Ввод [1]:

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

Ввод [2]:

```
loan_data = pd.read_csv('C:\\Users\\hemfa\\Downloads\\Customers_of_Bank.
```

Ввод [3]:

```
type(loan_data)
```

Out[3]:

```
pandas.core.frame.DataFrame
```

Ввод [4]:

```
loan_data = pd.read_csv('C:\\Users\\hemfa\\Downloads\\Customers_of_Bank.

loan_data.drop(columns=['ID'], axis=1, inplace=True)
loan_data.rename(columns={"Experience": "Work_experience", "CCAvg": "Avg
                   "Mortgage": "Mortgage_value_k", "Personal Loan": "Acc
                   "ZIP Code": "ZIP_Code", "Online": "Online_transaction
loan_data['Total_bank_relationships'] = (loan_data['Accepted_personal_lo
                                         loan_data['CreditCard'] + loan_

loan_data.head()
```

Out[4]:

	Age	Work_experience	Income	ZIP_Code	Family	Avg_credit_card_monthly	Education	Mort
0	25	1	49	91107	4	1.6	1	
1	45	19	34	90089	3	1.5	1	
2	39	15	11	94720	1	1.0	1	
3	35	9	100	94112	1	2.7	2	
4	35	8	45	91330	4	1.0	2	

Ввод [5]:

```
loan_data.shape
```

Out[5]:

```
(5000, 14)
```

Ввод [6]:
```
loan_data.describe()
```

Out[6]:

	Age	Work_experience	Income	ZIP_Code	Family	Avg_credit_card_r
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000
mean	45.338400	20.104600	73.774200	93152.503000	2.396400	1
std	11.463166	11.467954	46.033729	2121.852197	1.147663	1
min	23.000000	-3.000000	8.000000	9307.000000	1.000000	0
25%	35.000000	10.000000	39.000000	91911.000000	1.000000	0
50%	45.000000	20.000000	64.000000	93437.000000	2.000000	1
75%	55.000000	30.000000	98.000000	94608.000000	3.000000	2
max	67.000000	43.000000	224.000000	96651.000000	4.000000	10

avg age = 45, min and max = 23 and 67 avg income = $73

Ввод [7]:
```
loan_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 14 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Age                       5000 non-null   int64  
 1   Work_experience           5000 non-null   int64  
 2   Income                    5000 non-null   int64  
 3   ZIP_Code                  5000 non-null   int64  
 4   Family                    5000 non-null   int64  
 5   Avg_credit_card_monthly   5000 non-null   float64
 6   Education                 5000 non-null   int64  
 7   Mortgage_value_k          5000 non-null   int64  
 8   Accepted_personal_loan    5000 non-null   int64  
 9   Securities Account        5000 non-null   int64  
 10  CD Account                5000 non-null   int64  
 11  Online_transactions       5000 non-null   int64  
 12  CreditCard                5000 non-null   int64  
 13  Total_bank_relationships  5000 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 547.0 KB
```

Ввод [8]:

```
loan_data.isnull()
```

Out[8]:

	Age	Work_experience	Income	ZIP_Code	Family	Avg_credit_card_monthly	Education
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
4995	False	False	False	False	False	False	False
4996	False	False	False	False	False	False	False
4997	False	False	False	False	False	False	False
4998	False	False	False	False	False	False	False
4999	False	False	False	False	False	False	False

5000 rows × 14 columns

Ввод [9]:

```
loan_data.isnull().sum()
```

Out[9]:

```
Age                        0
Work_experience            0
Income                     0
ZIP_Code                   0
Family                     0
Avg_credit_card_monthly    0
Education                  0
Mortgage_value_k           0
Accepted_personal_loan     0
Securities Account         0
CD Account                 0
Online_transactions        0
CreditCard                 0
Total_bank_relationships   0
dtype: int64
```

Ввод [10]:

```
neg_Workexp = loan_data.Work_experience < 0
neg_Workexp.value_counts()
```

Out[10]:

```
False    4948
True       52
Name: Work_experience, dtype: int64
```

Ввод [11]:

```
# loan_data.abs(column = ['Work_experience'], inplace=True)

# loan_data.loc[loan_data.Work_experience.abs(), inplace = True]

# loan_data.Work_experience = abs(Work_experience)
# loan_data.Work_experience = loan_data['Work_experience'].abs()
# loan_data['Work_experience']=loan_data['Work_experience'].abs()
# loan_data[loan_data['Work_experience'] < 0]['Work_experience'].abs()
loan_data.Work_experience = abs(loan_data.Work_experience)
```

Ввод [12]:

```
loan_data[loan_data['Work_experience'] < 0]['Work_experience'].count()
```

Out[12]:

```
0
```

Ввод [13]:

```
loan_data.Work_experience = pd.to_numeric(loan_data.Work_experience)
```

Ввод [14]:

```
loan_data.describe()
```

Out[14]:

	Age	Work_experience	Income	ZIP_Code	Family	Avg_credit_card_r
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000
mean	45.338400	20.134600	73.774200	93152.503000	2.396400	1
std	11.463166	11.415189	46.033729	2121.852197	1.147663	1
min	23.000000	0.000000	8.000000	9307.000000	1.000000	0
25%	35.000000	10.000000	39.000000	91911.000000	1.000000	0
50%	45.000000	20.000000	64.000000	93437.000000	2.000000	1
75%	55.000000	30.000000	98.000000	94608.000000	3.000000	2
max	67.000000	43.000000	224.000000	96651.000000	4.000000	10

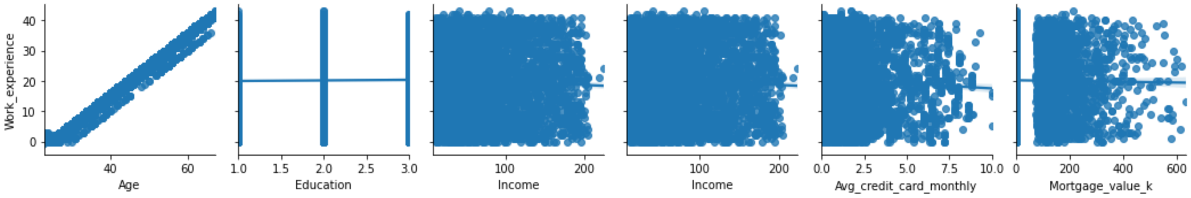
min work_experience is 0 contrary to the previous data of -3

Ввод [15]:

```
quantitivecor = ['Age', 'Education', 'Income', 'Income', 'Avg_credit_ca
experience_Grid = sns.PairGrid(loan_data, y_vars = 'Work_experience', x
experience_Grid.map(sns.regplot)
```

Out[15]:

```
<seaborn.axisgrid.PairGrid at 0x214ddc30130>
```



The higher the age and education level, the longer work experience. This means bank can offer loans to the elder customers with advanced professional experience.

Ввод [16]:

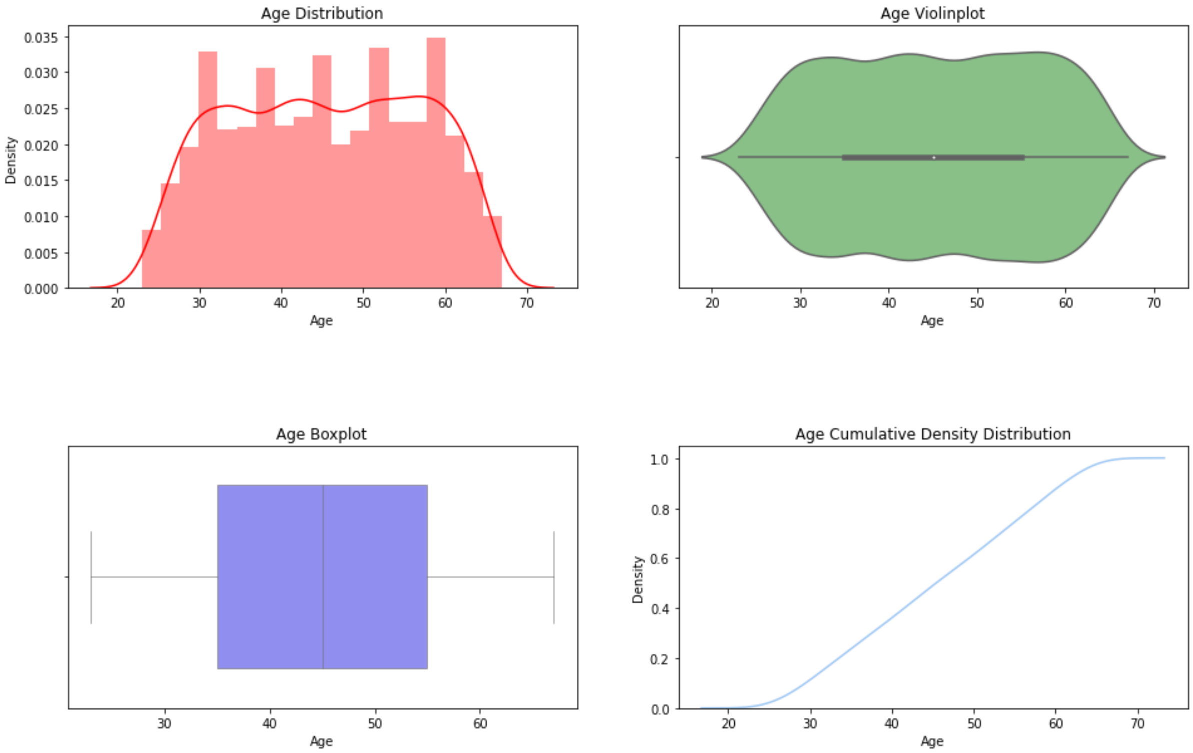
```
def summary(values):
    values_min = loan_data[values].min()
    values_max = loan_data[values].max()
    q1 = loan_data[values].quantile(0.25)
    q2 = loan_data[values].quantile(0.50)
    q3 = loan_data[values].quantile(0.75)
    print(f'5 Point Summary of {values.capitalize()} Attribute:\n'
          f'{values.capitalize()}(min) : {values_min}\n'
          f'q1                    : {q1}\n'
          f'q2(Median)            : {q2}\n'
          f'q3                    : {q3}\n'
          f'{values.capitalize()}(max) : {values_max}')

    fig = plt.figure(figsize=(16, 10))
    plt.subplots_adjust(hspace = 0.6)
    sns.set_palette('pastel')

    plt.subplot(221)
    ax1 = sns.distplot(loan_data[values], color = 'r')
    plt.title(f'{values.capitalize()} Distribution')

    plt.subplot(222)
    ax2 = sns.violinplot(loan_data[values], palette = 'Accent', split =
    plt.title(f'{values.capitalize()} Violinplot')

    plt.subplot(223)
    ax2 = sns.boxplot(loan_data[values], palette = 'cool', width=0.7, l
    plt.title(f'{values.capitalize()} Boxplot')

    plt.subplot(224)
    ax3 = sns.kdeplot(loan_data[values], cumulative=True)
    plt.title(f'{values.capitalize()} Cumulative Density Distribution')

    plt.show()
```

Ввод [17]:

```
summary('Age')
```

```
5 Point Summary of Age Attribute:
Age(min) : 23
q1                      : 35.0
q2(Median)              : 45.0
q3                      : 55.0
Age(max) : 67
```

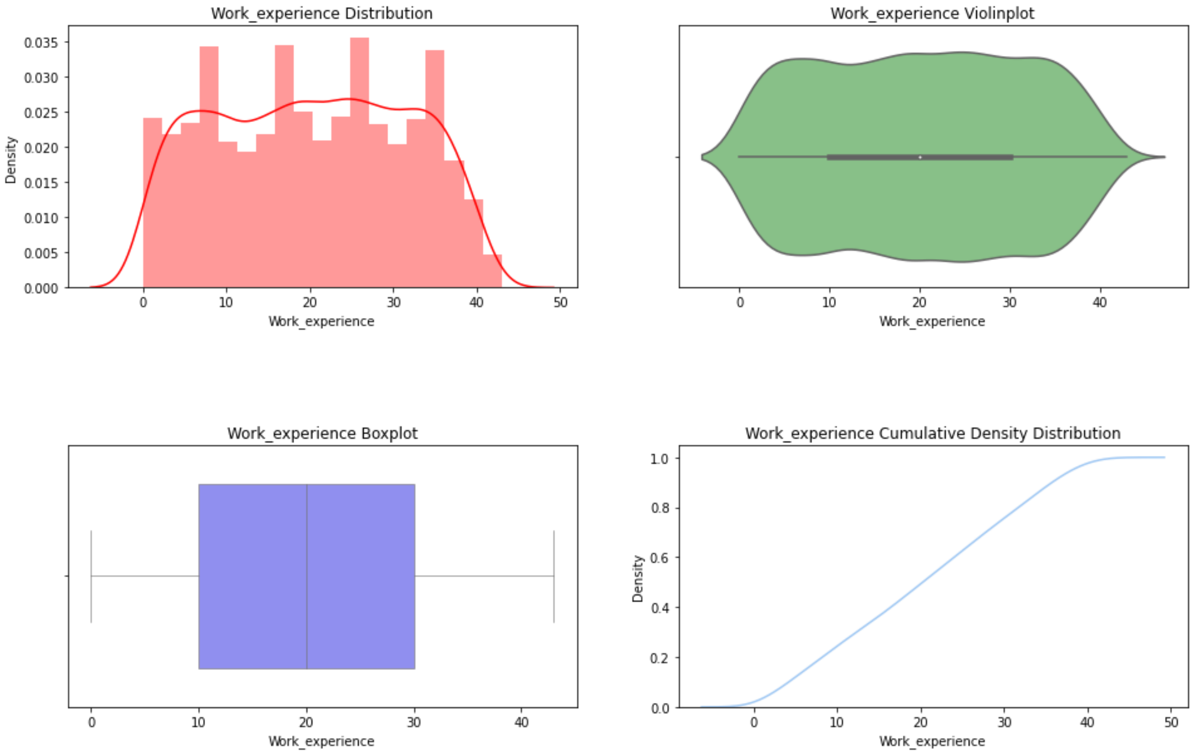
```
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecat
ed function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flex
ibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argum
ent will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argum
ent will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```



Ввод [18]:

```
summary('Work_experience')
```

```
5 Point Summary of Work_experience Attribute:
Work_experience(min) : 0
q1                     : 10.0
q2(Median)             : 20.0
q3                     : 30.0
Work_experience(max) : 43

c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecat
ed function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flex
ibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argum
ent will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argum
ent will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```

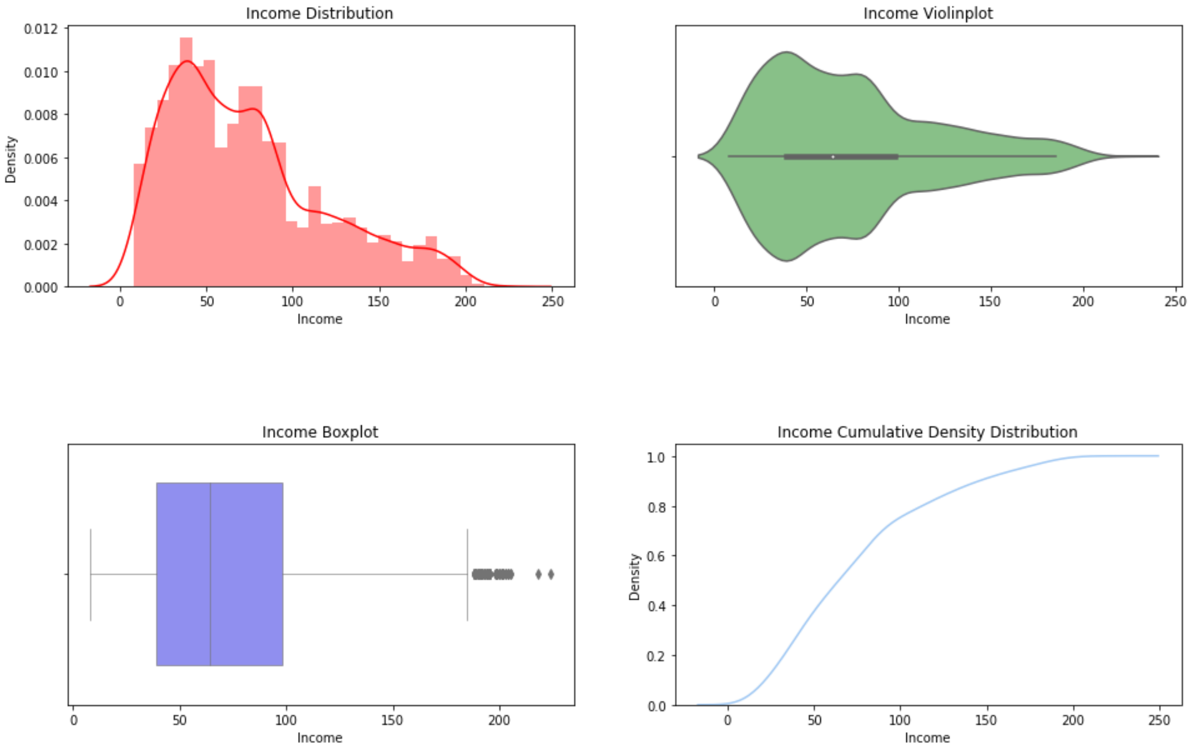


The experience is normally distributed

Ввод [19]:

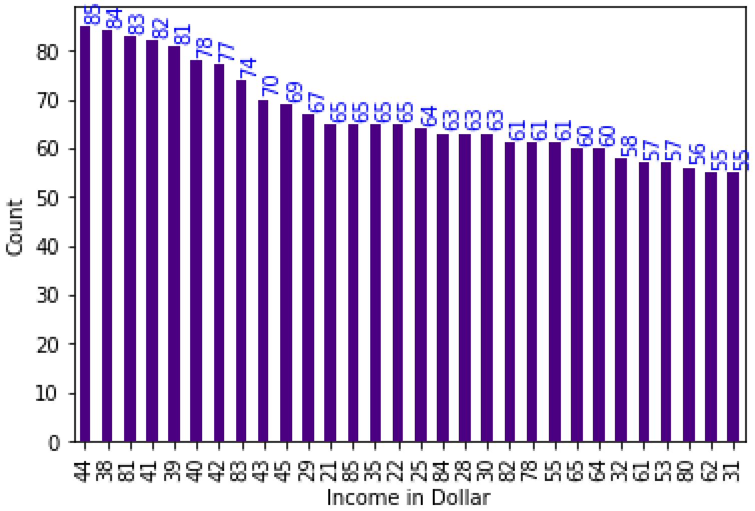
```
summary('Income')
```

```
5 Point Summary of Income Attribute:
Income(min) : 8
q1                    : 39.0
q2(Median)            : 64.0
q3                    : 98.0
Income(max) : 224

c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecat
ed function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flex
ibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argum
ent will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argum
ent will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```



Ввод [20]:

```
# sns.distplot(loan_data['Income'], color = 'red')
# plt.title('Income')


income=loan_data['Income'].value_counts().head(30)
ax=income.plot.bar(width=0.5,color="indigo")
plt.xlabel("Income in Dollar")
plt.ylabel("Count")
for i, j in income.reset_index().iterrows():
    ax.text(i, j.Income + 1.5, j.Income, color='blue',rotation=90)
```



The distribution is higher than normal, which is positive

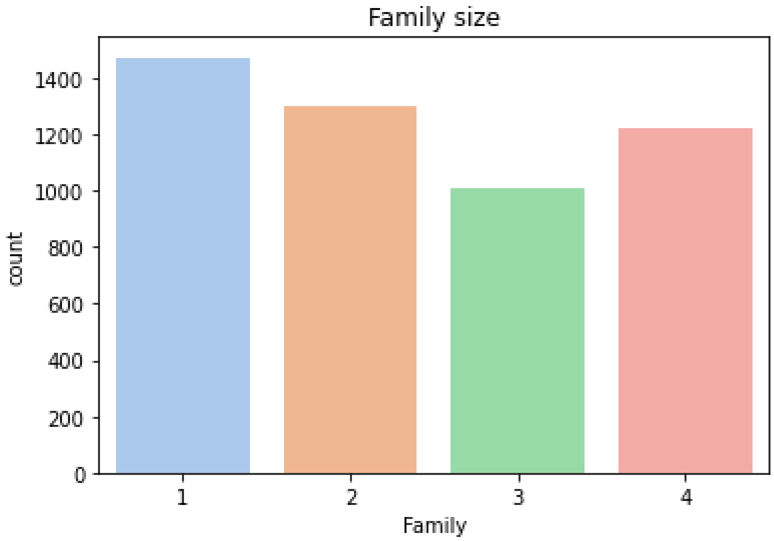
Ввод [21]:

```
sns.countplot(loan_data['Family'])
plt.title('Family size')
```

```
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argum
ent will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```

Out[21]:

```
Text(0.5, 1.0, 'Family size')
```



Most of the customers have family member of 1. Customers with family size of 3 are comparatively less.

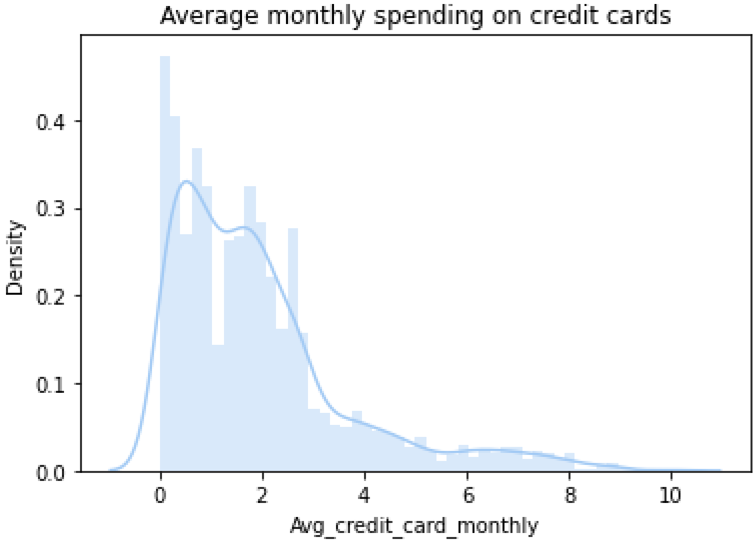
Ввод [22]:

```
sns.distplot(loan_data['Avg_credit_card_monthly'])
plt.title('Average monthly spending on credit cards')
```

```
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecat
ed function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flex
ibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[22]:

```
Text(0.5, 1.0, 'Average monthly spending on credit cards')
```

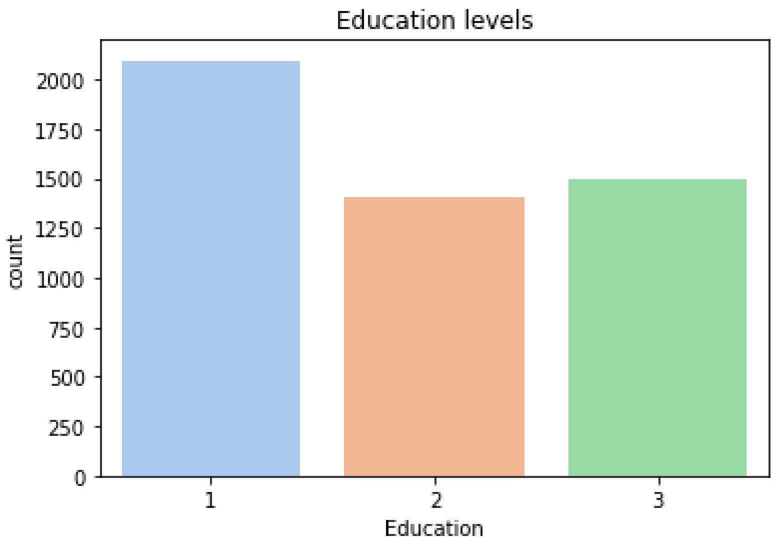


Most of the customers average monthly spending on credit cards is between 0 $to2000$. There are very few customers whose monthly spending is between $8000-$ 10000.

```
Ввод [23]:  sns.countplot(loan_data['Education'])
            plt.title('Education levels')
```

```
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argum
ent will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[23]: Text(0.5, 1.0, 'Education levels')
```



1 = Undergraduate, 2 = Graduate, 3 =Advanced/Professional. Majority of customers has only undergraduate degree.

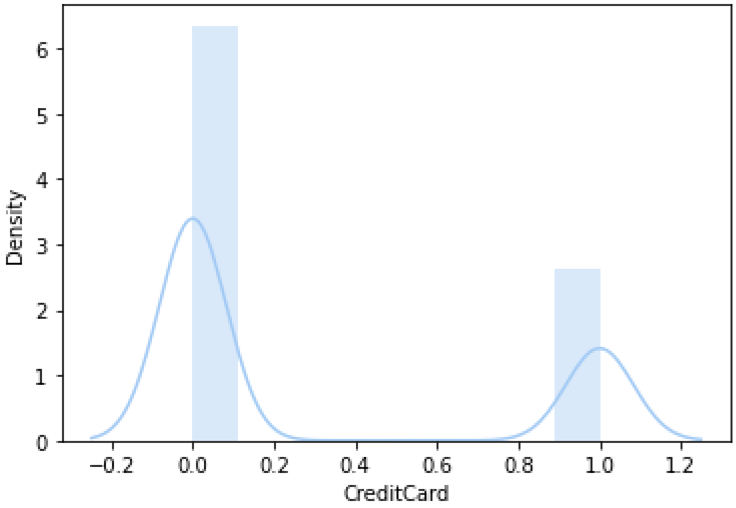
Ввод [24]:

```
loan_data_cc = loan_data['CreditCard']
loan_data_cc = loan_data_cc.astype({'CreditCard': 'float64'})
sns.distplot(loan_data_cc)
```

```
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecat
ed function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flex
ibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[24]:

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



Those who do not own credit card are half as many as credit card holders.

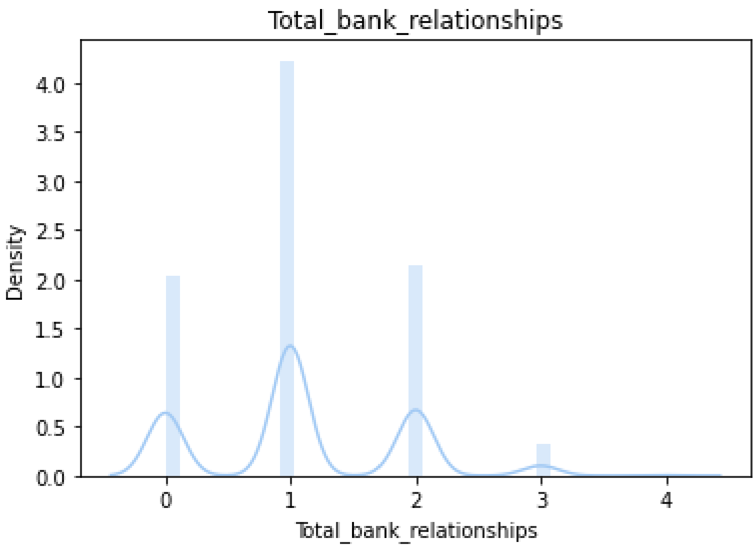
Ввод [25]:

```
sns.distplot(loan_data['Total_bank_relationships'])
plt.title('Total_bank_relationships')
```

```
c:\users\hemfa\appdata\local\programs\python\python38\lib\site-packages
\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecat
ed function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flex
ibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[25]:

```
Text(0.5, 1.0, 'Total_bank_relationships')
```



According to the graph most of the customers use only one service of the bank and no one uses 4 services.

Ввод [26]:

```
n_true = len(loan_data.loc[loan_data['Accepted_personal_loan'] == True]
n_false = len(loan_data.loc[loan_data['Accepted_personal_loan'] == Fals
print('Number of customers who accepted the Loan offer: {0} ({1:2.2f}%)
      .format(n_true, (n_true / (n_true + n_false)) * 100))
print('Number of customers who did not accept the Loan offer: {0} ({1:2
      .format(n_false, (n_false / (n_true + n_false)) * 100))
```

```
Number of customers who accepted the Loan offer: 480 (9.60%)
Number of customers who did not accept the Loan offer: 4520 (90.40%)
```

Ввод [27]:

```
loan_acceptance_count = pd.DataFrame(loan_data['Accepted_personal_loan'
loan_acceptance_count.columns = ['Labels', 'Accepted_personal_loan']
loan_acceptance_count
```

Out[27]:

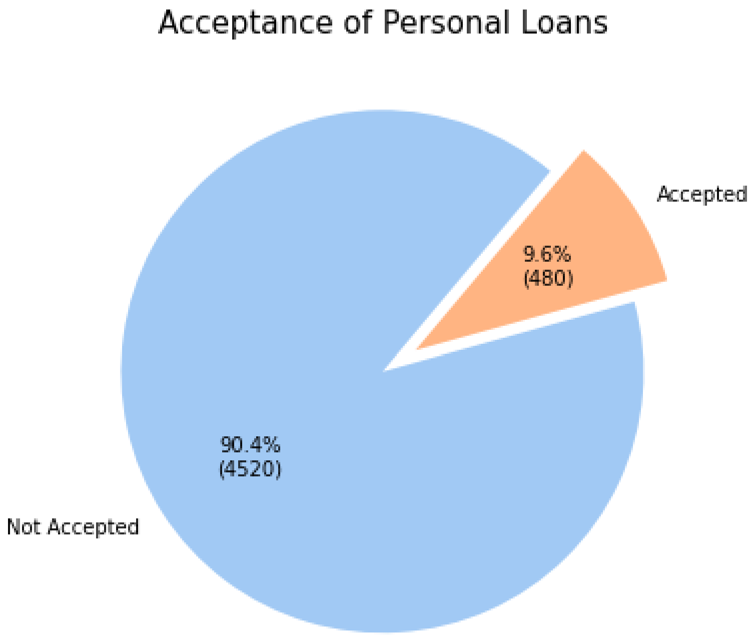
	Labels	Accepted_personal_loan
0	0	4520
1	1	480

Ввод [28]:

```
pie_labels = loan_acceptance_count['Labels']
pie_labels = ['Not Accepted' if i == 0 else 'Accepted' for i in pie_labe

pie_loan_data = loan_acceptance_count['Accepted_personal_loan']

explode = (0, 0.15)
fig, ax = plt.subplots(figsize =(12, 6))
wp = {}
def func(pct, allvalues):
    absolute = int(np.round(pct / 100.*np.sum(allvalues)))
    return "{:.1f}%\n({:d})".format(pct, absolute)


ax.pie(pie_loan_data,
       autopct = lambda pct: func(pct, pie_loan_data),
       labels = pie_labels,
       explode = explode,
       startangle = 50,
       wedgeprops = wp)


plt.title('Acceptance of Personal Loans', size=15)
```

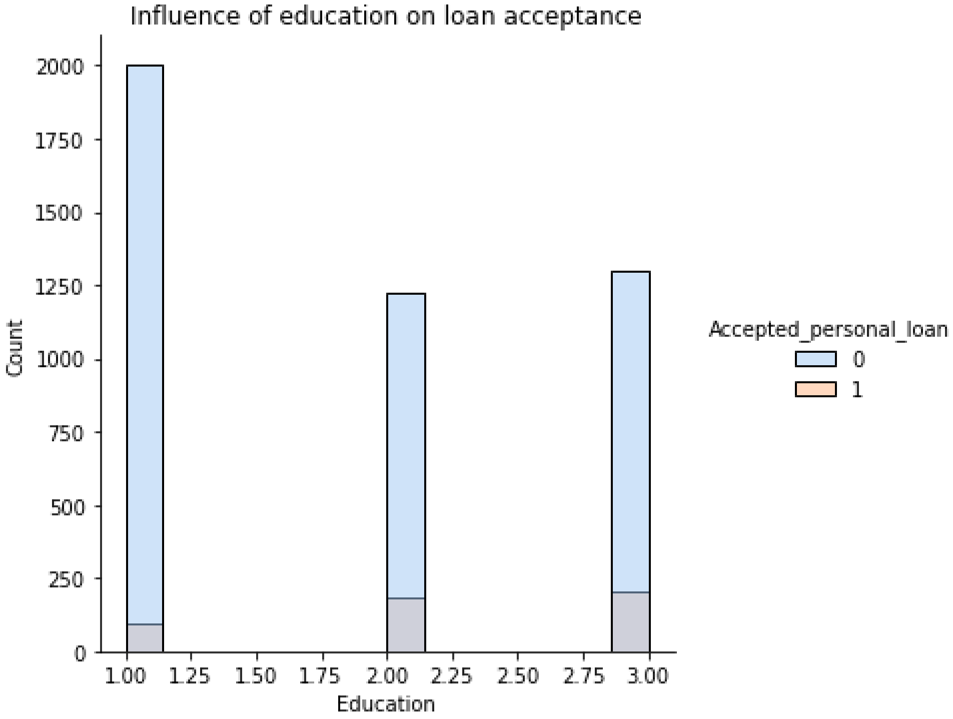
Out[28]:

```
Text(0.5, 1.0, 'Acceptance of Personal Loans')
```



```
Ввод [29]:  sns.displot(x ='Education', data = loan_data, hue = 'Accepted_personal_
            plt.title('Influence of education on loan acceptance')
```

```
Out[29]: Text(0.5, 1.0, 'Influence of education on loan acceptance')
```



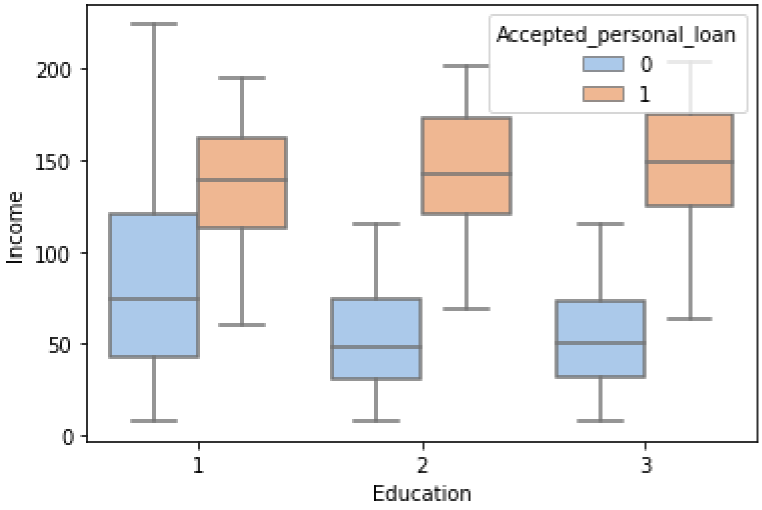
There is a slight influence of education on loan acceptance, as most of the customers with undergraduate degree did not accept the offered loan. However, the majority of customers who accepted the loans holds advanced degree level.

Ввод [30]:

```
sns.boxplot( x ='Education', y='Income', hue='Accepted_personal_loan',
```

Out[30]:

```
<AxesSubplot:xlabel='Education', ylabel='Income'>
```



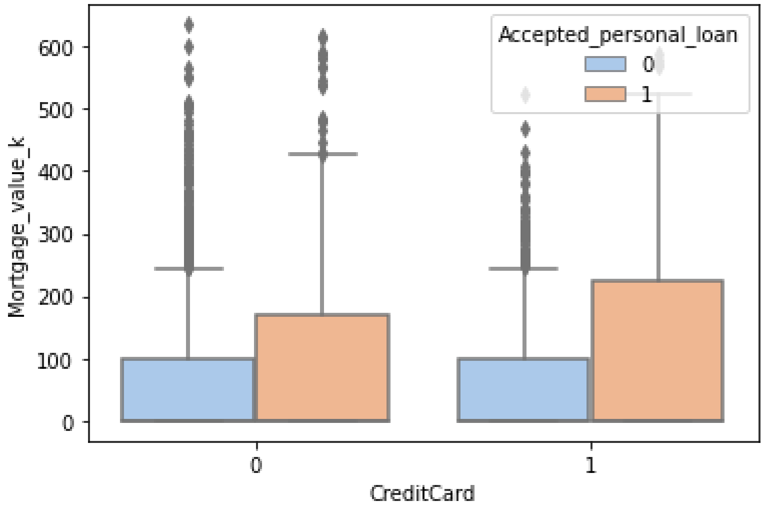
Customers who accepted the loans had the same income level, in spite of those who did not accept.

Ввод [31]:

```
sns.boxplot(x='CreditCard', y = 'Mortgage_value_k', hue = 'Accepted_per
```

Out[31]:

```
<AxesSubplot:xlabel='CreditCard', ylabel='Mortgage_value_k'>
```



The customers who have accepted the offered loans have higher mortgage debt and hold credit cards.

Ввод [32]:

```
loan_data.corr()
```

Out[32]:

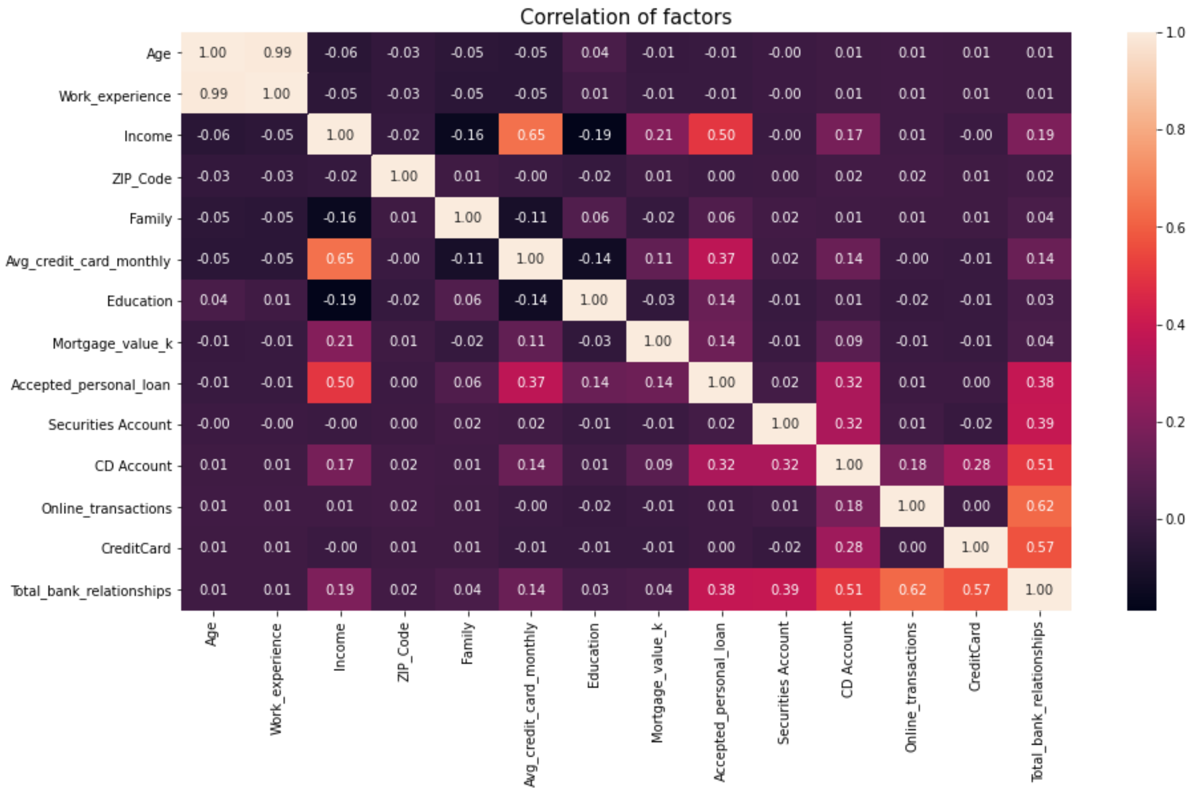
	Age	Work_experience	Income	ZIP_Code	Family	Avg_credi
Age	1.000000	0.993991	-0.055269	-0.029216	-0.046418	
Work_experience	0.993991	1.000000	-0.046876	-0.028936	-0.051851	
Income	-0.055269	-0.046876	1.000000	-0.016410	-0.157501	
ZIP_Code	-0.029216	-0.028936	-0.016410	1.000000	0.011778	
Family	-0.046418	-0.051851	-0.157501	0.011778	1.000000	
Avg_credit_card_monthly	-0.052012	-0.049738	0.645984	-0.004061	-0.109275	
Education	0.041334	0.013919	-0.187524	-0.017377	0.064929	
Mortgage_value_k	-0.012539	-0.011097	0.206806	0.007383	-0.020445	
Accepted_personal_loan	-0.007726	-0.008304	0.502462	0.000107	0.061367	
Securities Account	-0.000436	-0.000989	-0.002616	0.004704	0.019994	
CD Account	0.008043	0.009735	0.169738	0.019972	0.014110	
Online_transactions	0.013702	0.014051	0.014206	0.016990	0.010354	
CreditCard	0.007681	0.008851	-0.002385	0.007691	0.011588	
Total_bank_relationships	0.009792	0.010249	0.191907	0.016682	0.043307	

Ввод [33]:

```
plt.figure(figsize = (15, 8))
plt.title('Correlation of factors', y=1, size=15)
sns.heatmap(loan_data.corr(), annot = True, fmt='.2f')
```

Out[33]:

```
<AxesSubplot:title={'center':'Correlation of factors'}>
```



The correlation matrix illustrated high correlation between customers' income and their average monthly spending on credit cards. Obviously there is the highest correlation between age and work experience of customers. Interestingly, there is a strong correlation relationship between the users of total bank services and credit card holders, suggesting that if the bank focuses on credit card holders and attracts perpective customers using cc, it can increase the number of customers and turn them lenders.

Ввод [ ]: