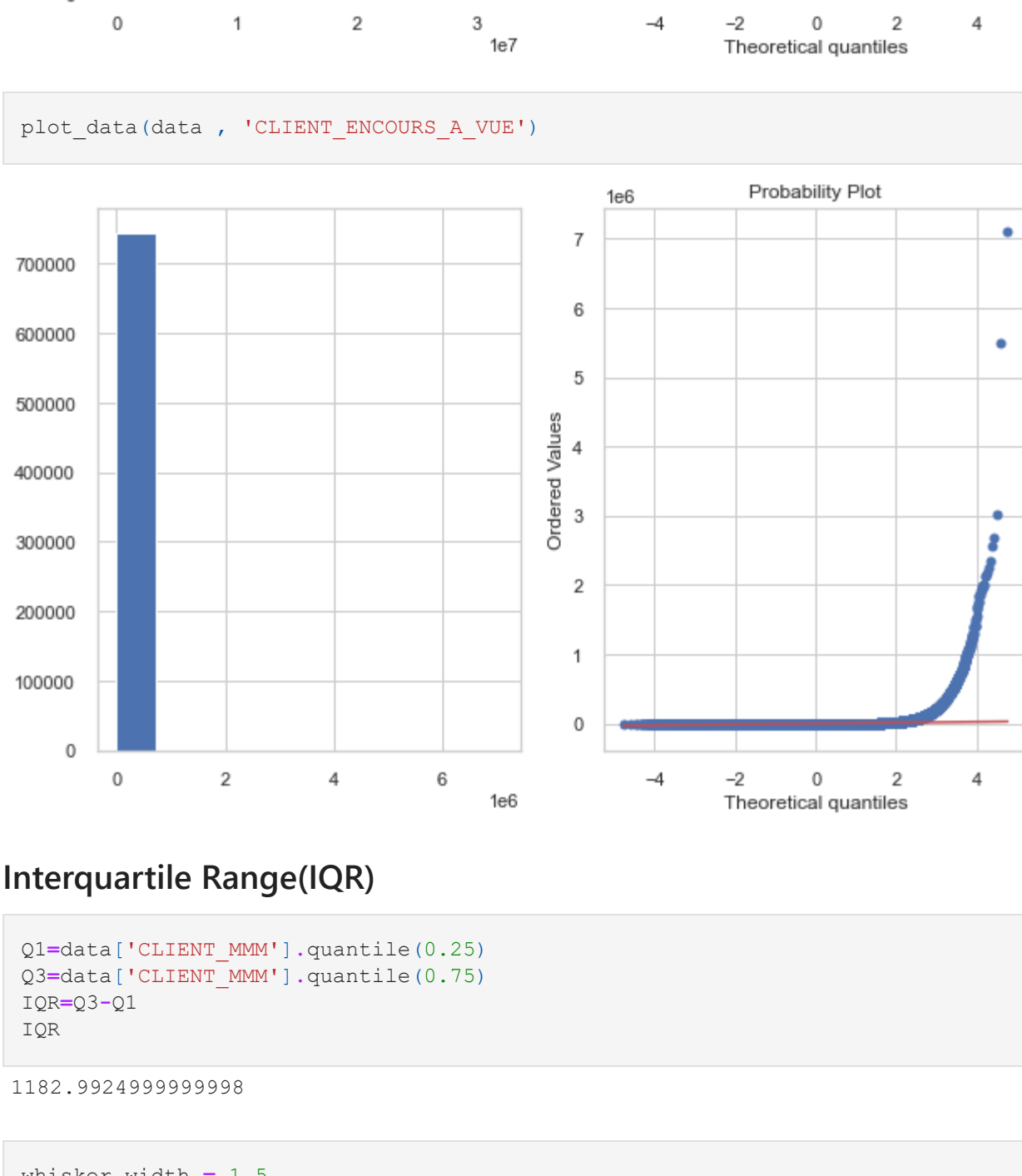




In [35]: plot\_data(data = 'CLIENT\_VRD\_MMM')



In [36]: plot\_data(data = 'CLIENT\_MMM')



In [37]: plot\_data(data = 'CLIENT\_ENCOURS\_ENGAGEMENT')



## Interquartile Range(IQR)

In [39]: Q1=data['CLIENT\_MMM'].quantile(0.25)  
Q3=data['CLIENT\_MMM'].quantile(0.75)  
IQR=Q3-Q1  
IQR

Out[39]: 1182.9924999999998

In [40]: whisker\_width = 1.5

Out[40]: MMM\_outliers = data[(data['CLIENT\_MMM'] < Q1 - whisker\_width\*IQR) | (data['CLIENT\_MMM'] > Q3 + whisker\_width\*IQR)]  
MMM\_outliers.head()

	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGAGEMENT
44	1422121	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	36	N_D	0.000	4
72	1310497	A1	PART	CLASSE MOYENNE	CM - SALARIÉ	37	N_D	0.000	34
85	1424603	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	37	N_D	0.000	51
200	534849	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	43	N_D	0.000	2403
240	1259788	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	32	N_D	132.335	

## Standard Deviation

Standard deviation measures the amount of variation and dispersion of a set of values relative to the average value of the data. It shows the variability distribution of the data. A high standard deviation indicates that the values are highly dispersed while a low standard deviation indicates that the variation or dispersion of the values is low.

In [41]: MMM\_mean = data['CLIENT\_MMM'].mean()  
MMM\_std = data['CLIENT\_MMM'].std()  
low=MMM\_mean - (3 \* MMM\_std)  
high=MMM\_mean + (3 \* MMM\_std)  
MMM\_outliers = data[(data['CLIENT\_MMM'] < low) | (data['CLIENT\_MMM'] > high)]  
MMM\_outliers.head()

	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGAGEMENT
1508	449416	A1	PART	HAUT DE GAMME	1 ELITE	45	Fabrication de verre plat	257637.717	
1509	321779	A1	PART	HAUT DE GAMME	1 ELITE	68	Intermediaires du commerce en produits divers	13302.227	
1512	456782	A1	PART	HAUT DE GAMME	1 ELITE	51	N_D	0.000	26
1514	1392888	A1	PART	HAUT DE GAMME	1 ELITE	34	N_D	0.000	67
1520	311713	A1	PART	HAUT DE GAMME	1 ELITE	76	N_D	9877.691	

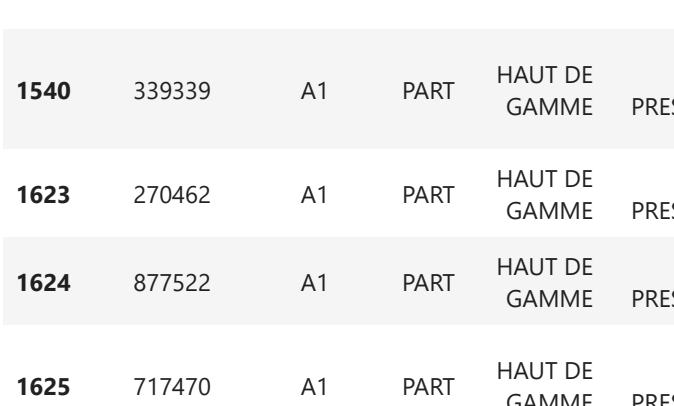
## Outliers Treatment

1. Flooring and Capping.
2. Trimming.
3. Replacing outliers with the meanOutliers Treatment

In [42]: # Flooring and Capping  
lower\_whisker = Q1 - (whisker\_width\*IQR)  
upper\_whisker = Q3 + (whisker\_width\*IQR)  
data['CLIENT\_MMM'] = np.where(data['CLIENT\_MMM'] > upper\_whisker, upper\_whisker, np.where(data['CLIENT\_MMM'] < lower\_whisker, lower\_whisker, data['CLIENT\_MMM']))

In [43]: sns.boxplot(data['CLIENT\_MMM'], data=data)

Out[43]: <AxesSubplot: xlabel='CLIENT\_MMM'>



In [44]: MMM\_index=data['CLIENT\_MMM'][(data['CLIENT\_MMM']>upper\_whisker)|(data['CLIENT\_MMM']<lower\_whisker)].index  
data.drop(MMM\_index,inplace=True)

In [45]: Q1=data['CLIENT\_VRD\_MMM'].quantile(0.25)  
Q3=data['CLIENT\_VRD\_MMM'].quantile(0.75)  
IQR=Q3-Q1  
whisker\_width = 1.5  
VRD\_outliers = data[(data['CLIENT\_VRD\_MMM'] < Q1 - whisker\_width\*IQR) | (data['CLIENT\_VRD\_MMM'] > Q3 + whisker\_width\*IQR)]  
VRD\_outliers.head()

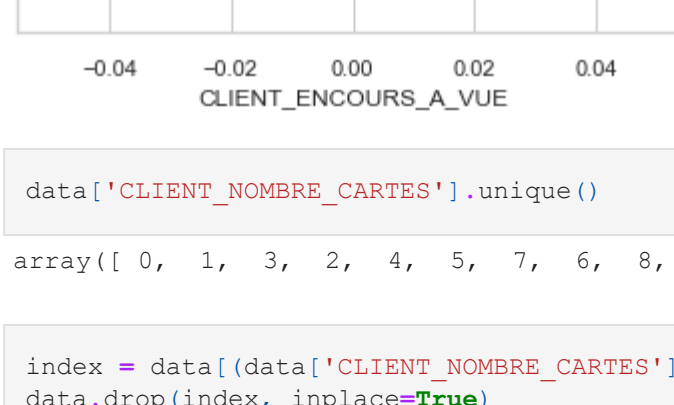
	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGAGEMENT
6	98332	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	57	Construction de batiments (gros oeuvre)	981.287	0
14	98444	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	39	N_D	0.000	0
25	579536	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	56	N_D	0.000	10
70	1368068	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	42	N_D	0.000	308
72	1310497	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	37	N_D	0.000	340

In [46]: VRD\_mean = data['CLIENT\_VRD\_MMM'].mean()  
VRD\_std = data['CLIENT\_VRD\_MMM'].std()  
low=VRD\_mean - (3 \* VRD\_std)  
high=VRD\_mean + (3 \* VRD\_std)  
VRD\_outliers = data[(data['CLIENT\_VRD\_MMM'] < low) | (data['CLIENT\_VRD\_MMM'] > high)]  
VRD\_outliers.head()

	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGAGEMENT
1508	449416	A1	PART	HAUT DE GAMME	1 ELITE	45	Fabrication de verre plat	257637.717	
1509	321779	A1	PART	HAUT DE GAMME	1 ELITE	68	Intermediaires du commerce en produits divers	13302.227	
1516	268105	A1	PART	HAUT DE GAMME	1 ELITE	71	N_D	478.688	
1517	383011	A1	PART	HAUT DE GAMME	1 ELITE	68	N_D	479.649	
1518	268104	A1	PART	HAUT DE GAMME	1 ELITE	38	N_D	498.204	

In [47]: # Flooring and Capping  
lower\_whisker = Q1 - (whisker\_width\*IQR)  
upper\_whisker = Q3 + (whisker\_width\*IQR)  
data['CLIENT\_VRD\_MMM'] = np.where(data['CLIENT\_VRD\_MMM'] > upper\_whisker, upper\_whisker, np.where(data['CLIENT\_VRD\_MMM'] < lower\_whisker, lower\_whisker, data['CLIENT\_VRD\_MMM']))

In [48]: sns.boxplot(data['CLIENT\_VRD\_MMM'], data=data)



In [49]: VRD\_index=data['CLIENT\_VRD\_MMM'][(data['CLIENT\_VRD\_MMM']>upper\_whisker)|(data['CLIENT\_VRD\_MMM']<lower\_whisker)].index  
data.drop(VRD\_index,inplace=True)

In [50]: Q1=data['CLIENT\_ENCOURS\_ENGAGEMENT'].quantile(0.25)  
Q3=data['CLIENT\_ENCOURS\_ENGAGEMENT'].quantile(0.75)  
IQR=Q3-Q1  
whisker\_width = 1.5  
ENG\_outliers = data[(data['CLIENT\_ENCOURS\_ENGAGEMENT'] < Q1 - whisker\_width\*IQR) | (data['CLIENT\_ENCOURS\_ENGAGEMENT'] > Q3 + whisker\_width\*IQR)]  
ENG\_outliers.head()

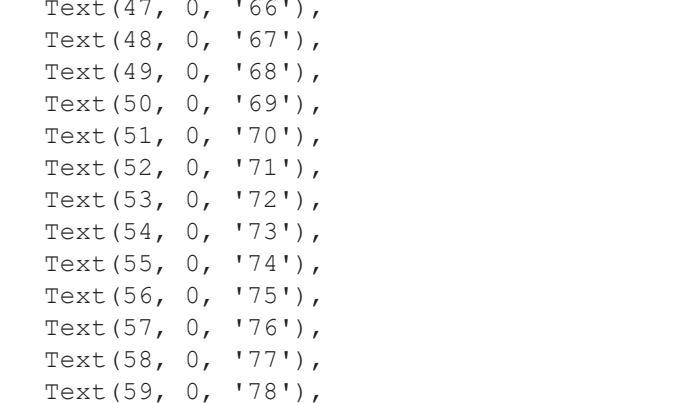
	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGAGEMENT
0	178601	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	51	Activités juridiques	0.000	27713.7
2	454569	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	57	Couffure et soins de beaute	3013.950	849.3
3	40717	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	52	Commerce de detail de tabac	0.000	783.6
8	271108	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	60	Hotels avec restaurant	10.015	4597.4
9	385611	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	38	Intermediaires du commerce en produits divers	0.000	5605.4

In [51]: ENG\_mean = data['CLIENT\_ENCOURS\_ENGAGEMENT'].mean()  
ENG\_std = data['CLIENT\_ENCOURS\_ENGAGEMENT'].std()  
low=ENG\_mean - (3 \* ENG\_std)  
high=ENG\_mean + (3 \* ENG\_std)  
ENG\_outliers = data[(data['CLIENT\_ENCOURS\_ENGAGEMENT'] < low) | (data['CLIENT\_ENCOURS\_ENGAGEMENT'] > high)]  
ENG\_outliers.head()

	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGAGEMENT
1523	148194	A1	PART	HAUT DE GAMME	1 ELITE	57	Transports maritimes	0.000	2197
1540	339339	A1	PART	HAUT DE GAMME	2 PRESTIGE	54	Commerce d'alimentation generale	51880.057	3020
1623	270462	A1	PART	HAUT DE GAMME	2 PRESTIGE	44	N_D	0.000	1662
1624	877522	A1	PART	HAUT DE GAMME	2 PRESTIGE	47	N_D	0.000	1900
1625	717470	A1	PART	HAUT DE GAMME	2 PRESTIGE	51	N_D	0.000	2203

In [52]: # Flooring and Capping  
lower\_whisker = Q1 - (whisker\_width\*IQR)  
upper\_whisker = Q3 + (whisker\_width\*IQR)  
data['CLIENT\_ENCOURS\_ENGAGEMENT'] = np.where(data['CLIENT\_ENCOURS\_ENGAGEMENT'] > upper\_whisker, upper\_whisker, np.where(data['CLIENT\_ENCOURS\_ENGAGEMENT'] < lower\_whisker, lower\_whisker, data['CLIENT\_ENCOURS\_ENGAGEMENT']))

In [53]: sns.boxplot(data['CLIENT\_ENCOURS\_ENGAGEMENT'], data=data)



In [54]: Q1=data['CLIENT\_ENCOURS\_A\_VUE'].quantile(0.25)  
Q3=data['CLIENT\_ENCOURS\_A\_VUE'].quantile(0.75)  
IQR=Q3-Q1  
whisker\_width = 1.5  
ENG\_outliers = data[(data['CLIENT\_ENCOURS\_A\_VUE'] < Q1 - whisker\_width\*IQR) | (data['CLIENT\_ENCOURS\_A\_VUE'] > Q3 + whisker\_width\*IQR)]  
ENG\_outliers.head()

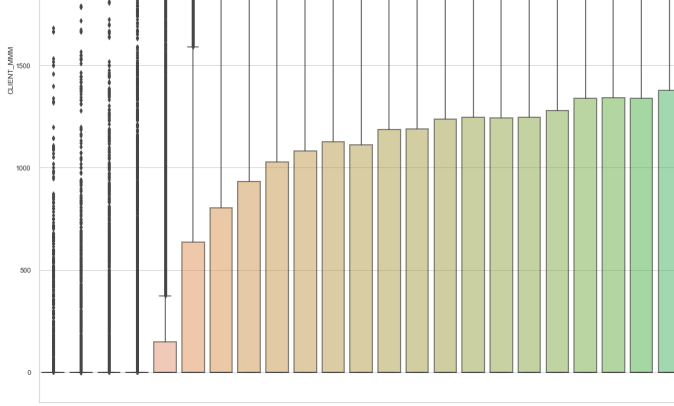
	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGAGEMENT
2	454569	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	57	Couffure et soins de beaute	3013.950	176.481
4	442731	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	58	Commerce de gros d'equipements pour la constru...	631.940	0.000
5	270519	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	70	Commerce de gros d'habillement	4025.130	0.000
6	98332	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	57	Construction de batiments (gros oeuvre)	981.287	0.000
8	271108	A1	PART	CLASSE MOYENNE	1 CM - SALARIÉ	60	Hotels avec restaurant	10.015	176.481

In [55]: AV\_mean = data['CLIENT\_ENCOURS\_A\_VUE'].mean()  
AV\_std = data['CLIENT\_ENCOURS\_A\_VUE'].std()  
low=AV\_mean - (3 \* AV\_std)  
high=AV\_mean + (3 \* AV\_std)  
ENG\_outliers = data[(data['CLIENT\_ENCOURS\_A\_VUE'] < low) | (data['CLIENT\_ENCOURS\_A\_VUE'] > high)]  
ENG\_outliers.head()

	CLIENT_ID	AGENCE	MARCHE	SEGMENT	SOUS SEGMENT	CLIENT_AGE	CLIENT_ACTIVITE	CLIENT_ENCOURS_A_VUE	CLIENT_ENCOURS_ENGAGEMENT
1508	449416	A1	PART	HAUT DE GAMME	1 ELITE	45	Fabrication de verre plat	257637.717	0
1537	328325	A1	PART	HAUT DE GAMME	2 PRESTIGE	49	Autres services personnels	121221.556	176
1543	385603	A1	PART	HAUT DE GAMME	2 PRESTIGE	64	Commerce de gros de biens de consommation non ...	151442.243	0
1545	442896	A1	PART	HAUT DE GAMME	2 PRESTIGE	47	Commerces de detail divers en magasin specialise	303863.881	176
1761	270593	A1	PART	HAUT DE GAMME	2 PRESTIGE	58	N_D	81522.380	0

In [56]: # Flooring and Capping  
lower\_whisker = Q1 - (whisker\_width\*IQR)  
upper\_whisker = Q3 + (whisker\_width\*IQR)  
data['CLIENT\_ENCOURS\_A\_VUE'] = np.where(data['CLIENT\_ENCOURS\_A\_VUE'] > upper\_whisker, upper\_whisker, np.where(data['CLIENT\_ENCOURS\_A\_VUE'] < lower\_whisker, lower\_whisker, data['CLIENT\_ENCOURS\_A\_VUE']))

In [57]: sns.boxplot(data['CLIENT\_ENCOURS\_A\_VUE'], data=data)



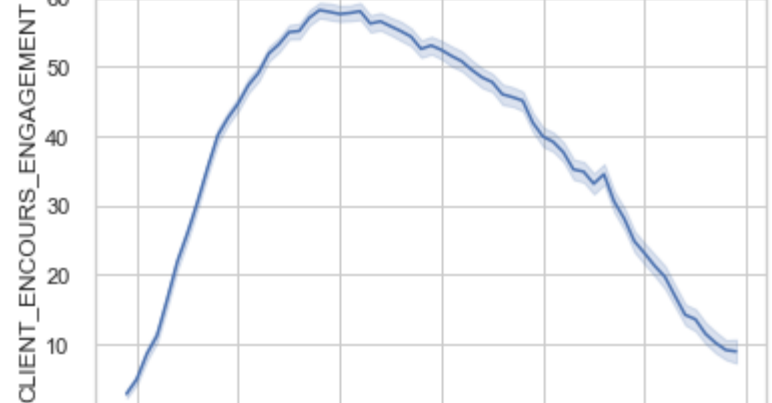
In [58]: data['CLIENT\_NOMBRE\_CARTES'].unique()

Out[58]: array([0, 1, 3, 2, 4, 5, 7, 6, 8, 12, 9, 10, 72])

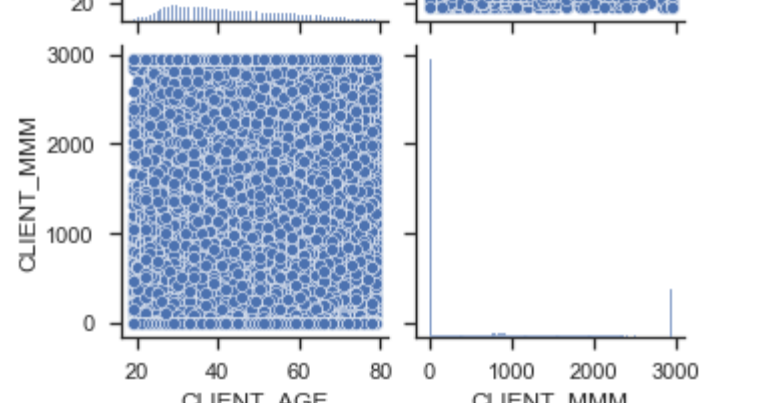
In [59]: index = data[(data['CLIENT\_NOMBRE\_CARTES'] > 5)].index  
data.drop(index, inplace=True)  
data['CLIENT\_NOMBRE\_CARTES'].describe()

Out[59]: count 743339.000000  
mean 0.878349  
std 1.019329  
min 0.000000  
25% 0.000000  
50% 1.000000  
75% 1.000000  
max 5.000000  
Name: CLIENT\_NOMBRE\_CARTES, dtype: float64

In [60]: # Histogram and density of clients MMM  
sns.distplot(data['CLIENT\_MMM'], hist=True, kde=True,  
bins=int(180/5), color = 'darkblue',  
hist\_kws={'edgecolor':'black'},  
kde\_kws={'linewidth': 4})



In [61]: # Histogram and density of clients VRDs  
sns.distplot(data['CLIENT\_VRD\_MMM'], hist=True, kde=True,  
bins=int(180/5), color = 'darkblue',  
hist\_kws={'edgecolor':'black'},  
kde\_kws={'linewidth': 4})



In [62]: plt.figure(figsize = (50, 20))  
ax = sns.boxplot(x = data['CLIENT\_AGE'], y = data['CLIENT\_VRD\_MMM'], data= data)  
plt.setp(ax.artists, alpha=.5, linewidth=2, edgecolor='k')  
plt.xticks(rotation=45)

Out[62]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60]),  
(Text(0, 0, '19'),  
Text(1, 0, '20'),  
Text(2, 0, '21'),  
Text(3, 0, '22'),  
Text(4, 0, '23'),  
Text(5, 0, '24'),  
Text(6, 0, '25'),  
Text(7, 0, '26'),  
Text(8, 0, '27'),  
Text(9, 0, '28'),  
Text(10, 0, '29'),  
Text(11, 0, '30'),  
Text(12, 0, '31'),  
Text(13, 0, '32'),  
Text(14, 0, '33'),  
Text(15, 0, '34'),  
Text(16, 0, '35'),  
Text(17, 0, '36'),  
Text(18, 0, '37'),  
Text(19, 0, '38'),  
Text(20, 0, '39'),  
Text(21, 0, '40'),  
Text(22, 0, '41'),  
Text(23, 0, '42'),  
Text(24, 0, '43'),  
Text(25, 0, '44'),  
Text(26, 0, '45'),  
Text(27, 0, '46'),  
Text(28, 0, '47'),  
Text(29, 0, '48'),  
Text(30, 0, '49'),  
Text(31, 0, '50'),  
Text(32, 0, '51'),  
Text(33, 0, '52'),  
Text(34, 0, '53'),  
Text(35, 0, '54'),  
Text(36, 0, '55'),  
Text(37, 0, '56'),  
Text(38, 0, '57'),  
Text(39, 0, '58'),  
Text(40, 0, '59'),  
Text(41, 0, '60'),  
Text(42, 0, '61'),  
Text(43, 0, '62'),  
Text(44, 0, '63'),  
Text(45, 0, '64'),  
Text(46, 0, '65'),  
Text(47, 0, '66'),  
Text(48, 0, '67'),  
Text(49, 0, '68'),  
Text(50, 0, '69'),  
Text(51, 0, '70'),  
Text(52, 0, '71'),  
Text(53, 0, '72'),  
Text(54, 0, '73'),  
Text(55, 0, '74'),  
Text(56, 0, '75'),  
Text(57, 0, '76'),  
Text(58, 0, '77'),  
Text(59, 0, '78'),  
Text(60, 0, '79')))



In [63]: plt.figure(figsize = (50, 20))  
ax = sns.boxplot(x = data['CLIENT\_AGE'], y = data['CLIENT\_MMM'], data= data)  
plt.setp(ax.artists, alpha=.5, linewidth=2, edgecolor='k')  
plt.xticks(rotation=45)

Out[63]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60]),  
(Text(0, 0, '19'),  
Text(1, 0, '20'),  
Text(2, 0, '21'),  
Text(3, 0, '22'),  
Text(4, 0, '23'),  
Text(5, 0, '24'),  
Text(6, 0, '25'),  
Text(7, 0, '26'),  
Text(8, 0, '27'),  
Text(9, 0, '28'),  
Text(10, 0, '29'),  
Text(11, 0, '30'),  
Text(12, 0, '31'),  
Text(13, 0, '32'),  
Text(14, 0, '33'),  
Text(15, 0, '34'),  
Text(16, 0, '35'),  
Text(17, 0, '36'),  
Text(18, 0, '37'),  
Text(19, 0, '38'),  
Text(20, 0, '39'),  
Text(21, 0, '40'),  
Text(22, 0, '41'),  
Text(23, 0, '42'),  
Text(24, 0, '43'),  
Text(25, 0, '44'),  
Text(26, 0, '45'),  
Text(27, 0, '46'),  
Text(28, 0, '47'),  
Text(29, 0, '48'),  
Text(30, 0, '49'),  
Text(31, 0, '50'),  
Text(32, 0, '51'),  
Text(33, 0, '52'),  
Text(34, 0, '53'),  
Text(35, 0, '54'),  
Text(36, 0, '55'),  
Text(37, 0, '56'),  
Text(38, 0, '57'),  
Text(39, 0, '58'),  
Text(40, 0, '59'),  
Text(41, 0, '60'),  
Text(42, 0, '61'),  
Text(43, 0, '62'),  
Text(44, 0, '63'),  
Text(45, 0, '64'),  
Text(46, 0, '65'),  
Text(47, 0, '66'),  
Text(48, 0, '67'),  
Text(49, 0, '68'),  
Text(50, 0, '69'),  
Text(51, 0, '70'),  
Text(52, 0, '71'),  
Text(53, 0, '72'),  
Text(54, 0, '73'),  
Text(55, 0, '74'),  
Text(56, 0, '75'),  
Text(57, 0, '76'),  
Text(58, 0, '77'),  
Text(59, 0, '78'),  
Text(60, 0, '79')))



In [64]: fig,ax = plt.subplots()  
ax = sns.lineplot(data = data, x='CLIENT\_AGE', y='CLIENT\_MMM')  
ax.set\_title('MMM by AGE')



In [65]: fig,ax = plt.subplots()  
ax = sns.lineplot(data = data, x='CLIENT\_AGE', y='CLIENT\_VRD\_MMM')  
ax.set\_title('VRD by AGE')



In [66]: fig,ax = plt.subplots()  
ax = sns.lineplot(data = data, x='CLIENT\_AGE', y='CLIENT\_ENCOURS\_ENGAGEMENT')  
ax.set\_title('ENGAGEMENT by AGE')



In [67]: g = sns.pairplot(data, vars=['CLIENT\_AGE', 'CLIENT\_MMM'])  
plt.show()



In [68]: sns.set(style='ticks', color\_codes=True)  
gg = sns.pairplot(data, vars=['CLIENT\_AGE', 'TOTAL\_PACK'])  
plt.show()









