finance-project-1

June 17, 2024

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
df= pd.read_csv(r"D:\Data Science & AI\1. Python␣
 ↪Module\Bank_Personal_Loan_Modelling.csv")
df.head()
```

[2]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	\
0	1	25	1	49	91107	4	1/60	1	0	
1	2	45	19	34	90089	3	1/50	1	0	
2	3	39	15	11	94720	1	1/00	1	0	
3	4	35	9	100	94112	1	2/70	2	0	
4	5	35	8	45	91330	4	1/00	2	0	

	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	0	1	0	0	0
1	0	1	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	1

[3]:
```
df.shape
```

[3]:
```
(5000, 14)
```

[4]:
```
df.isnull().sum()
```

[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
```

[5]:
```
df.columns
```

[5]:
```
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal Loan', 'Securities Account',
       'CD Account', 'Online', 'CreditCard'],
      dtype='object')
```

[6]:
```
df.drop(['ID','ZIP Code'], axis=1, inplace=True)
```

[7]:
```
df.columns
```

[7]:
```
Index(['Age', 'Experience', 'Income', 'Family', 'CCAvg', 'Education',
       'Mortgage', 'Personal Loan', 'Securities Account', 'CD Account',
       'Online', 'CreditCard'],
      dtype='object')
```

[8]:
```
#5 Number Summary

import plotly.express as px
```

[9]:
```
fig=px.box(df,y=["Age", "Experience","Family", "Income","Education"])
fig.show()
```

[10]:
```
df.dtypes
```

[10]:
```
Age                  int64
Experience           int64
Income               int64
Family               int64
CCAvg               object
Education            int64
Mortgage             int64
Personal Loan        int64
Securities Account   int64
CD Account           int64
Online               int64
CreditCard           int64
```

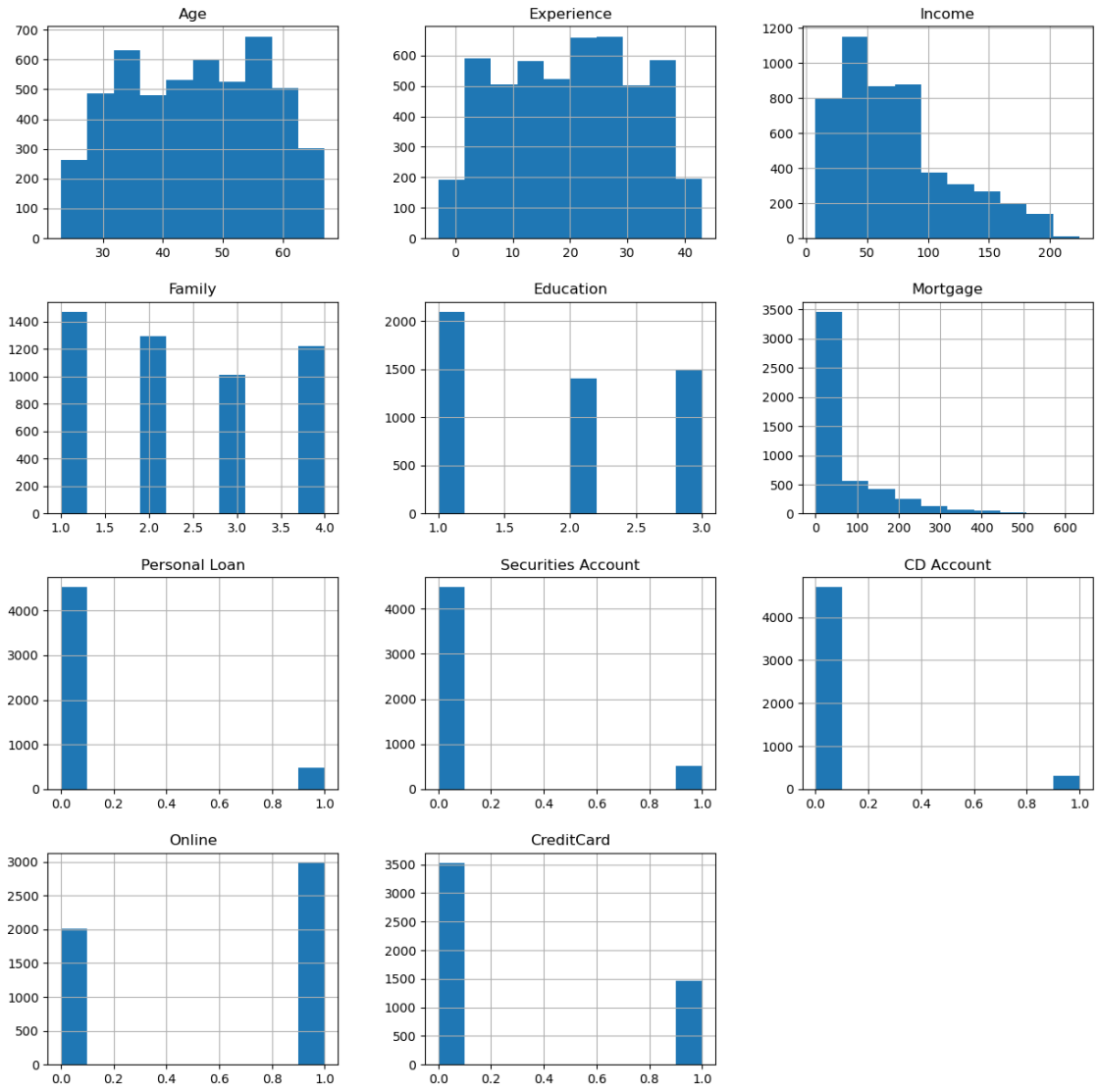
```
dtype: object
```

[11]:
```
df.skew
```

[11]:
```
<bound method NDFrame._add_numeric_operations.<locals>.skew of       Age
Experience  Income  Family CCAvg  Education  Mortgage  \
0      25           1      49       4  1/60          1         0
1      45          19      34       3  1/50          1         0
2      39          15      11       1  1/00          1         0
3      35           9     100       1  2/70          2         0
4      35           8      45       4  1/00          2         0
…   …          …      …      …   …         …       …
4995   29           3      40       1  1/90          3         0
4996   30           4      15       4  0/40          1        85
4997   63          39      24       2  0/30          3         0
4998   65          40      49       3  0/50          2         0
4999   28           4      83       3  0/80          1         0

      Personal Loan  Securities Account  CD Account  Online  CreditCard
0                 0                   1           0       0           0
1                 0                   1           0       0           0
2                 0                   0           0       0           0
3                 0                   0           0       0           0
4                 0                   0           0       0           1
…               …                   …         …     …         …
4995              0                   0           0       1           0
4996              0                   0           0       1           0
4997              0                   0           0       0           0
4998              0                   0           0       1           0
4999              0                   0           0       1           1

[5000 rows x 12 columns]>
```

[12]:
```
df.hist(figsize=(15,15))
```

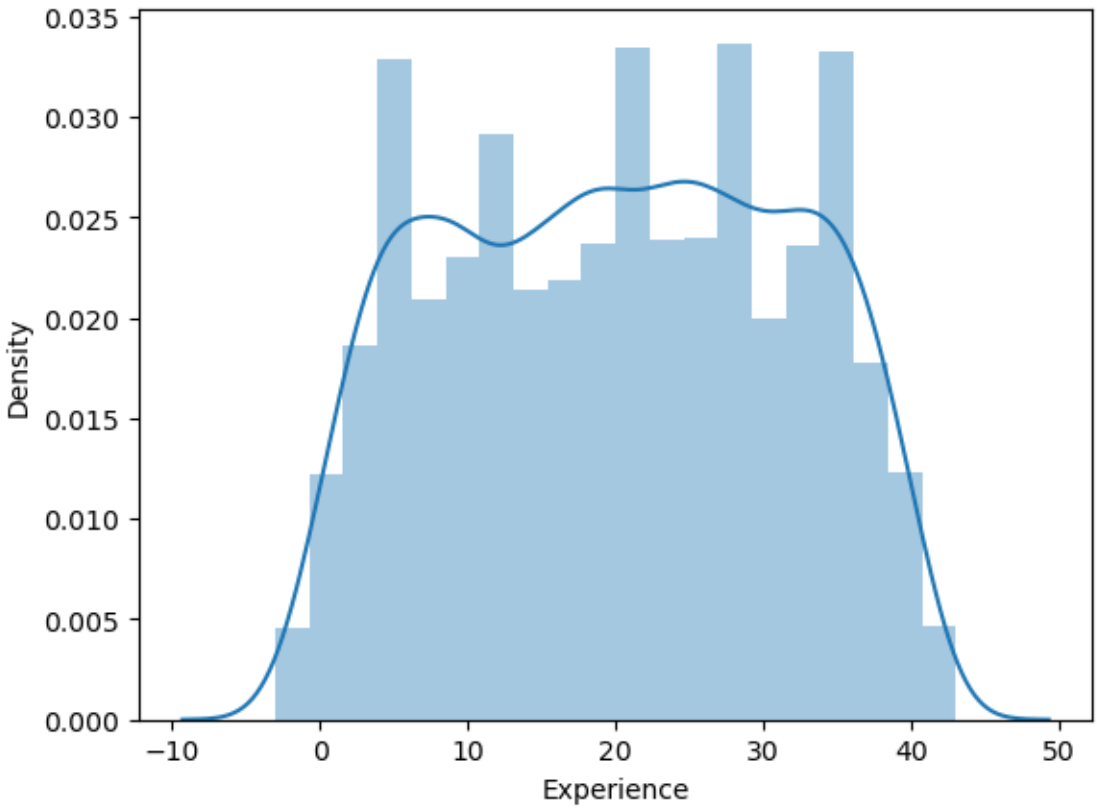
[12]:
```
array([[<Axes: title={'center': 'Age'}>,
        <Axes: title={'center': 'Experience'}>,
        <Axes: title={'center': 'Income'}>],
       [<Axes: title={'center': 'Family'}>,
        <Axes: title={'center': 'Education'}>,
        <Axes: title={'center': 'Mortgage'}>],
       [<Axes: title={'center': 'Personal Loan'}>,
        <Axes: title={'center': 'Securities Account'}>,
        <Axes: title={'center': 'CD Account'}>],
       [<Axes: title={'center': 'Online'}>,
        <Axes: title={'center': 'CreditCard'}>, <Axes: >]], dtype=object)
```



[13]:
```
import seaborn as sns
```

[14]:
```
sns.distplot(df['Experience'])
```

[14]:
```
<Axes: xlabel='Experience', ylabel='Density'>
```



[15]:
```
df['Experience'].mean()
```

[15]: 20.1046

[16]:
```
Negative_Exp=df[df['Experience']<0]
Negative_Exp.head()
```

[16]:

	Age	Experience	Income	Family	CCAvg	Education	Mortgage	\
89	25	-1	113	4	2/30	3	0	
226	24	-1	39	2	1/70	2	0	
315	24	-2	51	3	0/30	3	0	
451	28	-2	48	2	1/75	3	89	
524	24	-1	75	4	0/20	1	0	

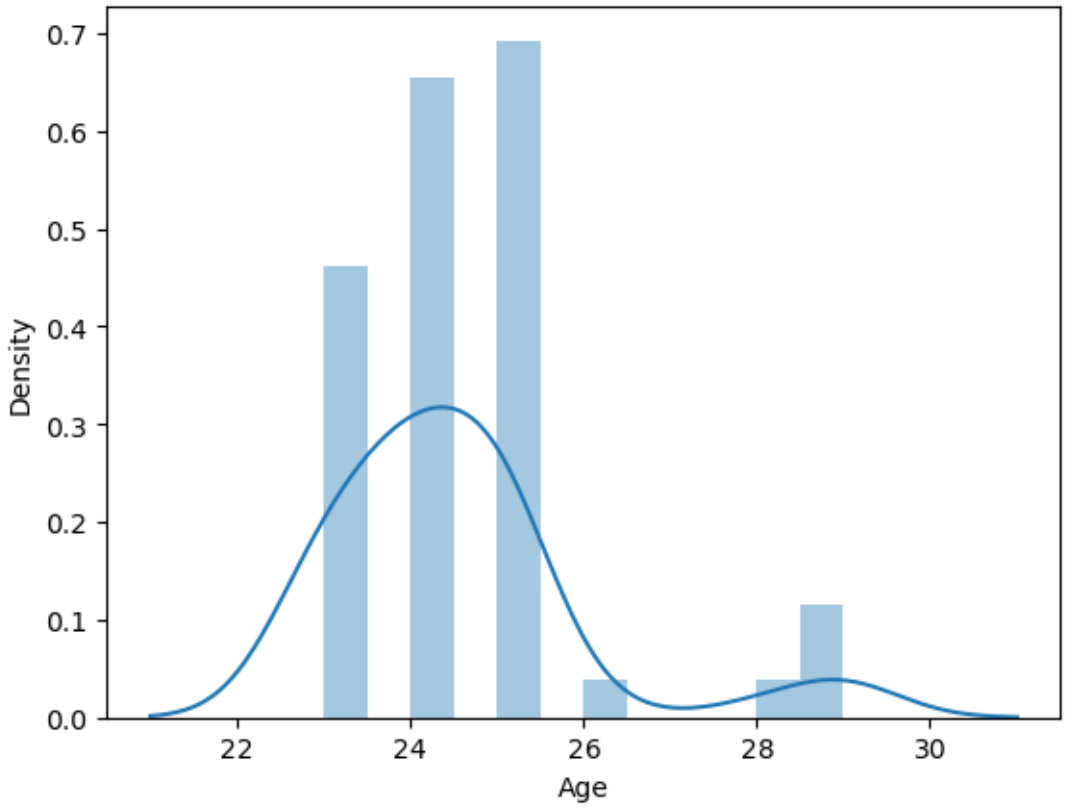
	Personal Loan	Securities Account	CD Account	Online	CreditCard
89	0	0	0	0	1
226	0	0	0	0	0
315	0	0	0	1	0
451	0	0	0	1	0
524	0	0	0	1	0

[17]:
```
Negative_Exp.shape
```

[17]:
```
(52, 12)
```

[18]:
```
sns.distplot(Negative_Exp['Age'])
```

[18]:
```
<Axes: xlabel='Age', ylabel='Density'>
```



[19]:
```
Negative_Exp['Experience'].mean(    )
```

[19]:
```
-1.4423076923076923
```

[20]:
```
Negative_Exp.size
```

[20]:
```
624
```

[21]:
```
print('There are {} records which has negative value for experience, approx{}␣
↪%'.format(Negative_Exp.size, ((Negative_Exp.size/df.size)*100)))
```

```
There are 624 records which has negative value for experience, approx1.04 %
```

[22]:
```
data = df.copy()
```

[23]:
```
data
```

[23]:
```
      Age  Experience  Income  Family CCAvg  Education  Mortgage  \
0      25           1      49       4  1/60          1         0
1      45          19      34       3  1/50          1         0
2      39          15      11       1  1/00          1         0
3      35           9     100       1  2/70          2         0
4      35           8      45       4  1/00          2         0
…    …         …      …      …    …         …       …
4995   29           3      40       1  1/90          3         0
4996   30           4      15       4  0/40          1        85
4997   63          39      24       2  0/30          3         0
4998   65          40      49       3  0/50          2         0
4999   28           4      83       3  0/80          1         0

      Personal Loan  Securities Account  CD Account  Online  CreditCard
0                 0                   1           0       0           0
1                 0                   1           0       0           0
2                 0                   0           0       0           0
3                 0                   0           0       0           0
4                 0                   0           0       0           1
…               …                 …         …     …         …
4995              0                   0           0       1           0
4996              0                   0           0       1           0
4997              0                   0           0       0           0
4998              0                   0           0       1           0
4999              0                   0           0       1           1

[5000 rows x 12 columns]
```

[24]:
```
data.shape
```

[24]:
```
(5000, 12)
```

[25]:
```
import numpy as np
```

[26]:
```
data['Experience']=np.where(data['Experience']<0,
                            data['Experience'].mean(),
                            data['Experience'])
```

[27]:
```
data[data['Experience']<0]
```

[27]:
```
Empty DataFrame
Columns: [Age, Experience, Income, Family, CCAvg, Education, Mortgage, Personal
Loan, Securities Account, CD Account, Online, CreditCard]
```

```
Index: []
```

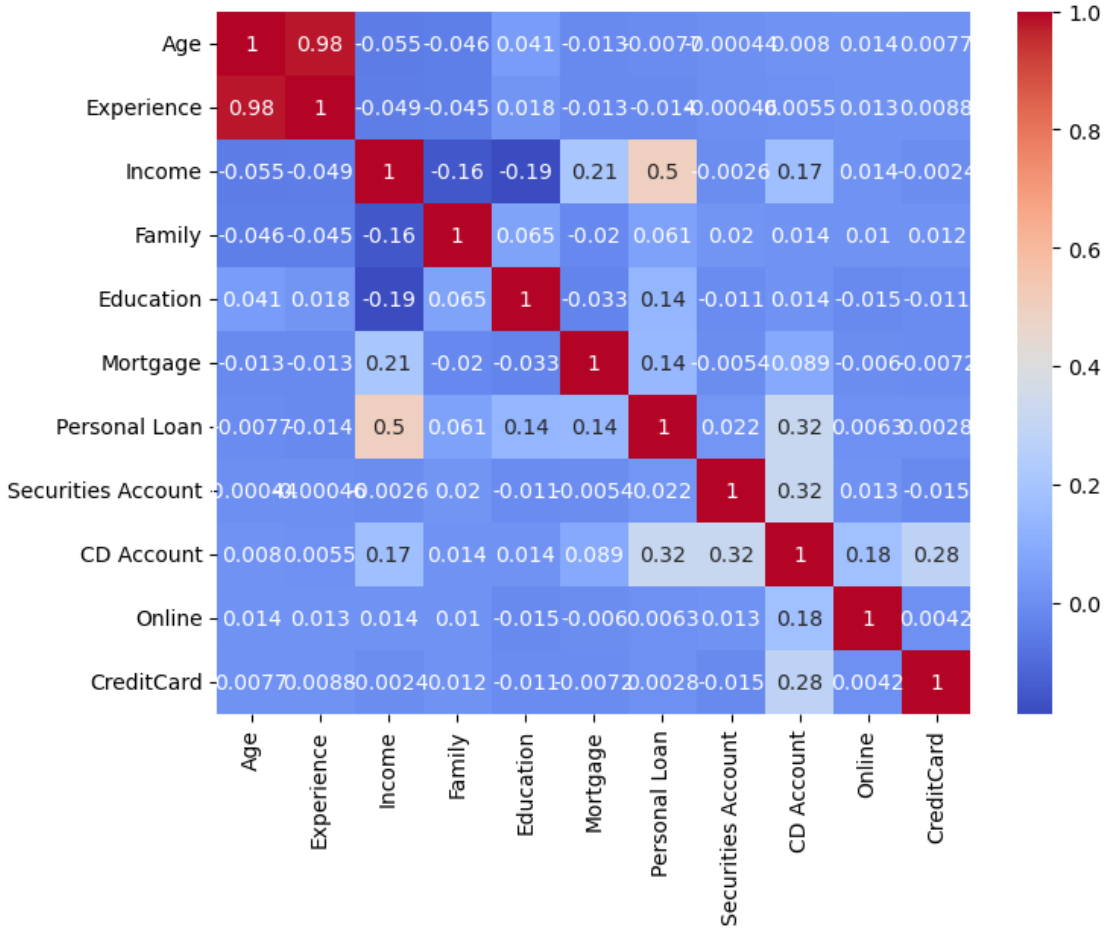
[28]:
```
data.corr()
```

[28]:
```
                          Age  Experience    Income    Family  Education  \
Age                  1.000000    0.977008 -0.055269 -0.046418   0.041334
Experience           0.977008    1.000000 -0.049054 -0.045488   0.018097
Income              -0.055269   -0.049054  1.000000 -0.157501  -0.187524
Family              -0.046418   -0.045488 -0.157501  1.000000   0.064929
Education            0.041334    0.018097 -0.187524  0.064929   1.000000
Mortgage            -0.012539   -0.013378  0.206806 -0.020445  -0.033327
Personal Loan       -0.007726   -0.014045  0.502462  0.061367   0.136722
Securities Account  -0.000436   -0.000462 -0.002616  0.019994  -0.010812
CD Account           0.008043    0.005502  0.169738  0.014110   0.013934
Online               0.013702    0.013455  0.014206  0.010354  -0.015004
CreditCard           0.007681    0.008833 -0.002385  0.011588  -0.011014

                     Mortgage  Personal Loan  Securities Account  CD Account  \
Age                 -0.012539      -0.007726           -0.000436    0.008043
Experience          -0.013378      -0.014045           -0.000462    0.005502
Income               0.206806       0.502462           -0.002616    0.169738
Family              -0.020445       0.061367            0.019994    0.014110
Education           -0.033327       0.136722           -0.010812    0.013934
Mortgage             1.000000       0.142095           -0.005411    0.089311
Personal Loan        0.142095       1.000000            0.021954    0.316355
Securities Account  -0.005411       0.021954            1.000000    0.317034
CD Account           0.089311       0.316355            0.317034    1.000000
Online              -0.005995       0.006278            0.012627    0.175880
CreditCard          -0.007231       0.002802           -0.015028    0.278644

                       Online  CreditCard
Age                  0.013702    0.007681
Experience           0.013455    0.008833
Income               0.014206   -0.002385
Family               0.010354    0.011588
Education           -0.015004   -0.011014
Mortgage            -0.005995   -0.007231
Personal Loan        0.006278    0.002802
Securities Account   0.012627   -0.015028
CD Account           0.175880    0.278644
Online               1.000000    0.004210
CreditCard           0.004210    1.000000
```

[32]:
```
import matplotlib.pyplot as plt
import seaborn as sns
# Generate a correlation matrix
corr = data.corr()
```

```
# Create a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm')
```

[32]: <Axes: >



[33]:
```
data.drop(['Experience'], axis=1, inplace=True)
```

[34]:
```
data.head()
```

[34]:

	Age	Income	Family	CCAvg	Education	Mortgage	Personal Loan	\
0	25	49	4	1/60	1	0	0	
1	45	34	3	1/50	1	0	0	
2	39	11	1	1/00	1	0	0	
3	35	100	1	2/70	2	0	0	
4	35	45	4	1/00	2	0	0	

	Securities Account	CD Account	Online	CreditCard

```
0                    1           0       0           0
1                    1           0       0           0
2                    0           0       0           0
3                    0           0       0           0
4                    0           0       0           1
```

[35]:
```
#Education
data["Education"].unique()
```

[35]:
```
array([1, 2, 3], dtype=int64)
```

[36]:
```
def experiece(x):
    if x==1 :
        return "Undergrade"
    if x==2 :
        return "Graduate"
    if x==3 :
        return "Preofessional Person"
```

[40]:
```
data['EDU'] = data['Education'].apply(experiece)
```

[41]:
```
data.head()
```

[41]:
```
   Age  Income  Family CCAvg  Education  Mortgage  Personal Loan  \
0   25      49       4  1/60          1         0              0
1   45      34       3  1/50          1         0              0
2   39      11       1  1/00          1         0              0
3   35     100       1  2/70          2         0              0
4   35      45       4  1/00          2         0              0

   Securities Account  CD Account  Online  CreditCard         EDU
0                   1           0       0           0  Undergrade
1                   1           0       0           0  Undergrade
2                   0           0       0           0  Undergrade
3                   0           0       0           0    Graduate
4                   0           0       0           1    Graduate
```

[42]:
```
data['EDU'].unique()
```

[42]:
```
array(['Undergrade', 'Graduate', 'Preofessional Person'], dtype=object)
```

[43]:
```
education_dis = data.groupby('EDU')['Age'].count()
```

[44]:
```
education_dis
```

[44]:
```
EDU
Graduate                1403
```

```
Preofessional Person    1501
Undergrade              2096
Name: Age, dtype: int64
```

[45]:
```
px.pie(data,values=education_dis, names=education_dis.index, title='pie chart')
```

[46]:
```
data.columns
```

[46]:
```
Index(['Age', 'Income', 'Family', 'CCAvg', 'Education', 'Mortgage',
       'Personal Loan', 'Securities Account', 'CD Account', 'Online',
       'CreditCard', 'EDU'],
      dtype='object')
```

[47]:
```
data['Income'].unique()
```

[47]:
```
array([ 49,  34,  11, 100,  45,  29,  72,  22,  81, 180, 105, 114,  40,
       112, 130, 193,  21,  25,  63,  62,  43, 152,  83, 158,  48, 119,
        35,  41,  18,  50, 121,  71, 141,  80,  84,  60, 132, 104,  52,
       194,   8, 131, 190,  44, 139,  93, 188,  39, 125,  32,  20, 115,
        69,  85, 135,  12, 133,  19,  82, 109,  42,  78,  51, 113, 118,
        64, 161,  94,  15,  74,  30,  38,   9,  92,  61,  73,  70, 149,
        98, 128,  31,  58,  54, 124, 163,  24,  79, 134,  23,  13, 138,
       171, 168,  65,  10, 148, 159, 169, 144, 165,  59,  68,  91, 172,
        55, 155,  53,  89,  28,  75, 170, 120,  99, 111,  33, 129, 122,
       150, 195, 110, 101, 191, 140, 153, 173, 174,  90, 179, 145, 200,
       183, 182,  88, 160, 205, 164,  14, 175, 103, 108, 185, 204, 154,
       102, 192, 202, 162, 142,  95, 184, 181, 143, 123, 178, 198, 201,
       203, 189, 151, 199, 224, 218], dtype=int64)
```

[48]:
```
data['Securities Account'].value_counts()
```

[48]:
```
0    4478
1     522
Name: Securities Account, dtype: int64
```

[49]:
```
data['CD Account'].value_counts()
```

[49]:
```
0    4698
1     302
Name: CD Account, dtype: int64
```

[50]:
```
def security(y):
    if(y['Securities Account'] == 1) & (y['CD Account']==1):
        return "Hold Securities & Deposite"
    if(y['Securities Account'] == 0) & (y['CD Account']==0):
        return "Does not Hold Securities & Dposite"
    if(y['Securities Account'] == 1) & (y['CD Account']==0):
```

```
        return "Holds only  Securities"
    if(y['Securities Account'] == 0) & (y['CD Account']==1):
        return "Holds only Deposit account"
```

[51]:
```
data['Account_holder_category']=data.apply(security, axis=1)
```

[52]:
```
data.head()
```

[52]:

	Age	Income	Family	CCAvg	Education	Mortgage	Personal Loan	\
0	25	49	4	1/60	1	0	0	
1	45	34	3	1/50	1	0	0	
2	39	11	1	1/00	1	0	0	
3	35	100	1	2/70	2	0	0	
4	35	45	4	1/00	2	0	0	

	Securities Account	CD Account	Online	CreditCard	EDU	\
0	1	0	0	0	Undergrade	
1	1	0	0	0	Undergrade	
2	0	0	0	0	Undergrade	
3	0	0	0	0	Graduate	
4	0	0	0	1	Graduate	

	Account_holder_category
0	Holds only Securities
1	Holds only Securities
2	Does not Hold Securities & Dposite
3	Does not Hold Securities & Dposite
4	Does not Hold Securities & Dposite

[53]:
```
values=data['Account_holder_category'].value_counts()
values.index
```

[53]:
```
Index(['Does not Hold Securities & Dposite', 'Holds only  Securities',
       'Holds only Deposit account', 'Hold Securities & Deposite'],
      dtype='object')
```

[54]:
```
fig=px.pie(data,values=values, names=values.index, title='pie chart')
fig.show()
```

[55]:
```
data.columns
```

[55]:
```
Index(['Age', 'Income', 'Family', 'CCAvg', 'Education', 'Mortgage',
       'Personal Loan', 'Securities Account', 'CD Account', 'Online',
       'CreditCard', 'EDU', 'Account_holder_category'],
      dtype='object')
```

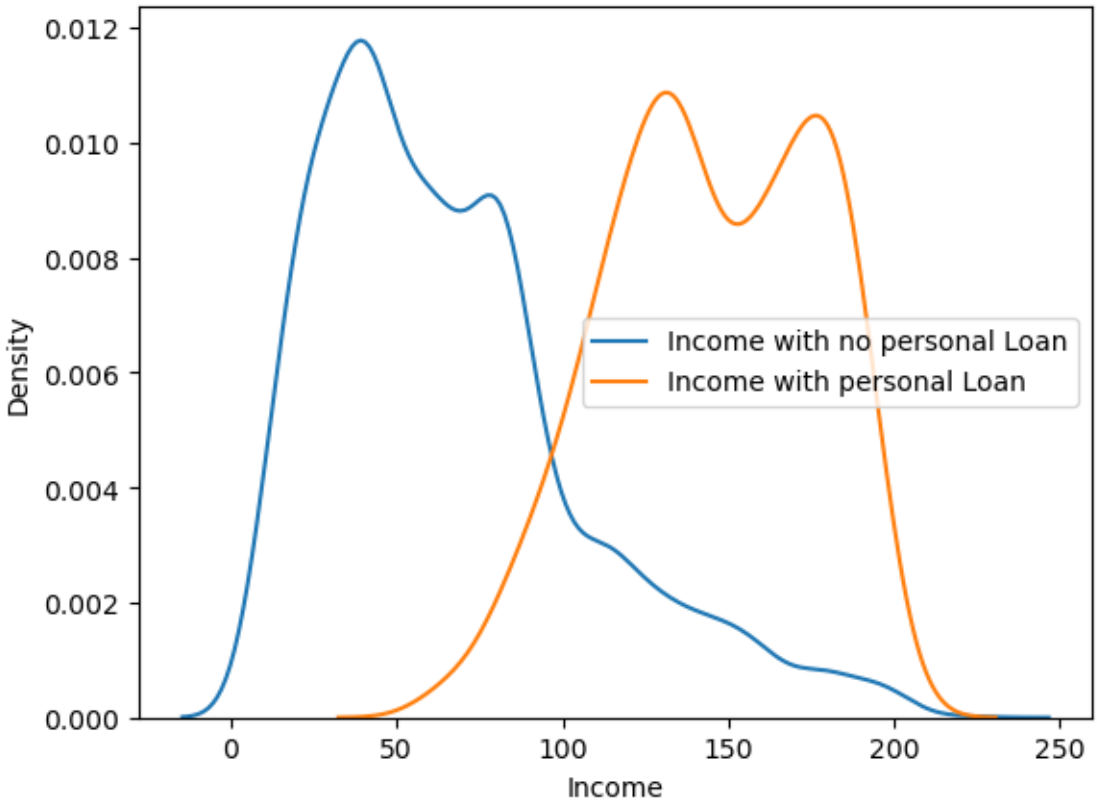
[56]:
```
px.box(data, x='Education',y='Income',facet_col='Personal Loan')
```

[57]:
```
data.columns
```

[57]:
```
Index(['Age', 'Income', 'Family', 'CCAvg', 'Education', 'Mortgage',
       'Personal Loan', 'Securities Account', 'CD Account', 'Online',
       'CreditCard', 'EDU', 'Account_holder_category'],
      dtype='object')
```

[58]:
```
sns.distplot(data[data['Personal Loan'] == 0]['Income'],␣
 ↪hist=False,label="Income with no personal Loan")
sns.distplot(data[data['Personal Loan'] == 1]['Income'], hist=False,␣
 ↪label="Income with personal Loan")
plt.legend()
```

[58]:
```
<matplotlib.legend.Legend at 0x262cfaa6950>
```

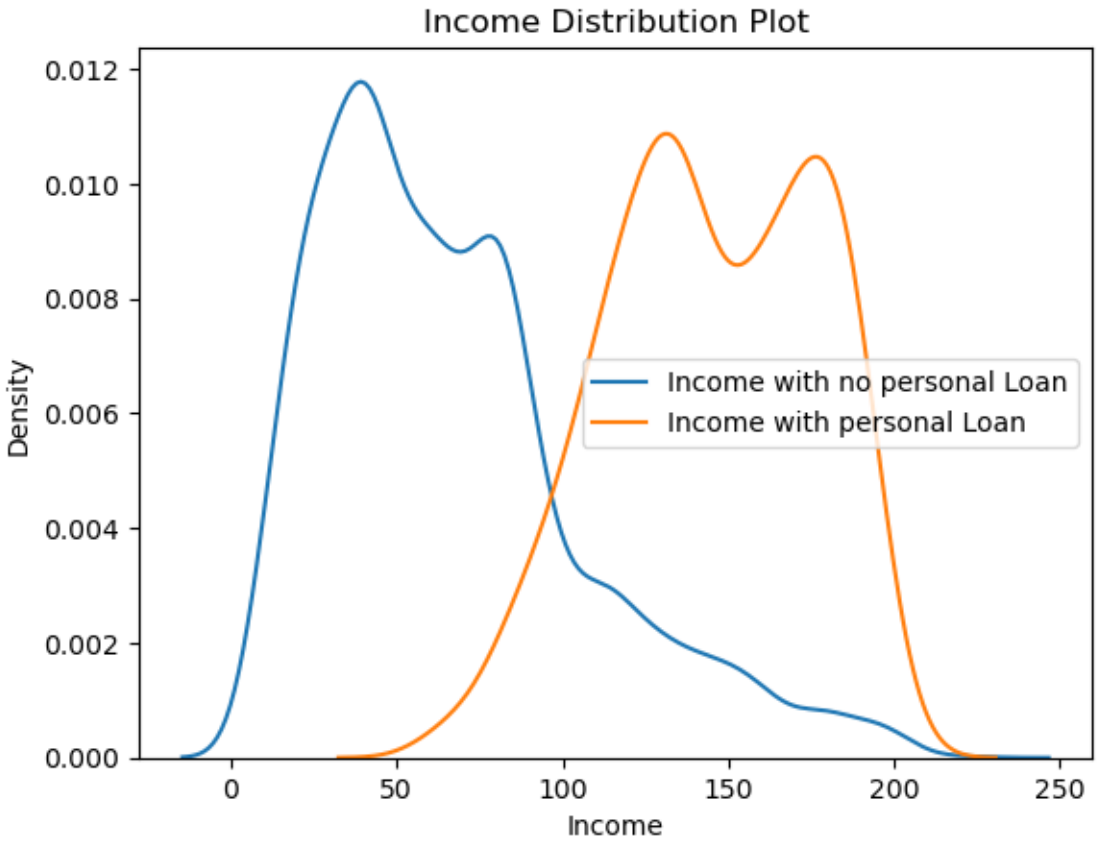


[59]:
```
data.columns
```

[59]:
```
Index(['Age', 'Income', 'Family', 'CCAvg', 'Education', 'Mortgage',
       'Personal Loan', 'Securities Account', 'CD Account', 'Online',
       'CreditCard', 'EDU', 'Account_holder_category'],
      dtype='object')
```

[87]:
```
def plot(col1, col2, label1, label2, title):
    sns.distplot(data[data[col2]==0][col1], hist=False,
                 label=label1)

    sns.distplot(data[data[col2]==1][col1], hist=False,
             label=label2)
    plt.legend()
    plt.title(title)
```

[88]:
```
plot('Income', 'Personal Loan','Income with no personal Loan', 'Income with␣
 ↪personal Loan', 'Income Distribution Plot')
```



[89]:
```
def plot(col1, col2, label1, label2, title):

    sns.distplot(data[data[col2]==0][col1], hist=False,
                 label=label1)

    sns.distplot(data[data[col2]==1][col1], hist=False,
             label=label2)
    plt.legend()
```

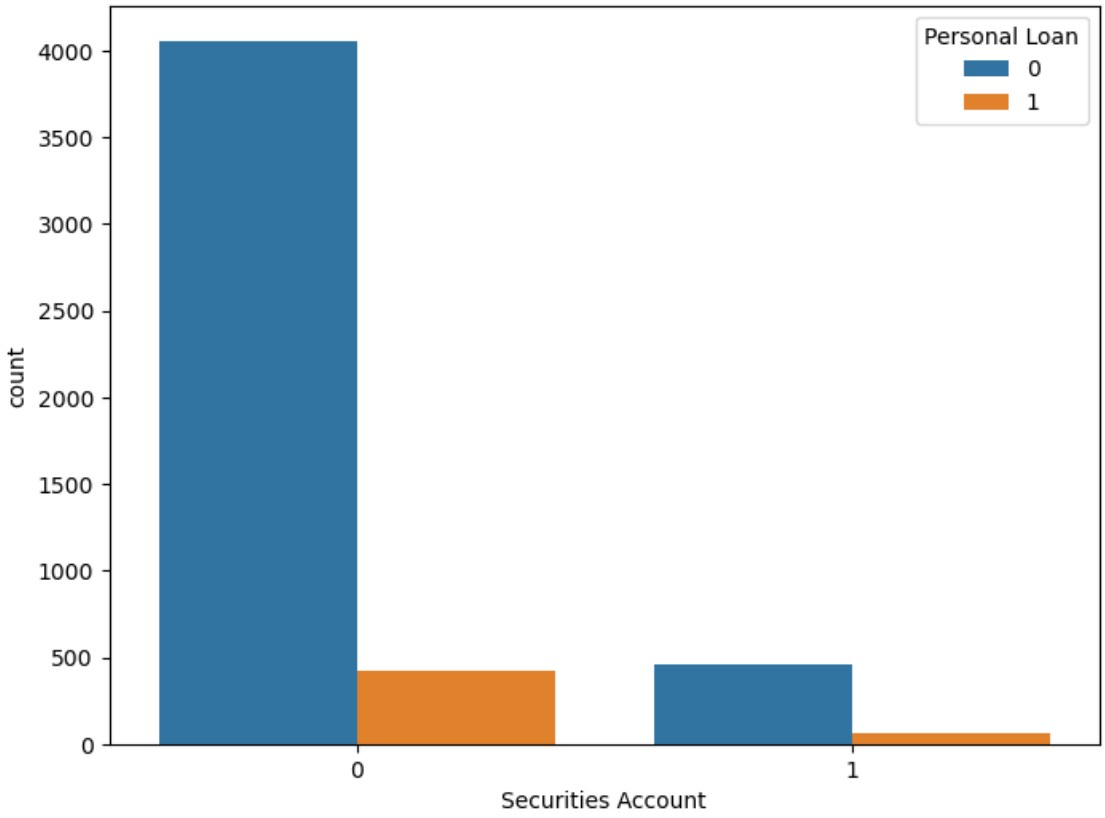
```
    plt.title(title)
    plt.show()
```

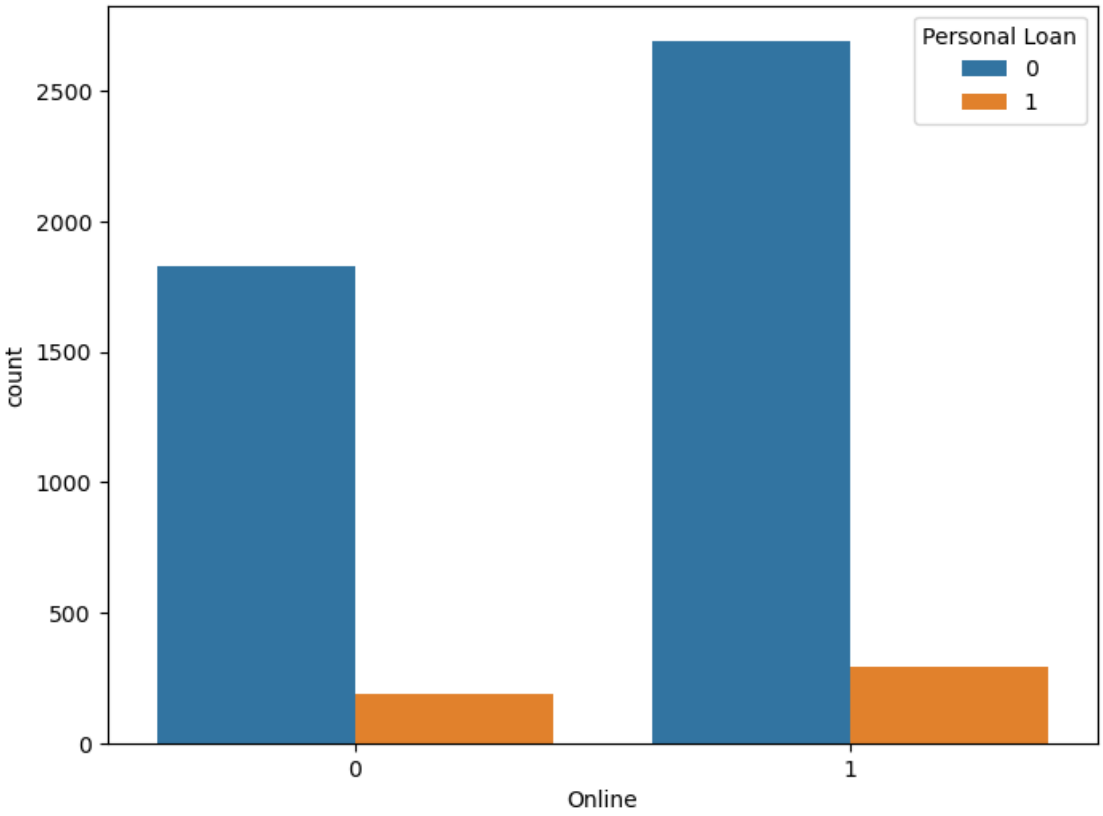
[62]:
```
data.columns
```

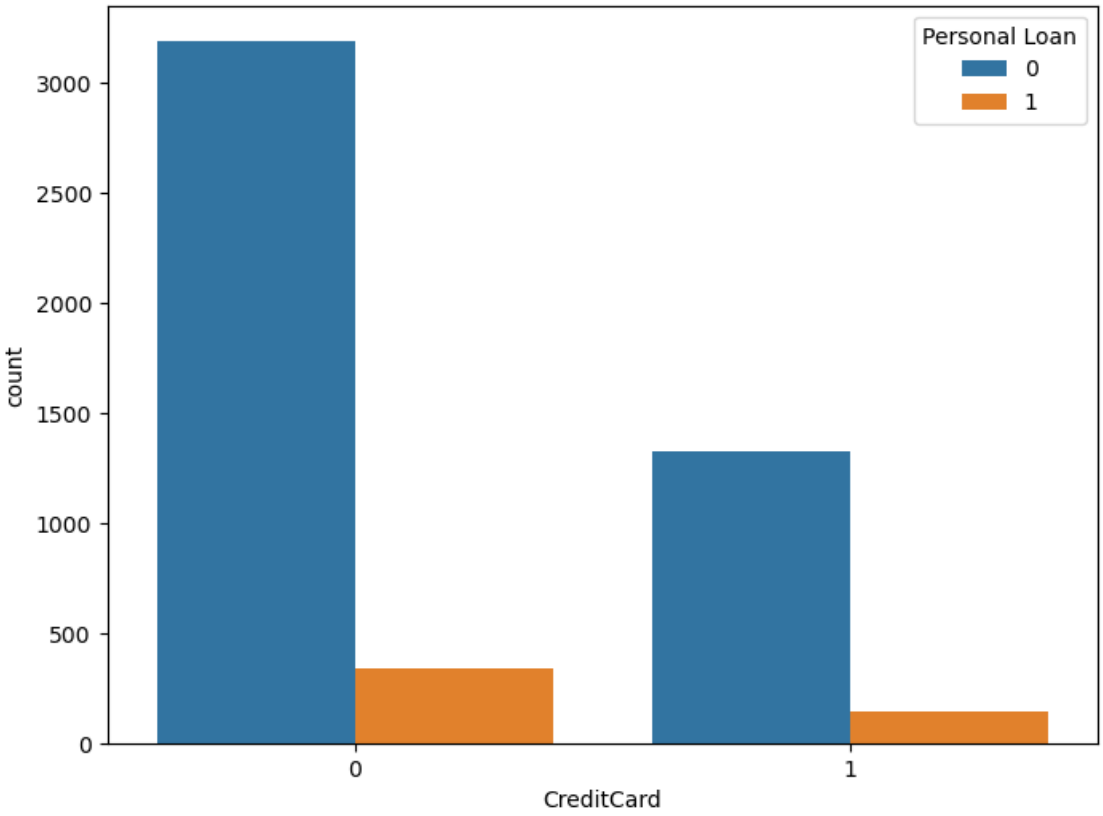
[62]:
```
Index(['Age', 'Income', 'Family', 'CCAvg', 'Education', 'Mortgage',
       'Personal Loan', 'Securities Account', 'CD Account', 'Online',
       'CreditCard', 'EDU', 'Account_holder_category'],
      dtype='object')
```

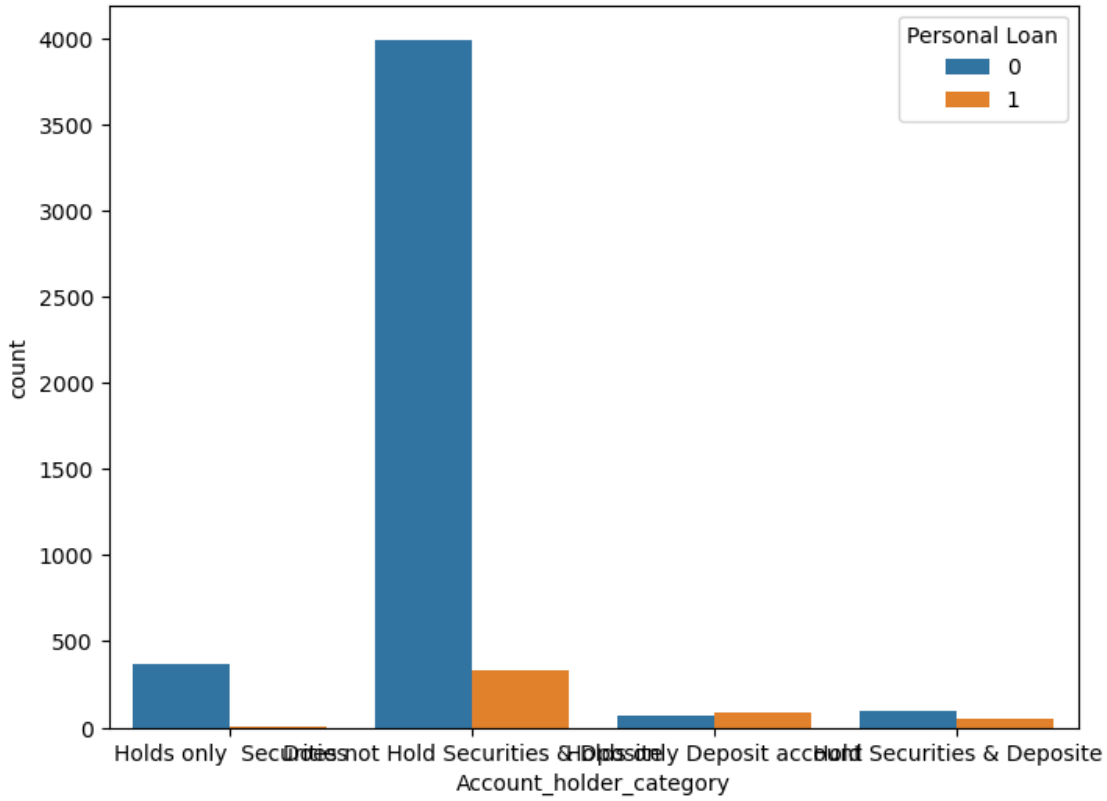
[65]:
```
col = ['Securities Account',
      'Online',
       'CreditCard',
       'Account_holder_category',]
```

[66]:
```
for i in col:
    plt.figure(figsize=(8,6))
    sns.countplot(x=i, data=data, hue='Personal Loan')
```




Personal Loan
0
1
2500
2000
1500
1000
500
0
count
0
1
Online


Personal Loan
0
1
3000
2500
2000
1500
1000
500
0
count
0
1
CreditCard



[67]:
```
from scipy.stats import zscore
```

[68]:
```
q1=data.quantile(0.25)
q3=data.quantile(0.75)
IQR=q3-q1

print(IQR)
```

```
Age                  20.0
Income               59.0
Family                2.0
Education             2.0
Mortgage            101.0
Personal Loan         0.0
Securities Account    0.0
CD Account            0.0
Online                1.0
CreditCard            1.0
dtype: float64
```

[75]:
```
# Log Normal Transform

data_1 = data[['Income']]
data_1 = np.log(data_1+1)
data_1
```

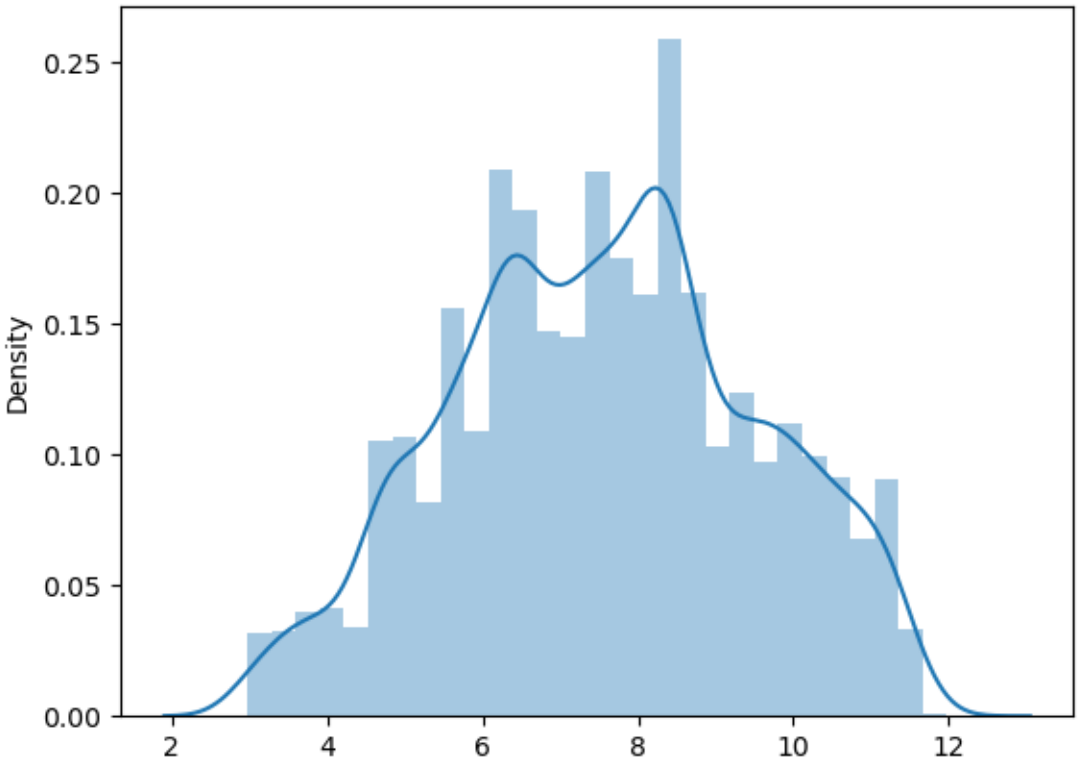
[75]:

	Income
0	3.912023
1	3.555348
2	2.484907
3	4.615121
4	3.828641
…	…
4995	3.713572
4996	2.772589
4997	3.218876
4998	3.912023
4999	4.430817

[5000 rows x 1 columns]

[70]:
```
#Power Transformer
from sklearn.preprocessing import PowerTransformer
```

[73]:
```
pt= PowerTransformer(method='yeo-johnson',standardize=False )
pt.fit(data['Income'].values.reshape(-1,1))
Income = pt.transform(data['Income'].values.reshape(-1,1))
sns.distplot(Income)
plt.show()
```


Density
0.25
0.20
0.15
0.10
0.05
0.00
2
4
6
8
10
12