

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

```
import warnings
warnings.filterwarnings("ignore")
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Churn_Modelling.csv

- **Churn_Modelling.csv**(text/csv) - 684858 bytes, last modified: 11/9/2022 - 100% done

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

```
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.describe()
```

Churn_Modelling.csv X

1 to 10 of 10000 entries

Filter



RowNumber	CustomerId	Surname	CreditScore
1	15634602	Hargrave	6
2	15647311	Hill	6
3	15619304	Onio	5
4	15701354	Boni	6
5	15737888	Mitchell	8
6	15574012	Chu	6
7	15592531	Bartlett	8
8	15656148	Obinna	3
9	15792365	He	5
10	15592389	H?	6

◀ ▶

Show 10 per page

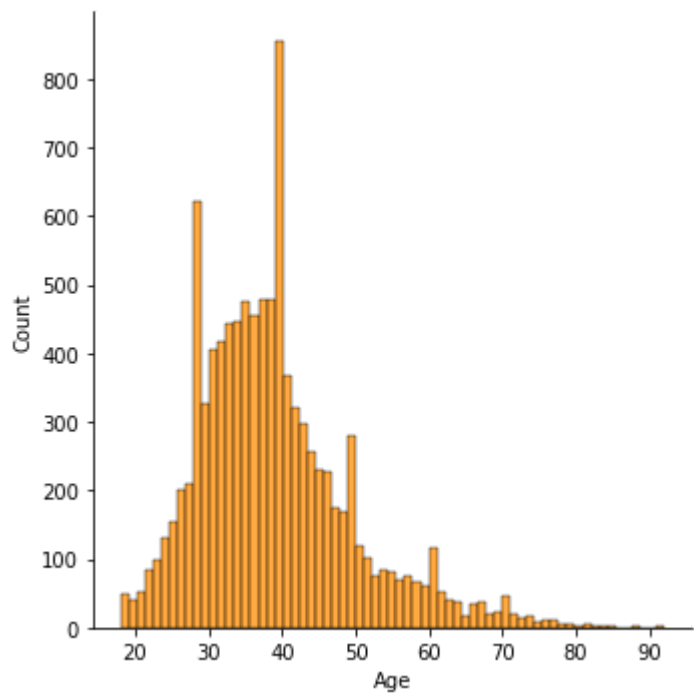
1 2 10 100 900 990

1000

	RowNumber	CustomerId	CreditScore	
count	10000.00000	1.000000e+04	10000.000000	1000
mean	5000.50000	1.569094e+07	650.528800	3
std	2886.89568	7.193619e+04	96.653299	1
min	1.00000	1.556570e+07	350.000000	1
25%	2500.75000	1.562853e+07	584.000000	3
50%	5000.50000	1.569074e+07	652.000000	3
75%	7500.25000	1.575323e+07	718.000000	4
max	10000.00000	1.581569e+07	850.000000	9

```
sns.displot(df["Age"], color='darkorange')
```

<seaborn.axisgrid.FacetGrid at 0x7f2039a8fe10>



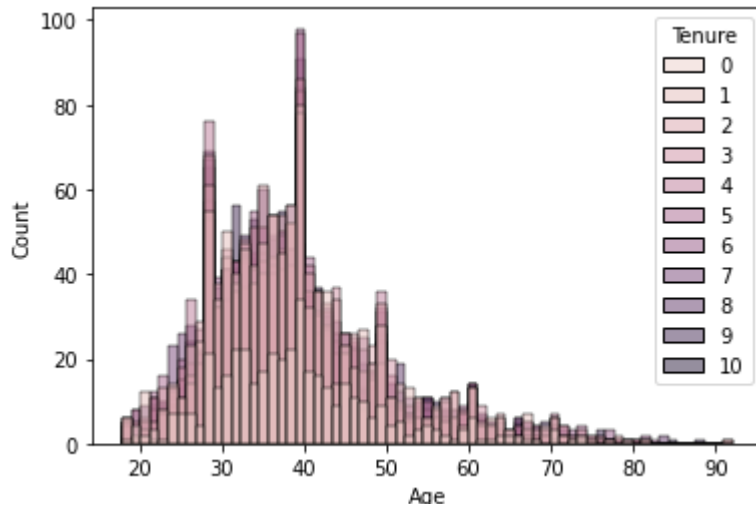
```
sns.histplot(y="Age",data=df,color='darkorange')
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f203939fe90>
```



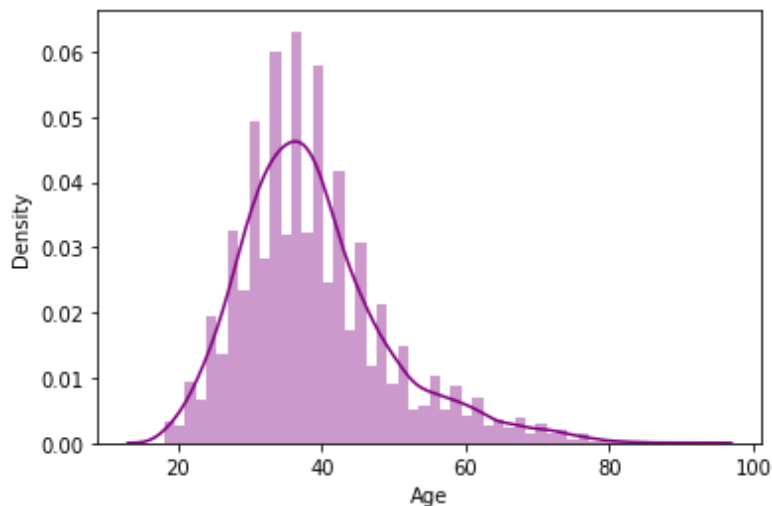
```
sns.histplot(x='Age',data=df,hue=df['Tenure'])
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f203c3ee850>
```



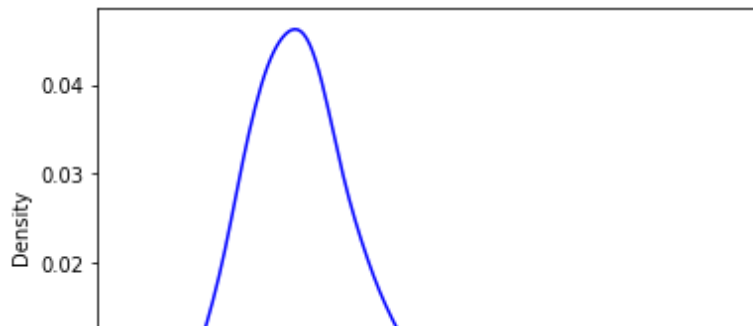
```
sns.distplot(df["Age"],color='purple')
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f2037fbf450>
```



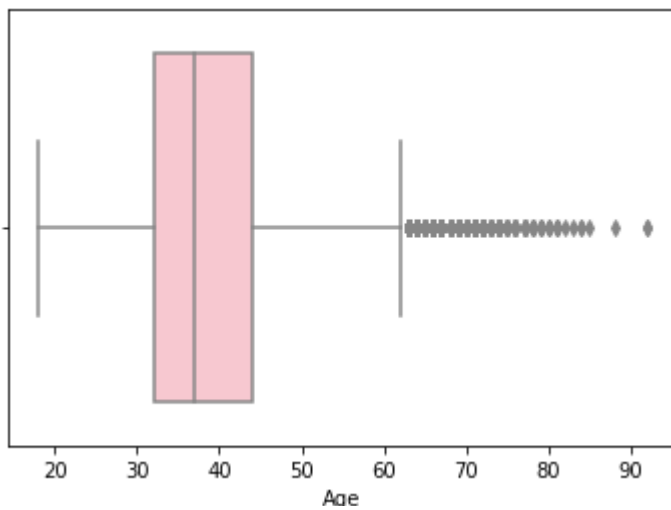
```
sns.distplot(df["Age"],hist=False,color='blue')
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f2037de4dd0>
```



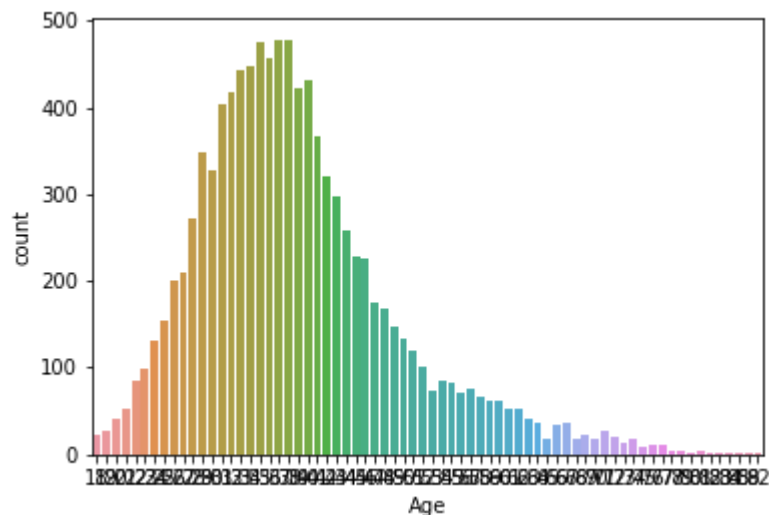
```
sns.boxplot(df["Age"],color='pink')
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f2037d3dfd0>
```



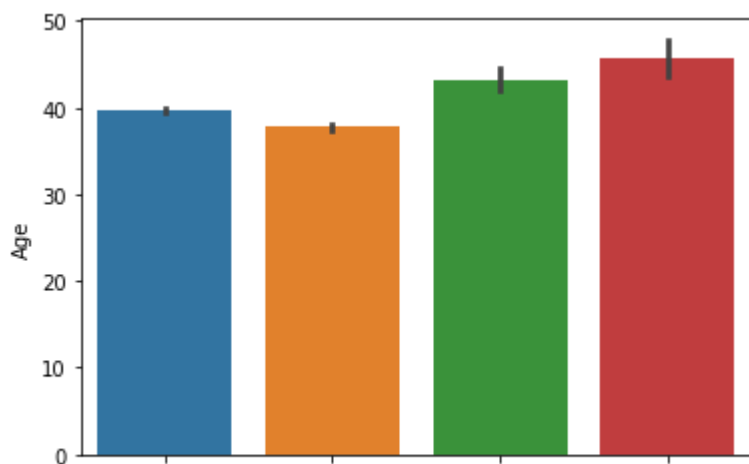
```
sns.countplot(df['Age'])
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f2037a23590>
```



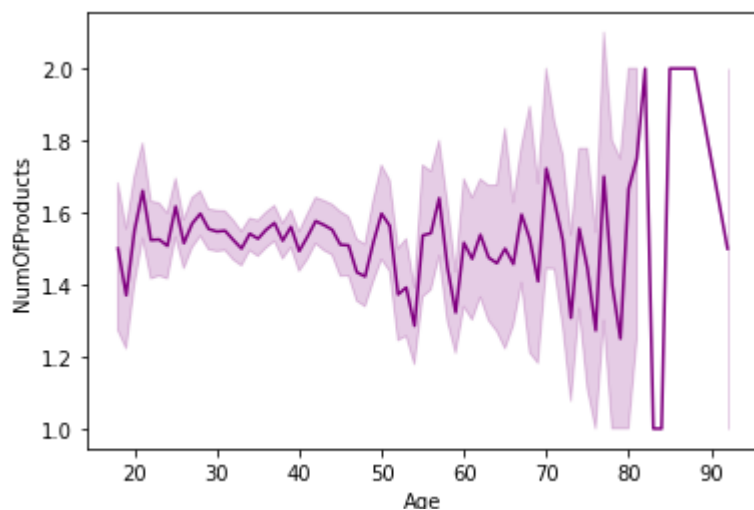
```
sns.barplot(df["NumOfProducts"],df["Age"])
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f20377dafd0>
```



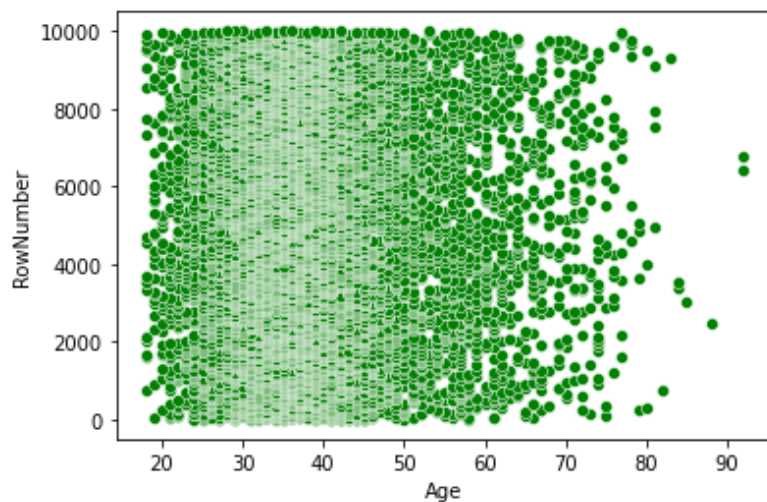
```
sns.lineplot(df["Age"],df["NumOfProducts"], color='purple')
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f20376dd310>
```



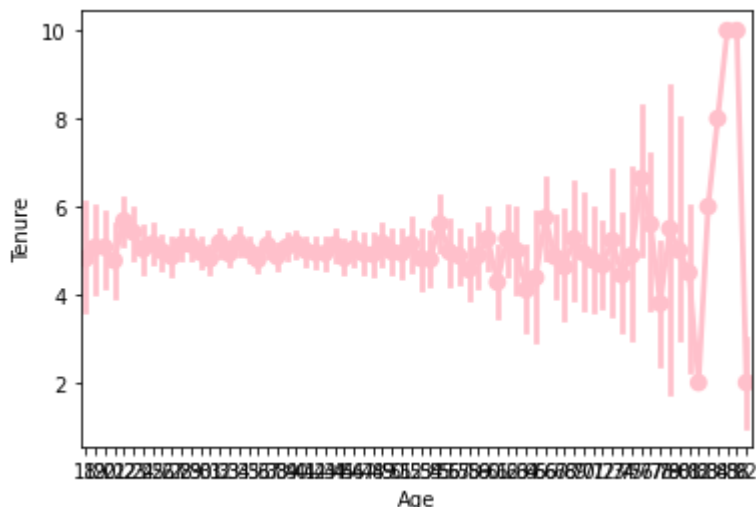
```
sns.scatterplot(x=df.Age,y=df.RowNumber,color='green')
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f20375c7510>
```



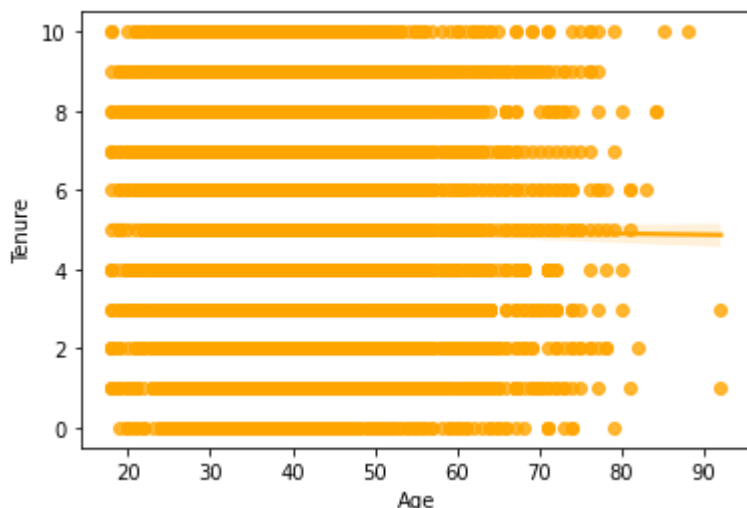
```
sns.pointplot(x='Age',y='Tenure',data=df,color='pink')
```

```
<matplotlib.axes._subplots.AxesSubplot at  
0x7f2035d4eb90>
```



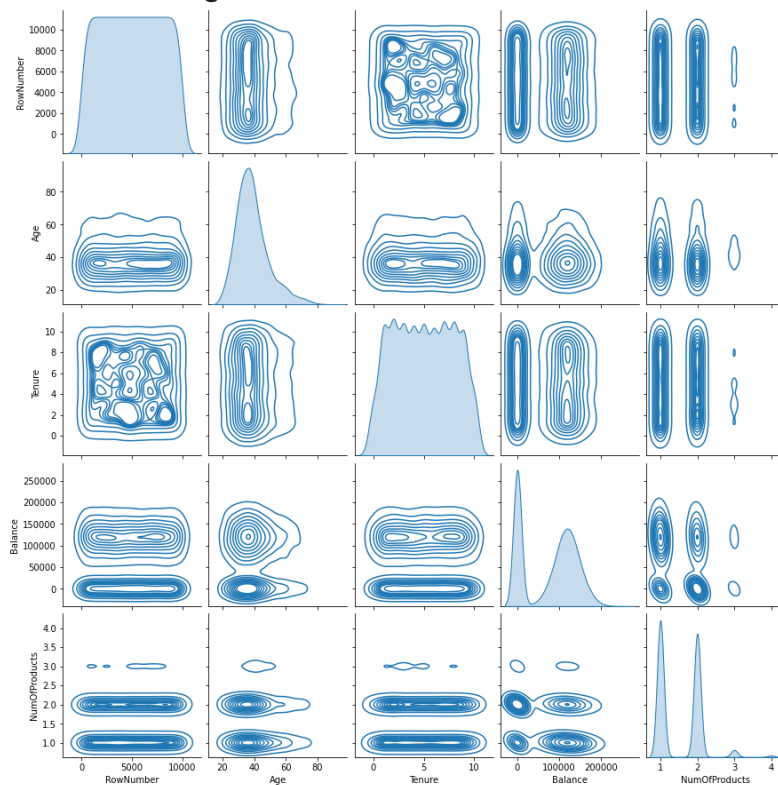
```
sns.regplot(df['Age'],df['Tenure'],color='orange')
```

```
<matplotlib.axes._subplots.AxesSubplot at  
0x7f2033276150>
```



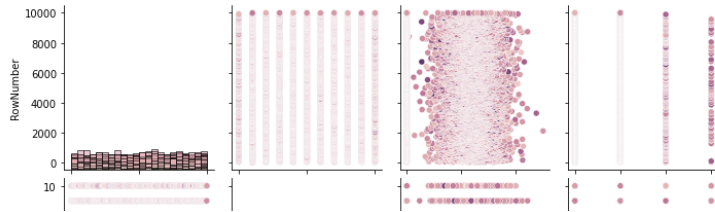
```
sns.pairplot(data=df[["RowNumber","Age","Tenure","Balance"]])
```

<seaborn.axisgrid.PairGrid at 0x7f2032603d10>



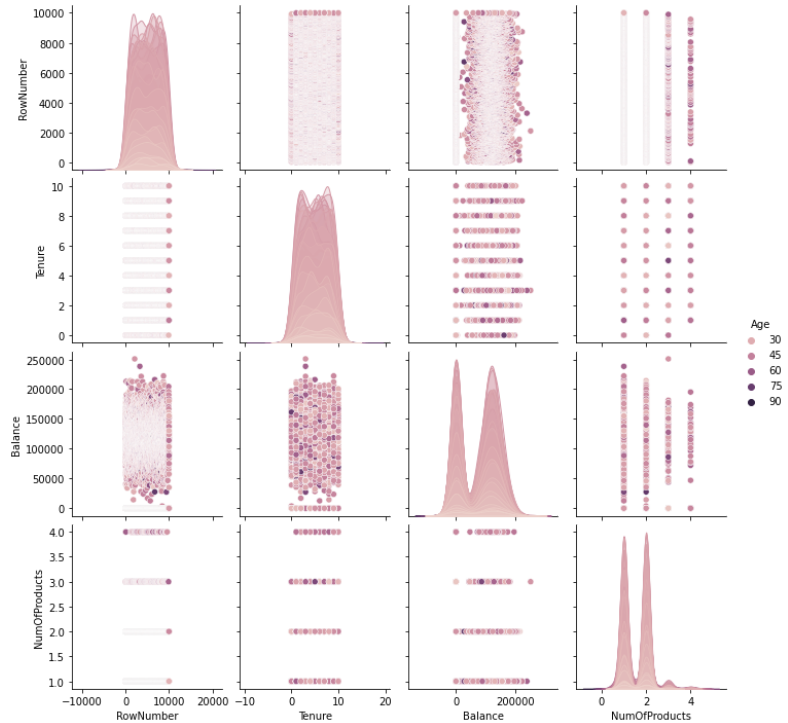
```
sns.pairplot(data=df[["RowNumber", "Age", "Tenure", "Balance", "NumOfProducts"]])
```

```
<seaborn.axisgrid.PairGrid at 0x7f203219ab50>
```



```
sns.pairplot(data=df[["RowNumber", "Age", "Tenure", "Balance"]])
```

```
<seaborn.axisgrid.PairGrid at 0x7f2025efda50>
```



```
df.describe()
```


	RowNumber	CustomerId	CreditScore	
count	10000.00000	1.000000e+04	10000.000000	1000
mean	5000.50000	1.569094e+07	650.528800	3
std	2886.89568	7.193619e+04	96.653299	1
min	1.00000	1.556570e+07	350.000000	1
25%	2500.75000	1.562853e+07	584.000000	3
50%	5000.50000	1.569074e+07	652.000000	3

```
data=pd.DataFrame({"a":[1,2,np.nan],"b":[1,np.nan,np.nan]})
data
```

	a	b	c
0	1.0	1.0	1
1	2.0	NaN	2
2	NaN	NaN	4

```
data.isnull().any()
```

```
a      True
b      True
c     False
dtype: bool
```

```
data.isnull().sum()
```

```
a      1
b      2
c      0
dtype: int64
```

```
data.fillna(value = "S")
```

	a	b	c
0	1.0	1.0	1
1	2.0	S	2
2	S	S	4

```
data["a"].mean()
```

```
1.5
```

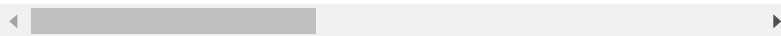
```
data["a"].median()
```

```
1.5
```

```
outlierss=df.quantile(q=(0.25,0.75))
```

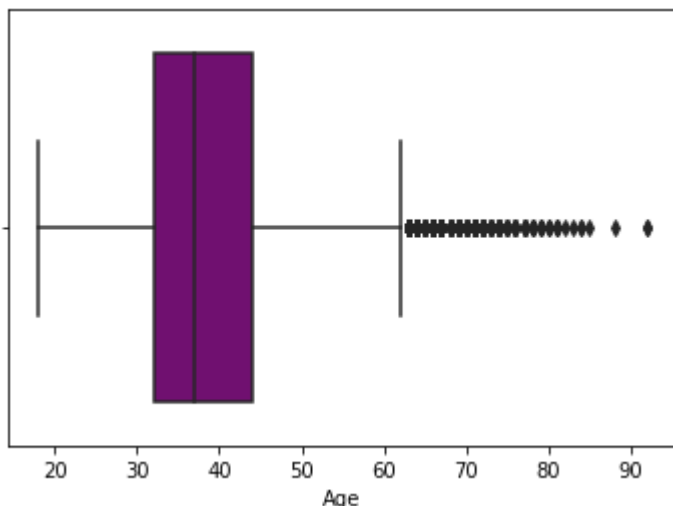
```
outlierss
```

	RowNumber	CustomerId	CreditScore	Age	Ten
0.25	2500.75	15628528.25	584.0	32.0	
0.75	7500.25	15753233.75	718.0	44.0	



```
sns.boxplot(df["Age"],color='purple')
```

```
<matplotlib.axes._subplots.AxesSubplot at  
0x7f20248f4b50>
```



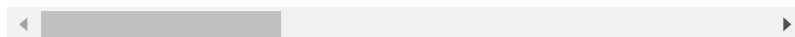
```
sns.boxplot(df["Age"],color='pink')
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f20248454d0>
```



```
df.head(4)
```

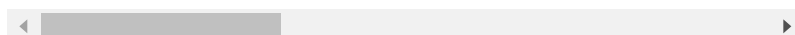
	RowNumber	CustomerId	Surname	CreditScore	Geography
0	1	15634602	Hargrave	619	France
1	2	15647311	Hill	608	Spain
2	3	15619304	Onio	502	Germany
3	4	15701354	Boni	699	France



```
df["Gender"].replace({"Female":0,"Male":1},inplace = True)
df["Geography"].replace({"France":1,"Spain":2,"Germany":3},inplace = True)
df["Gender"].replace({"Female":0,"Male":1},inplace = True)
df["Geography"].replace({"France":1,"Spain":2,"Germany":3},inplace = True)
```

```
df.head(4)
```

	RowNumber	CustomerId	Surname	CreditScore	Geography
0	1	15634602	Hargrave	619	France
1	2	15647311	Hill	608	Spain
2	3	15619304	Onio	502	Germany
3	4	15701354	Boni	699	France

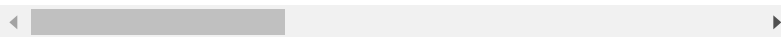


```
y = df["Surname"]
```

```
x=df.drop(columns=["Surname"],axis=1)
```

```
x.head()
```

	RowNumber	CustomerId	CreditScore	Geography	Gender
0	1	15634602	619	France	Male
1	2	15647311	608	France	Female
2	3	15619304	502	France	Female
3	4	15701354	699	France	Female
4	5	15737888	850	France	Female



```
names=x.columns
```

```
names
```

```
Index(['RowNumber', 'CustomerId', 'CreditScore',
      'Geography', 'Gender', 'Age',
      'Tenure', 'Balance', 'NumOfProducts',
      'HasCrCard', 'IsActiveMember',
      'EstimatedSalary', 'Exited'],
      dtype='object')
```

```
from sklearn.preprocessing import scale
```

```
X=scale(x)
```

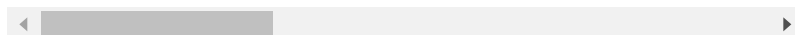
```
X
```

```
array([[ -1.73187761, -0.78321342, -0.32622142, ...,
         0.97024255,
         0.02188649,  1.97716468],
       [ -1.7315312 , -0.60653412, -0.44003595, ...,
         0.97024255,
         0.21653375, -0.50577476],
       [ -1.73118479, -0.99588476, -1.53679418, ...,
        -1.03067011,
         0.2406869 ,  1.97716468],
       ...,
       [  1.73118479, -1.47928179,  0.60498839, ...,
         0.97024255,
        -1.00864308,  1.97716468],
       [  1.7315312 , -0.11935577,  1.25683526, ...,
        -1.03067011,
        -0.12523071,  1.97716468],
       [  1.73187761, -0.87055909,  1.46377078, ...,
        -1.03067011,
        -1.07636976, -0.50577476]])
```

```
x = pd.DataFrame(X,columns = names )
x
```

	RowNumber	CustomerId	CreditScore	Geography
0	-1.731878	-0.783213	-0.326221	-0.902587
1	-1.731531	-0.606534	-0.440036	0.301665
2	-1.731185	-0.995885	-1.536794	-0.902587
3	-1.730838	0.144767	0.501521	-0.902587
4	-1.730492	0.652659	2.063884	0.301665
...
9995	1.730492	-1.177652	1.246488	-0.902587
9996	1.730838	-1.682806	-1.391939	-0.902587
9997	1.731185	-1.479282	0.604988	-0.902587
9998	1.731531	-0.119356	1.256835	1.505917
9999	1.731878	-0.870559	1.463771	-0.902587

10000 rows × 13 columns



```
from sklearn.model_selection import train_test_split
```

```
train,x_test,y_train,y_test=train_test_split(x,y,test_size=
```

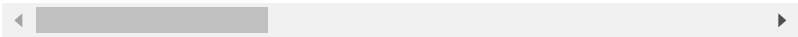
```
x_train.head()
```

RowNumber CustomerId CreditScore Geography

x_train.shape,y_train.shape,x_test.shape,y_test.shape

((8000, 13), (8000,), (2000, 13), (2000,))

2995	-0.694379	-1.507642	-1.195351	-0.902587
5316	0.109639	1.243462	0.035916	0.301665
356	-1.608556	-1.100775	2.063884	0.301665



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 11:02 AM

