1.Importing the Reqiured Package

```
import pandas as pd
import seaborn as sns
import numpy as np
from matplotlib import pyplot as pyplot
%matplotlib inline
```

2.Loading the Dataset

```
import pandas as pd
```

```
df = pd.read_csv("/content/Churn_Modelling.csv")
df
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProc
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	

3.Visualization

3.1 Univariate Analysis

9995 9996 15606229 Obijiaku 771 France Male 39 5 0.00

```
import seaborn as sns
import pandas as pd
```

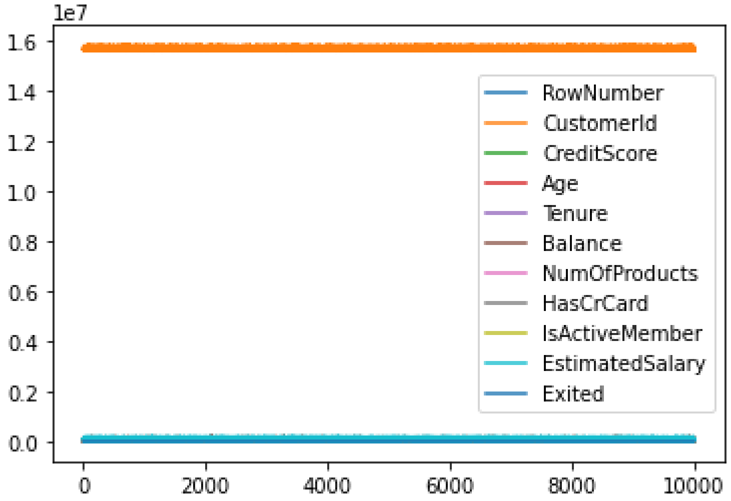
```
sns.displot(df.Gender)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fbce8c90810>
```

3.2 Bi-Variate Analysis

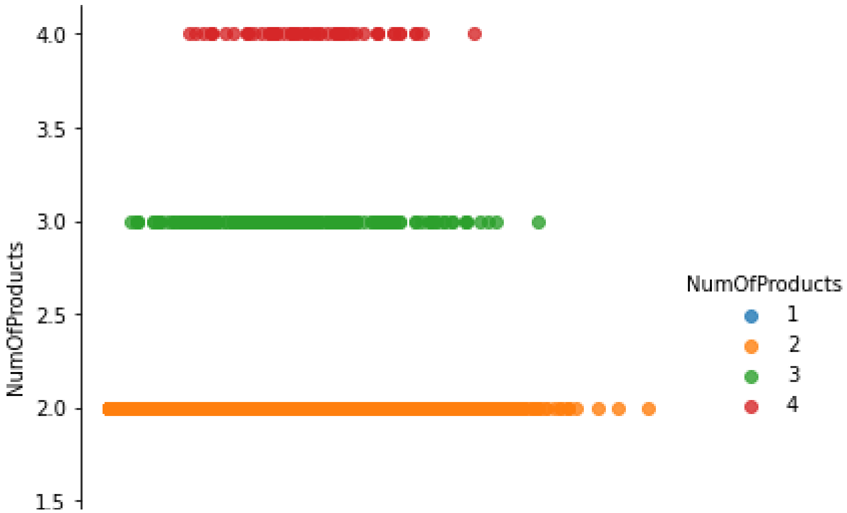
```
df.plot.line()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbce5595210>
```



3.3 Multi-Variate Analysis

```
sns.lmplot("Age","NumOfProducts",df,hue="NumOfProducts",fit_reg=False);
```



4.Perform description statics on the dataset

```
df.describe()
```

```
          RowNumber   CustomerId   CreditScore           Age        Tenure        Balance  NumOfProducts

count   10000.00000  1.000000e+04  10000.000000  10000.000000  10000.000000   10000.000000   10000.000000
mean     5000.50000  1.569094e+07    650.528800     38.921800      5.012800   76485.889288       1.530200
std      2886.89568  7.193619e+04     96.653299     10.487806      2.892174   62397.405202       0.581654
min         1.00000  1.556570e+07    350.000000     18.000000      0.000000       0.000000       1.000000
25%      2500.75000  1.562853e+07    584.000000     32.000000      3.000000       0.000000       1.000000
50%      5000.50000  1.569074e+07    652.000000     37.000000      5.000000   97198.540000       1.000000
75%      7500.25000  1.575323e+07    718.000000     44.000000      7.000000  127644.240000       2.000000
max     10000.00000  1.581569e+07    850.000000     92.000000     10.000000  250898.090000       4.000000
```

5.Handle the Missing values

```
data=pd.read_csv("Churn_Modelling.csv")
pd.isnull(data["Gender"])
```

```
0       False
1       False
2       False
3       False
4       False
        ...
9995    False
9996    False
9997    False
9998    False
9999    False
Name: Gender, Length: 10000, dtype: bool
```

6.Find the outliners and replace the outliners

```
import numpy as np
```

```
df["Tenure"] = np.where(df["Tenure"] >10,np.median,df['Tenure'])
df["Tenure"]
```

```
0        2
1        1
2        8
3        1
4        2
        ..
9995     5
```

```
9996    10
9997     7
9998     3
9999     4
Name: Tenure, Length: 10000, dtype: object
```

7.Check for Categorical columns and perform encoding

```
pd.get_dummies(df,columns =["Gender","Age"],prefix=["Age","Gender"]).head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Tenure	Balance	NumOfProducts	HasCrCard
0	1	15634602	Hargrave	619	France	2	0.00	1	1
1	2	15647311	Hill	608	Spain	1	83807.86	1	0
2	3	15619304	Onio	502	France	8	159660.80	3	1
3	4	15701354	Boni	699	France	1	0.00	2	0
4	5	15737888	Mitchell	850	Spain	2	125510.82	1	1

5 rows × 84 columns

8.Split the data into dependent and independent variables

8.1.Split the data into Independent variables

```
#independant
```

```
x = df.iloc[:,:-1].values
print(x)
```

```
[[1 15634602 'Hargrave' ... 1 1 101348.88]
 [2 15647311 'Hill' ... 0 1 112542.58]
 [3 15619304 'Onio' ... 1 0 113931.57]
 ...
 [9998 15584532 'Liu' ... 0 1 42085.58]
 [9999 15682355 'Sabbatini' ... 1 0 92888.52]
 [10000 15628319 'Walker' ... 1 0 38190.78]]
```

8.2.Split the data into Depenedent variables

```
#dependant
y = df.iloc[:,:-1].values
print(y)
```

```
[[1 15634602 'Hargrave' ... 1 1 101348.88]
 [2 15647311 'Hill' ... 0 1 112542.58]
 [3 15619304 'Onio' ... 1 0 113931.57]
 ...
 [9998 15584532 'Liu' ... 0 1 42085.58]
 [9999 15682355 'Sabbatini' ... 1 0 92888.52]
 [10000 15628319 'Walker' ... 1 0 38190.78]]
```

9.Scale the independent variables

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
Scaler = MinMaxScaler()
df[["RowNumber"]] = Scaler.fit_transform(df[["RowNumber"]])
print(x)
```

```
[[1 15634602 'Hargrave' ... 1 1 101348.88]
 [2 15647311 'Hill' ... 0 1 112542.58]
 [3 15619304 'Onio' ... 1 0 113931.57]
 ...
 [9998 15584532 'Liu' ... 0 1 42085.58]
 [9999 15682355 'Sabbatini' ... 1 0 92888.52]
 [10000 15628319 'Walker' ... 1 0 38190.78]]
```

10.Split the data into training and testing

```
from sklearn.model_selection import train_test_split
train_size=0.8
x = df.drop(columns = ['Tenure']).copy()
y = df['Tenure']
x_train,x_rem,y_train,y_rem = train_test_split(x,y,train_size=0.8)
test_size=0.5
x_valid,x_test,y_valid,y_test = train_test_split(x,y,test_size=0.5)
print(x_train.shape),print(y_train.shape)
print(x_valid.shape),print(y_valid.shape)
print(x_test.shape),print(y_test.shape)
```

```
(8000, 13)
(8000,)
(5000, 13)
(5000,)
(5000, 13)
(5000,)
(None, None)
```

Colab paid products - Cancel contracts here

✓ 0s completed at 1:34 AM