Bank Loan Analyzer

Aishik Dasgupta

```
Linkedin : https://www.linkedin.com/in/aishik-dasgupta/
Github : https://github.com/AishikDasgupta
```

Importing Libraries

In [1]:

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

Reading the .csv File

In [4]:

```python
df = pd.read_csv("Bank_Personal_Loan_Modelling.csv")
df
```

Out[4]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan
0	1	25	1	49	91107	4	1.6	1	0	0
1	2	45	19	34	90089	3	1.5	1	0	0
2	3	39	15	11	94720	1	1.0	1	0	0
3	4	35	9	100	94112	1	2.7	2	0	0
4	5	35	8	45	91330	4	1.0	2	0	0
...	...	...	...	...	...	...	...	...	...	...
4995	4996	29	3	40	92697	1	1.9	3	0	0
4996	4997	30	4	15	92037	4	0.4	1	85	0
4997	4998	63	39	24	93023	2	0.3	3	0	0
4998	4999	65	40	49	90034	3	0.5	2	0	0
4999	5000	28	4	83	92612	3	0.8	1	0	0

5000 rows × 14 columns

Top 5 Values

In [5]:

```python
df.head(5)
```

Out[5]:

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

Bottom 5 Values

In [6]:

```
df.tail(5)
```

Out[6]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan
4995	4996	29	3	40	92697	1	1.9	3	0	0
4996	4997	30	4	15	92037	4	0.4	1	85	0
4997	4998	63	39	24	93023	2	0.3	3	0	0
4998	4999	65	40	49	90034	3	0.5	2	0	0
4999	5000	28	4	83	92612	3	0.8	1	0	0

Shape of Data

In [7]:

```
df.shape
```

Out[7]:

```
(5000, 14)
```

Cheking null Values

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
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
```

Columns Available

In [9]:

```
df.columns
```

Out[9]:

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

In [ ]:

```
## Data Types
```

In [10]:

```
df.dtypes
```

Out[10]:

```
ID                      int64
Age                     int64
Experience              int64
Income                  int64
ZIP Code                int64
Family                  int64
CCAvg                 float64
Education               int64
Mortgage                int64
Personal Loan           int64
Securities Account      int64
CD Account              int64
Online                  int64
CreditCard              int64
dtype: object
```

In [ ]:

```
## Summary of Data
```

In [11]:

```
summary_stats = df[['Age','Experience','Income','Family','Education']].desc
summary_stats
```

Out[11]:

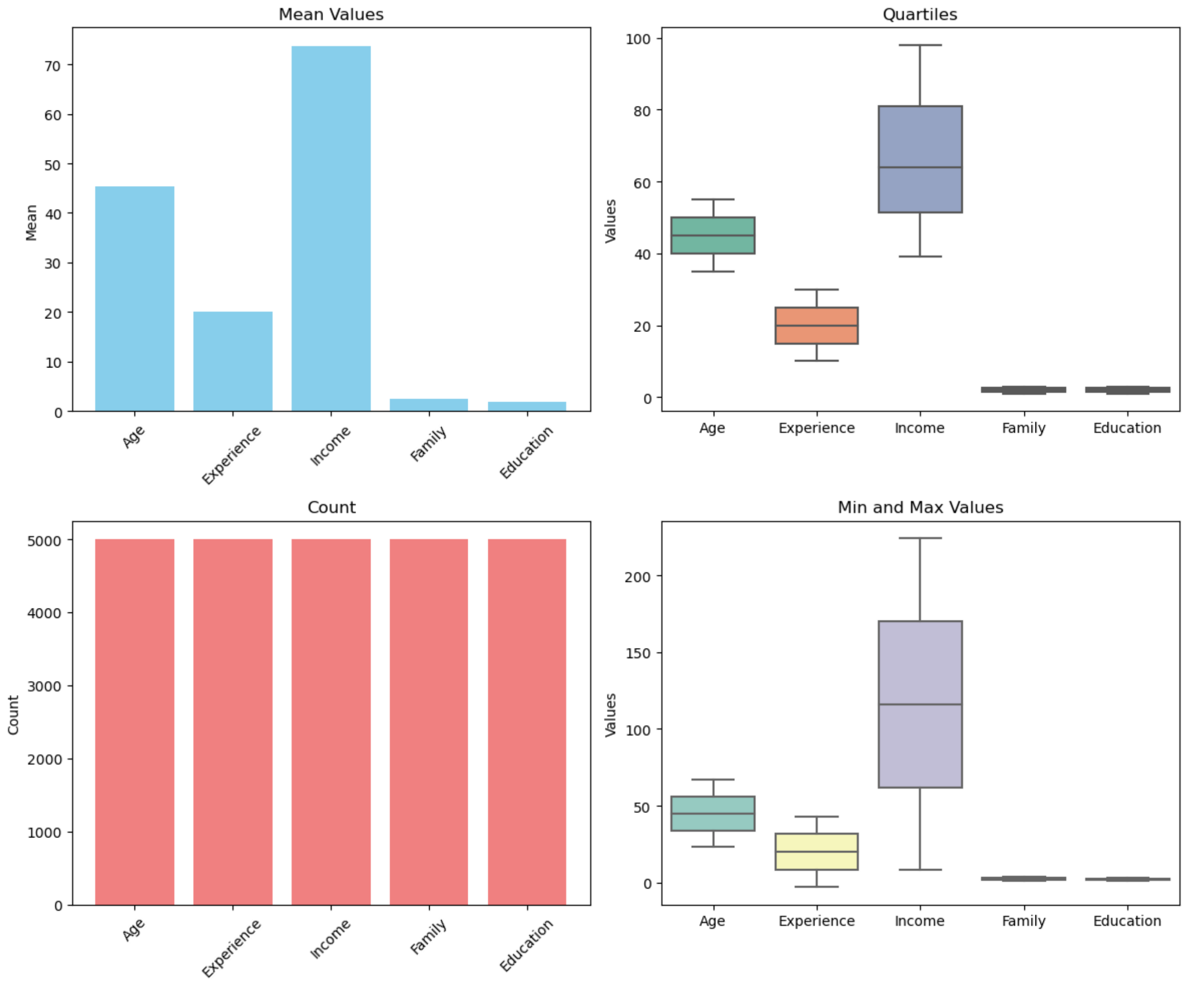
	Age	Experience	Income	Family	Education
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	45.338400	20.104600	73.774200	2.396400	1.881000
std	11.463166	11.467954	46.033729	1.147663	0.839869
min	23.000000	-3.000000	8.000000	1.000000	1.000000
25%	35.000000	10.000000	39.000000	1.000000	1.000000
50%	45.000000	20.000000	64.000000	2.000000	2.000000
75%	55.000000	30.000000	98.000000	3.000000	3.000000
max	67.000000	43.000000	224.000000	4.000000	3.000000

Subplots of the Summary

In [12]:

```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))

axes[0, 0].bar(summary_stats.columns, summary_stats.loc['mean'], color='sky
axes[0, 0].set_title('Mean Values')
axes[0, 0].set_ylabel('Mean')
axes[0, 0].tick_params(axis='x', rotation=45)
```

```
sns.boxplot(data=summary_stats.loc[['25%', '50%', '75%']], ax=axes[0, 1], pa
axes[0, 1].set_title('Quartiles')
axes[0, 1].set_ylabel('Values')

axes[1, 0].bar(summary_stats.columns, summary_stats.loc['count'], color='lig
axes[1, 0].set_title('Count')
axes[1, 0].set_ylabel('Count')
axes[1, 0].tick_params(axis='x', rotation=45)

sns.boxplot(data=summary_stats.loc[['min', 'max']], ax=axes[1, 1], palette=
axes[1, 1].set_title('Min and Max Values')
axes[1, 1].set_ylabel('Values')

plt.tight_layout()
plt.show()
```



Calculating the skewness of numerical columns

In [13]:

```
df.skew()
```

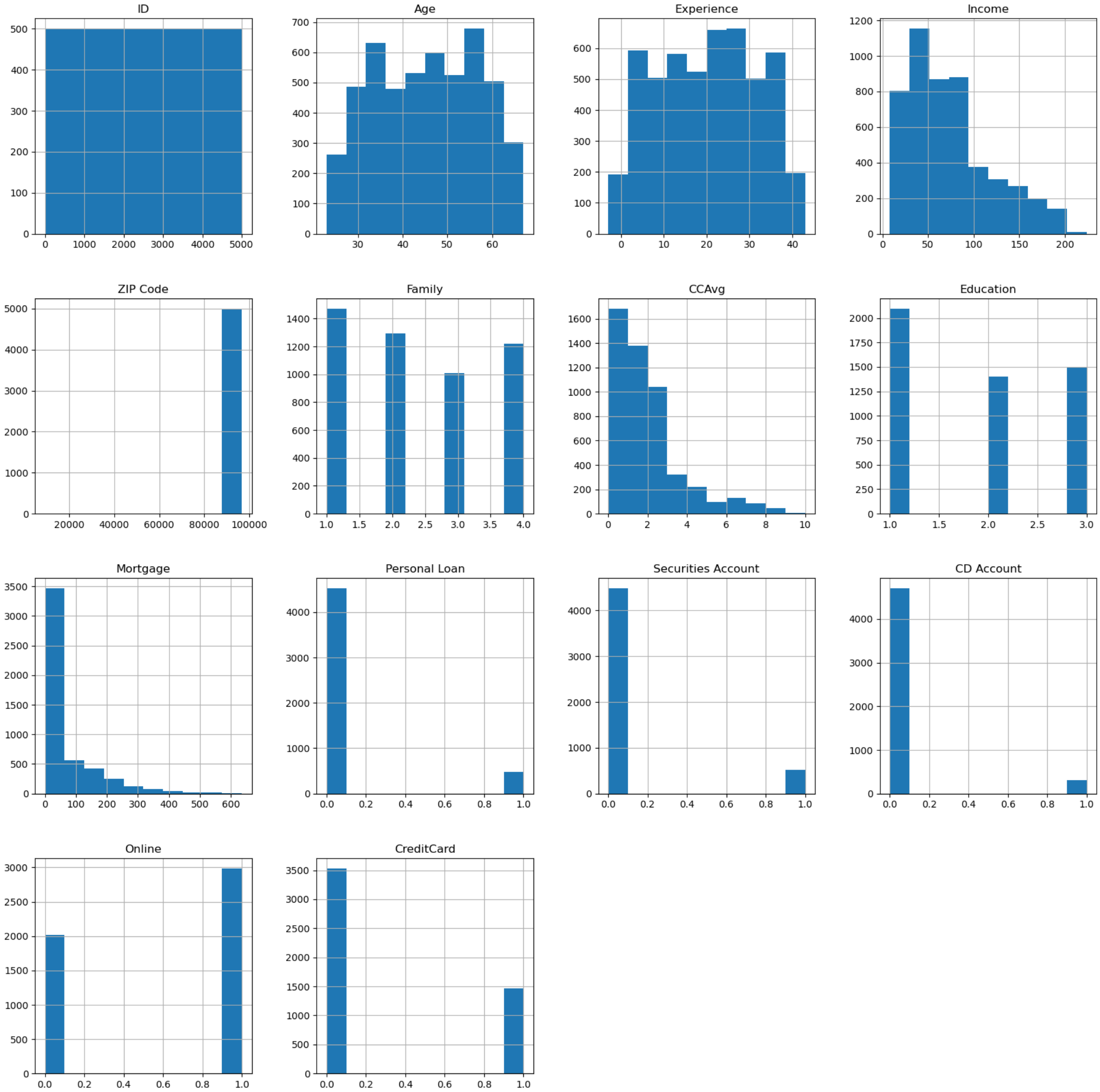
Out[13]:

```
ID                    0.000000
Age                  -0.029341
Experience           -0.026325
Income                0.841339
ZIP Code            -12.500221
Family                0.155221
CCAvg                 1.598443
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
```

Creating histograms for each column

In [14]:

```
df.hist(figsize = (20,20))
plt.show()
```

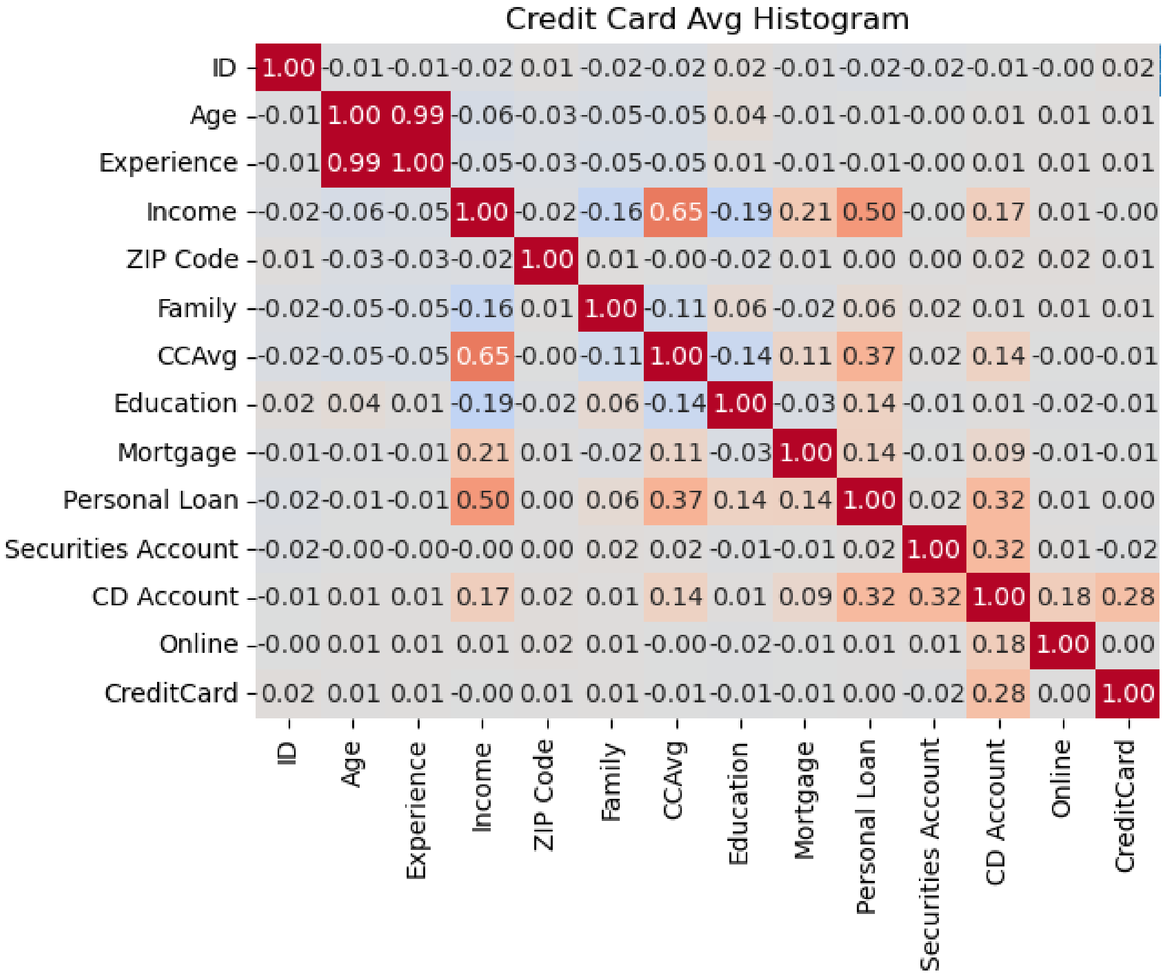


In [ ]:

```
#
```

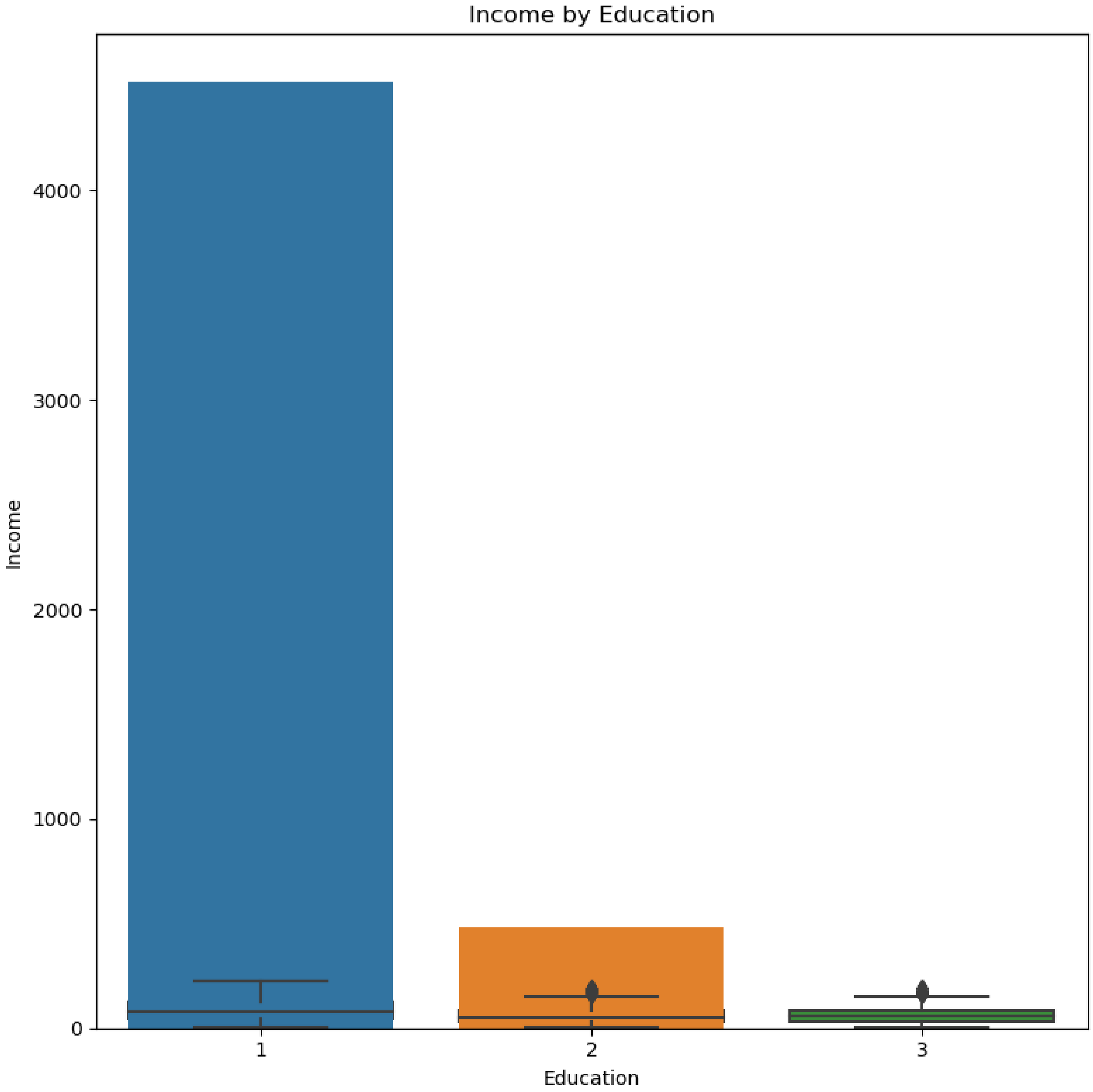
In [61]:

```
sns.boxplot(x='Education', y='Age', data=df)
plt.ylim(15, 80)
plt.title('Age by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Age')

plt.scatter(x='Income', y='CCAvg', data=df, s=5, alpha=0.5)
plt.xlim(0, 250)
plt.ylim(0, 10)
plt.title('Income vs Credit Card Avg')
plt.xlabel('Income')
plt.ylabel('Credit Card Avg')

plt.bar(x=[0, 1], height=df['Securities Account'].value_counts(), width=0.8
plt.title('Securities Account Values')
plt.xlabel('Securities Account')
plt.ylabel('Count')

sns.histplot(df['CCAvg'], kde=False, bins=20)
plt.xlim(0, 10)
plt.title('Credit Card Avg Histogram')
plt.xlabel('Credit Card Avg')
plt.ylabel('Frequency')

corr = df.corr()
ax = sns.heatmap(corr, annot=True, fmt=".2f", cmap="coolwarm",
                 vmin=-1, vmax=1, cbar=False)
plt.figure(figsize=(8,8))
plt.title('Correlations')

sns.countplot(x='Personal Loan', data=df)
plt.title('Personal Loan Count')

plt.hist(df['Age'], bins=20)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')

sns.boxplot(x='Education', y='Income', data=df)
plt.title('Income by Education')



plt.tight_layout()
plt.show()
```



Credit Card Avg Histogram

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
ID	1.00	-0.01	-0.01	-0.02	0.01	-0.02	-0.02	0.02	-0.01	-0.02	-0.02	-0.01	-0.00	0.02
Age	-0.01	1.00	0.99	-0.06	-0.03	-0.05	-0.05	0.04	-0.01	-0.01	-0.00	0.01	0.01	0.01
Experience	-0.01	0.99	1.00	-0.05	-0.03	-0.05	-0.05	0.01	-0.01	-0.01	-0.00	0.01	0.01	0.01
Income	-0.02	-0.06	-0.05	1.00	-0.02	-0.16	0.65	-0.19	0.21	0.50	-0.00	0.17	0.01	-0.00
ZIP Code	0.01	-0.03	-0.03	-0.02	1.00	0.01	-0.00	-0.02	0.01	0.00	0.00	0.02	0.02	0.01
Family	-0.02	-0.05	-0.05	-0.16	0.01	1.00	-0.11	0.06	-0.02	0.06	0.02	0.01	0.01	0.01
CCAvg	-0.02	-0.05	-0.05	0.65	-0.00	-0.11	1.00	-0.14	0.11	0.37	0.02	0.14	-0.00	-0.01
Education	0.02	0.04	0.01	-0.19	-0.02	0.06	-0.14	1.00	-0.03	0.14	-0.01	0.01	-0.02	-0.01
Mortgage	-0.01	-0.01	-0.01	0.21	0.01	-0.02	0.11	-0.03	1.00	0.14	-0.01	0.09	-0.01	-0.01
Personal Loan	-0.02	-0.01	-0.01	0.50	0.00	0.06	0.37	0.14	0.14	1.00	0.02	0.32	0.01	0.00
Securities Account	-0.02	-0.00	-0.00	-0.00	0.00	0.02	0.02	-0.01	-0.01	0.02	1.00	0.32	0.01	-0.02
CD Account	-0.01	0.01	0.01	0.17	0.02	0.01	0.14	0.01	0.09	0.32	0.32	1.00	0.18	0.28
Online	-0.00	0.01	0.01	0.01	0.02	0.01	-0.00	-0.02	-0.01	0.01	0.01	0.18	1.00	0.00
CreditCard	0.02	0.01	0.01	-0.00	0.01	0.01	-0.01	-0.01	-0.01	0.00	-0.02	0.28	0.00	1.00

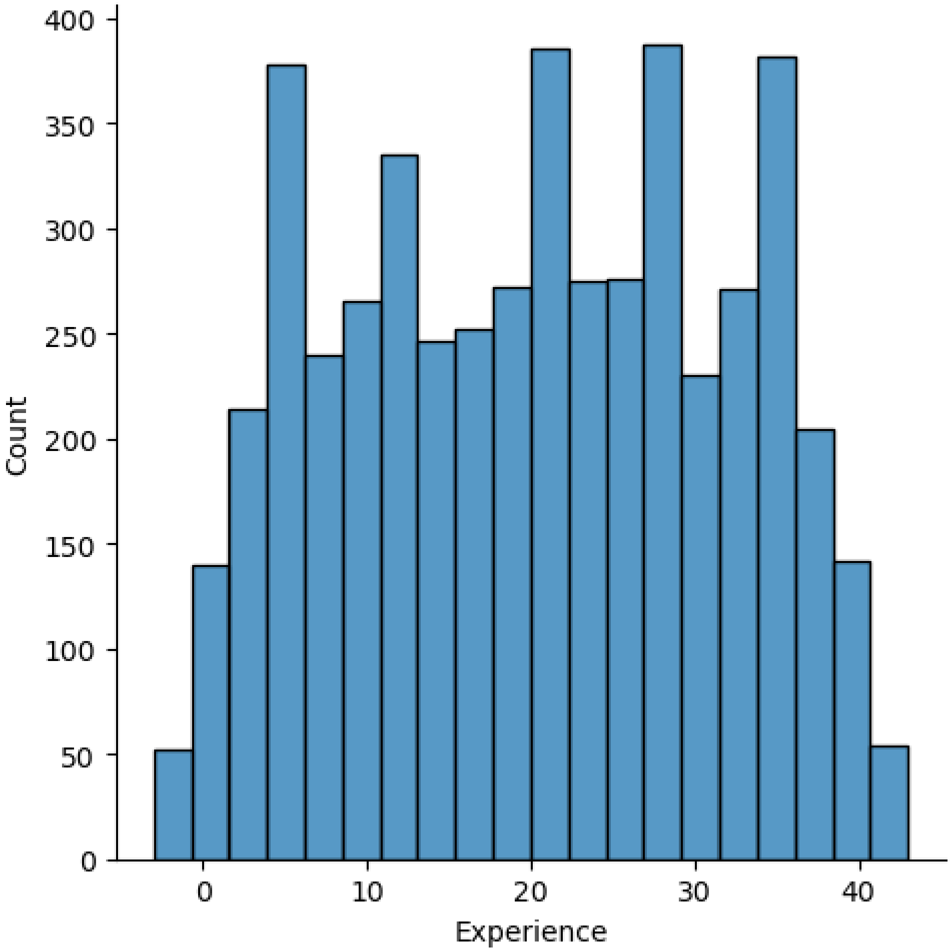


Creating a displot for 'Experience' Column

In [18]:

```
plt.figure(figsize = (16,12))
sns.displot(df['Experience'])
plt.show()
```

```
<Figure size 1600x1200 with 0 Axes>
```



Negative data in Experience Column

In [19]:
```
negative_exp = df[df['Experience']<0]
negative_exp
```

Out[19]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan
89	90	25	-1	113	94303	4	2.30	3	0	0
226	227	24	-1	39	94085	2	1.70	2	0	0
315	316	24	-2	51	90630	3	0.30	3	0	0
451	452	28	-2	48	94132	2	1.75	3	89	0
524	525	24	-1	75	93014	4	0.20	1	0	0
536	537	25	-1	43	92173	3	2.40	2	176	0
540	541	25	-1	109	94010	4	2.30	3	314	0
576	577	25	-1	48	92870	3	0.30	3	0	0
583	584	24	-1	38	95045	2	1.70	2	0	0
597	598	24	-2	125	92835	2	7.20	1	0	0
649	650	25	-1	82	92677	4	2.10	3	0	0
670	671	23	-1	61	92374	4	2.60	1	239	0
686	687	24	-1	38	92612	4	0.60	2	0	0
793	794	24	-2	150	94720	2	2.00	1	0	0
889	890	24	-2	82	91103	2	1.60	3	0	0
909	910	23	-1	149	91709	1	6.33	1	305	0
1173	1174	24	-1	35	94305	2	1.70	2	0	0
1428	1429	25	-1	21	94583	4	0.40	1	90	0
1522	1523	25	-1	101	94720	4	2.30	3	256	0
1905	1906	25	-1	112	92507	2	2.00	1	241	0
2102	2103	25	-1	81	92647	2	1.60	3	0	0
2430	2431	23	-1	73	92120	4	2.60	1	0	0
2466	2467	24	-2	80	94105	2	1.60	3	0	0
2545	2546	25	-1	39	94720	3	2.40	2	0	0
2618	2619	23	-3	55	92704	3	2.40	2	145	0
2717	2718	23	-2	45	95422	4	0.60	2	0	0
2848	2849	24	-1	78	94720	2	1.80	2	0	0
2876	2877	24	-2	80	91107	2	1.60	3	238	0
2962	2963	23	-2	81	91711	2	1.80	2	0	0
2980	2981	25	-1	53	94305	3	2.40	2	0	0
3076	3077	29	-1	62	92672	2	1.75	3	0	0
3130	3131	23	-2	82	92152	2	1.80	2	0	0
3157	3158	23	-1	13	94720	4	1.00	1	84	0
3279	3280	26	-1	44	94901	1	2.00	2	0	0
3284	3285	25	-1	101	95819	4	2.10	3	0	0
3292	3293	25	-1	13	95616	4	0.40	1	0	0
3394	3395	25	-1	113	90089	4	2.10	3	0	0
3425	3426	23	-1	12	91605	4	1.00	1	90	0

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan
3626	3627	24	-3	28	90089	4	1.00	3	0	0
3796	3797	24	-2	50	94920	3	2.40	2	0	0
3824	3825	23	-1	12	95064	4	1.00	1	0	0
3887	3888	24	-2	118	92634	2	7.20	1	0	0
3946	3947	25	-1	40	93117	3	2.40	2	0	0
4015	4016	25	-1	139	93106	2	2.00	1	0	0
4088	4089	29	-1	71	94801	2	1.75	3	0	0
4116	4117	24	-2	135	90065	2	7.20	1	0	0
4285	4286	23	-3	149	93555	2	7.20	1	0	0
4411	4412	23	-2	75	90291	2	1.80	2	0	0
4481	4482	25	-2	35	95045	4	1.00	3	0	0
4514	4515	24	-3	41	91768	4	1.00	3	0	0
4582	4583	25	-1	69	92691	3	0.30	3	0	0
4957	4958	29	-1	50	95842	2	1.75	3	0	0

In [20]:

```
negative_exp.head()
```

Out[20]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Se A
89	90	25	-1	113	94303	4	2.30	3	0	0	
226	227	24	-1	39	94085	2	1.70	2	0	0	
315	316	24	-2	51	90630	3	0.30	3	0	0	
451	452	28	-2	48	94132	2	1.75	3	89	0	
524	525	24	-1	75	93014	4	0.20	1	0	0	

Total number of negative data

In [21]:

```
negative_exp.shape
```

Out[21]:

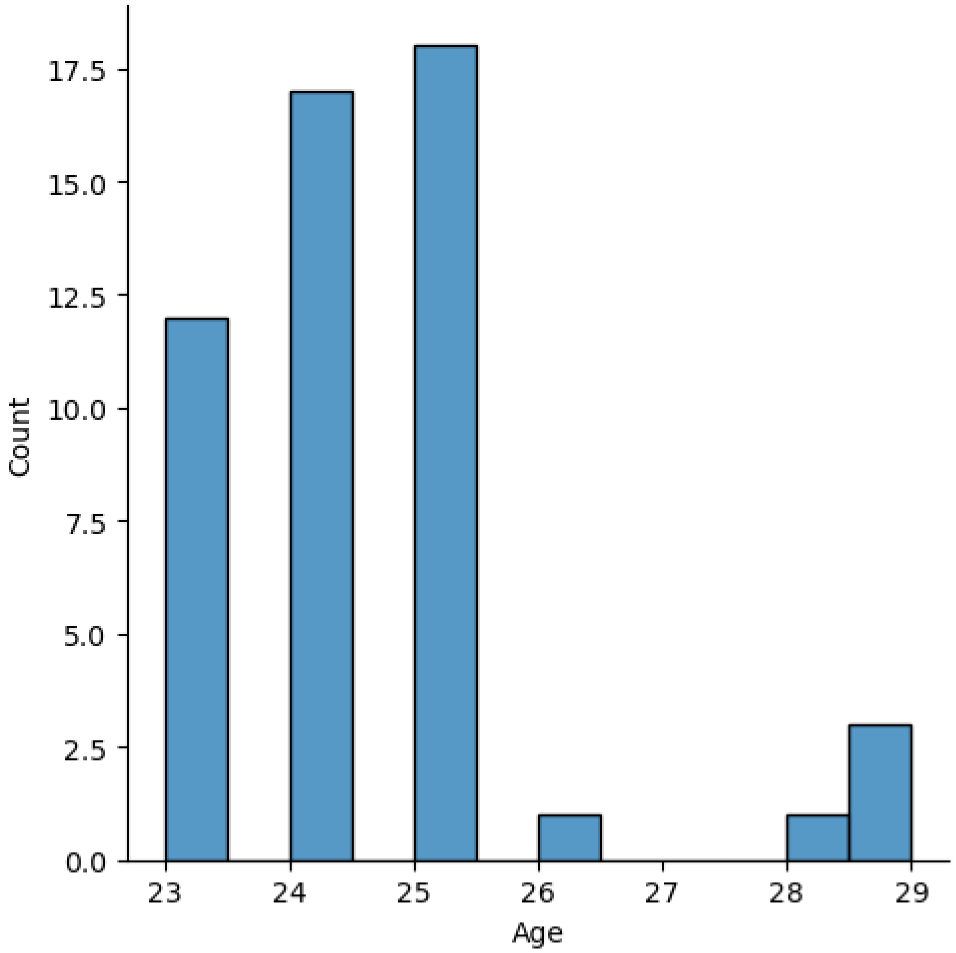
```
(52, 14)
```

In [ ]:

```
## Creating a distribution plot (displot) for the 'Age' column
```

In [22]:

```
sns.displot(negative_exp['Age'])
plt.show()
```



Mean Count of negative experice data

In [23]:
```
negative_exp['Experience'].mean()
```

Out[23]:
```
-1.4423076923076923
```

Size of negative experience data

In [24]:
```
negative_exp.size
```

Out[24]:
```
728
```

In [25]:
```
print('There are {} records which has negative values for experince, approx
```

```
There are 728 records which has negative values for experince, approx 1.04
%
```

Creating a copy of a DataFrame df and assign it to a new variable data

In [26]:
```
data = df.copy()
data
```

Out[26]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan
0	1	25	1	49	91107	4	1.6	1	0	0
1	2	45	19	34	90089	3	1.5	1	0	0
2	3	39	15	11	94720	1	1.0	1	0	0
3	4	35	9	100	94112	1	2.7	2	0	0
4	5	35	8	45	91330	4	1.0	2	0	0
...	...	...	...	...	...	...	...	...	...	...
4995	4996	29	3	40	92697	1	1.9	3	0	0
4996	4997	30	4	15	92037	4	0.4	1	85	0
4997	4998	63	39	24	93023	2	0.3	3	0	0
4998	4999	65	40	49	90034	3	0.5	2	0	0
4999	5000	28	4	83	92612	3	0.8	1	0	0

5000 rows × 14 columns

In [27]:

```
data.shape
```

Out[27]:

```
(5000, 14)
```

Using NumPy function to replace values in the 'Experience' column of the DataFrame 'data'

with the mean of the 'Experience'

column where the original values are less than 0

In [28]:

```
data['Experience'] = np.where(data['Experience']<0 , data['Experience'].mea
```

Filter rows in the DataFrame 'data' where the 'Experience' column has negative values

In [29]:

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

Out[29]:

ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securiti Accou

Calculating the correlation matrix

In [30]:
```
data.corr()
```

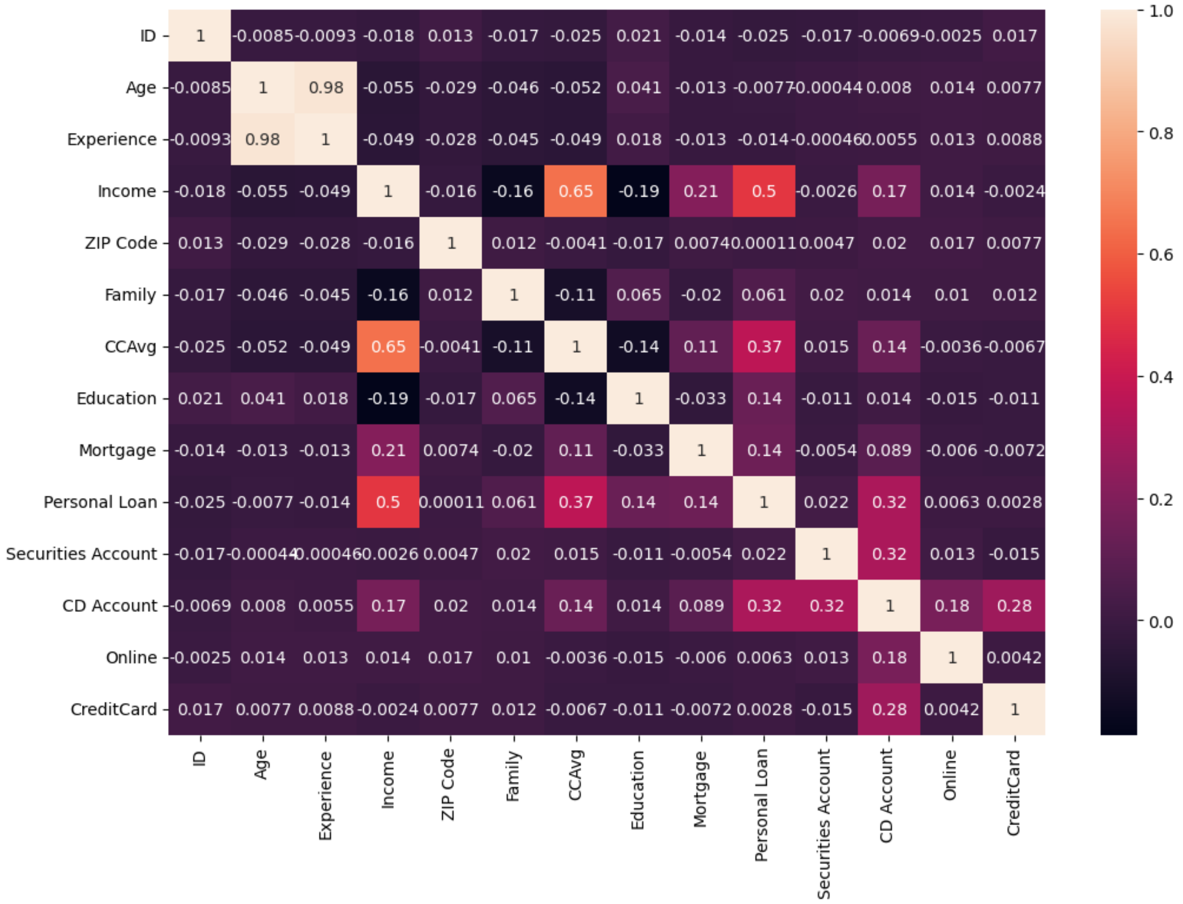
Out[30]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Educat
ID	1.000000	-0.008473	-0.009344	-0.017695	0.013432	-0.016797	-0.024675	0.0214
Age	-0.008473	1.000000	0.977008	-0.055269	-0.029216	-0.046418	-0.052012	0.0413
Experience	-0.009344	0.977008	1.000000	-0.049054	-0.028488	-0.045488	-0.048708	0.0180
Income	-0.017695	-0.055269	-0.049054	1.000000	-0.016410	-0.157501	0.645984	-0.1875
ZIP Code	0.013432	-0.029216	-0.028488	-0.016410	1.000000	0.011778	-0.004061	-0.0173
Family	-0.016797	-0.046418	-0.045488	-0.157501	0.011778	1.000000	-0.109275	0.0649
CCAvg	-0.024675	-0.052012	-0.048708	0.645984	-0.004061	-0.109275	1.000000	-0.1361
Education	0.021463	0.041334	0.018097	-0.187524	-0.017377	0.064929	-0.136124	1.0000
Mortgage	-0.013920	-0.012539	-0.013378	0.206806	0.007383	-0.020445	0.109905	-0.0333
Personal Loan	-0.024801	-0.007726	-0.014045	0.502462	0.000107	0.061367	0.366889	0.1367
Securities Account	-0.016972	-0.000436	-0.000462	-0.002616	0.004704	0.019994	0.015086	-0.0108
CD Account	-0.006909	0.008043	0.005502	0.169738	0.019972	0.014110	0.136534	0.0139
Online	-0.002528	0.013702	0.013455	0.014206	0.016990	0.010354	-0.003611	-0.0150
CreditCard	0.017028	0.007681	0.008833	-0.002385	0.007691	0.011588	-0.006689	-0.0110

Creating a heatmap of the correlation matrix using Seaborn (sns) and Matplotlib (plt)

In [31]:
```
plt.figure(figsize = (12,8))
sns.heatmap(data.corr(), annot = True)
```

Out[31]:
```
<AxesSubplot:>
```



Dropping the 'Experience' column from DataFrame 'data' using the data.drop()

In [32]:

```
data = data.drop(['Experience'], axis=1)
data
```

Out[32]:

	ID	Age	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	A
0	1	25	49	91107	4	1.6	1	0	0	1	
1	2	45	34	90089	3	1.5	1	0	0	1	
2	3	39	11	94720	1	1.0	1	0	0	0	
3	4	35	100	94112	1	2.7	2	0	0	0	
4	5	35	45	91330	4	1.0	2	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	
4995	4996	29	40	92697	1	1.9	3	0	0	0	
4996	4997	30	15	92037	4	0.4	1	85	0	0	
4997	4998	63	24	93023	2	0.3	3	0	0	0	
4998	4999	65	49	90034	3	0.5	2	0	0	0	
4999	5000	28	83	92612	3	0.8	1	0	0	0	

5000 rows × 13 columns

In [33]:

```python
data.head()
```

Out[33]:

	ID	Age	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CI Accoun
0	1	25	49	91107	4	1.6	1	0	0	1	(
1	2	45	34	90089	3	1.5	1	0	0	1	(
2	3	39	11	94720	1	1.0	1	0	0	0	(
3	4	35	100	94112	1	2.7	2	0	0	0	(
4	5	35	45	91330	4	1.0	2	0	0	0	(

In [34]:

```python
data['Education'].unique()
```

Out[34]:

```
array([1, 2, 3])
```

Defining a Python function called experience(x) that takes an input x

In [35]:

```python
def experience(x):
    if x==1:
        return "UnderGraduate"
    if x==2:
        return "Graduate"
    else:
        return "Working Professionals"
```

In [36]:

```python
data['EDU'] = data['Education'].apply(experience)
```

In [37]:

```python
data.head()
```

Out[37]:

	ID	Age	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CI Accoun
0	1	25	49	91107	4	1.6	1	0	0	1	(
1	2	45	34	90089	3	1.5	1	0	0	1	(
2	3	39	11	94720	1	1.0	1	0	0	0	(
3	4	35	100	94112	1	2.7	2	0	0	0	(
4	5	35	45	91330	4	1.0	2	0	0	0	(

In [38]:

```python
data['EDU'].unique()
```

Out[38]:

```
array(['UnderGraduate', 'Graduate', 'Working Professionals'], dtype=object)
```

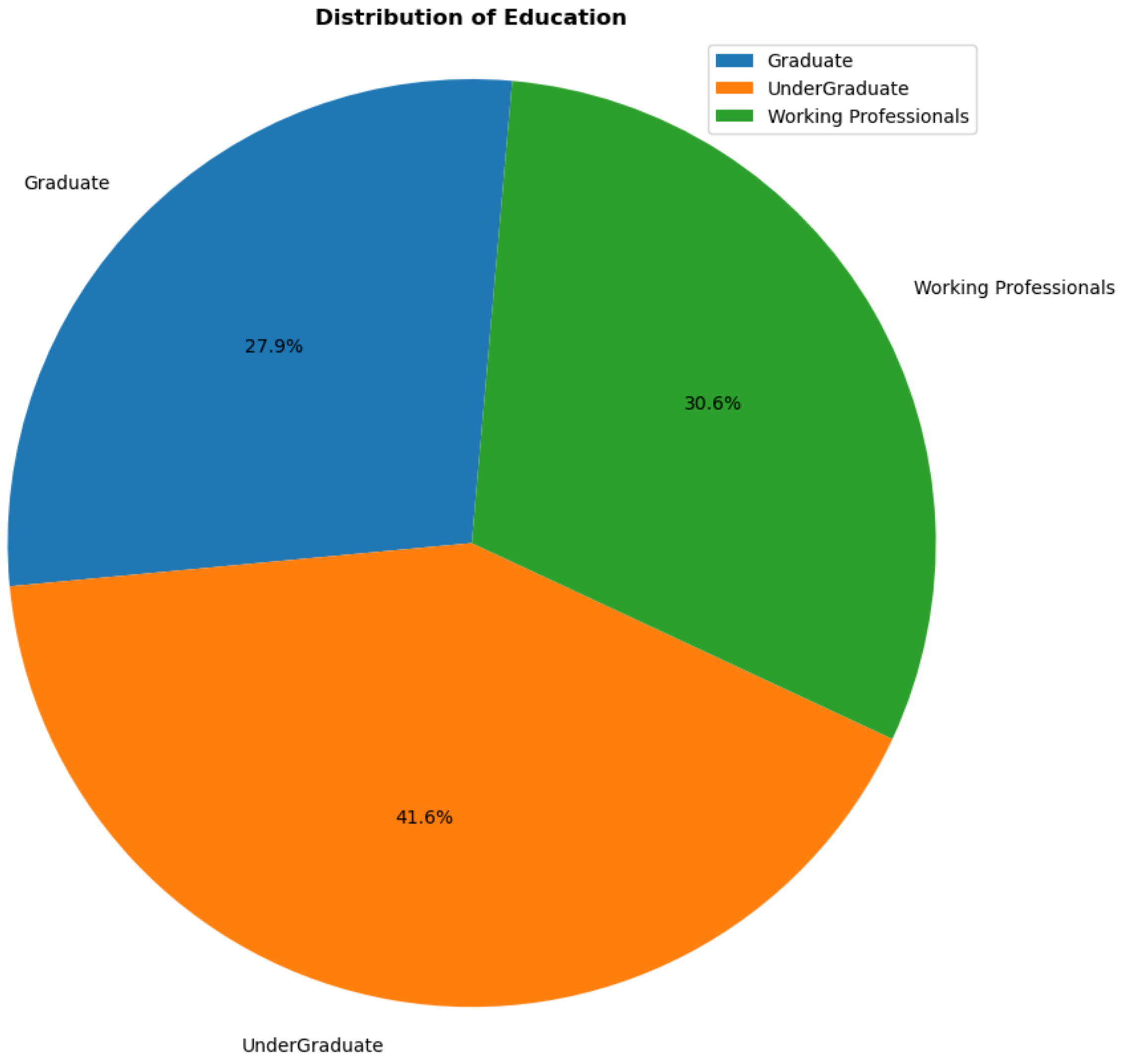
Grouping DataFrame 'data' by the 'EDU' column and then calculating the sum of the 'Age' column within each group

```
In [39]: education_dis = data.groupby('EDU')['Age'].sum()
         education_dis
```

```
Out[39]: EDU
         Graduate                 63191
         UnderGraduate            94244
         Working Professionals    69257
         Name: Age, dtype: int64
```

Creating a pie chart to visualize the distribution of education categories based on the'EDU' column

```
In [40]: plt.figure(figsize = (10,10))
         plt.pie(education_dis, labels = education_dis.index, autopct = '%1.1f%%', s
         plt.axis('equal')
         plt.title('Distribution of Education', fontweight = 'bold')
         plt.legend(education_dis.index, loc='upper right')
         plt.show()
```



Retrieving the unique values present in the 'Income' column

In [41]:

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

Out[41]:

```
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])
```

In [42]:

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

Out[42]:

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

In [43]:

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

Out[43]:

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

Defining a Python function called security(y) that takes a DataFrame y as input and categorizes individuals into different groups based on the values of the 'Securities Account' and 'CD Account' columns

In [44]:

```
def security(y):
    if(y['Securities Account'] == 1) & (y['CD Account'] == 1):
        return "Both Security and Deposit Account"
    if(y['Securities Account'] == 0) & (y['CD Account'] == 0):
        return "No Account"
    if(y['Securities Account'] == 1) & (y['CD Account'] == 0):
        return "Only Security Account"
    if(y['Securities Account'] == 0) & (y['CD Account'] == 1):
        return "Only Deposit Account"
```

In [45]:

```
data['Account_Holder_Category'] = data.apply(security, axis = 1)
```

In [46]:

```
data.head()
```

Out[46]:

	ID	Age	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CI Accoun
0	1	25	49	91107	4	1.6	1	0	0	1	(
1	2	45	34	90089	3	1.5	1	0	0	1	(
2	3	39	11	94720	1	1.0	1	0	0	0	(
3	4	35	100	94112	1	2.7	2	0	0	0	(
4	5	35	45	91330	4	1.0	2	0	0	0	(

Count the occurrences of unique values in the 'Account_Holder_Category' column

In [47]:

```
account_values = data['Account_Holder_Category'].value_counts()
account_values
```

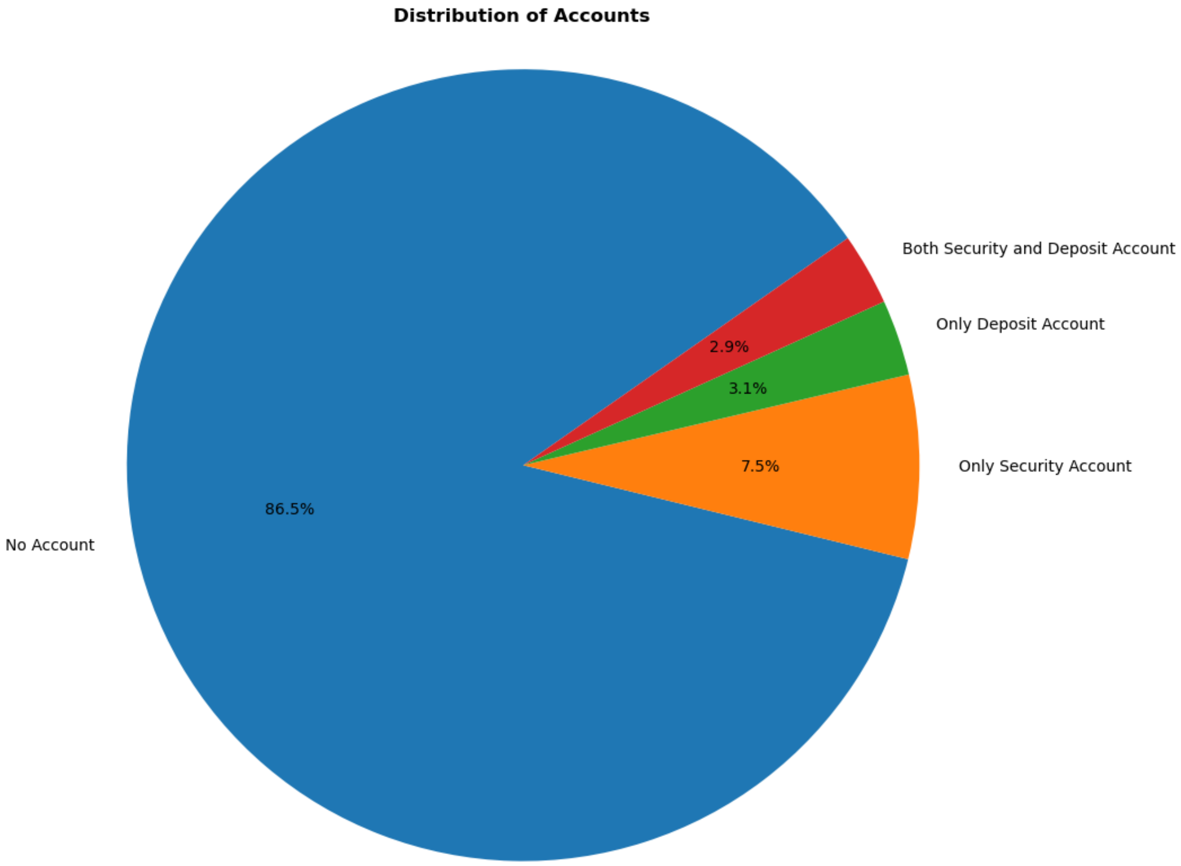
Out[47]:

```
No Account                          4323
Only Security Account                375
Only Deposit Account                 155
Both Security and Deposit Account    147
Name: Account_Holder_Category, dtype: int64
```

Creating a pie chart to visualize the distribution of account holder categories based on the 'Account_Holder_Category' column

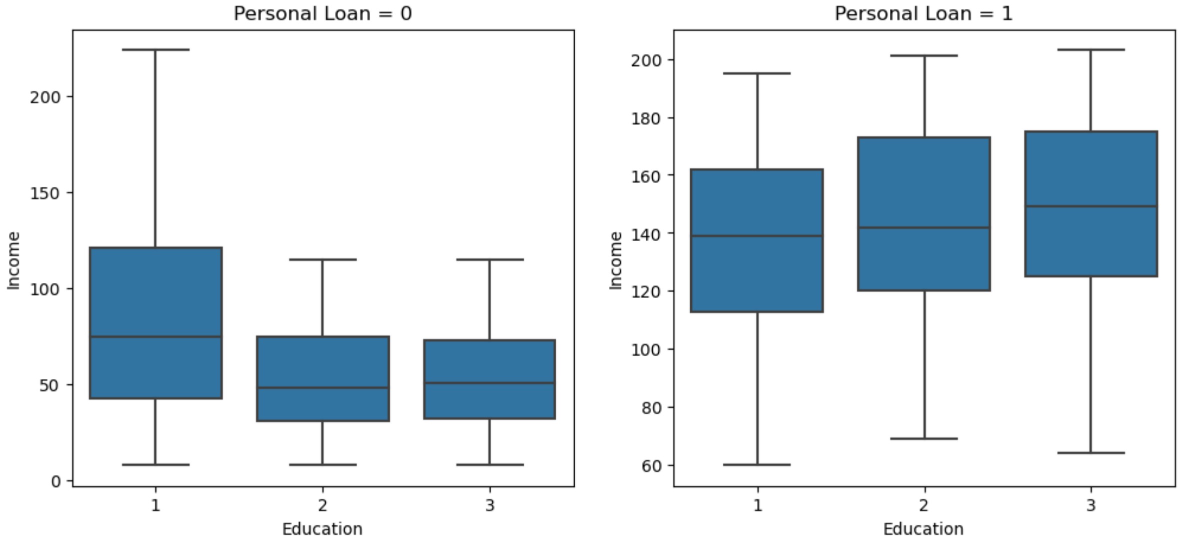
In [48]:

```
plt.figure(figsize = (10,10))
plt.pie(account_values, labels = account_values.index, autopct = '%1.1f%%',
plt.axis('equal')
plt.title('Distribution of Accounts', fontweight = 'bold')

plt.show()
```



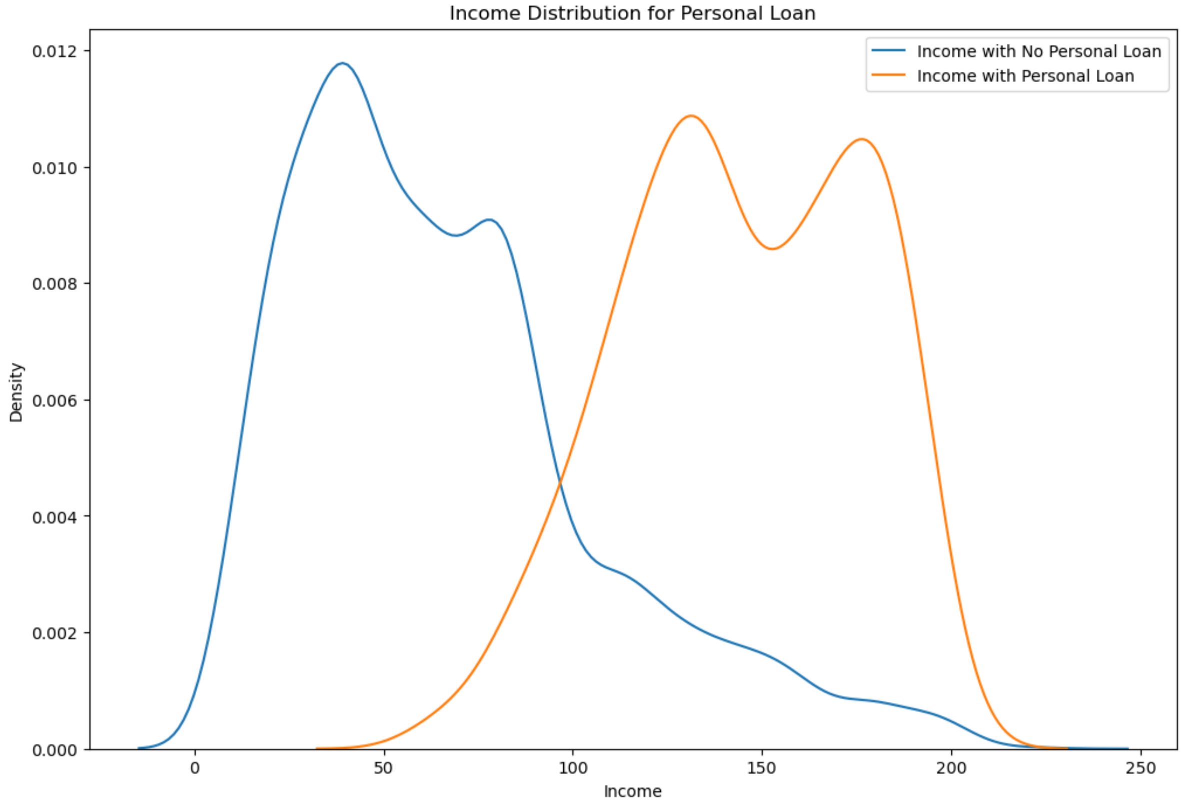
Creating two separate boxplot visualizations based on the 'Personal Loan' column

In [49]:
```
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

sns.boxplot(data=data[data['Personal Loan'] == 0], x='Education', y='Income
axes[0].set_title("Personal Loan = 0")
axes[0].legend().set_visible(False)

sns.boxplot(data=data[data['Personal Loan'] == 1], x='Education', y='Income
axes[1].set_title("Personal Loan = 1")
axes[1].legend().set_visible(False)

plt.show()
```



Creating a Kernel Density Estimation (KDE) plot to visualize the distribution of income for two groups: individuals with no personal loan ('Personal Loan' equals 0) and individuals with a personal loan ('Personal Loan' equals 1)

```
In [50]: plt.figure(figsize=(12, 8))
         sns.kdeplot(data=data[data['Personal Loan'] == 0]['Income'] ,label='Income
         sns.kdeplot(data=data[data['Personal Loan'] == 1]['Income'] ,label='Income
         plt.title("Income Distribution for Personal Loan")
         plt.xlabel("Income")
         plt.ylabel("Density")
         plt.legend()
         plt.show()
```



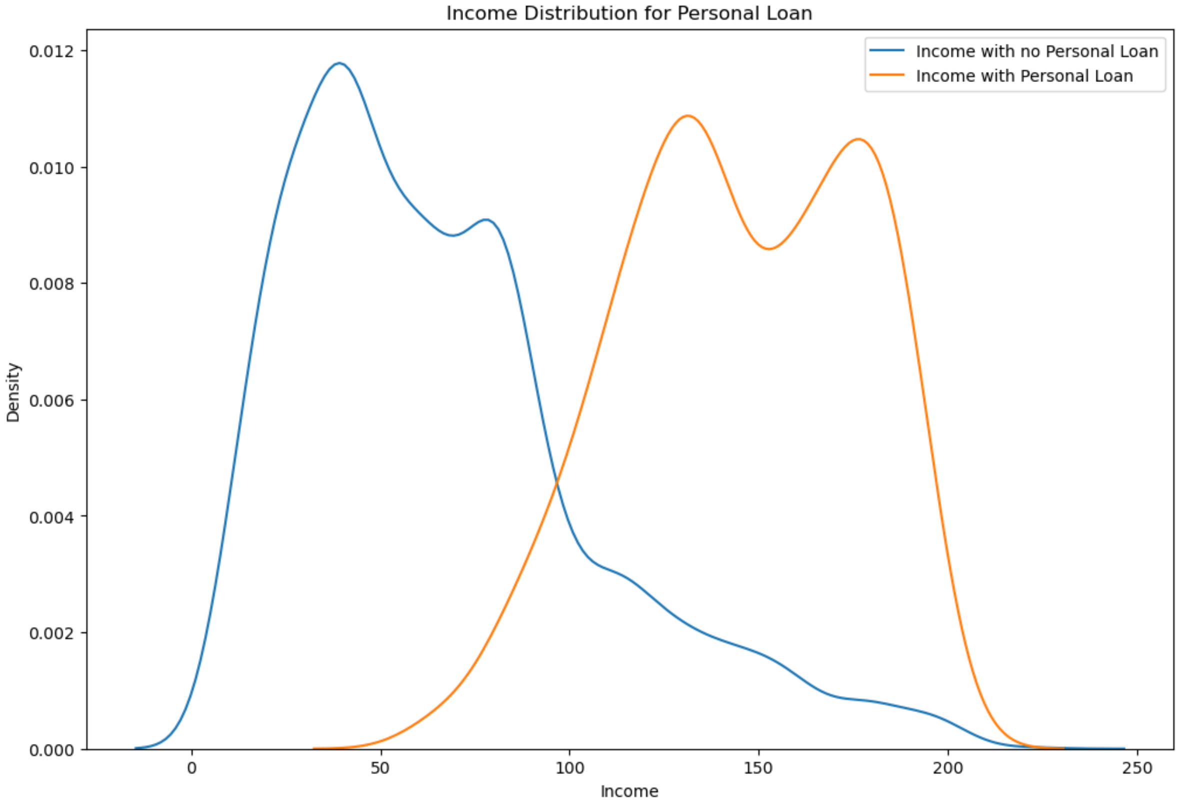
In [51]:

```
def plot(col1, col2, label1, label2, title):
    plt.figure(figsize=(12, 8))

    sns.kdeplot(data=data[data[col2] == 0][col1], label=label1)
    sns.kdeplot(data=data[data[col2] == 1][col1], label=label2)

    plt.legend()
    plt.title(title)
    plt.xlabel(col1)
    plt.ylabel("Density")
    plt.show()
```

Calling the plot function to create a KDE plot that visualizes the distribution of 'Income' based on the presence or absence of a 'Personal Loan'

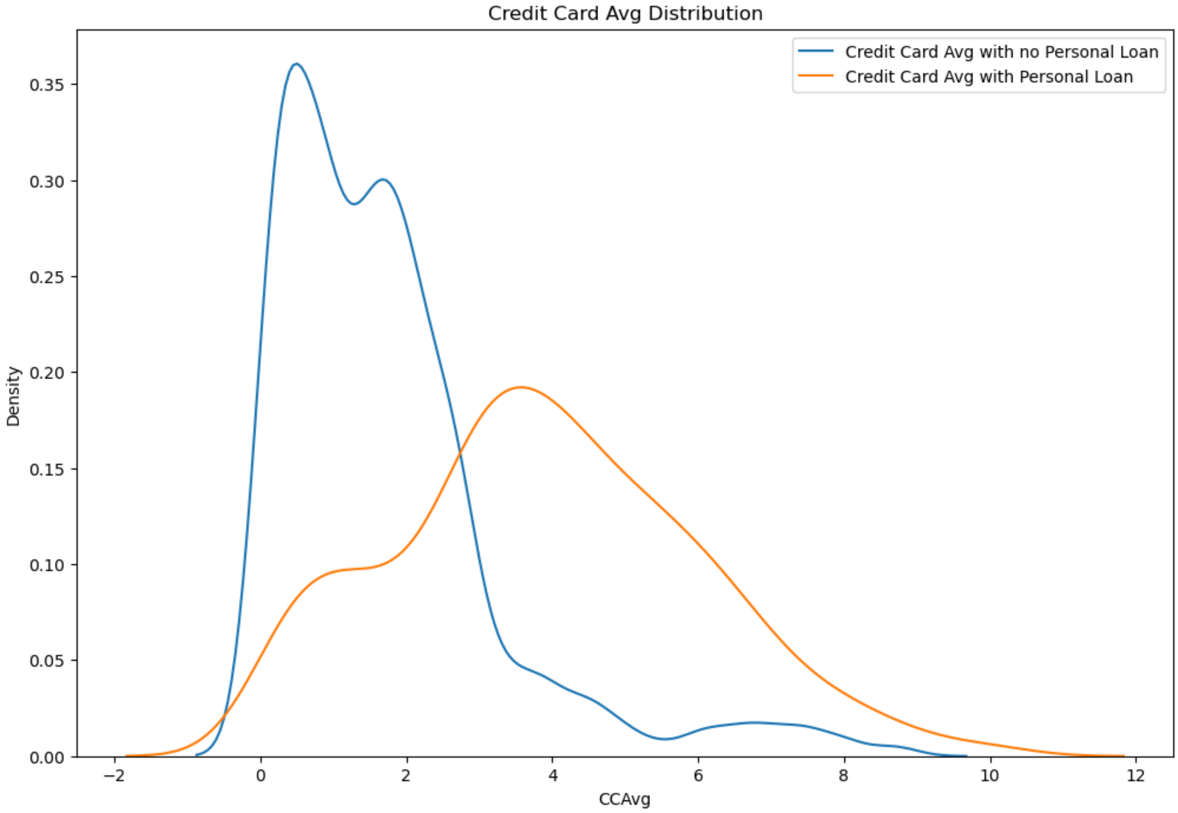
In [52]:

```
plot('Income', 'Personal Loan', 'Income with no Personal Loan', 'Income wit
```



Calling the plot function to create a KDE plot that visualizes the distribution of 'CCAvg' (Credit Card Average) based on the presence or absence of a 'Personal Loan.'

```
In [53]: plot('CCAvg', 'Personal Loan',
              'Credit Card Avg with no Personal Loan', 'Credit Card Avg with Personal
```



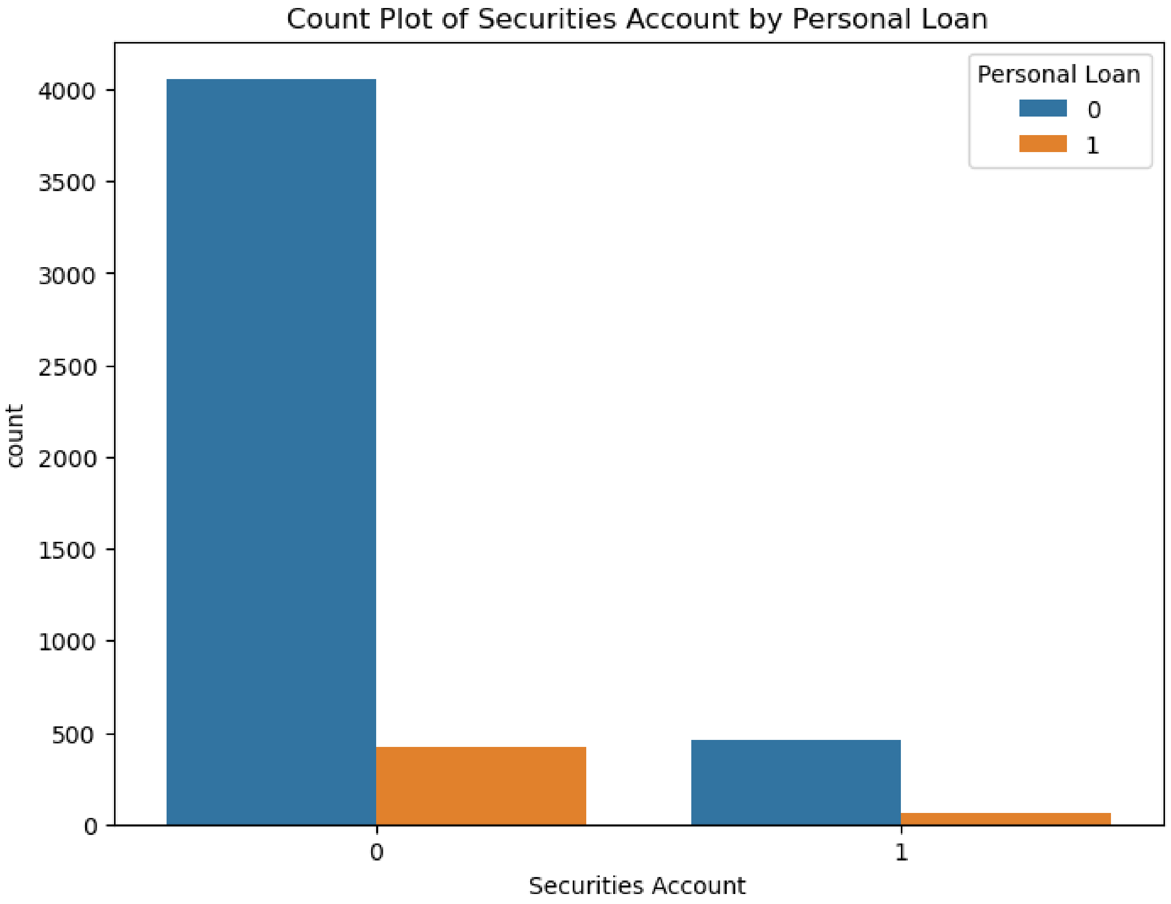
In [54]:

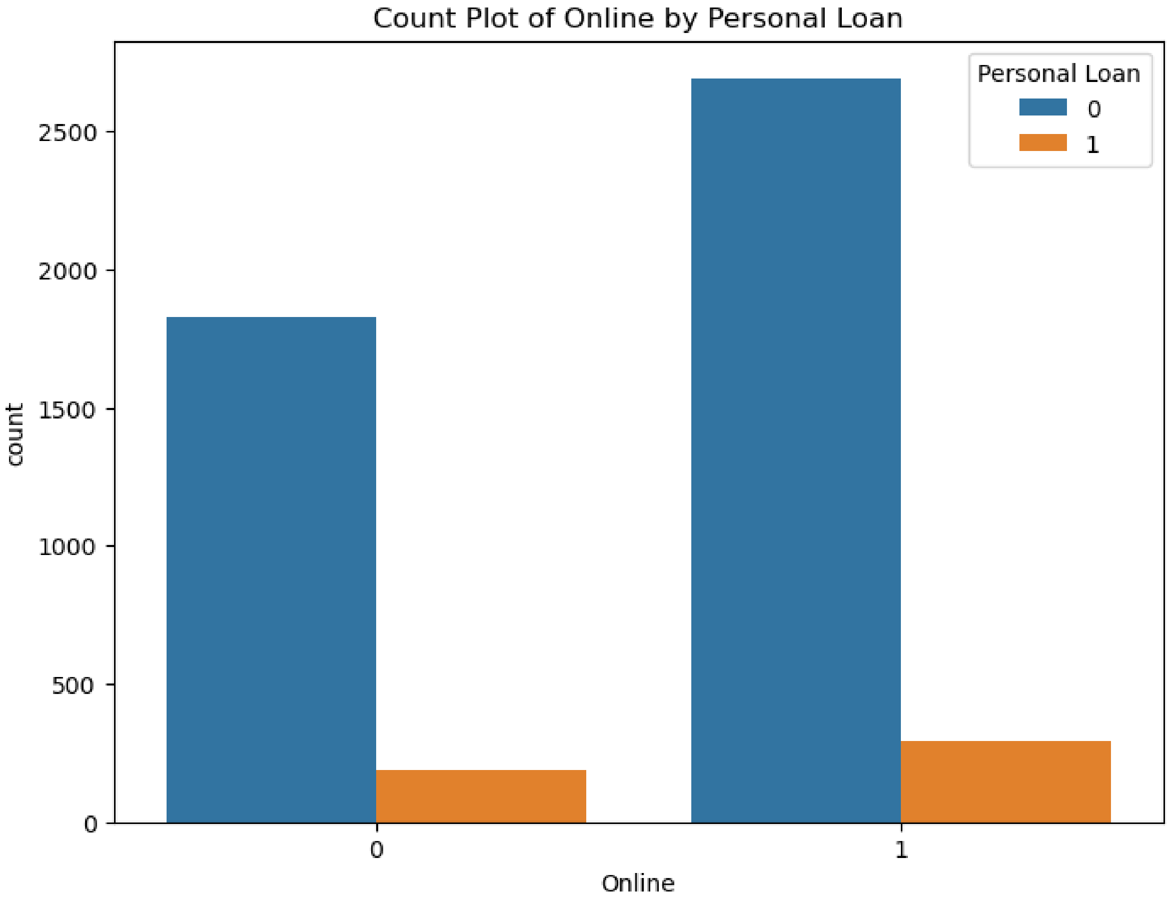
```python
col = ['Securities Account', 'Online', 'Account_Holder_Category', 'CreditCa
```

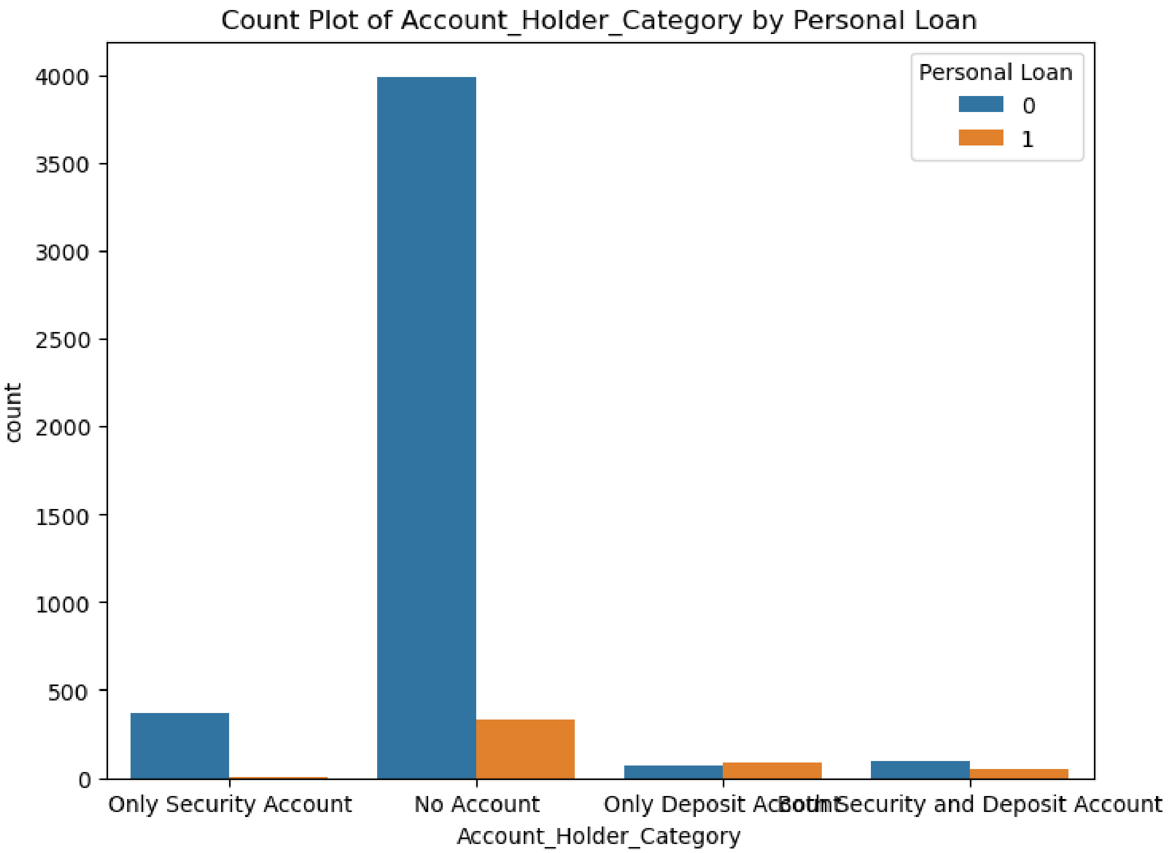
Creating count plots for each of the columns listed in the col list, and you're visualizing how the counts vary with respect to the 'Personal Loan' column

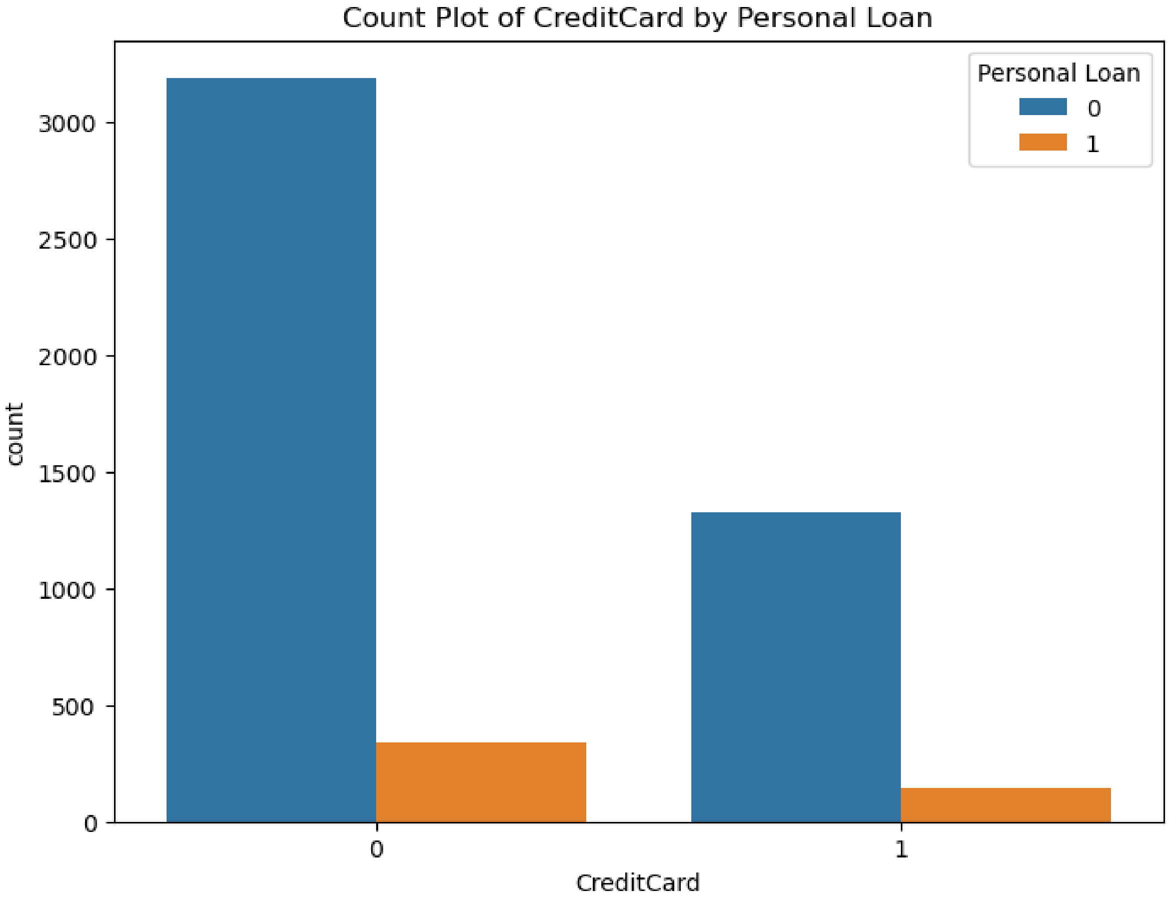
In [55]:

```python
for i in col:
    plt.figure(figsize = (8, 6))
    sns.countplot(x = i, data = data, hue = 'Personal Loan')
    plt.title(f'Count Plot of {i} by Personal Loan')
    plt.show()
```


Count Plot of Securities Account by Personal Loan
Personal Loan
0
1
4000
3500
3000
2500
2000
1500
1000
500
0
count
0
1
Securities Account

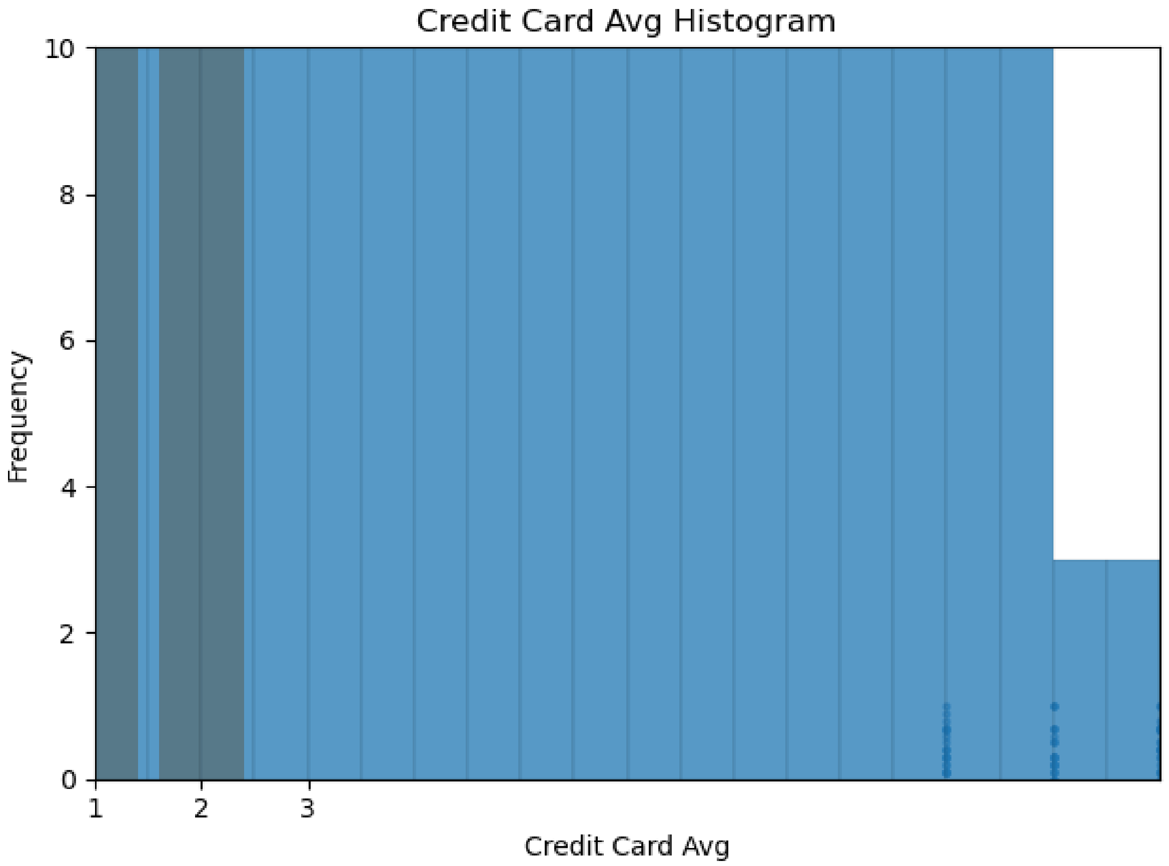

Count Plot of Online by Personal Loan
Personal Loan
0
1
2500
2000
1500
1000
500
0
count
0
1
Online





Creating a Age Distribution by Education Level (Boxplot)

In [59]:
```
sns.boxplot(x='Education', y='Age', data=df)
plt.ylim(15, 80)
plt.title('Age by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Age')

plt.scatter(x='Income', y='CCAvg', data=df, s=5, alpha=0.5)
plt.xlim(0, 250)
plt.ylim(0, 10)
plt.title('Income vs Credit Card Avg')
plt.xlabel('Income')
plt.ylabel('Credit Card Avg')

plt.bar(x=[0, 1], height=df['Securities Account'].value_counts(), width=0.8
plt.title('Securities Account Values')
plt.xlabel('Securities Account')
plt.ylabel('Count')

sns.histplot(df['CCAvg'], kde=False, bins=20)
plt.xlim(0, 10)
plt.title('Credit Card Avg Histogram')
plt.xlabel('Credit Card Avg')
plt.ylabel('Frequency')



plt.tight_layout()
plt.show()
```

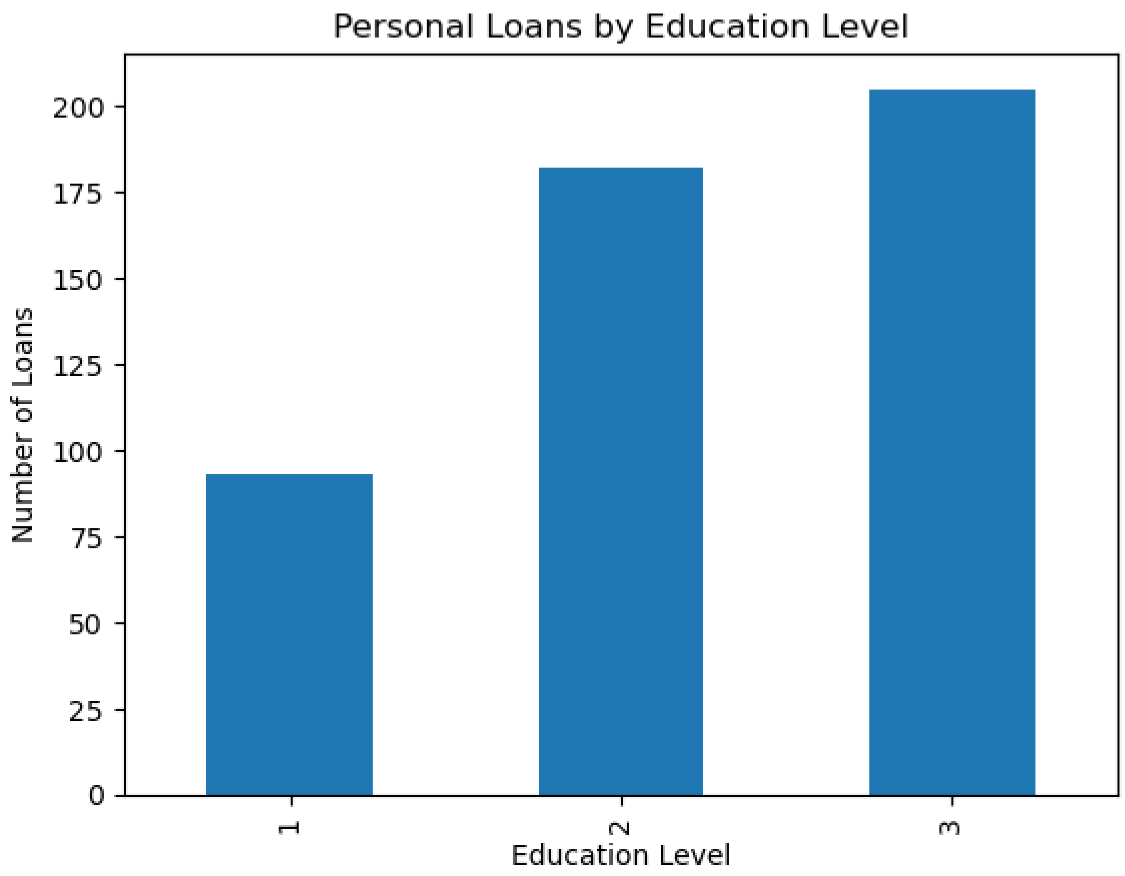


Education Chart

In [10]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('Bank_Personal_Loan_Modelling(ADG).csv')

print(data.shape)
print(data.head())

plot_data = data.groupby('Education')['Personal Loan'].sum()
plot_data.plot(kind='bar')

plt.title('Personal Loans by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Number of Loans')

plt.show()
```

```
(5000, 14)
   ID  Age  Experience  Income  ZIP Code  Family  CCAvg  Education  Mortgag
e  \
0   1   25           1      49     91107       4    1.6          1
0
1   2   45          19      34     90089       3    1.5          1
0
2   3   39          15      11     94720       1    1.0          1
0
3   4   35           9     100     94112       1    2.7          2
0
4   5   35           8      45     91330       4    1.0          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
```

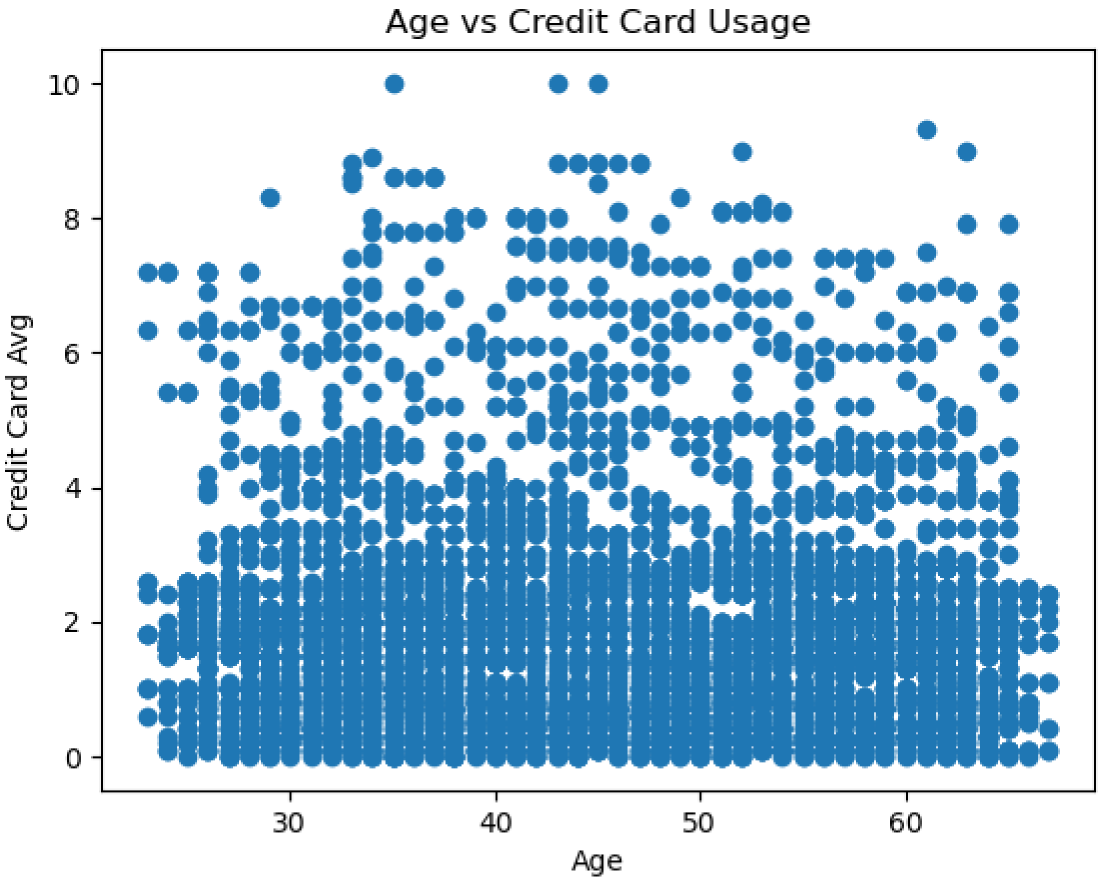


Creating, Data Loading, Exploration, and Scatter Plot Visualization.

In [14]:

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('Bank_Personal_Loan_Modelling(ADG).csv')

print(data.shape)
print(data.columns)
print(data.head())

plt.scatter(data['Age'], data['CCAvg'])

plt.xlabel('Age')
plt.ylabel('Credit Card Avg')
plt.title('Age vs Credit Card Usage')

plt.show()
```

```
(5000, 14)
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal Loan', 'Securities Account',
       'CD Account', 'Online', 'CreditCard'],
      dtype='object')
   ID  Age  Experience  Income  ZIP Code  Family  CCAvg  Education  Mortgag
e  \
0   1   25           1      49     91107       4    1.6          1
0
1   2   45          19      34     90089       3    1.5          1
0
2   3   39          15      11     94720       1    1.0          1
0
3   4   35           9     100     94112       1    2.7          2
0
4   5   35           8      45     91330       4    1.0          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
```

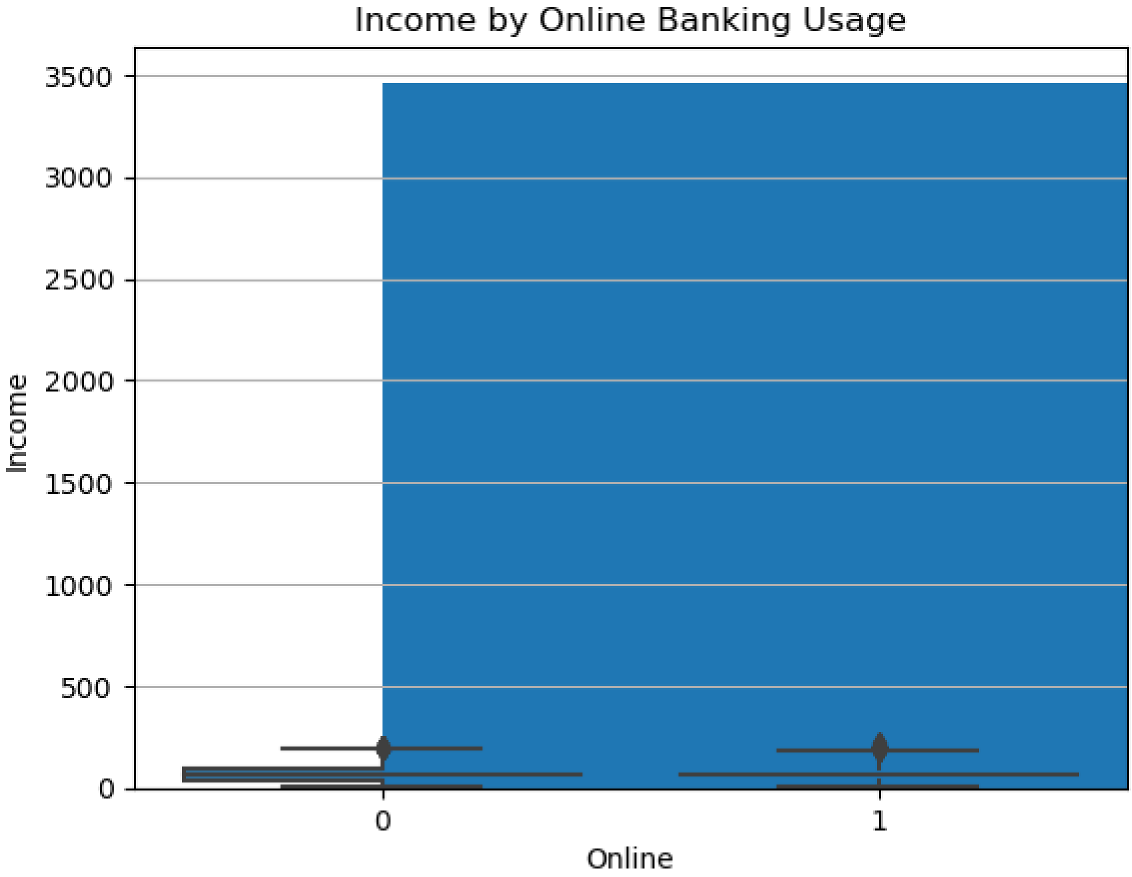


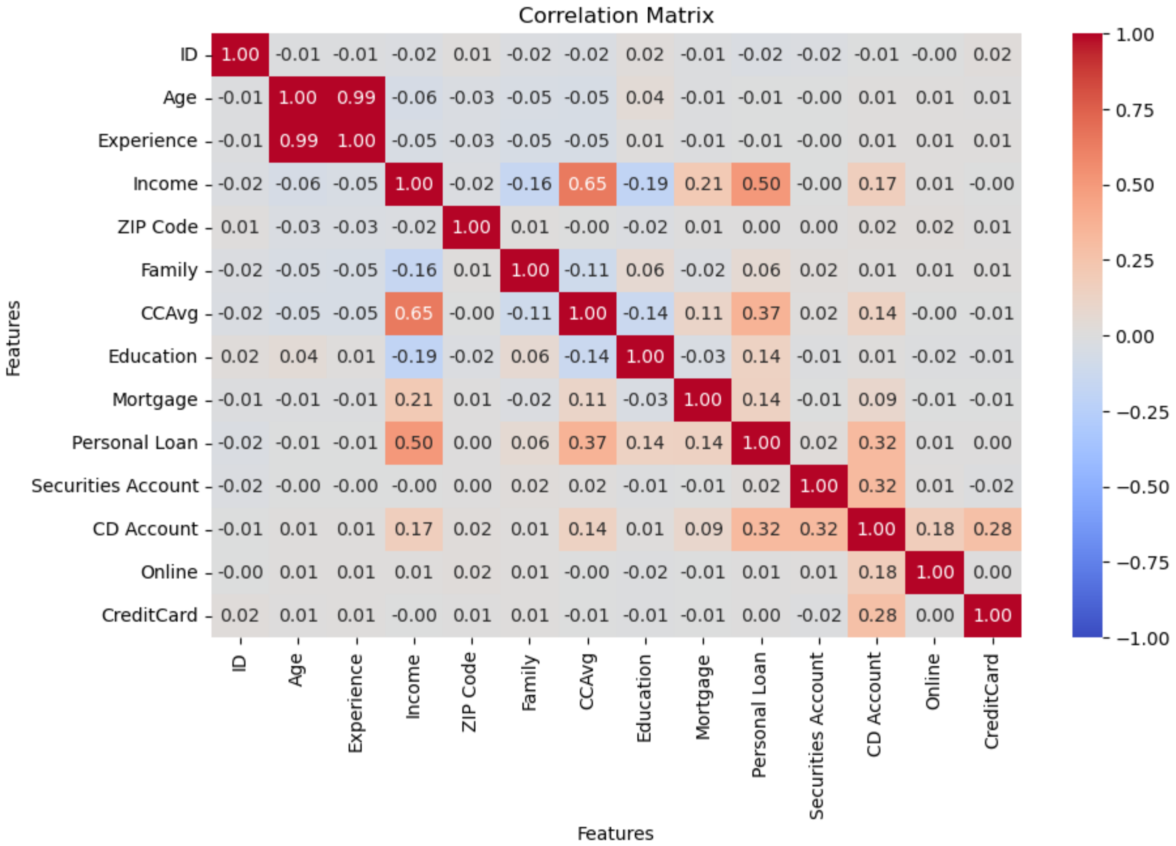
Data Exploration, Visualization, and Correlation Analysis

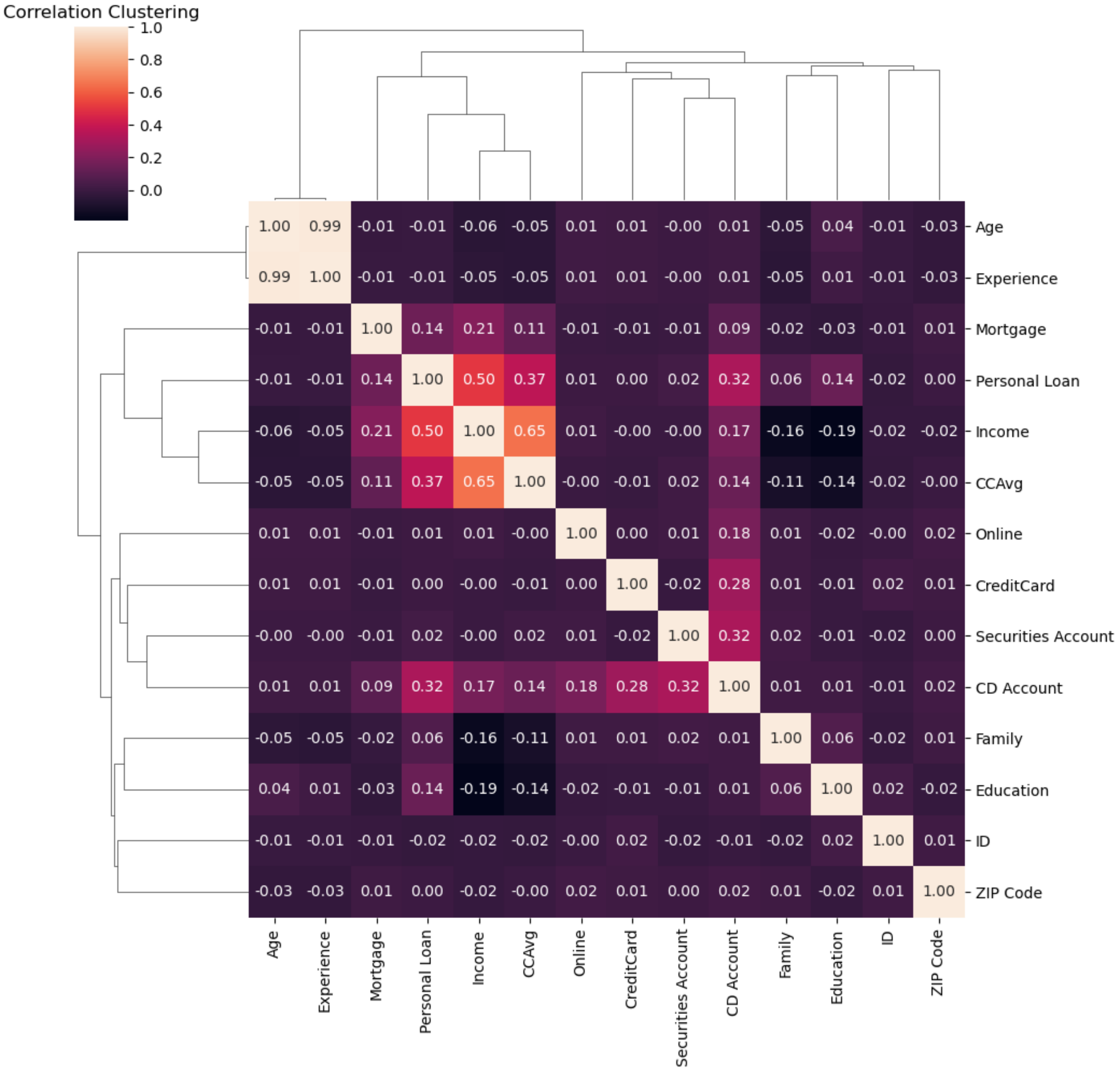
In [16]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv('Bank_Personal_Loan_Modelling(ADG).csv')

print(data.shape)
print(data.head())

data['Mortgage'].hist(bins=30)
plt.title('Distribution of Mortgage Amounts')
plt.xlabel('Mortgage Amount')
plt.ylabel('Frequency')

sns.boxplot(x='Online', y='Income', data=data)
plt.title('Income by Online Banking Usage')

correlations = data.corr()

def format_heatmap(corr):
  sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f', vmin=-1, vmax=1

plt.figure(figsize=(10, 6))
format_heatmap(correlations)

plt.title('Correlation Matrix')
plt.xlabel('Features')
plt.ylabel('Features')
plt.xticks(rotation=90)

plt.show()

sns.clustermap(corr, annot=True, fmt=".2f")
plt.title('Correlation Clustering')
plt.xticks(rotation=90)

plt.show()
```

```
(5000, 14)
   ID  Age  Experience  Income  ZIP Code  Family  CCAvg  Education  Mortgag
e  \
0   1   25           1      49     91107       4    1.6          1
0
1   2   45          19      34     90089       3    1.5          1
0
2   3   39          15      11     94720       1    1.0          1
0
3   4   35           9     100     94112       1    2.7          2
0
4   5   35           8      45     91330       4    1.0          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
```




Correlation Matrix
ID 1.00 -0.01 -0.01 -0.02 0.01 -0.02 -0.02 0.02 -0.01 -0.02 -0.02 -0.01 -0.00 0.02
Age -0.01 1.00 0.99 -0.06 -0.03 -0.05 -0.05 0.04 -0.01 -0.01 -0.00 0.01 0.01 0.01
Experience -0.01 0.99 1.00 -0.05 -0.03 -0.05 -0.05 0.01 -0.01 -0.01 -0.00 0.01 0.01 0.01
Income -0.02 -0.06 -0.05 1.00 -0.02 -0.16 0.65 -0.19 0.21 0.50 -0.00 0.17 0.01 -0.00
ZIP Code 0.01 -0.03 -0.03 -0.02 1.00 0.01 -0.00 -0.02 0.01 0.00 0.00 0.02 0.02 0.01
Family -0.02 -0.05 -0.05 -0.16 0.01 1.00 -0.11 0.06 -0.02 0.06 0.02 0.01 0.01 0.01
CCAvg -0.02 -0.05 -0.05 0.65 -0.00 -0.11 1.00 -0.14 0.11 0.37 0.02 0.14 -0.00 -0.01
Education 0.02 0.04 0.01 -0.19 -0.02 0.06 -0.14 1.00 -0.03 0.14 -0.01 0.01 -0.02 -0.01
Mortgage -0.01 -0.01 -0.01 0.21 0.01 -0.02 0.11 -0.03 1.00 0.14 -0.01 0.09 -0.01 -0.01
Personal Loan -0.02 -0.01 -0.01 0.50 0.00 0.06 0.37 0.14 0.14 1.00 0.02 0.32 0.01 0.00
Securities Account -0.02 -0.00 -0.00 -0.00 0.00 0.02 0.02 -0.01 -0.01 0.02 1.00 0.32 0.01 -0.02
CD Account -0.01 0.01 0.01 0.17 0.02 0.01 0.14 0.01 0.09 0.32 0.32 1.00 0.18 0.28
Online -0.00 0.01 0.01 0.01 0.02 0.01 -0.00 -0.02 -0.01 0.01 0.01 0.18 1.00 0.00
CreditCard 0.02 0.01 0.01 -0.00 0.01 0.01 -0.01 -0.01 -0.01 0.00 -0.02 0.28 0.00 1.00
ID Age Experience Income ZIP Code Family CCAvg Education Mortgage Personal Loan Securities Account CD Account Online CreditCard
Features
Features
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


Correlation Clustering
1.0
0.8
0.6
0.4
0.2
0.0
1.00 0.99 -0.01 -0.01 -0.06 -0.05 0.01 0.01 -0.00 0.01 -0.05 0.04 -0.01 -0.03 Age
0.99 1.00 -0.01 -0.01 -0.05 -0.05 0.01 0.01 -0.00 0.01 -0.05 0.01 -0.01 -0.03 Experience
-0.01 -0.01 1.00 0.14 0.21 0.11 -0.01 -0.01 -0.01 0.09 -0.02 -0.03 -0.01 0.01 Mortgage
-0.01 -0.01 0.14 1.00 0.50 0.37 0.01 0.00 0.02 0.32 0.06 0.14 -0.02 0.00 Personal Loan
-0.06 -0.05 0.21 0.50 1.00 0.65 0.01 -0.00 -0.00 0.17 -0.16 -0.19 -0.02 -0.02 Income
-0.05 -0.05 0.11 0.37 0.65 1.00 -0.00 -0.01 0.02 0.14 -0.11 -0.14 -0.02 -0.00 CCAvg
0.01 0.01 -0.01 0.01 0.01 -0.00 1.00 0.00 0.01 0.18 0.01 -0.02 -0.00 0.02 Online
0.01 0.01 -0.01 0.00 -0.00 -0.01 0.00 1.00 -0.02 0.28 0.01 -0.01 0.02 0.01 CreditCard
-0.00 -0.00 -0.01 0.02 -0.00 0.02 0.01 -0.02 1.00 0.32 0.02 -0.01 -0.02 0.00 Securities Account
0.01 0.01 0.09 0.32 0.17 0.14 0.18 0.28 0.32 1.00 0.01 0.01 -0.01 0.02 CD Account
-0.05 -0.05 -0.02 0.06 -0.16 -0.11 0.01 0.01 0.02 0.01 1.00 0.06 -0.02 0.01 Family
0.04 0.01 -0.03 0.14 -0.19 -0.14 -0.02 -0.01 -0.01 0.01 0.06 1.00 0.02 -0.02 Education
-0.01 -0.01 -0.01 -0.02 -0.02 -0.02 -0.00 0.02 -0.02 -0.01 -0.02 0.02 1.00 0.01 ID
-0.03 -0.03 0.01 0.00 -0.02 -0.00 0.02 0.01 0.00 0.02 0.01 -0.02 0.01 1.00 ZIP Code
Age Experience Mortgage Personal Loan Income CCAvg Online CreditCard Securities Account CD Account Family Education ID ZIP Code

```
In [ ]:
```