

Assignment -2

Data Visualization & Pre-processing

Assignment Date	22 September 2022
Student Name	Miss.Madhumitha N
Student Roll Number	620119106049
Maximum Marks	2 Marks

Question-1:

1.Dataset downloaded as "model.csv"

2.Load the dataset

#importing libraries

import pandas as pd

#load the dataset

df=pd.read_csv("model.csv")

df

Out[7]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

10000 rows × 14 columns

3. Perform Below Visualizations

3.1 Univariate Analysis

#scatterplot

import matplotlib.pyplot as plt

import pandas as pd

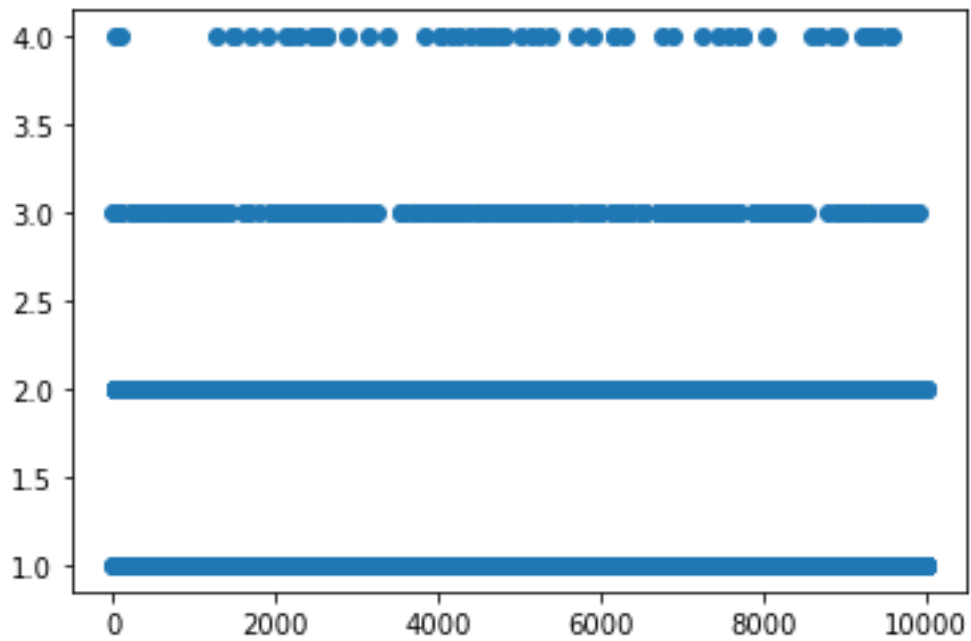
import seaborn as sns

#load the dataset

df=pd.read_csv("model.csv")

plt.scatter(df.index,df['NumOfProducts'])

plt.show()



```
#strip plot
sns.stripplot(y=df['NumOfProducts'])
<AxesSubplot:ylabel='NumOfProducts'>
```

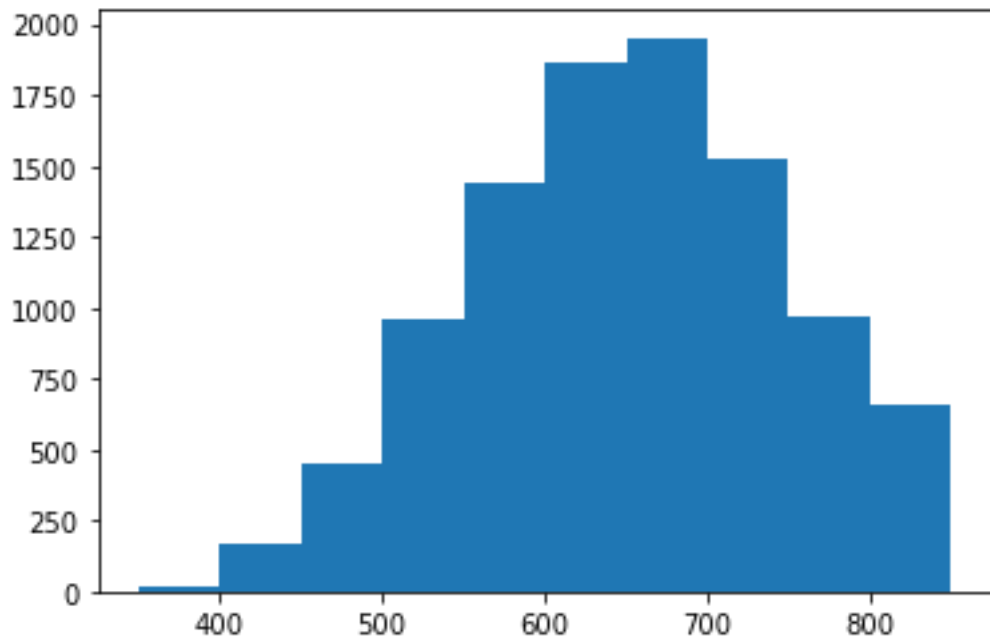
Out[7]



```
#histogram
plt.hist(df['CreditScore'])
```

Out[8]:

```
(array([ 19., 166., 447., 958., 1444., 1866., 1952., 1525., 968.,
        655.]),
 array([350., 400., 450., 500., 550., 600., 650., 700., 750., 800., 850.]),
 <BarContainer object of 10 artists>)
```



```
#boxplot
```

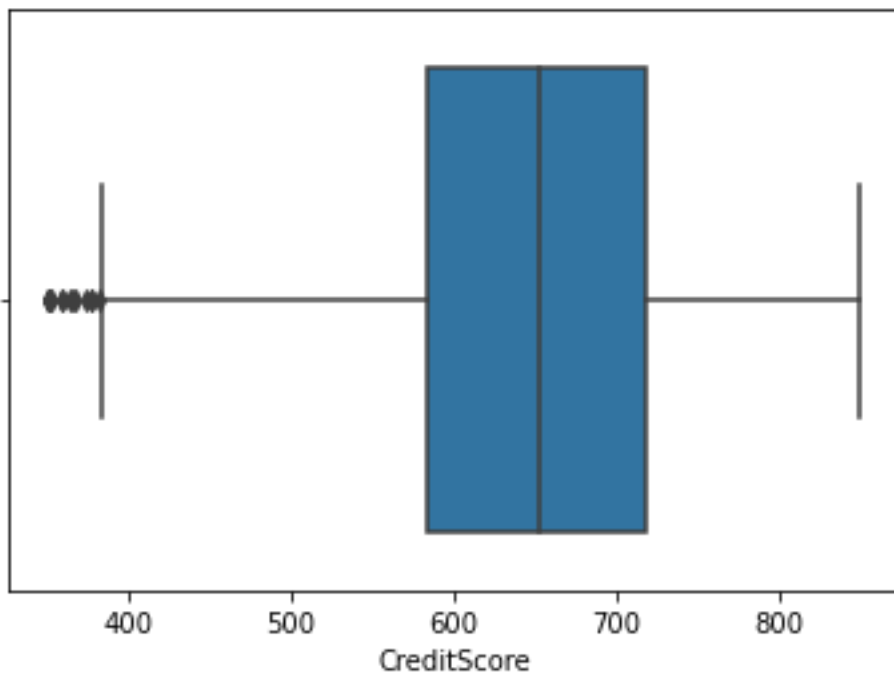
```
sns.boxplot(df['CreditScore'])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: 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.

```
warnings.warn(
```

```
Out[10]:
```

```
<AxesSubplot:xlabel='CreditScore'>
```



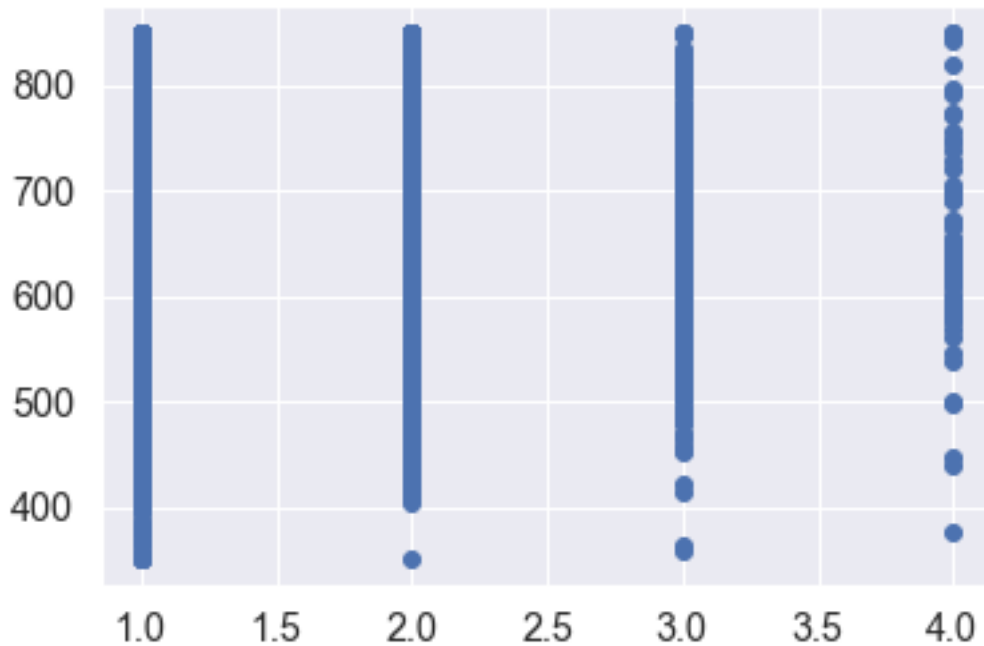
3.2 Bivariate Analysis

In [21]:

```
#scatter plot
```

```
plt.scatter(df.NumOfProducts,df.CreditScore)
```

```
plt.show()
```



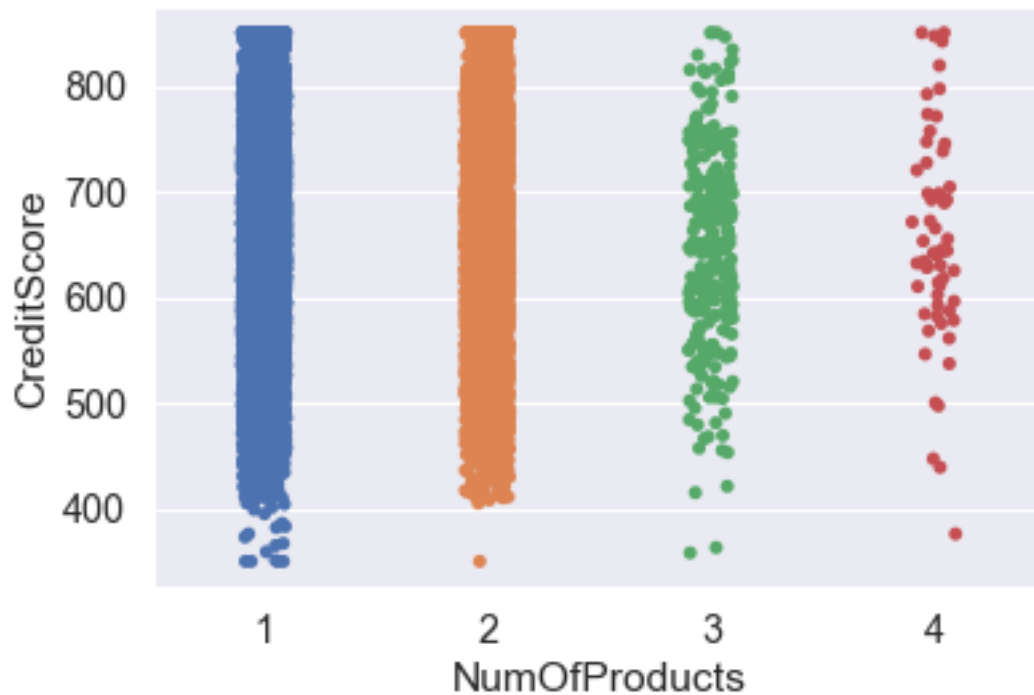
In [22]:

```
#strip plot
```

```
sns.stripplot(x=df['NumOfProducts'],y=df['CreditScore'])
```

Out[22]:

```
<AxesSubplot:xlabel='NumOfProducts', ylabel='CreditScore'>
```



3.3 Multivariate Analysis

In [12]:

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

```
sns.set_style('darkgrid')
sns.set(font_scale=1.3)
```

```
df=pd.read_csv('model.csv')
df
```

```
Out[12]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

10000 rows x 14 columns

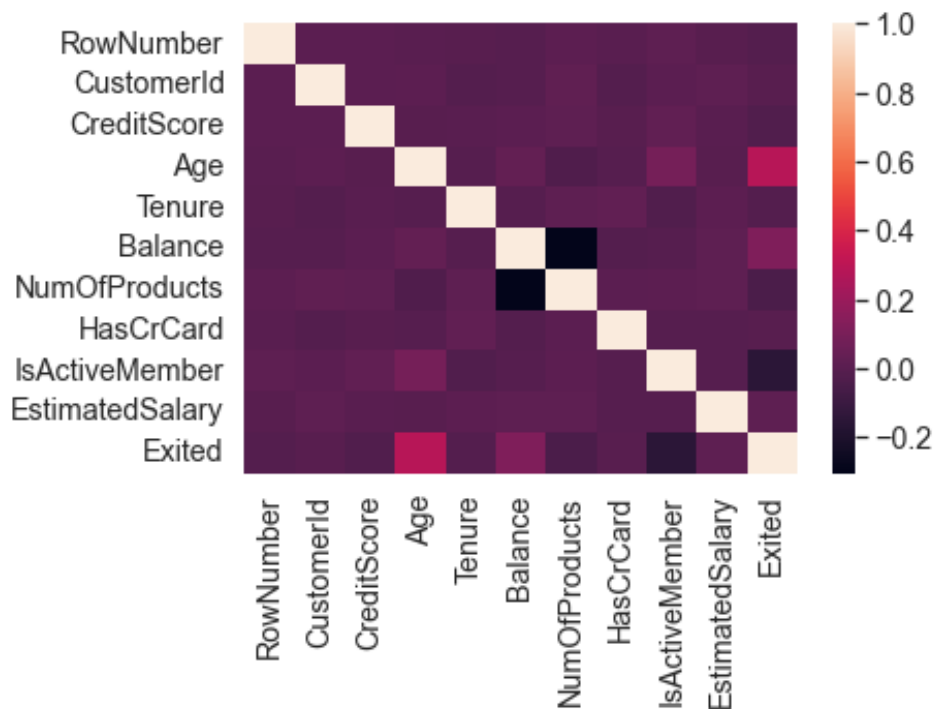
```
#pairplot
sns.pairplot(df);
```



```
sns.heatmap(df.corr())
```

```
<AxesSubplot:>
```

```
Out[14]:
```



4. Perform descriptive statistics on the dataset.

#load the dataset

import pandas as pd

data=pd.read_csv("model.csv")

data.head()

8	9	10/16/2005	HE	501	France	Male	44	4	14205.10/	2	0	1	74940.50	0
9	10	15502389	H7	604	France	Male	27	2	134003.88	1	1	1	71725.73	0

In [6]:	data.tail()
---------	-------------

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
9995	9995	15500229	Obijaku	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabatini	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	France	Female	28	4	130142.79	1	1	0	38190.78	0

In [7]:	data.tail(10)
---------	---------------

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
9990	9991	15798954	Nikamalcom	Germany	Male	33	3	35016.60	1	1	0	53667.08	0
9991	9992	15769959	Ajuluohukwu	France	Female	53	4	88361.21	1	1	0	69364.71	1
9992	9993	15657105	Chukwuka	Spain	Male	36	2	0.00	1	1	0	195192.40	0
9993	9994	15569266	Rahman	France	Male	28	7	155060.41	1	1	0	29179.52	0
9994	9995	15719294	Wood	France	Female	29	2	0.00	2	0	0	167773.55	0
9995	9996	15800229	Obijaku	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabatini	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	France	Female	28	4	130142.79	1	1	0	38190.78	0

In [8]:	data.info()
---------	-------------

```
Out[4]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

```
In [5]: data.head(10)
```

```
Out[5]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
5	6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1
6	7	15592531	Bartlett	822	France	Male	50	7	0.00	2	1	1	10062.80	0
7	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
8	9	15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.50	0

```
<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

In [9]: data.shape

Out[9]: (10000, 14)

In [10]: data.mean()

C:\Users\janar vijay\AppData\Local\Temp\ipykernel_5088\533903386.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
  data.mean()

Out[10]: RowNumber      5.000500e+03
CustomerId    1.569094e+07
CreditScore   6.505288e+02
Age           3.852180e+01
Tenure        5.012800e+00
Balance       7.648593e+04
NumOfProducts 1.536200e+00
HasCrCard     7.055000e-01
IsActiveMember 5.151000e-01
```

```
C:\Users\janar vijay\AppData\Local\Temp\ipykernel_5088\4184645713.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
  data.median()

Out[11]: RowNumber      5.000500e+03
CustomerId    1.569094e+07
CreditScore   6.520000e+02
Age           3.700000e+01
Tenure        5.000000e+00
Balance       9.721854e+04
NumOfProducts 1.000000e+00
HasCrCard     1.000000e+00
IsActiveMember 1.000000e+00
EstimatedSalary 1.001339e+05
Exited        0.000000e+00
dtype: float64

In [12]: data.mode()

Out[12]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15565701	Smith	850.0	France	Male	37.0	2.0	0.0	1.0	1.0	1.0	24924.92	0.0
1	2	15565705	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	3	15565714	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	4	15565779	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	5	15565795	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...
9995	9996	15815628	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9996	9997	15815645	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9997	9998	15815656	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9998	9999	15815660	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9999	10000	15815690	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

10000 rows x 14 columns

we can see the wealthier passengers in the higher classes tend to be older, which makes sense average age

values to impute based on Pclass for Age

```
In [29]: def impute_age(cols):
        Age=cols[0]
        Pclass=cols[1]

        if pd.isnull(Age):

            if Pclass==1:
                return 37

            elif Pclass ==2:
                return 29

            else:
                return 24
        else:
            return Age
```

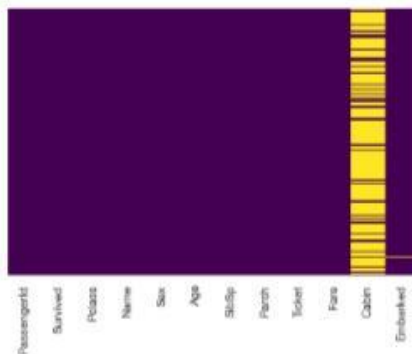
Now Apply This Function

```
In [30]: train['Age'] = train[['Age', 'Pclass']].apply(impute_age,axis=1)
```

Now let's check Heatmap Again..

```
In [31]: sns.heatmap(train.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[31]: <AxesSubplot:>



Now The Age Missing Values Can be Handled.

6. Find the outliers and replace the outliers

```
In [16]: #import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

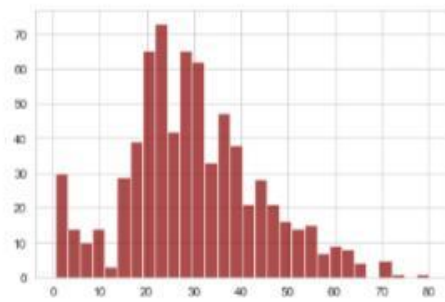
#load the dataset
df=pd.read_csv('model.csv')
df
```

```
Out[16]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	
...
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	2	1	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.71	2	1	
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	

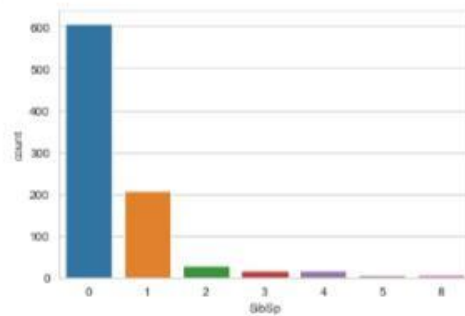
10000 rows x 14 columns

Out[18]: <AxesSubplot:>



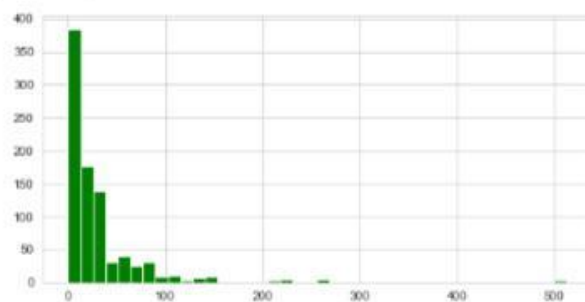
In [19]: `sns.countplot(x='SibSp', data=train)`

Out[19]: <AxesSubplot: xlabel='SibSp', ylabel='count'>



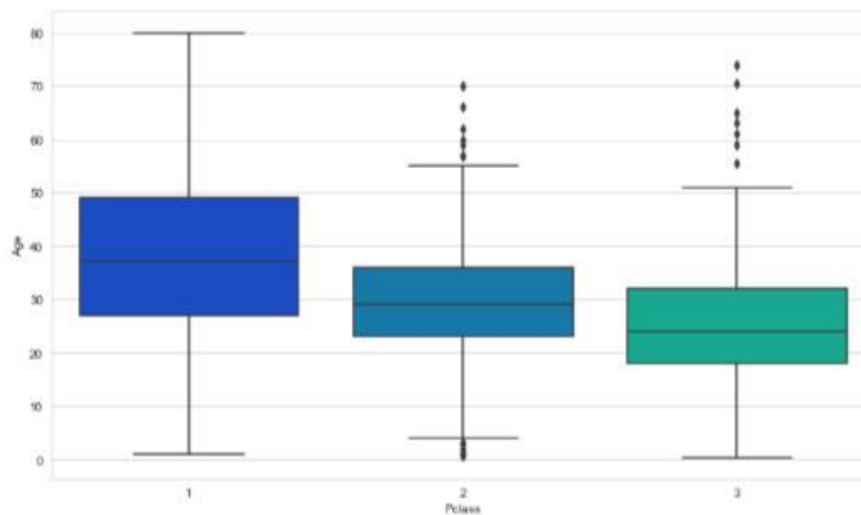
In [20]: `train['Fare'].hist(color='green', bins=40, figsize=(8,4))`

Out[20]: <AxesSubplot:>



In [21]: `#Data cleaning
plt.figure(figsize=(12,7))
sns.boxplot(x='Pclass', y='Age', data=train, palette='winter')`

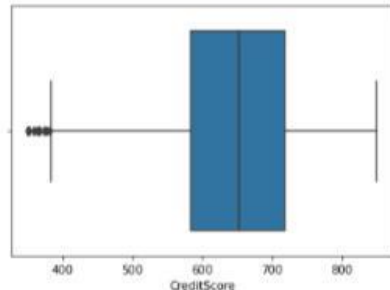
Out[21]: <AxesSubplot: xlabel='Pclass', ylabel='Age'>



```
In [11]: #plotting outliers
sns.boxplot(df["CreditScore"])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: 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.

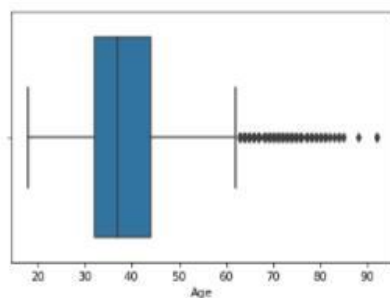
```
Out[11]: <AxesSubplot:xlabel='CreditScore'>
```



```
In [39]: sns.boxplot(df["Age"])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: 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.

```
Out[39]: <AxesSubplot:xlabel='Age'>
```



```
In [12]: qnt=df.quantile(q=[0.75,0.25])
qnt
```

```
Out[12]:
```

	RowNumber	CustomerId	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary
0.75	7500.25	15753237.75	718.0	44.0	7.0	127644.24	2.0	1.0	1.0	149388.2475
0.25	2500.75	15628928.25	584.0	32.0	3.0	0.00	1.0	0.0	0.0	91002.1100

```
In [14]: iqr = qnt.loc[0.75]-qnt.loc[0.25] #iqr calculations
iqr
```

```
Out[14]:
```

RowNumber	4999.5000
CustomerId	124705.5000
CreditScore	134.0000
Age	12.0000
Tenure	4.0000
Balance	127644.2400
NumOfProducts	1.0000
HasCrCard	1.0000
IsActiveMember	1.0000
EstimatedSalary	98386.1375
Exited	0.0000
dtype:	float64

```
In [26]: #lower extreme values
lower=qnt.loc[0.25] - 1.5*iqr
lower
```

```
Out[26]:
```

RowNumber	-4.998500e+03
CustomerId	1.544147e+07
CreditScore	3.830000e+02
Age	1.400000e+01
Tenure	-3.000000e+00
Balance	-1.914664e+05
NumOfProducts	-5.000000e-01
HasCrCard	-1.500000e+00
IsActiveMember	-1.500000e+00
EstimatedSalary	-9.657710e+04
Exited	0.000000e+00
dtype:	float64

```
In [27]: #upper extreme values
upper=qnt.loc[0.75] + 1.5*iqr
upper
```

```
Out[27]: RowNumber      1.499950e+04
CustomerId    1.594029e+07
CreditScore   9.190000e+02
Age           6.200000e+01
Tenure        1.300000e+01
Balance       3.191106e+05
NumOfProducts 3.500000e+00
HasCrCard     2.500000e+00
IsActiveMember 2.500000e+00
EstimatedSalary 2.969675e+05
Exited        0.000000e+00
dtype: float64
```

```
In [18]: df.mean()
```

C:\Users\janar vijay\AppData\Local\Temp\ipykernel_10016\3698961737.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.

```
df.mean()
Out[18]: RowNumber      5.000500e+03
CustomerId    1.569094e+07
CreditScore   6.505288e+02
Age           3.892180e+01
Tenure        5.012800e+00
Balance       7.648589e+04
NumOfProducts 1.530200e+00
HasCrCard     7.055000e-01
IsActiveMember 5.151000e-01
EstimatedSalary 1.000902e+05
Exited        2.037000e-01
dtype: float64
```

Replacing outlier

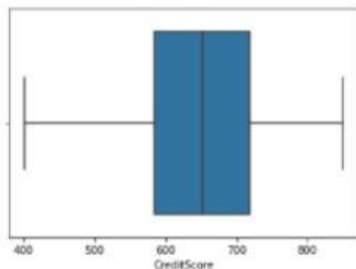
```
In [45]: #import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#load the dataset
df=pd.read_csv('model.csv')
df['CreditScore']=np.where(df['CreditScore']<400,402,df['CreditScore'])
```

```
In [49]: #remove outlier on the CreditScore column
sns.boxplot(df["CreditScore"])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: 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.

```
warnings.warn(
Out[49]: <AxesSubplot:xlabel='CreditScore'>
```

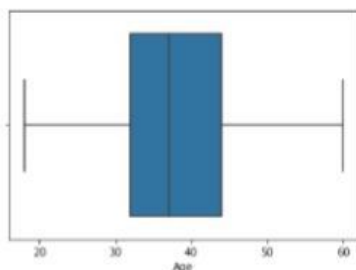


```
In [50]: #remove outlier on the Age column
df['Age']=np.where(df['Age']>60,50,df['Age'])
```

```
In [51]: sns.boxplot(df["Age"])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: 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.

```
warnings.warn(
Out[51]: <AxesSubplot:xlabel='Age'>
```



7. Check for Categorical columns and perform encoding

```
In [53]: df.head()
```

```
Out[53]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	1	15674602	Hargrave	619	France	Female	42	2	0.00	1	1	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1

encoding

```
In [57]: #manually handling categorincal data
df['Gender'].replace({'Female':1,'Male':2},inplace=True)
df.head()
```

```
Out[57]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	1	15674602	Hargrave	619	France	1	42	2	0.00	1	1	1
1	2	15647311	Hill	608	Spain	1	41	1	83807.86	1	0	0
2	3	15619304	Onio	502	France	1	42	8	159660.80	3	1	1
3	4	15701354	Boni	699	France	1	39	1	0.00	2	0	0
4	5	15737888	Mitchell	850	Spain	1	43	2	125510.82	1	1	1

```
In [62]: df['Geography'].replace({'France':100,'Spain':200},inplace=True)
df.head()
```

```
Out[62]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	1	15674602	Hargrave	619	100	1	42	2	0.00	1	1	1
1	2	15647311	Hill	608	200	1	41	1	83807.86	1	0	0
2	3	15619304	Onio	502	100	1	42	8	159660.80	3	1	1
3	4	15701354	Boni	699	100	1	39	1	0.00	2	0	0
4	5	15737888	Mitchell	850	200	1	43	2	125510.82	1	1	1

```
In [60]: #dummy variable function
df_main=pd.get_dummies(df,columns=['Geography'])
df_main
```

```
Out[60]:
```

	RowNumber	CustomerId	Surname	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	1	15674602	Hargrave	619	1	42	2	0.00	1	1	1
1	2	15647311	Hill	608	1	41	1	83807.86	1	0	1
2	3	15619304	Onio	502	1	42	8	159660.80	3	1	0
3	4	15701354	Boni	699	1	39	1	0.00	2	0	0
4	5	15737888	Mitchell	850	1	43	2	125510.82	1	1	1
...
9995	9996	15606229	Obijaku	731	2	39	5	0.00	2	1	0
9996	9997	15569892	Johnstone	516	2	35	10	57369.61	1	1	1
9997	9998	15584532	Liu	709	1	36	7	0.00	1	0	1
9998	9999	15682355	Sabbatini	732	2	42	3	75075.31	2	1	0
9999	10000	15628319	Walker	792	1	28	4	130142.79	1	1	0

10000 rows x 16 columns

8. Split the data into dependent and independent variables.

```
In [64]: #target variable or dependent variable.
import pandas as pd
df=pd.read_csv('model.csv')
y=df['EstimatedSalary']
y.head()
```

```
Out[64]:
```

```
0    101348.88
1    112542.58
2    113931.57
3    93826.63
4    79084.10
Name: EstimatedSalary, dtype: float64
```

```
In [74]: #independent variables
x = df.drop(columns=['EstimatedSalary'],axis=1)
x.head()
```

```
Out[74]:
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActive
0	1	15674602	Hargrave	619	France	Female	42	2	0.00	1	1	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1

