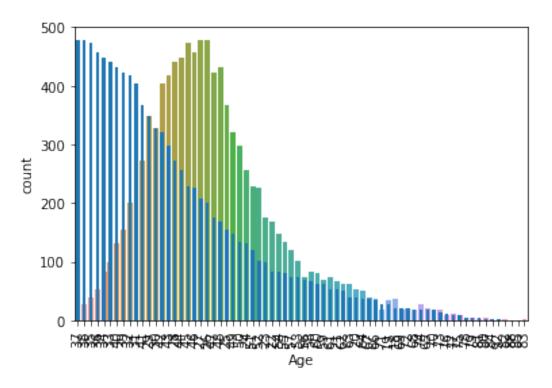
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read csv('/content/Churn Modelling.csv')
df.head()
   RowNumber CustomerId
                            Surname CreditScore Geography Gender Age
\
0
           1
                15634602 Hargrave
                                             619
                                                     France Female
                                                                      42
1
           2
                15647311
                               Hill
                                              608
                                                      Spain Female
                                                                       41
2
           3
                15619304
                               Onio
                                              502
                                                     France Female
                                                                      42
3
           4
                15701354
                               Boni
                                              699
                                                     France Female
                                                                       39
                                                      Spain Female
4
           5
                15737888 Mitchell
                                              850
                                                                      43
             Balance
                      NumOfProducts HasCrCard
                                                  IsActiveMember
   Tenure
0
                0.00
        2
                                   1
                                               1
                                                               1
1
        1
            83807.86
                                   1
                                              0
                                                               1
2
        8
                                   3
                                              1
           159660.80
                                                               0
3
        1
                0.00
                                   2
                                              0
                                                               0
4
        2
           125510.82
                                   1
                                               1
                                                               1
   EstimatedSalary
                    Exited
0
         101348.88
                          1
1
         112542.58
                          0
2
                          1
         113931.57
3
          93826.63
                          0
4
          79084.10
                          0
#univariate analysis
#categorical data
a.countplot
sns.countplot(df['Age'])
df['Age'].value counts().plot(kind='bar')
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be 'data', and
```

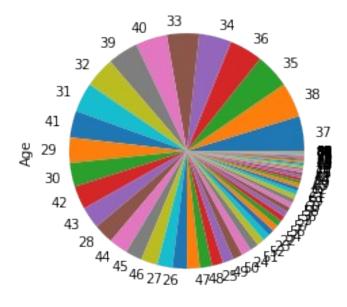
passing other arguments without an explicit keyword will result in an error or misinterpretation.
FutureWarning

<matplotlib.axes._subplots.AxesSubplot at 0x7f629de0ae10>



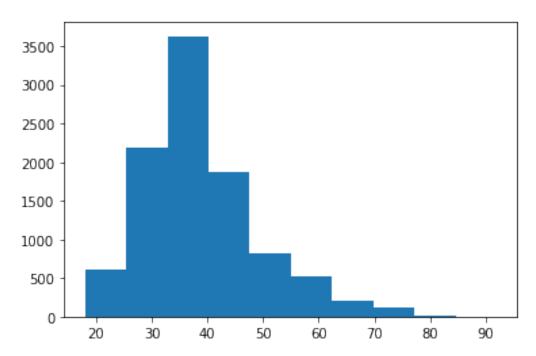
b.piechart

df['Age'].value_counts().plot(kind='pie')
<matplotlib.axes._subplots.AxesSubplot at 0x7f629db16490>



#numerical data

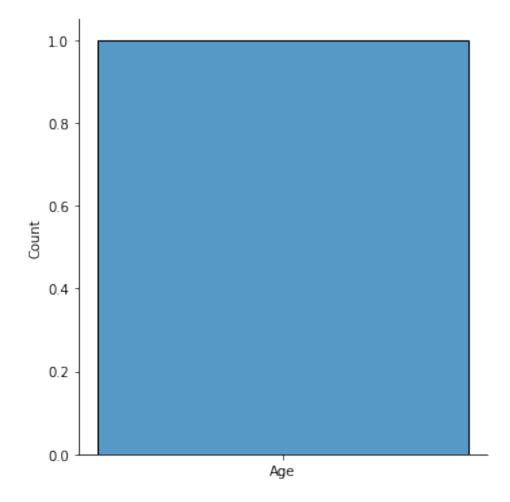
a. histogram



b.distplot

sns.displot(['Age'])

<seaborn.axisgrid.FacetGrid at 0x7f629d588c50>



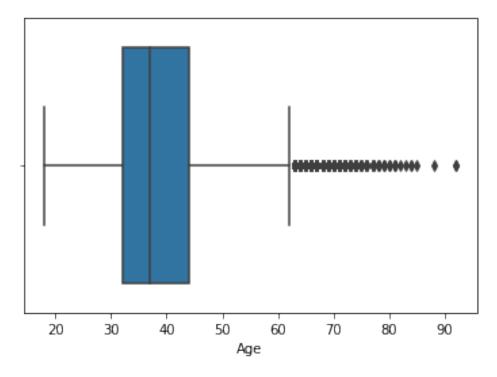
c.boxplot

sns.boxplot(df['Age'])

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

<matplotlib.axes._subplots.AxesSubplot at 0x7f629d527a10>



```
df['Age'].max()
92
df['Age'].min()
18
df['Age'].mean()
38.9218
```

bivariate analysis

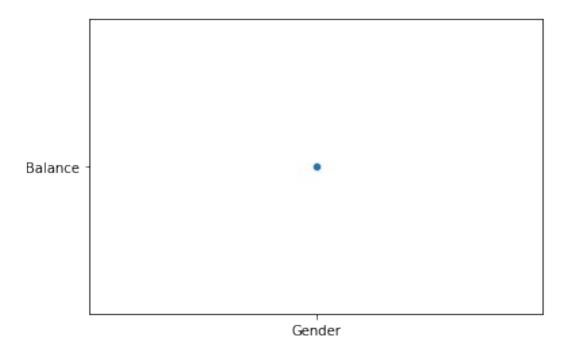
```
1. scatter plot (numerical - numerical\\
```

```
sns.scatterplot(['Gender'],['Balance'])
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

<matplotlib.axes. subplots.AxesSubplot at 0x7f629b604c50>



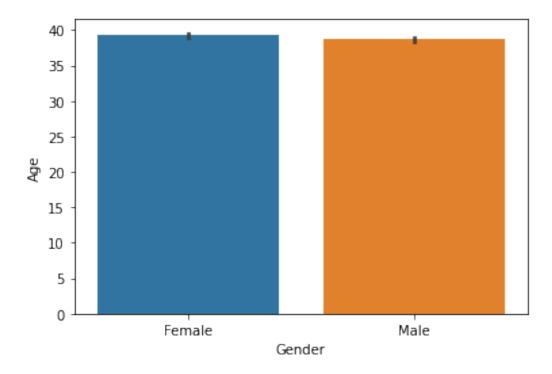
2.bar plot (numerical _categorical)

sns.barplot(df['Gender'],df['Age'])

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

<matplotlib.axes. subplots.AxesSubplot at 0x7f629b589250>



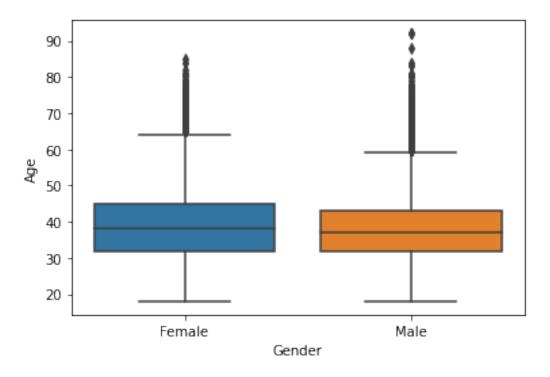
3.box plot(numerical_categorical)

sns.boxplot(df['Gender'],df['Age'])

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

<matplotlib.axes. subplots.AxesSubplot at 0x7f629b56afd0>



4.heatMap(categorical-categorical)

df.head(3)

`	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age
ò	1	15634602	Hargrave	619	France	Female	42
1	2	15647311	Hill	608	Spain	Female	41
2	3	15619304	Onio	502	France	Female	42

	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	\
0	2	0.00	1	1	1	
1	1	83807.86	1	0	1	
2	8	159660.80	3	1	0	

	EstimatedSalary	Exited
0	101348.88	1
1	112542.58	0
2	113931.57	1

pd.crosstab(df['Age'],df['Balance'])

Balance 0.00 3768.69 12459.19 14262.80 16893.59 23503.31 \ Age

18	8	Θ	0	0	6)
0 19 0	11	Θ	0	Θ	6)
20 0	17	Θ	Θ	Θ	6)
21 0	20	Θ	Θ	Θ	6)
22 0	36	0	0	0	e)
83 0	0	Θ	Θ	Θ	6)
84 0	0	Θ	Θ	Θ	6)
85	1	Θ	Θ	Θ	6)
0 88	1	0	0	0	6)
0 92 0	0	0	0	0	6)
Balance Age	24043.45	27288.43	27517.15	27755.97	2126	592.97 \
18 19	0 0	0 0	0 0	0 0		0 0
20 21	0 0	0 0	0 0	0 0		0 0
22	0	0	0	0		0
83 84	0 0	0 0	0 0	0 0		0 0
85 88	0 0	0 0	0 0	0 0		0 0
92	Θ	0	0	0		0
Balance 221532.8 Age		212778.20	213146.20	214346.96	216109.88	3
18	0	0	0	0	6)
0 19	0	0	0	0	6)
0 20	0	0	0	0	6)
0 21	0	0	0	0	6)
0 22	0	0	0	0	6)

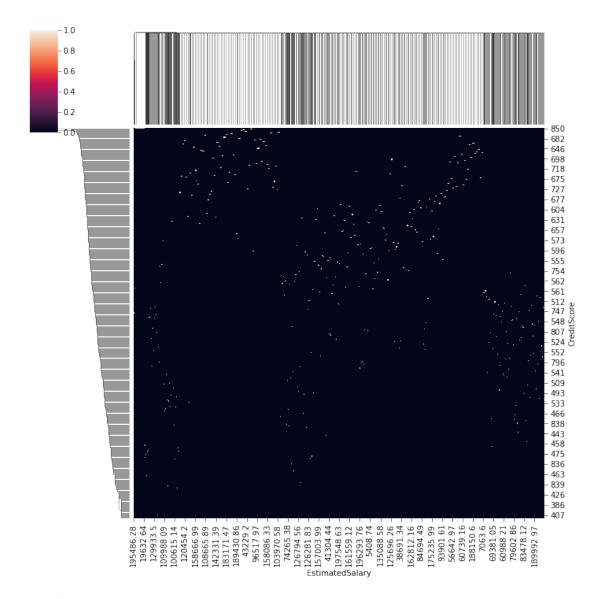
0						
83 0	0	0	0	0	0	
84 0	0	0	0	0	0	
85 0	0	0	0	0	0	
88 0	0	0	0	0	0	
92 0	0	0	0	0	0	
Balance Age 18 19 20 21 22 83 84 85 88 92	222267.63 0 0 0 0 0 0 0 0	238387.56 0 0 0 0 0 0 0 0	250898.09 0 0 0 0 0 0 0			
[70 rows	x 6382 col	umns]				

6.clusterMap(categorical_categorical)

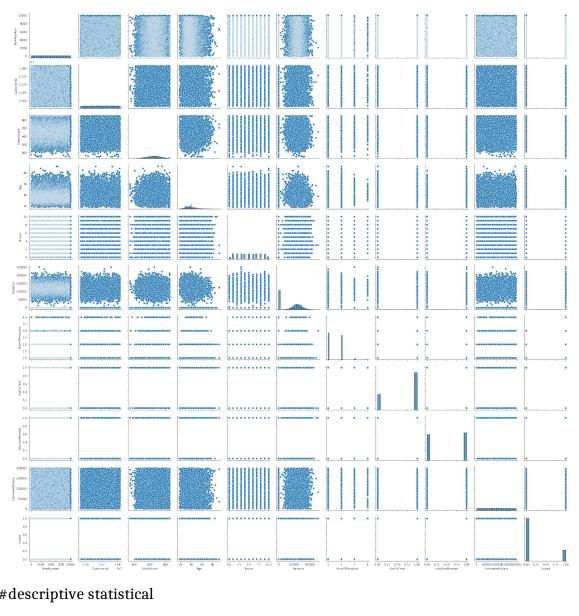
sns.clustermap(pd.crosstab(df['CreditScore'],df['EstimatedSalary']))

/usr/local/lib/python3.7/dist-packages/seaborn/matrix.py:654:
UserWarning: Clustering large matrix with scipy. Installing
`fastcluster` may give better performance.
warnings.warn(msg)

<seaborn.matrix.ClusterGrid at 0x7f629d5e6710>



7.pairplot
sns.pairplot(df)
<seaborn.axisgrid.PairGrid at 0x7f629d4ef150>



#descriptive statistical

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv('/content/Churn_Modelling.csv')
df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
Age 0 42	1	15634602	Hargrave	619	France	Female
42 1 41	2	15647311	Hill	608	Spain	Female
2	3	15619304	Onio	502	France	Female

42						
3	4	15701354	Boni	69	9 France	Female
39						
4	5	15737888	Mitchell	85	9 Spain	Female
43						
9995	9996	15606229	Obijiaku	77	1 France	Male
39 9996	9997	15569892	Johnstone	51	6 France	Male
35	3337	13303032	301113 20112	31	3 Trance	Hate
9997	9998	15584532	Liu	70	9 France	Female
36 9998	9999	15682355	Sabbatini	77.	2 Germany	Male
42	9999	13002333	Jabbatini	77.	2 Germany	riace
9999	10000	15628319	Walker	79:	2 France	Female
28						
	Tenure B	Balance Nur	nOfProducts	HasCrCard	IsActiveMem	nber \
0	2	0.00	1	1		1
1		8807.86	1	0		1
2 3	8 159 1	0.00 0.00	3 2	1 0		0 0
4		5510.82	1	1		1
	· · <u>·</u>					
9995 9996	5 10 57	0.00 369.61	2 1	1 1		0 1
9997	7	0.00	1	0		1
9998		075.31	2	1		0
9999	4 130	142.79	1	1		0
	EstimatedSa	ılarv Exite	ad			
0	10134	,	1			
1	11254	2.58	0			
2	11393		1			
3 4		26.63 84.10	0			
	7300					
9995		0.64	0			
9996 9997	10169	19.77 35.58	0 1			
9998		88.52	1			
9999		0.78	0			
[10000	rows x 14	columns]				
df.des	scribe()					
	RowNumbe	er Custor	merId Cred	litScore	Age	
Tenure	١ ،				-	

Tenure \

```
1.000000e+04
       10000.00000
                                    10000.000000
                                                   10000.000000
count
10000.000000
mean
        5000.50000
                     1.569094e+07
                                      650.528800
                                                       38.921800
5.012800
std
        2886.89568
                     7.193619e+04
                                       96.653299
                                                       10.487806
2.892174
            1.00000
                     1.556570e+07
                                      350.000000
                                                       18.000000
min
0.000000
25%
        2500.75000
                     1.562853e+07
                                      584.000000
                                                       32.000000
3.000000
50%
        5000.50000
                     1.569074e+07
                                      652.000000
                                                       37.000000
5.000000
75%
        7500.25000
                     1.575323e+07
                                      718.000000
                                                       44.000000
7.000000
       10000.00000
max
                     1.581569e+07
                                      850.000000
                                                       92.000000
10.000000
                       NumOfProducts
              Balance
                                          HasCrCard
                                                      IsActiveMember
                         10000.000000
                                                        10000.000000
        10000.000000
                                        10000.00000
count
        76485.889288
                             1.530200
                                            0.70550
                                                            0.515100
mean
        62397.405202
std
                             0.581654
                                            0.45584
                                                            0.499797
min
            0.000000
                             1.000000
                                            0.00000
                                                            0.00000
25%
            0.000000
                             1.000000
                                            0.00000
                                                            0.000000
        97198.540000
50%
                             1.000000
                                            1.00000
                                                            1.000000
75%
       127644.240000
                             2,000000
                                            1.00000
                                                            1.000000
       250898.090000
                             4.000000
                                            1.00000
                                                            1.000000
max
       EstimatedSalary
                                Exited
           10000.000000
                          10000.000000
count
         100090.239881
                              0.203700
mean
std
          57510.492818
                              0.402769
              11.580000
                              0.00000
min
          51002.110000
25%
                              0.000000
50%
         100193.915000
                              0.000000
75%
         149388.247500
                              0.000000
         199992.480000
                              1.000000
max
df.describe(include=['object'])
       Surname Geography Gender
         10000
                    10000
                            10000
count
unique
          2932
                        3
                                2
top
         Smith
                   France
                             Male
freq
            32
                     5014
                             5457
df['Age'].value_counts()
37
      478
38
      477
35
      474
36
      456
```

```
34
      447
92
        2
82
        1
88
        1
85
        1
83
        1
Name: Age, Length: 70, dtype: int64
df['Age'].value_counts().to_frame()
    Age
37
    478
38 477
35
   474
36
    456
34
   447
92
      2
82
      1
88
      1
85
      1
83
      1
[70 rows x 1 columns]
model counts.index.name='Balance'
model_counts
         Age
Balance
37
         478
38
         477
35
         474
36
         456
34
         447
. . .
           2
92
           1
82
88
           1
85
           1
83
           1
[70 rows x 1 columns]
#handling missing values
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

df=pd.read_csv('/content/Churn_Modelling.csv')
df

A	RowNumber	Custome	rId	Surname	CreditScore	Geography	Gender
Age 0	1	156346	602	Hargrave	619	France	Female
42 1	2	2 156473	311	Hill	608	Spain	Female
41	3	156193	304	Onio	502	France	Female
42 3	4	157013	354	Boni	699	France	Female
39 4	5	157378	888	Mitchell	850	Spain	Female
43 							
9995	9996	156062	229	Obijiaku	771	France	Male
39 9996	9997	155698	892	Johnstone	516	France	Male
35 9997	9998	155845	532	Liu	709	France	Female
36 9998	9999	156823	355	Sabbatini	772	Germany	Male
42 9999 28	10000	156283	319	Walker	792	France	Female
Z0							
	Tenure	Balance	Num	OfProducts		IsActiveMem	
0 1	2	Balance 0.00 83807.86	Num(OfProducts 1 1	HasCrCard 1	IsActiveMem	ber \ 1 1
0 1 2	2 1 8 1	0.00 83807.86 59660.80	Num(1 1 3	1 0 1	IsActiveMem	1 1 0
0 1 2 3	2 1 8 1 1	0.00 83807.86 59660.80 0.00	Num(1 1 3 2	1 0 1 0	IsActiveMem	1 1 0 0
0 1 2	2 1 8 1 1	0.00 83807.86 59660.80	Num(1 1 3	1 0 1	IsActiveMem	1 1 0
0 1 2 3 4	2 1 8 1 1 2 1 	0.00 83807.86 .59660.80 0.00 .25510.82 	Num	1 3 2 1	1 0 1 0 1 	IsActiveMem	1 1 0 0 1
0 1 2 3 4 9995 9996	2 1 8 1 1 2 1 5 10	0.00 83807.86 .59660.80 0.00 .25510.82 0.00 57369.61	Num(1 1 3 2 1 2 1	1 0 1 0 1 1	IsActiveMem	1 1 0 0 1
0 1 2 3 4 9995 9996 9997	2 1 8 1 1 2 1 5 10 7	0.00 83807.86 .59660.80 0.00 .25510.82 0.00 57369.61 0.00	Num	1 3 2 1 2 1	1 0 1 0 1 1 1 0	IsActiveMem	1 0 0 1 0 1
0 1 2 3 4 9995 9996	2 1 8 1 1 2 1 5 10 7 3	0.00 83807.86 .59660.80 0.00 .25510.82 0.00 57369.61	Num	1 1 3 2 1 2 1	1 0 1 0 1 1	IsActiveMem	1 1 0 0 1
0 1 2 3 4 9995 9996 9997 9998	2 1 8 1 1 2 1 5 10 7 3 4 1	0.00 83807.86 .59660.80 0.00 .25510.82 0.00 57369.61 0.00 75075.31		1 1 3 2 1 2 1 1 2	1 0 1 0 1 1 1 0	IsActiveMem	1 0 0 1 0 1 1 0
0 1 2 3 4 9995 9996 9997 9998	2 1 8 1 1 2 1 5 10 7 3 4 1	0.00 83807.86 .59660.80 0.00 .25510.82 0.00 57369.61 0.00 75075.31	xite	1 1 3 2 1 2 1 1 2	1 0 1 0 1 1 1 0	IsActiveMen	1 0 0 1 0 1 1 0
0 1 2 3 4 9995 9996 9997 9998 9999	2 1 8 1 1 2 1 5 10 7 3 4 1 Estimated 101 112	0.00 83807.86 .59660.80 0.00 .25510.82 0.00 57369.61 0.00 75075.31 .30142.79 ISalary Ex	xite	1 1 3 2 1 2 1 1 2 1	1 0 1 0 1 1 1 0	IsActiveMem	1 0 0 1 0 1 1 0
0 1 2 3 4 9995 9996 9997 9998 9999	2 1 8 1 1 2 1 5 10 7 3 4 1 Estimated 101 112 113	0.00 83807.86 .59660.80 0.00 .25510.82 0.00 57369.61 0.00 75075.31 .30142.79 ISalary Ex. 348.88 2542.58	xite (1 1 3 2 1 2 1 1 2 1	1 0 1 0 1 1 1 0	IsActiveMem	1 0 0 1 0 1 1 0
0 1 2 3 4 9995 9996 9997 9998 9999	2 1 8 1 1 2 1 5 10 7 3 4 1 Estimated 101 112 113 93	0.00 83807.86 .59660.80 0.00 .25510.82 0.00 57369.61 0.00 75075.31 .30142.79 ISalary Ex	xite (1 1 3 2 1 2 1 1 2 1	1 0 1 0 1 1 1 0	IsActiveMem	1 0 0 1 0 1 1 0
0 1 2 3 4 9995 9996 9997 9998 9999	2 1 8 1 1 2 1 5 10 7 3 4 1 Estimated 101 112 113 93 79	0.00 83807.86 .59660.80 0.00 .25510.82 0.00 57369.61 0.00 75075.31 .30142.79 ISalary Ex. 348.88 2542.58 8931.57	xite : : : (1 1 3 2 1 2 1 1 2 1	1 0 1 0 1 1 1 0	IsActiveMem	1 0 0 1 0 1 1 0

9997	42085.58	1
9998	92888.52	1
9999	38190.78	0

[10000 rows x 14 columns]

df.shape

(10000, 14)

df.isnull()

A a a		er Custo	merId	Surname	CreditScore	e Geography	Gender
0	\ Fals	se	False	False	False	e False	False
False	Fals	se	False	False	False	e False	False
False 2 False	Fals	se	False	False	False	e False	False
3	Fals	se	False	False	False	e False	False
False 4 False	Fals	se	False	False	False	e False	False
9995 Falso	Fals	se	False	False	False	e False	False
False 9996	Fals	se	False	False	False	e False	False
False 9997 False	Fals	se	False	False	False	e False	False
9998 False	Fals	se	False	False	False	e False	False
9999 False	Fals	se	False	False	False	e False	False
0 1 2 3 4	Tenure False False False False	Balance False False False False False	NumOf	Products False False False False False	HasCrCard False False False False False	IsActiveMemb Fal Fal Fal Fal	se se se se
9995 9996 9997 9998 9999	False False False False False	False False False False False		False False False False False	False False False False False	Fal Fal Fal Fal	se se se

EstimatedSalary Exited

```
0
                  False
                           False
1
                  False
                           False
2
                           False
                  False
3
                  False
                           False
4
                  False
                           False
. . .
                  False
9995
                           False
9996
                  False
                           False
9997
                  False
                           False
9998
                  False
                           False
9999
                  False
                           False
[10000 \text{ rows } \times 14 \text{ columns}]
df.isnull().sum()
RowNumber
                     0
CustomerId
                     0
Surname
                     0
CreditScore
                     0
Geography
                     0
Gender
                     0
Age
                     0
Tenure
                     0
Balance
                     0
NumOfProducts
                     0
HasCrCard
                     0
IsActiveMember
                     0
EstimatedSalary
                     0
Exited
                     0
dtype: int64
df.isnull().sum().sum()
0
fill the null values
df2 = df.fillna(value=0)
df2
      RowNumber CustomerId
                                   Surname CreditScore Geography
                                                                       Gender
Age
                1
                     15634602
                                 Hargrave
                                                      619
                                                              France
                                                                       Female
42
                2
                                      Hill
1
                     15647311
                                                      608
                                                               Spain
                                                                       Female
41
2
                3
                     15619304
                                      Onio
                                                      502
                                                              France
                                                                       Female
42
3
                4
                     15701354
                                      Boni
                                                      699
                                                              France
                                                                       Female
```

Mitchell

850

Spain

Female

39

5

15737888

```
43
. . .
                          . . .
9995
            9996
                    15606229
                                Obijiaku
                                                    771
                                                            France
                                                                      Male
39
9996
            9997
                    15569892
                               Johnstone
                                                    516
                                                            France
                                                                      Male
35
9997
            9998
                    15584532
                                      Liu
                                                    709
                                                            France Female
36
9998
            9999
                    15682355
                               Sabbatini
                                                    772
                                                           Germany
                                                                      Male
42
9999
           10000
                    15628319
                                  Walker
                                                    792
                                                            France
                                                                    Female
28
                 Balance
      Tenure
                           NumOfProducts
                                           HasCrCard
                                                       IsActiveMember
0
                    0.00
            2
                                                    1
            1
                83807.86
                                        1
                                                    0
                                                                      1
1
2
            8
               159660.80
                                        3
                                                    1
                                                                     0
3
                                        2
            1
                                                    0
                                                                     0
                    0.00
            2
4
               125510.82
                                        1
                                                    1
                                                                      1
                                        2
           5
                                                                     0
9995
                    0.00
                                                    1
9996
                                        1
                                                    1
                                                                     1
           10
                57369.61
                                                                      1
9997
            7
                    0.00
                                        1
                                                    0
9998
            3
                75075.31
                                        2
                                                    1
                                                                     0
9999
               130142.79
                                        1
                                                    1
                                                                     0
      EstimatedSalary
                        Exited
0
             101348.88
                              1
1
                              0
             112542.58
2
             113931.57
                              1
3
              93826.63
                              0
4
              79084.10
                              0
                             . . .
              96270.64
                              0
9995
9996
             101699.77
                              0
9997
              42085.58
                              1
              92888.52
                              1
9998
9999
              38190.78
[10000 rows x 14 columns]
df2.isnull().sum().sum()
0
df3 = df.fillna(value=5)
df3
      RowNumber CustomerId
                                 Surname CreditScore Geography
Age \
```

0 42		1	15634	602	Hargrav	е	619	France	Fem	nale
1		2	15647	311	Hil	ι	608	Spain	Fem	nale
41		3	15619	304	0ni	0	502	France	Fem	nale
42 3		4	15701	354	Bon	i	699	France	Fem	nale
39 4 43		5	15737	888	Mitchel	ι	850	Spain	Fem	nale
45										
9995	99	96	15606	229	0bijiak	u	771	France	M	lale
39 9996	99	97	15569	892	Johnston	е	516	France	M	lale
35 9997	99	98	15584	532	Li	u	709	France	Fem	nale
36 9998	99	99	15682	355	Sabbatin	i	772	Germany	M	lale
42 9999 28	100	00	15628	319	Walke	r	792	France	Fem	nale
0 1 2 3 4 9995 9996 9997 9998 9999	Tenure 2 1 8 1 2 5 10 7 3 4	8380 15960 1255 5730 750	lance 0.00 97.86 60.80 0.00 10.82 0.00 69.61 0.00 75.31 42.79	Num		1 1 3 2 1	 1 0 1 0 1	IsActiveMem	1 0 0 1 0 1 1 0	\
0 1 2 3 4 	1 1	edSala 01348 12542 13931 93826 79084	.88 .58 .57 .63		d 1 0 1 0 0					

[10000 rows x 14 columns]

filling null value with privious value

df4 = df.fillna(method='pad')
df4

۸۵۵	RowNumbe	r Custome	rId	Surname	CreditScore	Geography	Gender
Age 0	\	1 15634	602	Hargrave	619	France	Female
42 1 41 2		2 15647	311	Hill	608	Spain	Female
		3 15619	304	Onio	502	France	Female
42 3		4 15701	354	Boni	699	France	Female
39 4		5 15737	888	Mitchell	850	Spain	Female
43 							
 9995	999	6 15606	229	Obijiaku	771	France	Male
39 9996	999			Johnstone	516	France	Male
35						Trance	
9997 36	999	8 15584	532	Liu	709	France	Female
36 9998	9999 15682		355	Sabbatini	772	Germany	Male
42 9999 28	1000	0 15628	319	Walker	792	France	Female
0 1 2 3 4	1	Balance 0.00 83807.86 159660.80 0.00 125510.82	Num(OfProducts 1 1 3 2 1	HasCrCard 1 0 1 0 1	IsActiveMem	ber \ 1
9995 9996 9997 9998 9999	5 10 7 3 4	0.00 57369.61 0.00 75075.31 130142.79		2 1 1 2 1	1 1 0 1 1		 0 1 1 0 0
0 1 2 3 4	11 11 9	dSalary E 1348.88 2542.58 3931.57 3826.63 9084.10	([(d 1 9 1 9			

```
9995
             96270.64
                             0
9996
            101699.77
                             0
9997
             42085.58
                             1
                             1
9998
             92888.52
9999
             38190.78
                             0
[10000 rows x 14 columns]
df4.isnull().sum()
RowNumber
                    0
CustomerId
                    0
Surname
                    0
CreditScore
                    0
Geography
                    0
Gender
                    0
Age
                    0
Tenure
                    0
Balance
                    0
NumOfProducts
                    0
HasCrCard
                    0
IsActiveMember
                    0
EstimatedSalary
                    0
Exited
                    0
dtype: int64
#filling null values with next values
df5 = df.fillna(method='bfill')
df5
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender
Age 0 42	1	15634602	Hargrave	619	France	Female
42 1 41	2	15647311	Hill	608	Spain	Female
2 42	3	15619304	Onio	502	France	Female
3 39	4	15701354	Boni	699	France	Female
4 43	5	15737888	Mitchell	850	Spain	Female
9995 39	9996	15606229	0bijiaku	771	France	Male
9996 35	9997	15569892	Johnstone	516	France	Male
9997 36	9998	15584532	Liu	709	France	Female
9998	9999	15682355	Sabbatini	772	Germany	Male

```
42
9999
          10000
                    15628319
                                  Walker
                                                    792
                                                            France Female
28
                                                       IsActiveMember \
                           NumOfProducts HasCrCard
      Tenure
                 Balance
0
                     0.00
            2
                                        1
                                                    1
1
            1
                83807.86
                                        1
                                                    0
                                                                      1
2
                                        3
            8
               159660.80
                                                    1
                                                                      0
                                        2
3
            1
                                                                      0
                    0.00
                                                    0
4
            2
               125510.82
                                        1
                                                    1
                                                                      1
          . . .
                                       . . .
                                                   . . .
                                                                    . . .
           5
                                        2
                                                                      0
9995
                    0.00
                                                    1
9996
           10
                57369.61
                                        1
                                                    1
                                                                      1
                                        1
                                                                      1
9997
            7
                    0.00
                                                    0
            3
                75075.31
                                        2
9998
                                                    1
                                                                      0
                                        1
                                                                      0
9999
               130142.79
                                                    1
      EstimatedSalary Exited
0
             101348.88
1
             112542.58
                              0
2
             113931.57
                              1
3
              93826.63
                              0
4
              79084.10
                              0
              96270.64
                              0
9995
9996
             101699.77
                              0
9997
              42085.58
                              1
              92888.52
9998
                              1
9999
              38190.78
[10000 \text{ rows } \times 14 \text{ columns}]
df6 =df.fillna(method='pad',axis=1)
df6
     RowNumber CustomerId Surname CreditScore Geography Gender Age
Tenure \
0
              1
                  15634602
                              Hargrave
                                                 619
                                                         France Female 42
2
1
              2
                  15647311
                                  Hill
                                                 608
                                                          Spain Female 41
1
2
              3
                                   Onio
                  15619304
                                                 502
                                                         France Female 42
8
3
              4
                  15701354
                                   Boni
                                                 699
                                                         France Female 39
1
```

Mitchell

Obijiaku

. . .

Spain Female 43

Male 39

. . .

France

5							
9996 10	9997	15569892	Johnstone	516	France	Male	35
9997 7	9998	15584532	Liu	709	France	Female	36
9998	9999	15682355	Sabbatini	772	Germany	Male	42
3 9999 4	10000	15628319	Walker	792	France	Female	28
		NumOfProdu	cts HasCrC	ard IsActive	Member Esti	.matedSal	ary
Exite 0	0.0		1	1	1	101348	.88
1	83807.86		1	0	1	112542	.58
0 2	159660.8		3	1	0	113931	.57
1	0.0		2	0	0	93826	.63
0 4	125510.82		1	1	1	7908	4.1
0 							
9995	0.0		2	1	0	96270	.64
0 9996	57369.61		1	1	1	101699	. 77
0 9997	0.0		1	0	1	42085	.58
1 9998	75075.31		2	1	0	92888	.52
1 9999 0	130142.79		1	1	0	38190	.78
[1000	0 rows x 14	4 columns]					
df7 = df7	df.fillna(r	method='bfi	ll',axis=1)			
	RowNumber (CustomerId	Surname	CreditScore	Geography	Gender	Age
Tenur 0	e \ 1	15634602	Hargrave	619	France	Female	42
2	2	15647311	Hill	608	Spain	Female	41
1 2	3	15619304	Onio	502	France	Female	42
2 8 3 1	4	15701354	Boni	699	France	Female	39

4 2	5	15737888	M	itchell		850	Spain	Female	43
9995	9996	15606229	01	oijiaku		771	France	Male	39
5 9996	9997	15569892	Jol	nnstone		516	France	Male	35
10 9997	9998	15584532		Liu		709	France	Female	36
7 9998	9999	15682355	Sal	obatini		772	Germany	Male	42
3 9999 4	10000	15628319		Walker		792	France	Female	28
Γγ i +ο.		NumOfProdu	cts	HasCrCa	rd	IsActiveMer	nber Esti	matedSal	ary
Exited 0 1	0.0		1		1		1	101348	.88
1 1 0	83807.86		1		0		1	112542	.58
2	159660.8		3		1		0	113931	. 57
1 3 0	0.0		2		0		0	93826	.63
4 0	125510.82		1		1		1	7908	4.1
				•					
9995	0.0		2		1		0	96270	.64
0 9996	57369.61		1		1		1	101699	. 77
0 9997	0.0		1		0		1	42085	.58
1 9998	75075.31		2		1		0	92888	.52
1 9999 0	130142.79		1		1		0	38190	.78
[1000	o rows x 14	4 columns]							
filling different values in null in different columns									
<pre>df8=df.fillna({'HasCrCard' : 'abcd',</pre>									
df8		paralice :	ueı	1)					
Age `	RowNumber	CustomerI	d	Surname	е	CreditScore	e Geograp	hy Gend	er
0	1	1563460	2	Hargrave	е	619	9 Fran	ce Fema	le

42 1		2	15647	311	Hill	60	08 Spair	n Female
41		3	15619	304	Onio	50	2 France	e Female
42 3		4	15701	354	Boni	69	9 France	e Female
39 4		5	15737	888	Mitchell	85	0 Spair	n Female
43 								
9995	99	96	15606	229	0bijiaku	77	'1 France	e Male
39 9996	99	97	15569	892	Johnstone	51	.6 France	e Male
35 9997	99	98	15584	532	Liu	70	9 France	e Female
36 9998	99	99	15682	355	Sabbatini	77	'2 Germany	, Male
42 9999 28	100	00	15628	319	Walker	79	2 France	e Female
0 1 2 3 4	Tenure 2 1 8 1 2	838 1596	lance 0.00 07.86 60.80 0.00 10.82	Num	OfProducts 1 1 3 2 1	HasCrCard 1 0 1 0 1	IsActiveMe	ember \ 1 0 0 1
9995 9996 9997 9998 9999	5 10 7 3 4	750	0.00 69.61 0.00 75.31 42.79		2 1 1 2 1	1 1 0 1		0 1 1 0 0
0 1 2 3 4 9995 9996 9997 9998 9999	1 1	edSal 01348 12542 13931 93826 79084 96270 01699 42085 92888 38190	.88 .58 .57 .63 .10 .64 .77 .58		d 1 0 1 0 0 0 0 0 1 1			

[10000 rows x 14 columns]

```
#interpolate()
df['Age']=df['Age'].interpolate(method='linear')
df
```

٨٥٥	RowNumber	Customer	Id	Surname	CreditScore	Geography	Gender
Age 0	1	156346	92	Hargrave	619	France	Female
42 1	2	156473	11	Hill	608	S Spain	Female
41 2	3	156193	94	Onio	502	: France	Female
42 3	4	157013	54	Boni	699	France	Female
39 4	5	157378	38	Mitchell	850	Spain	Female
43 							
 9995	9996	156062	29	Obijiaku	771	. France	Male
39 9996	9997			Johnstone	516		Male
35							
9997 36	9998			Liu	709		Female
9998 42	9999	156823		Sabbatini	772	•	Male
9999 28	10000	156283	19	Walker	792	! France	Female
0 1 2 3 4 9995 9996 9997 9998 9999	8 1 1 2 1 5 10 7 3	Balance 0.00 83807.86 59660.80 0.00 25510.82 0.00 57369.61 0.00 75075.31 30142.79	NumO	fProducts 1 1 3 2 1 2 1 1 2 1	HasCrCard 1 0 1 0 1 1 0 1 1	IsActiveMer	nber \ 1 0 0 1 0 1 1 0 0
0 1 2 3 4 	112 113 93 79	Salary Ex. 348.88 542.58 931.57 826.63 084.10 270.64	ited 1 0 1 0 0				

9996	101699.77	0
9997	42085.58	1
9998	92888.52	1
9999	38190.78	0

[10000 rows x 14 columns]

#find the outliers and replace the outliers

df.describe()

50%

75%

max

100193.915000

149388.247500 199992.480000

	` '				
Tenu	RowNumber ire \	CustomerId	CreditScore	Age	
coun	it 10000.00000	1.000000e+04	10000.000000	10000.000000	
mean		1.569094e+07	650.528800	38.921800	
std	.2800 2886.89568	7.193619e+04	96.653299	10.487806	
min	1.00000	1.556570e+07	350.000000	18.000000	
25%	00000 2500.75000	1.562853e+07	584.000000	32.000000	
50%	5000.50000	1.569074e+07	652.000000	37.000000	
75%	7500.25000	1.575323e+07	718.000000	44.000000	
max	10000 10000.00000 10000	1.581569e+07	850.000000	92.000000	
coun mean std min 25% 50% 75% max		10000.0000 1.5302 2.0.5816 1.0000 1.0000 2.0000	1000 10000.00000 200 0.70550 554 0.45584 000 0.00000 000 0.00000 000 1.00000 000 1.00000	10000.000000 0.515100 4 0.499797 0 0.000000 0 0.000000 1.000000 0 1.000000	\
coun mean std min 25%		000 10000.000 381 0.203 318 0.402 000 0.000	3700 2769 0000		

0.000000

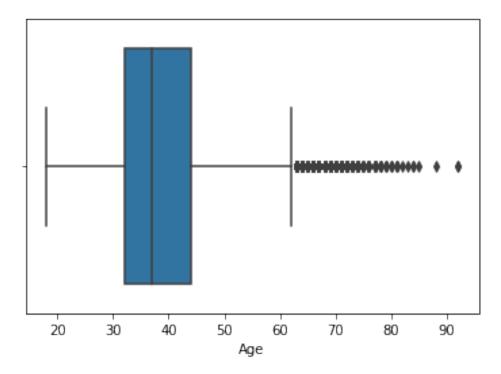
1.000000

```
#to see outliers clearly
sns.boxplot(df['Age'])
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

<matplotlib.axes._subplots.AxesSubplot at 0x7f9eafe7a850>



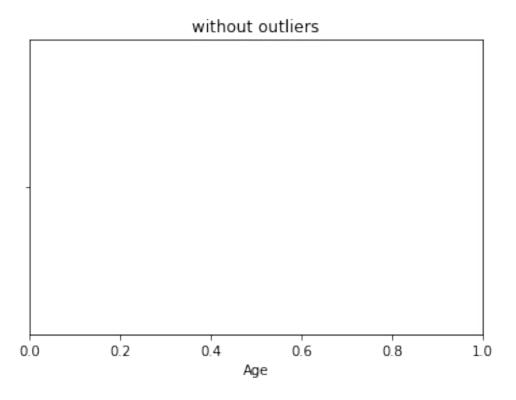
```
#z-score method
upper_limit=df['Age'].mean()+3*df['Age'].std()
lower_limit=df['Age'].mean()-3*df['Age'].std()
print('upper limit', upper_limit)
print('lower limit', lower_limit)
upper limit 70.38521935511383
lower limit 7.458380644886169

df_new=df[(df.Age<=7.458380644886169)&(df.NumOfProducts>=4)]
df_new.shape
(0, 14)
sns.boxplot(df_new['Age']).set_title('without outliers')
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
```

version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

Text(0.5, 1.0, 'without outliers')



#check for categorical columns and perform encoding

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):

		· · · · · · · · · · · · · · · · ·	
#	Column	Non-Null Count	Dtype
0	RowNumber	10000 non-null	int64
1	CustomerId	10000 non-null	int64
2	Surname	10000 non-null	object
3	CreditScore	10000 non-null	int64
4	Geography	10000 non-null	object
5	Gender	10000 non-null	object
6	Age	10000 non-null	int64
7	Tenure	10000 non-null	int64
8	Balance	10000 non-null	float64
9	NumOfProducts	10000 non-null	int64
10	HasCrCard	10000 non-null	int64
11	IsActiveMember	10000 non-null	int64

```
12
    EstimatedSalary
                      10000 non-null
                                       float64
 13
     Exited
                      10000 non-null
                                      int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
df.isnull().sum()
RowNumber
                   0
CustomerId
                   0
Surname
                   0
CreditScore
                   0
Geography
                   0
Gender
                   0
Age
                   0
Tenure
                   0
Balance
                   0
NumOfProducts
                   0
HasCrCard
                   0
IsActiveMember
                   0
EstimatedSalary
                   0
Exited
                   0
dtype: int64
df['Geography'].value counts()
France
           5014
           2509
Germany
Spain
           2477
Name: Geography, dtype: int64
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Geography'] = le.fit_transform(df['Geography'])
df.head()
   RowNumber CustomerId
                           Surname
                                     CreditScore Geography
                                                             Gender
Age \
0
           1
                15634602
                          Hargrave
                                             619
                                                           0 Female
42
           2
                               Hill
1
                15647311
                                             608
                                                           2 Female
41
2
           3
                15619304
                               Onio
                                             502
                                                           0 Female
42
3
           4
                15701354
                               Boni
                                             699
                                                             Female
39
           5
                15737888
                          Mitchell
                                             850
                                                           2 Female
4
43
                      NumOfProducts HasCrCard
   Tenure
             Balance
                                                 IsActiveMember
0
                0.00
```

```
83807.86
1
          1
                                                       0
2
          8
                                         3
                                                       1
             159660.80
                                         2
3
                   0.00
                                                       0
          1
4
          2 125510.82
                                                       1
   EstimatedSalary Exited
0
           101348.88
           112542.58
                              0
1
2
                              1
           113931.57
3
            93826.63
                              0
            79084.10
                              0
#split the data into dependent and independent variables
x = df.iloc[:,0:7].values
y = df.iloc[:,7:8].values
array([[1, 15634602, 'Hargrave', ..., 0, 'Female', 42],
        [2, 15647311, 'Hill', ..., 2, 'Female', 41], [3, 15619304, 'Onio', ..., 0, 'Female', 42],
        [9998, 15584532, 'Liu', ..., 0, 'Female', 36], [9999, 15682355, 'Sabbatini', ..., 1, 'Male', 42],
        [10000, 15628319, 'Walker', ..., 0, 'Female', 28]],
dtype=object)
У
array([[2],
         [1],
         [8],
         . . . ,
         [7],
        [3],
        [4]])
x.shape
(10000, 7)
print(type(x))
<class 'numpy.ndarray'>
y.shape
(10000, 1)
print(type(y))
<class 'numpy.ndarray'>
```

1

0

0

1

```
#split the data into testing and training
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test =
train_test_split(x,y,test_size=0.30,random_state=0)
x_train.shape
(7000, 7)
x_test.shape
(3000, 7)
y_train.shape
(7000, 1)
print(y_train.shape)
(7000, 1)
print(y_test.shape)
(3000, 1)
```