In [1]:

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

In [2]:

```
data = pd.read_excel(r'D:\Career\Udemy\DA 2\Finance Data Analysis/Bank.xlsx',1)
data.head()
```

Out[2]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
1	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
2	3	39	15	11	94720	1	1.0	1	0	0	0	0	0	0
3	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
4	5	35	8	45	91330	4	1.0	2	0	0	0	0	0	1

In [3]:

```
data.shape
```

Out[3]:

```
(5000, 14)
```

In [4]:

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

Out[4]:

```
ID                    0
Age                   0
Experience            0
Income                0
ZIP Code              0
Family                0
CCAvg                 0
Education             0
Mortgage              0
Personal Loan         0
Securities Account    0
CD Account            0
Online                0
CreditCard            0
dtype: int64
```

In [5]:

```
data.drop(['ID','ZIP Code'], axis = 1, inplace = True)
```

In [6]:

```
data.columns
```

Out[6]:

```
Index(['Age', 'Experience', 'Income', 'Family', 'CCAvg', 'Education',
       'Mortgage', 'Personal Loan', 'Securities Account', 'CD Account',
       'Online', 'CreditCard'],
      dtype='object')
```

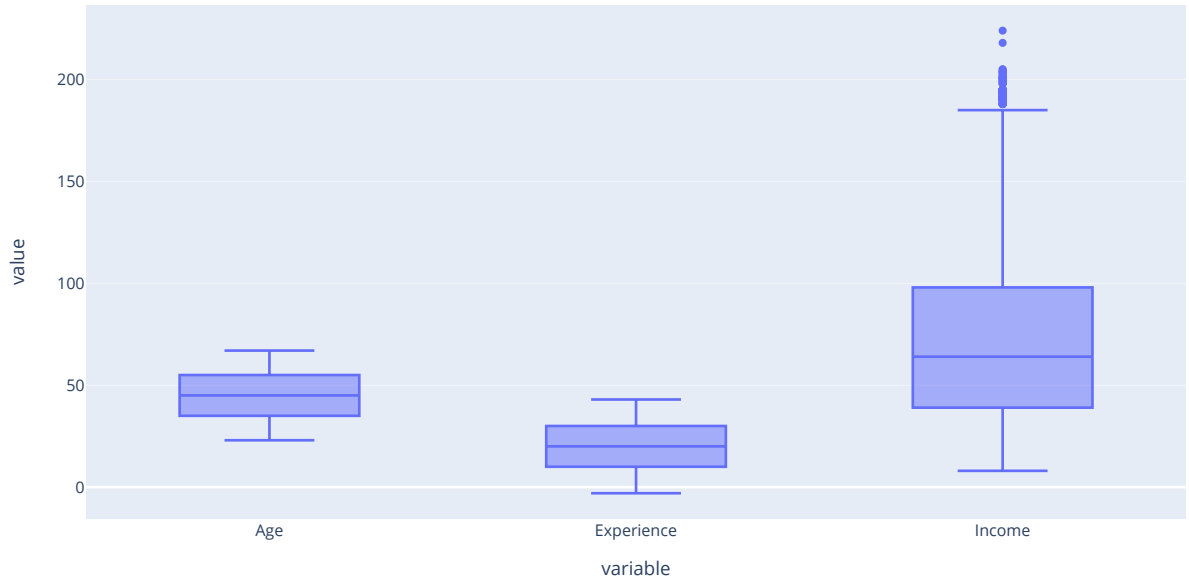
Five point summary concept for data description

In [7]:

```
import plotly.express as px
```

In [8]:

```
px.box(data, y = ['Age','Experience','Income'])
```



Distribution of Data

In [9]:

```
data.skew()
```

Out[9]:

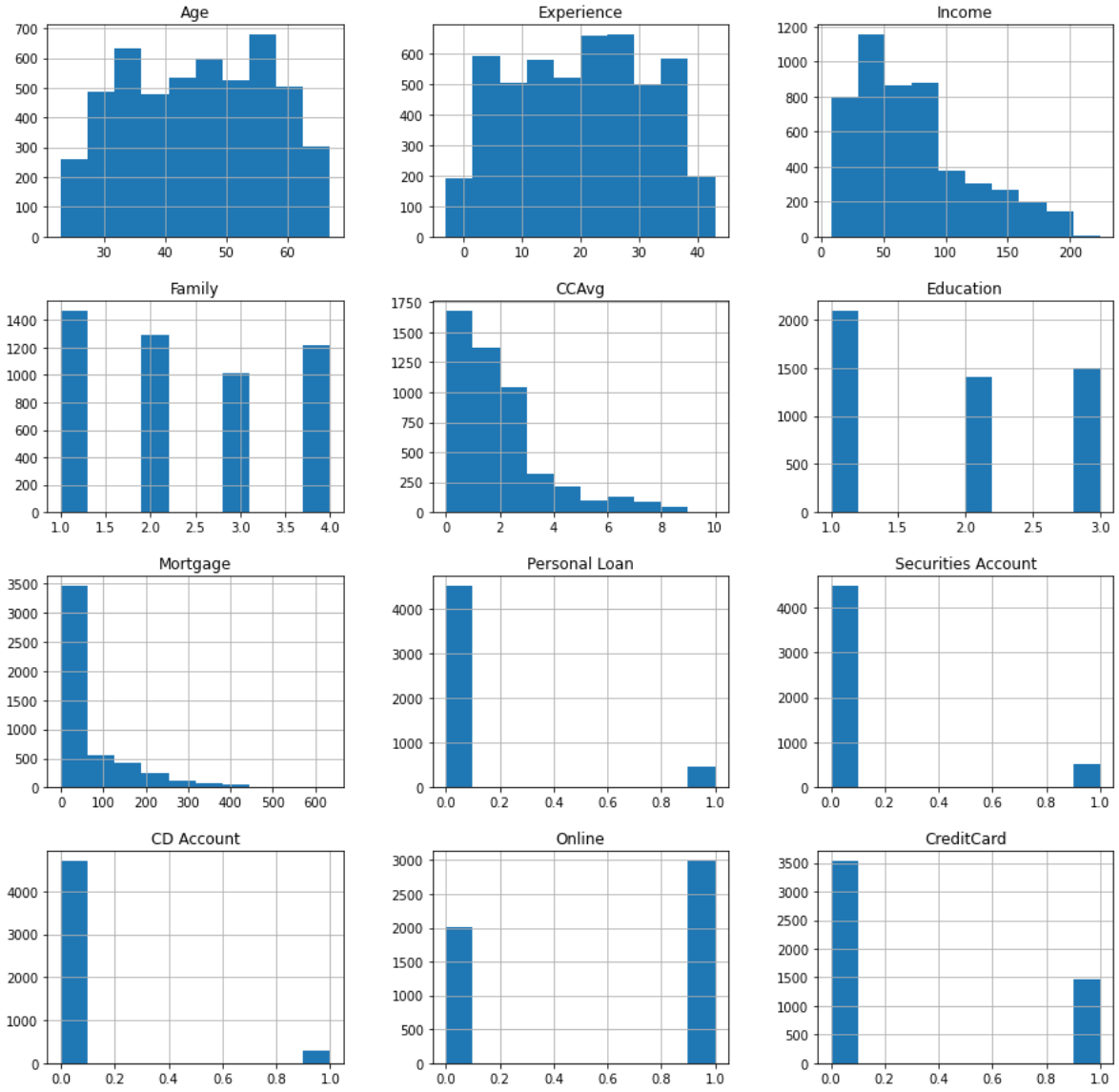
```
Age                 -0.029341
Experience          -0.026325
Income               0.841339
Family               0.155221
CCAvg                1.598457
Education            0.227093
Mortgage             2.104002
Personal Loan        2.743607
Securities Account   2.588268
CD Account           3.691714
Online              -0.394785
CreditCard           0.904589
dtype: float64
```

In [10]:

```
data.hist(figsize = (15,15))
```

Out[10]:

```
array([[<AxesSubplot: title={'center': 'Age'}>,
        <AxesSubplot: title={'center': 'Experience'}>,
        <AxesSubplot: title={'center': 'Income'}>],
       [<AxesSubplot: title={'center': 'Family'}>,
        <AxesSubplot: title={'center': 'CCAvg'}>,
        <AxesSubplot: title={'center': 'Education'}>],
       [<AxesSubplot: title={'center': 'Mortgage'}>,
        <AxesSubplot: title={'center': 'Personal Loan'}>,
        <AxesSubplot: title={'center': 'Securities Account'}>],
       [<AxesSubplot: title={'center': 'CD Account'}>,
        <AxesSubplot: title={'center': 'Online'}>,
        <AxesSubplot: title={'center': 'CreditCard'}>]], dtype=object)
```



In [11]:

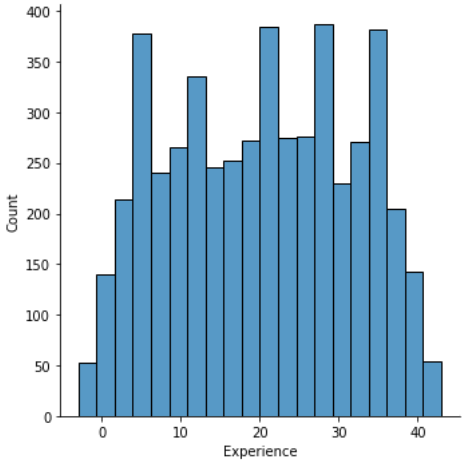
```
import seaborn as sns
```

In [12]:

```
sns.displot(data['Experience'])
```

Out[12]:

```
<seaborn.axisgrid.FacetGrid at 0x238c6578250>
```



In [13]:

```
data['Experience'].mean()
```

Out[13]:

```
20.1046
```

In [14]:

```
neg_exp = data[data['Experience'] < 0]
neg_exp.head()
```

Out[14]:

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
89	25	-1	113	4	2.30	3	0	0	0	0	0	1
226	24	-1	39	2	1.70	2	0	0	0	0	0	0
315	24	-2	51	3	0.30	3	0	0	0	0	1	0
451	28	-2	48	2	1.75	3	89	0	0	0	1	0
524	24	-1	75	4	0.20	1	0	0	0	0	1	0

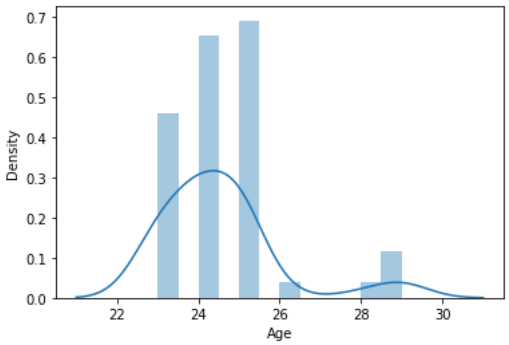
In [15]:

```
sns.distplot(neg_exp['Age'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
```

Out[15]:

```
<AxesSubplot: xlabel='Age', ylabel='Density'>
```



In [16]:

```
neg_exp['Experience'].mean()
```

Out[16]:

```
-1.4423076923076923
```

In [17]:

```
neg_exp.size
```

Out[17]:

624

In [18]:

```
print('There are around {} records which has negative values around {} %'. format(neg_exp.size, ((neg_exp.size/data.size )*100)))
```

There are around 624 records which has negative values around 1.04 %

In [19]:

```
data2 = data.copy()
data2.head()
```

Out[19]:

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	25	1	49	4	1.6	1	0	0	1	0	0	0
1	45	19	34	3	1.5	1	0	0	1	0	0	0
2	39	15	11	1	1.0	1	0	0	0	0	0	0
3	35	9	100	1	2.7	2	0	0	0	0	0	0
4	35	8	45	4	1.0	2	0	0	0	0	0	1

In [20]:

```
data2['Experience'] = np.where(data2['Experience']<0, data2['Experience'].mean(), data2['Experience'])
```

In [21]:

```
data2.head()
```

Out[21]:

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	25	1.0	49	4	1.6	1	0	0	1	0	0	0
1	45	19.0	34	3	1.5	1	0	0	1	0	0	0
2	39	15.0	11	1	1.0	1	0	0	0	0	0	0
3	35	9.0	100	1	2.7	2	0	0	0	0	0	0
4	35	8.0	45	4	1.0	2	0	0	0	0	0	1

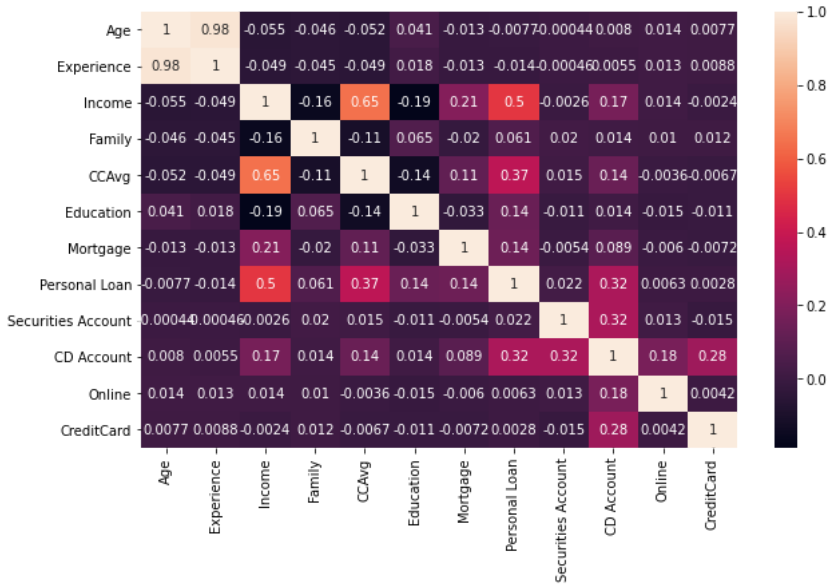
Co-relation of data

In [22]:

```
plt.figure(figsize = (10,6))
sns.heatmap(data2.corr(),annot = True)
```

Out[22]:

<AxesSubplot: >



In [23]:

```
data2.drop(['Experience'], axis = 1, inplace = True)
data2.head()
```

Out[23]:

	Age	Income	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	25	49	4	1.6	1	0	0	1	0	0	0
1	45	34	3	1.5	1	0	0	1	0	0	0
2	39	11	1	1.0	1	0	0	0	0	0	0
3	35	100	1	2.7	2	0	0	0	0	0	0
4	35	45	4	1.0	2	0	0	0	0	0	1

Education Status of Customers

In [24]:

```
def mark(x):
    if x==1:
        return 'UG'
    elif x==2:
        return 'PG'
    else:
        return 'Prof'
```

In [25]:

```
data2['Edu'] = data2['Education'].apply(mark)
```

In [26]:

```
data2.head()
```

Out[26]:

	Age	Income	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard	Edu
0	25	49	4	1.6	1	0	0	1	0	0	0	UG
1	45	34	3	1.5	1	0	0	1	0	0	0	UG
2	39	11	1	1.0	1	0	0	0	0	0	0	UG
3	35	100	1	2.7	2	0	0	0	0	0	0	PG
4	35	45	4	1.0	2	0	0	0	0	0	1	PG

In [27]:

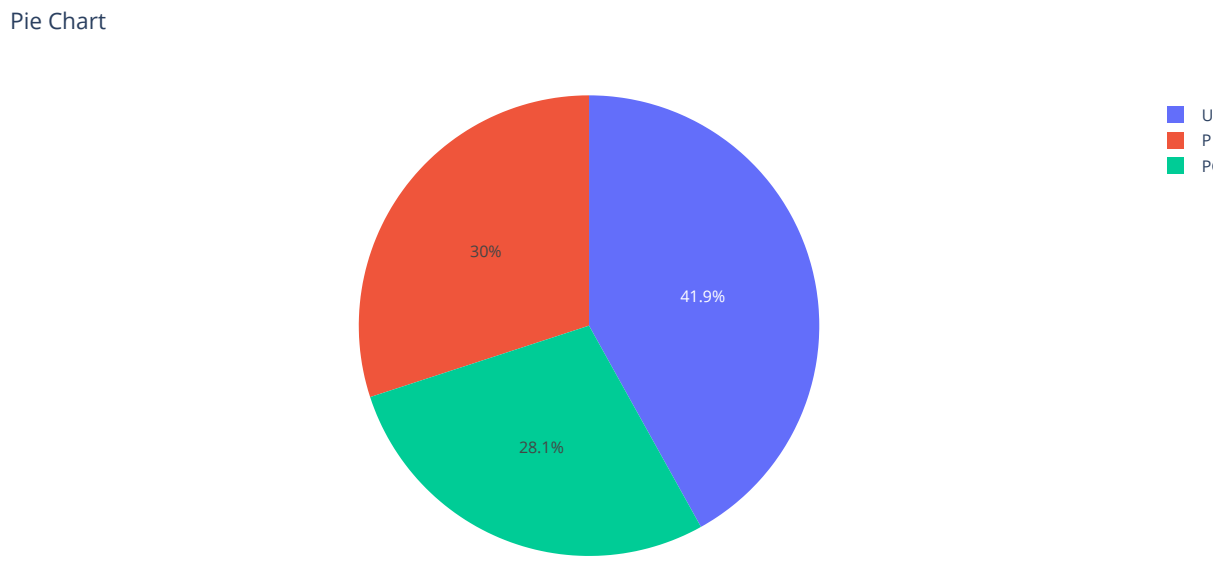
```
Edu_dis = data2.groupby('Edu')['Age'].count()
Edu_dis
```

Out[27]:

```
Edu
PG      1403
Prof    1501
UG      2096
Name: Age, dtype: int64
```

In [28]:

```
fig = px.pie(data2, values = Edu_dis, names = Edu_dis.index, title = 'Pie Chart')
fig.show()
```



Account Holder Distribution

In [29]:

```
def security_cd(row):
    if(row['Securities Account']==1) & (row['CD Account'] == 1):
        return 'Holds Security and deposit'
    elif(row['Securities Account']==1) & (row['CD Account'] == 0):
        return 'Holds only Security '
    elif(row['Securities Account']==0) & (row['CD Account'] == 1):
        return 'Holds only deposit'
    elif(row['Securities Account']==0) & (row['CD Account'] == 0):
        return 'Does not Hold Security or deposit'
```

In [30]:

```
data2['Account_holder_category'] = data2.apply(security_cd,axis = 1)
```

In [31]:

```
data2.head()
```

Out[31]:

	Age	Income	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard	Edu	Account_holder_category
0	25	49	4	1.6	1	0	0	1	0	0	0	UG	Holds only Security
1	45	34	3	1.5	1	0	0	1	0	0	0	UG	Holds only Security
2	39	11	1	1.0	1	0	0	0	0	0	0	UG	Does not Hold Security or deposit
3	35	100	1	2.7	2	0	0	0	0	0	0	PG	Does not Hold Security or deposit
4	35	45	4	1.0	2	0	0	0	0	0	1	PG	Does not Hold Security or deposit

In [32]:

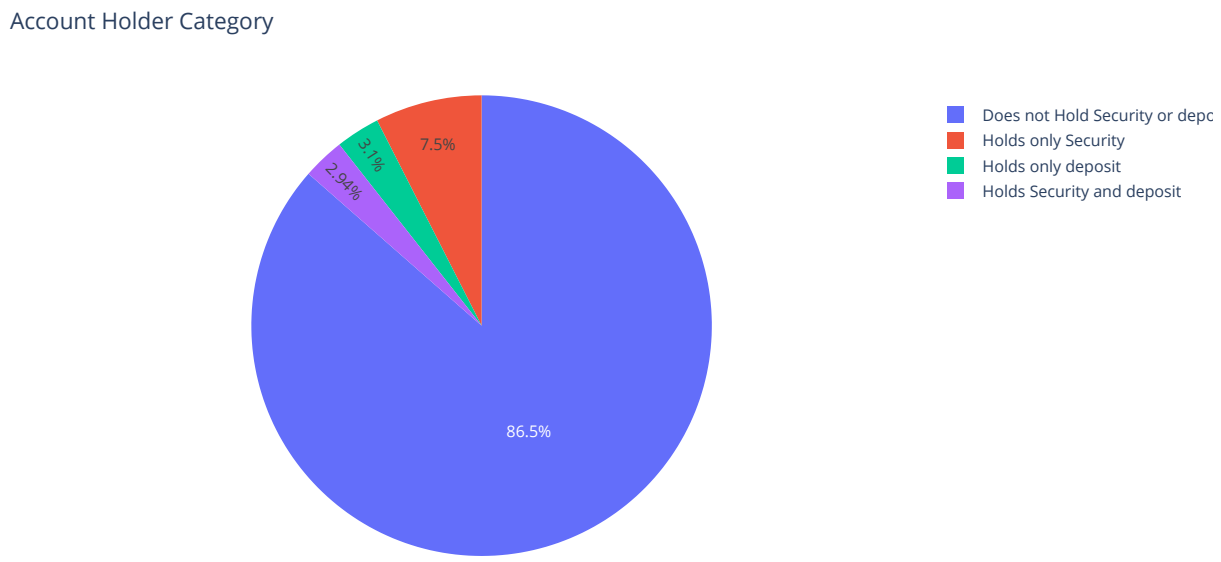
```
values = data2['Account_holder_category'].value_counts()
values.index
```

Out[32]:

```
Index(['Does not Hold Security or deposit', 'Holds only Security ',
       'Holds only deposit', 'Holds Security and deposit'],
      dtype='object')
```

In [33]:

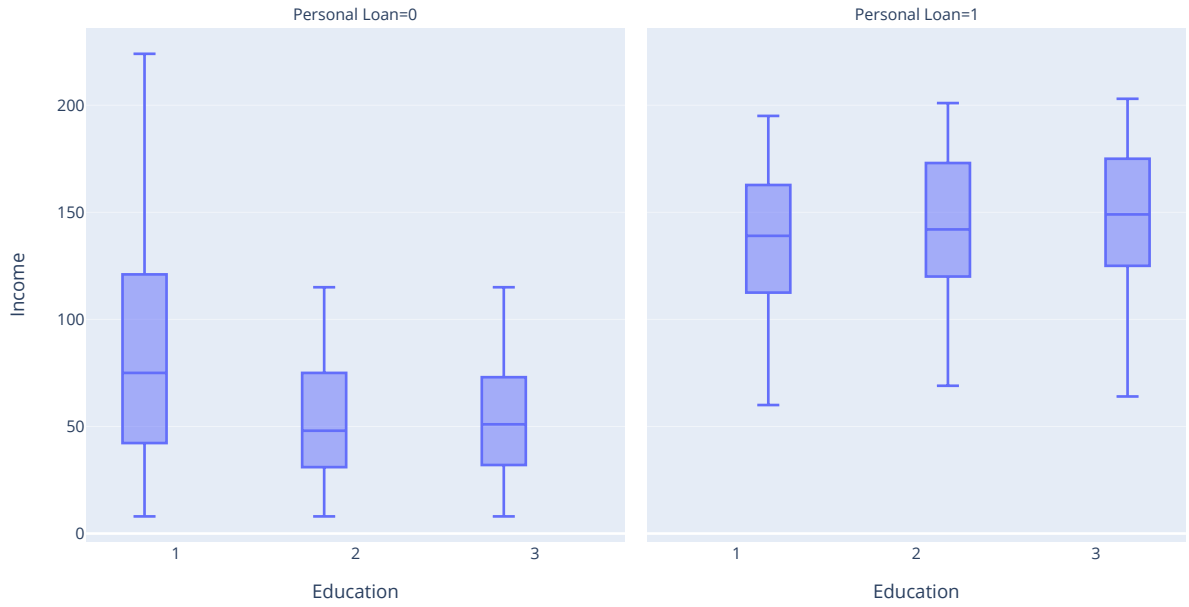
```
fig = px.pie(data2, values = values, names = values.index, title = 'Account Holder Category')
fig.show()
```



Customers based on Educational Status, Income and Personal Loan status

In [34]:

```
px.box(data2, x = 'Education', y = 'Income', facet_col = 'Personal Loan')
```



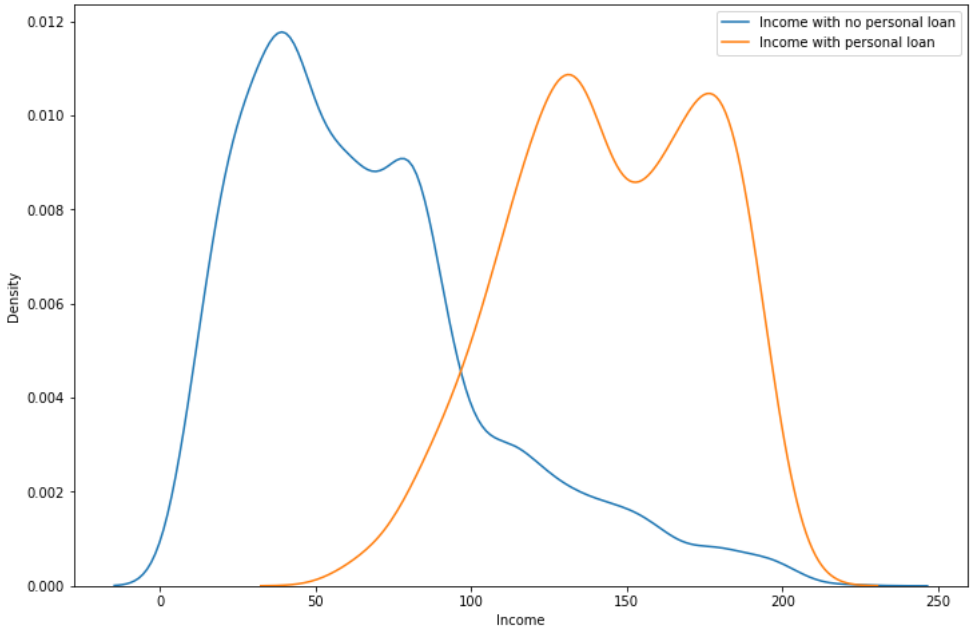
In [35]:

```
plt.figure(figsize = (12,8))
sns.distplot(data2[data2['Personal Loan'] == 0]['Income'],hist = False, label = 'Income with no personal loan')
sns.distplot(data2[data2['Personal Loan'] == 1]['Income'],hist = False, label = 'Income with personal loan')
plt.legend()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
```

Out[35]:

```
<matplotlib.legend.Legend at 0x238d6fe2520>
```



Automation of Analysis

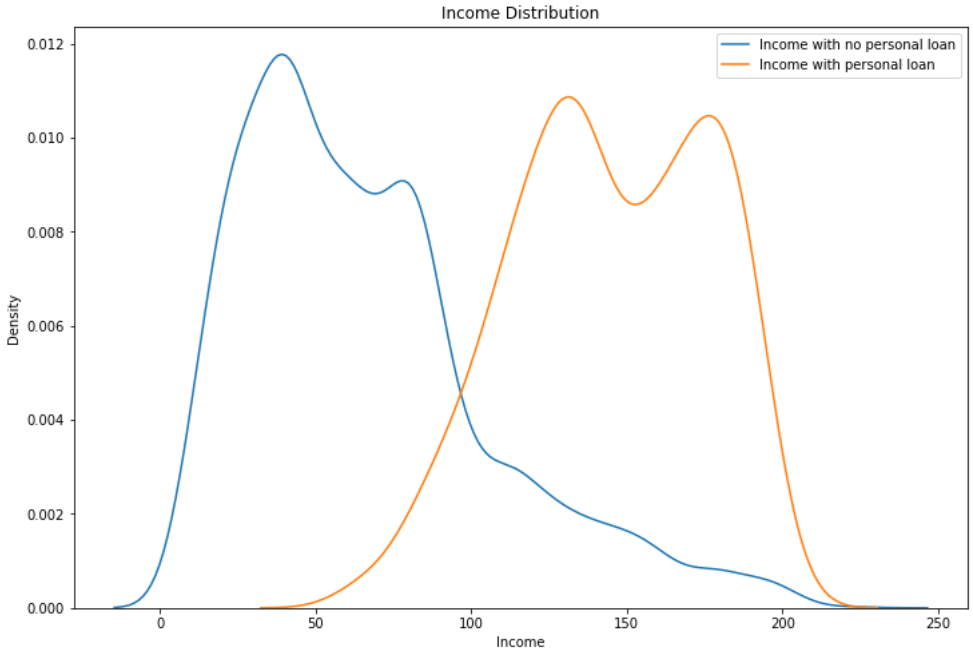
In [36]:

```
def plot(col1,col2,lab1,lab2,title):
    plt.figure(figsize = (12,8))
    sns.distplot(data2[data2[col2] == 0][col1],hist = False, label = lab1)
    sns.distplot(data2[data2[col2] == 1][col1],hist = False, label = lab2)
    plt.legend()
    plt.title(title)
```

In [37]:

```
plot('Income','Personal Loan','Income with no personal loan','Income with personal loan','Income Distribution')
```

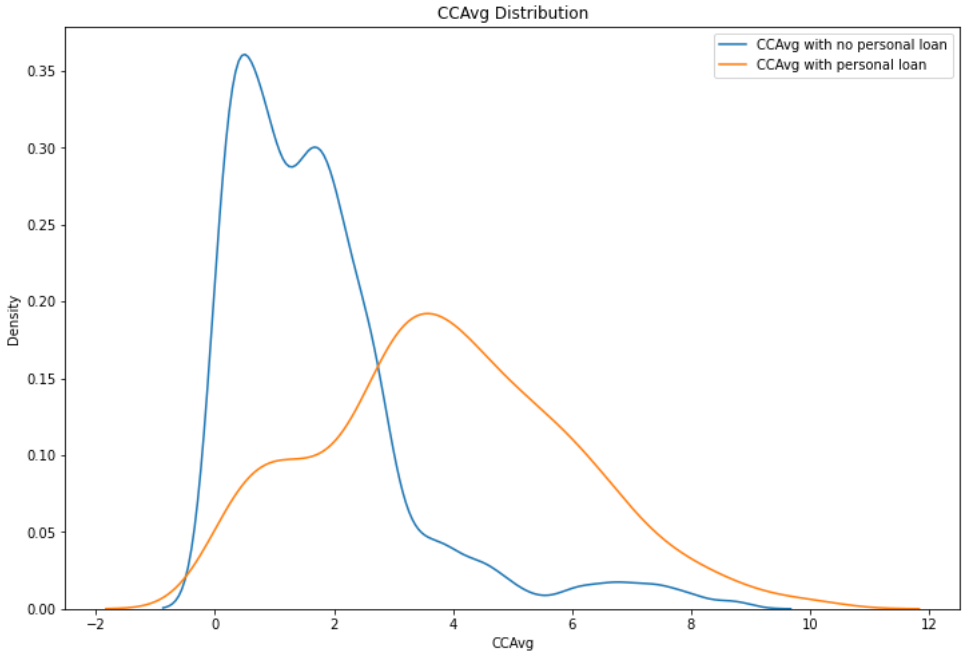
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
```



In [38]:

```
plot('CCAvg','Personal Loan','CCAvg with no personal loan','CCAvg with personal loan','CCAvg Distribution')
```

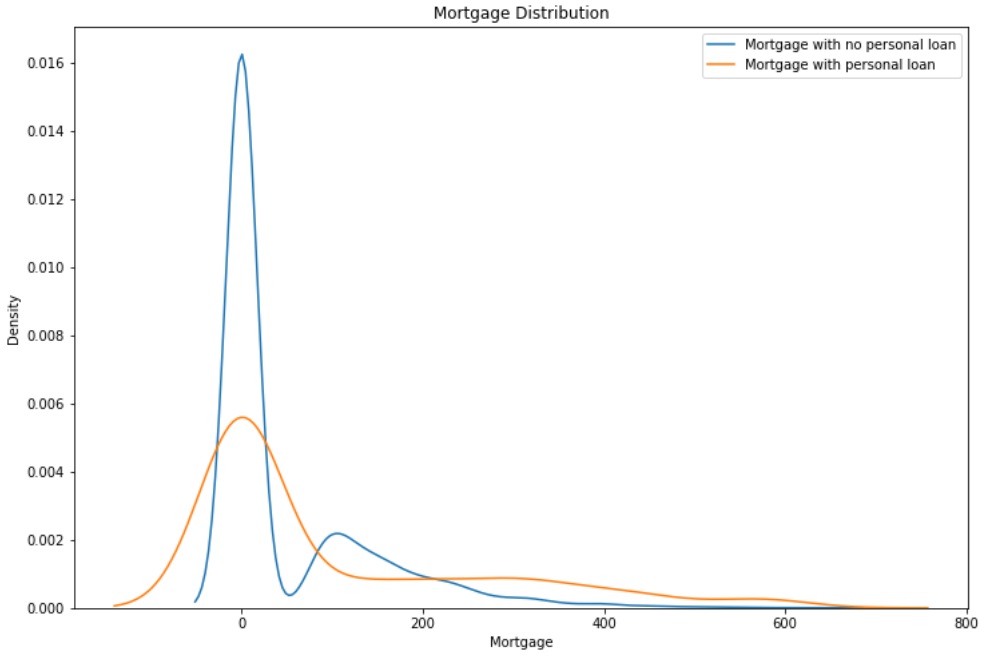
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
```



In [39]:

```
plot('Mortgage','Personal Loan','Mortgage with no personal loan','Mortgage with personal loan','Mortgage Distribution')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
```



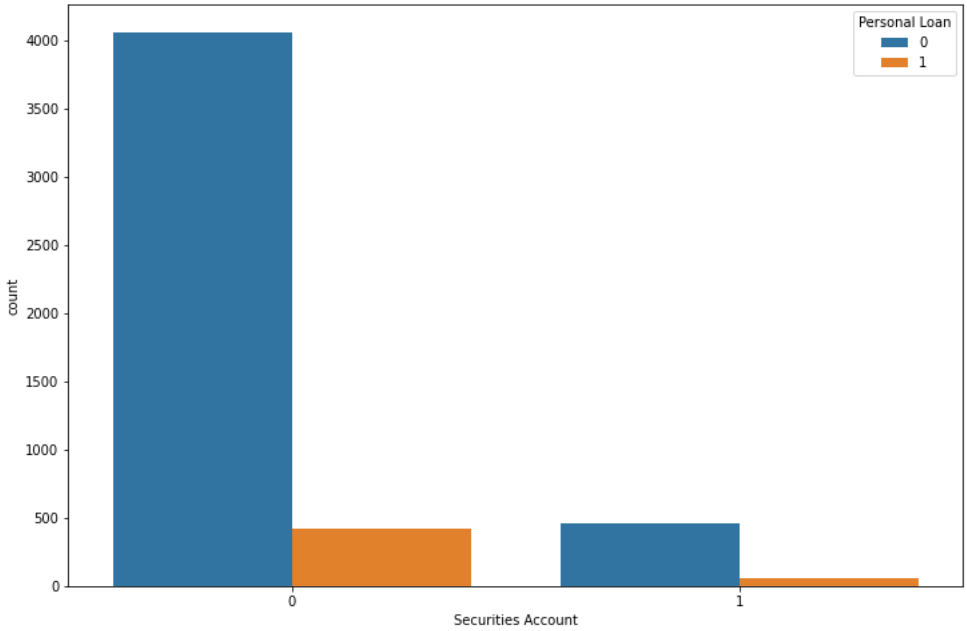
Categories of customers on the basis of security account, online, Account holder category and credit card

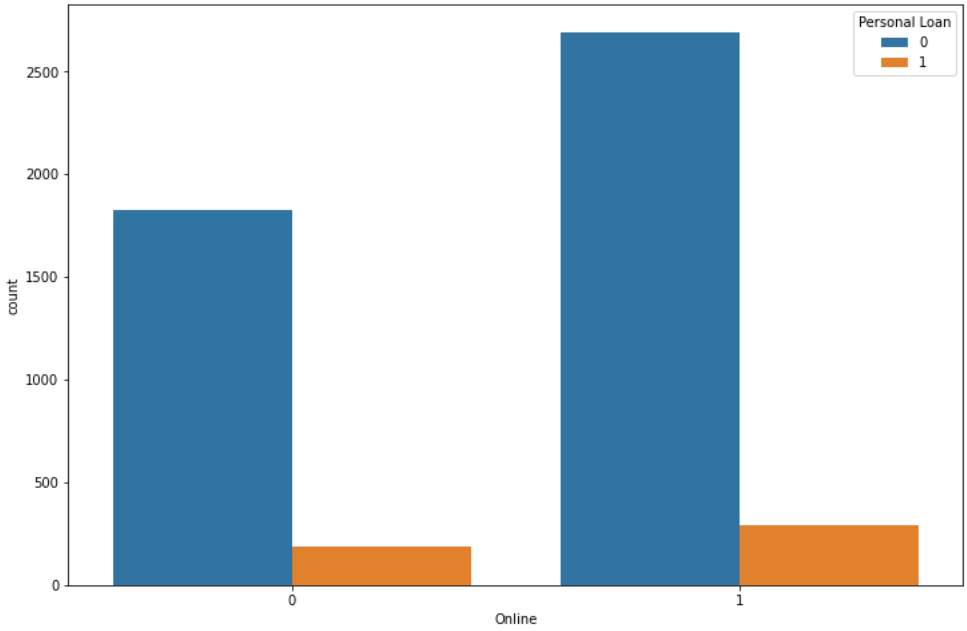
In [40]:

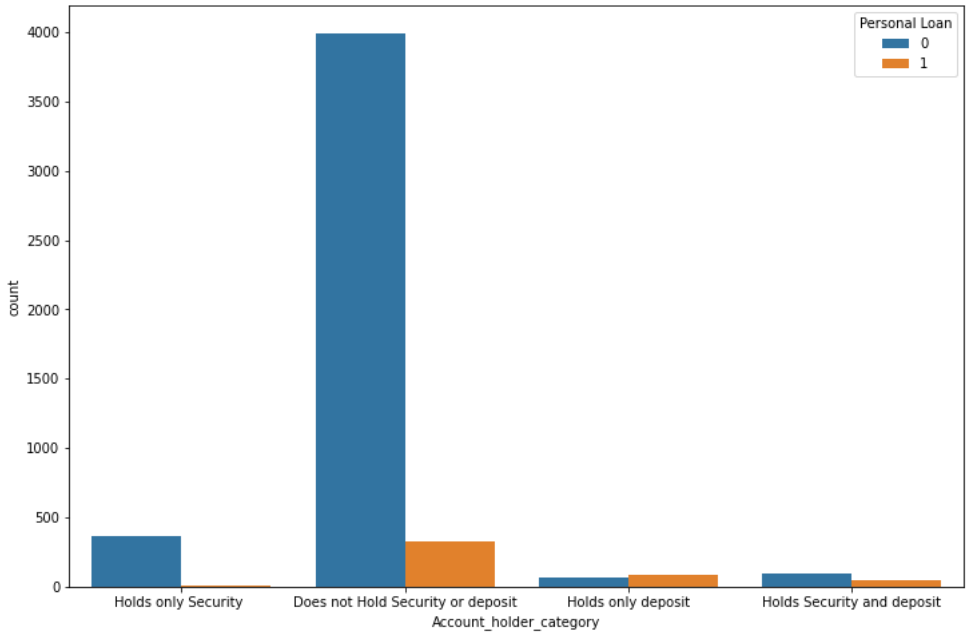
```
col_names = ['Securities Account','Online','Account_holder_category','CreditCard']
```

In [41]:

```
for i in col_names:
    plt.figure(figsize = (12,8))
    sns.countplot(x=i, hue = 'Personal Loan', data = data2)
```







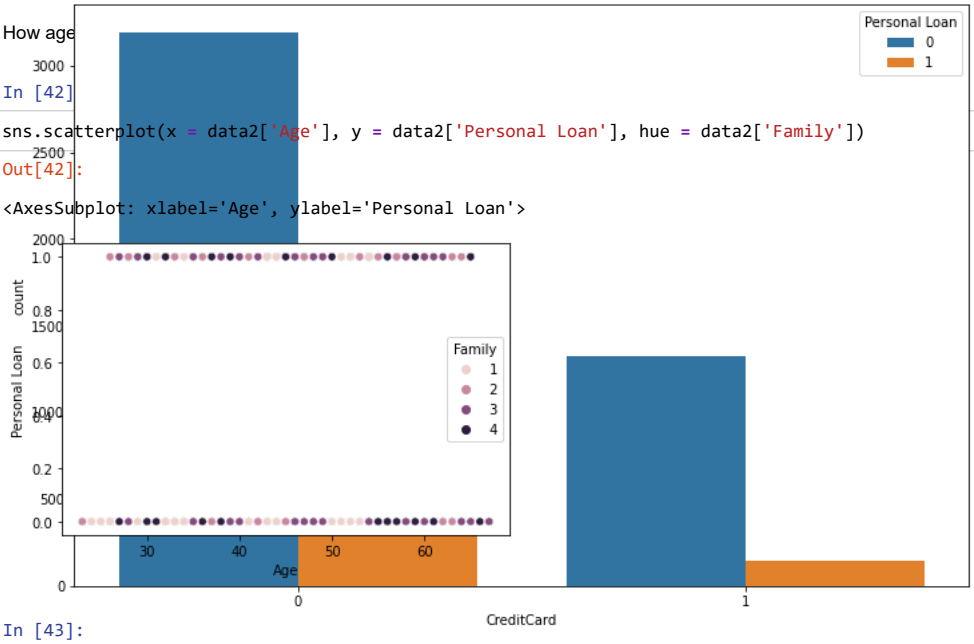
How age

In [42]:

```
sns.scatterplot(x = data2['Age'], y = data2['Personal Loan'], hue = data2['Family'])
```

Out[42]:

<AxesSubplot: xlabel='Age', ylabel='Personal Loan'>



In [43]:

```
import scipy.stats as stats
Ho = 'Age does not have impact on personal loan'
Ha = 'Age has impact on personal loan'
```

In [44]:

```
Age_no = np.array(data2[data2['Personal Loan']==0]['Age'])
Age_yes = np.array(data2[data2['Personal Loan']==1]['Age'])
```

In [45]:

```
t,p_value = stats.ttest_ind(Age_no, Age_yes, axis = 0)
if p_value < 0.05:
    print(Ha,'with a p_value {} which is lesser than 0.05'.format(p_value))
if p_value > 0.05:
    print(Ho,'with a p_value {} which is greater than 0.05'.format(p_value))
```

```
Age does not have impact on personal loan with a p_value 0.584959263705325 which is greater than 0.05
```

In [46]:

```
def hypo(col1,col2,ha,ho):
    arr1 = np.array(data2[data2[col1]==0][col2])
    arr2 = np.array(data2[data2[col1]==1][col2])
    t,p_value = stats.ttest_ind(arr1, arr2, axis = 0)
    if p_value < 0.05:
        print('{},with a p_value {} which is lesser than 0.05'.format(ha,p_value))
    if p_value > 0.05:
        print('{},with a p_value {} which is greater than 0.05'.format(ho,p_value))
```

In [47]:

```
hypo('Personal Loan','Age',ho = 'Age does not have impact on personal loan',ha = 'Age has impact on personal loan')
```

```
Age does not have impact on personal loan,with a p_value 0.584959263705325 which is greater than 0.05
```

In [48]:

```
hypo('Personal Loan','Income',ho = 'Income does not have impact on personal loan',ha = 'Income has impact on personal loan')
```

```
Income has impact on personal loan,with a p_value 0.0 which is lesser than 0.05
```

In [49]:

```
hypo('Personal Loan','Family',ho = 'Family size does not have impact on personal loan',ha = 'Family size has impact on personal loan')
```

```
Family size has impact on personal loan,with a p_value 1.4099040685673807e-05 which is lesser than 0.05
```