

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: loan_df=pd.read_csv(r"C:\Users\kisha\OneDrive\Desktop\BankLoan.csv")
```

```
In [3]: loan_df
```

```
Out[3]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
0	LP001002	Male	No	0	Graduate	No	5849
1	LP001003	Male	Yes	1	Graduate	No	4583
2	LP001005	Male	Yes	0	Graduate	Yes	3000
3	LP001006	Male	Yes	0	Not Graduate	No	2583
4	LP001008	Male	No	0	Graduate	No	6000
...
609	LP002978	Female	No	0	Graduate	No	2900
610	LP002979	Male	Yes	3+	Graduate	No	4106
611	LP002983	Male	Yes	1	Graduate	No	8072
612	LP002984	Male	Yes	2	Graduate	No	7583
613	LP002990	Female	No	0	Graduate	Yes	4583

614 rows × 13 columns



```
In [4]: loan_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Loan_ID          614 non-null    object  
 1   Gender           601 non-null    object  
 2   Married          611 non-null    object  
 3   Dependents       599 non-null    object  
 4   Education        614 non-null    object  
 5   Self_Employed    582 non-null    object  
 6   ApplicantIncome  614 non-null    int64  
 7   CoapplicantIncome 614 non-null    float64 
 8   LoanAmount       592 non-null    float64 
 9   Loan_Amount_Term 600 non-null    float64 
 10  Credit_History   564 non-null    float64 
 11  Property_Area    614 non-null    object  
 12  Loan_Status      614 non-null    object  
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

In [5]: `loan_df.isnull()`

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
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
...
609	False	False	False	False	False	False	False
610	False	False	False	False	False	False	False
611	False	False	False	False	False	False	False
612	False	False	False	False	False	False	False
613	False	False	False	False	False	False	False

614 rows × 13 columns



In [6]: `loan_df.shape`

Out[6]: (614, 13)

In [7]: `loan_df.size`

Out[7]: 7982

In [8]: loan_df.columns

```
Out[8]: Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

```
In [9]: cat=loan_df.select_dtypes(include='object')
num=loan_df.select_dtypes(exclude='object')
```

In [10]: cat

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Property_Area	L
0	LP001002	Male	No	0	Graduate	No	Urban	
1	LP001003	Male	Yes	1	Graduate	No	Rural	
2	LP001005	Male	Yes	0	Graduate	Yes	Urban	
3	LP001006	Male	Yes	0	Not Graduate	No	Urban	
4	LP001008	Male	No	0	Graduate	No	Urban	
...
609	LP002978	Female	No	0	Graduate	No	Rural	
610	LP002979	Male	Yes	3+	Graduate	No	Rural	
611	LP002983	Male	Yes	1	Graduate	No	Urban	
612	LP002984	Male	Yes	2	Graduate	No	Urban	
613	LP002990	Female	No	0	Graduate	Yes	Semiurban	

614 rows × 8 columns



In [11]: num

Out[11]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	5849	0.0	NaN	360.0	1.0
1	4583	1508.0	128.0	360.0	1.0
2	3000	0.0	66.0	360.0	1.0
3	2583	2358.0	120.0	360.0	1.0
4	6000	0.0	141.0	360.0	1.0
...
609	2900	0.0	71.0	360.0	1.0
610	4106	0.0	40.0	180.0	1.0
611	8072	240.0	253.0	360.0	1.0
612	7583	0.0	187.0	360.0	1.0
613	4583	0.0	133.0	360.0	0.0

614 rows × 5 columns

In [12]: `loan_df.dtypes`

Out[12]:

Loan_ID	object
Gender	object
Married	object
Dependents	object
Education	object
Self_Employed	object
ApplicantIncome	int64
CoapplicantIncome	float64
LoanAmount	float64
Loan_Amount_Term	float64
Credit_History	float64
Property_Area	object
Loan_Status	object
dtype:	object

In [15]: `loan_df.isnull().sum()`

```
Out[15]: Loan_ID      0  
Gender        13  
Married       3  
Dependents    15  
Education      0  
Self_Employed 32  
ApplicantIncome 0  
CoapplicantIncome 0  
LoanAmount     22  
Loan_Amount_Term 14  
Credit_History 50  
Property_Area   0  
Loan_Status     0  
dtype: int64
```

```
In [16]: new_gender=[]  
for i in loan_df['Gender']:   
    if i not in loan_df['Gender'].unique():  
        new_gender.append(loan_df['Gender'].mode().values[0])  
    else:  
        new_gender.append(i)
```

```
In [17]: loan_df['Gender']=new_gender
```

```
In [18]: loan_df['Gender'].value_counts()
```

```
Out[18]: Gender  
Male      502  
Female    112  
Name: count, dtype: int64
```

```
In [19]: loan_df.isnull().sum()
```

```
Out[19]: Loan_ID      0  
Gender        0  
Married       3  
Dependents    15  
Education      0  
Self_Employed 32  
ApplicantIncome 0  
CoapplicantIncome 0  
LoanAmount     22  
Loan_Amount_Term 14  
Credit_History 50  
Property_Area   0  
Loan_Status     0  
dtype: int64
```

```
In [20]: loan_df['Married'].value_counts()
```

```
Out[20]: Married  
Yes      398  
No       213  
Name: count, dtype: int64
```

```
In [21]: loan_df['Married'].unique()
```

```
Out[21]: array(['No', 'Yes', nan], dtype=object)
```

```
In [23]: new_Married_data=[]
for i in loan_df['Married']:
    if i not in loan_df['Married'].unique():
        new_Married_data.append(loan_df['Married'].mode().values[0])
    else:
        new_Married_data.append(i)
```

```
In [24]: loan_df['Married']=new_Married_data
```

```
In [25]: loan_df['Married'].value_counts()
```

```
Out[25]: Married
      Yes    401
      No     213
Name: count, dtype: int64
```

```
In [26]: loan_df.isnull().sum()
```

```
Out[26]: Loan_ID          0
Gender           0
Married          0
Dependents      15
Education        0
Self_Employed   32
ApplicantIncome  0
CoapplicantIncome 0
LoanAmount       22
Loan_Amount_Term 14
Credit_History   50
Property_Area    0
Loan_Status       0
dtype: int64
```

```
In [31]: loan_df['Dependents'].value_counts()
```

```
Out[31]: Dependents
      0    345
      1    102
      2    101
      3+    51
Name: count, dtype: int64
```

```
In [43]: new_data=[]
for i in loan_df['Dependents']:
    if i not in loan_df['Dependents'].unique():
        new_data.append(loan_df['Dependents'].mode().values[0])
    else:
        new_data.append(i)
```

```
In [41]: loan_df['Dependents']=new_data
```

```
In [42]: loan_df['Dependents'].value_counts()
```

```
Out[42]: Dependents
0      360
1      102
2      101
3+     51
Name: count, dtype: int64
```

```
In [47]: loan_df.isnull().sum()
```

```
Out[47]: Loan_ID          0
Gender           0
Married          0
Dependents       0
Education         0
Self_Employed    32
ApplicantIncome   0
CoapplicantIncome 0
LoanAmount        22
Loan_Amount_Term  14
Credit_History    50
Property_Area     0
Loan_Status        0
dtype: int64
```

```
In [49]: new_data1=[]
for i in loan_df['Dependents']:
    if len(i)>1:
        new_data1.append(int(float(i[0])))
    else:
        new_data1.append(int(float(i)))
```

```
In [50]: loan_df['Dependents']=new_data1
```

```
In [51]: loan_df['Dependents'].value_counts()
```

```
Out[51]: Dependents
0      360
1      102
2      101
3      51
Name: count, dtype: int64
```

```
In [52]: loan_df.isnull().sum()
```

```
Out[52]: Loan_ID      0  
Gender        0  
Married       0  
Dependents    0  
Education      0  
Self_Employed 32  
ApplicantIncome 0  
CoapplicantIncome 0  
LoanAmount     22  
Loan_Amount_Term 14  
Credit_History 50  
Property_Area   0  
Loan_Status     0  
dtype: int64
```

```
In [53]: loan_df['Self_Employed'].value_counts()
```

```
Out[53]: Self_Employed  
No      500  
Yes     82  
Name: count, dtype: int64
```

```
In [54]: loan_df['Self_Employed'].unique()
```

```
Out[54]: array(['No', 'Yes', nan], dtype=object)
```

```
In [55]: new_data2=[]  
for i in loan_df['Self_Employed']:  
    if i not in loan_df['Self_Employed'].unique():  
        new_data2.append(loan_df['Self_Employed'].mode().values[0])  
    else:  
        new_data2.append(i)
```

```
In [56]: loan_df['Self_Employed']=new_data2
```

```
In [57]: loan_df['Self_Employed'].value_counts()
```

```
Out[57]: Self_Employed  
No      532  
Yes     82  
Name: count, dtype: int64
```

```
In [58]: loan_df.isnull().sum()
```

```
Out[58]: Loan_ID      0
Gender        0
Married       0
Dependents    0
Education     0
Self_Employed 0
ApplicantIncome 0
CoapplicantIncome 0
LoanAmount    22
Loan_Amount_Term 14
Credit_History 50
Property_Area  0
Loan_Status    0
dtype: int64
```

```
In [59]: loan_df['LoanAmount'].unique()
```

```
Out[59]: array([ nan, 128., 66., 120., 141., 267., 95., 158., 168., 349., 70.,
   109., 200., 114., 17., 125., 100., 76., 133., 115., 104., 315.,
   116., 112., 151., 191., 122., 110., 35., 201., 74., 106., 320.,
   144., 184., 80., 47., 75., 134., 96., 88., 44., 286., 97.,
   135., 180., 99., 165., 258., 126., 312., 136., 172., 81., 187.,
   113., 176., 130., 111., 167., 265., 50., 210., 175., 131., 188.,
   25., 137., 160., 225., 216., 94., 139., 152., 118., 185., 154.,
   85., 259., 194., 93., 370., 182., 650., 102., 290., 84., 242.,
   129., 30., 244., 600., 255., 98., 275., 121., 63., 700., 87.,
   101., 495., 67., 73., 260., 108., 58., 48., 164., 170., 83.,
   90., 166., 124., 55., 59., 127., 214., 240., 72., 60., 138.,
   42., 280., 140., 155., 123., 279., 192., 304., 330., 150., 207.,
   436., 78., 54., 89., 143., 105., 132., 480., 56., 159., 300.,
   376., 117., 71., 490., 173., 46., 228., 308., 236., 570., 380.,
   296., 156., 103., 45., 65., 53., 360., 62., 218., 178., 239.,
   405., 148., 190., 149., 153., 162., 230., 86., 234., 246., 500.,
   186., 119., 107., 209., 208., 243., 40., 250., 311., 400., 161.,
   196., 324., 157., 145., 181., 26., 211., 9., 205., 36., 61.,
   146., 292., 142., 350., 496., 253.])
```

```
In [60]: loan_df['LoanAmount'].value_counts()
```

```
Out[60]: LoanAmount
120.0    20
110.0    17
100.0    15
160.0    12
187.0    12
...
240.0    1
214.0    1
59.0     1
166.0    1
253.0    1
Name: count, Length: 203, dtype: int64
```

```
In [62]: new_data3=[]
for i in loan_df['LoanAmount']:
    if i not in loan_df['LoanAmount'].unique():
```

```

    new_data3.append(loan_df['LoanAmount'].median())
else:
    new_data3.append(i)

```

In [63]: `loan_df['LoanAmount'] = new_data3`

In [64]: `loan_df.isnull().sum()`

Out[64]:

Loan_ID	0
Gender	0
Married	0
Dependents	0
Education	0
Self_Employed	0
ApplicantIncome	0
CoapplicantIncome	0
LoanAmount	0
Loan_Amount_Term	14
Credit_History	50
Property_Area	0
Loan_Status	0

dtype: int64

In [65]: `loan_df['Loan_Amount_Term'].value_counts()`

Out[65]:

Loan_Amount_Term	count
360.0	512
180.0	44
480.0	15
300.0	13
240.0	4
84.0	4
120.0	3
60.0	2
36.0	2
12.0	1

Name: count, dtype: int64

In [66]: `loan_df['Loan_Amount_Term'].unique()`

Out[66]: `array([360., 120., 240., nan, 180., 60., 300., 480., 36., 84., 12.])`

In [67]:

```

new_data4=[]
for i in loan_df['Loan_Amount_Term']:
    if i not in loan_df['Loan_Amount_Term'].unique():
        new_data4.append(loan_df['Loan_Amount_Term'].median())
    else:
        new_data4.append(i)

```

In [68]: `loan_df['Loan_Amount_Term'] = new_data4`

In [69]: `loan_df.isnull().sum()`

```
Out[69]: Loan_ID      0  
Gender        0  
Married       0  
Dependents    0  
Education     0  
Self_Employed 0  
ApplicantIncome 0  
CoapplicantIncome 0  
LoanAmount    0  
Loan_Amount_Term 0  
Credit_History 50  
Property_Area  0  
Loan_Status    0  
dtype: int64
```

```
In [70]: loan_df['Credit_History'].value_counts()
```

```
Out[70]: Credit_History  
1.0    475  
0.0     89  
Name: count, dtype: int64
```

```
In [71]: loan_df['Credit_History'].unique()
```

```
Out[71]: array([ 1.,  0., nan])
```

```
In [73]: new_data5=[]  
for i in loan_df['Credit_History']:  
    if i not in loan_df['Credit_History'].unique():  
        new_data5.append(loan_df['Credit_History'].median())  
    else:  
        new_data5.append(i)
```

```
In [74]: loan_df['Credit_History']=new_data5
```

```
In [75]: loan_df.isnull().sum()
```

```
Out[75]: Loan_ID      0  
Gender        0  
Married       0  
Dependents    0  
Education     0  
Self_Employed 0  
ApplicantIncome 0  
CoapplicantIncome 0  
LoanAmount    0  
Loan_Amount_Term 0  
Credit_History 0  
Property_Area  0  
Loan_Status    0  
dtype: int64
```

```
In [76]: loan_df.drop(['Loan_ID'],axis=1,inplace=True)
```

In [77]: loan_df

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Terms	Interest_Rate	Loan_Status
0	Male	No	0	Graduate	No	5849	2583	4583	Yes	12%	Approved
1	Male	Yes	1	Graduate	No	4583	3000	3000	Yes	12%	Approved
2	Male	Yes	0	Graduate	Yes	2583	6000	6000	Yes	12%	Approved
3	Male	Yes	0	Not Graduate	No	2583	2583	2583	Yes	12%	Approved
4	Male	No	0	Graduate	No	6000	2900	2900	Yes	12%	Approved
...
609	Female	No	0	Graduate	No	2900	4106	4106	Yes	12%	Approved
610	Male	Yes	3	Graduate	No	8072	7583	7583	Yes	12%	Approved
611	Male	Yes	1	Graduate	No	7583	4583	4583	Yes	12%	Approved
612	Male	Yes	2	Graduate	No	4583	2583	2583	Yes	12%	Approved
613	Female	No	0	Graduate	Yes	4583	4583	4583	Yes	12%	Approved

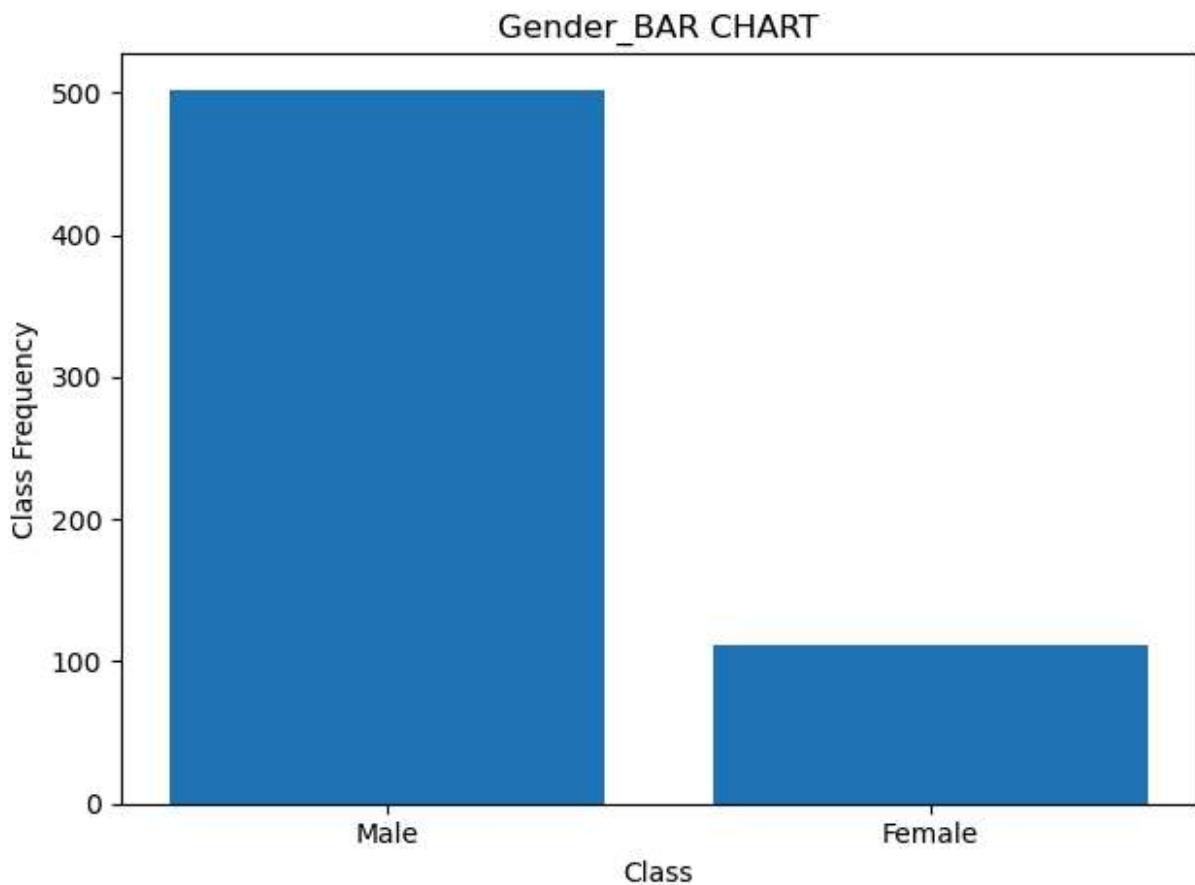
614 rows × 12 columns

In [78]: cat=loan_df.select_dtypes(include='object').columns
num=loan_df.select_dtypes(exclude='object').columns

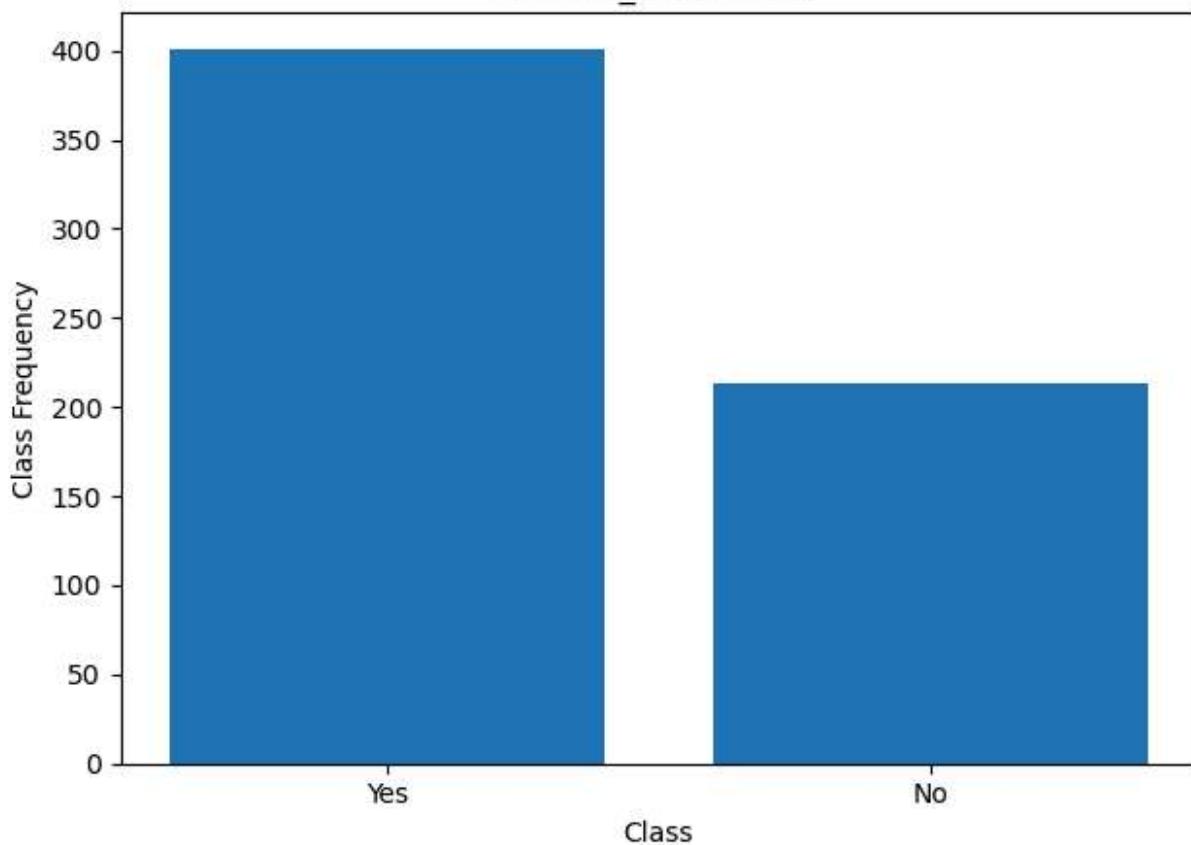
```
import os
dir=r'C:\Users\kisha\OneDrive\Desktop\Anudip\pythonProject'
folder='FREQUENCY TABLE'
new_dir=os.path.join(dir,folder)
os.makedirs(new_dir)
for i in cat:
    keys=loan_df[i].value_counts().keys()
    values=loan_df[i].value_counts().values
    df=pd.DataFrame(zip(keys,values),columns=[ 'LABELS ','COUNTS '])
    df.to_csv(os.path.join(new_dir,f'{i}_table.csv'),index=False)
```

```
import os
dir=r'C:\Users\kisha\OneDrive\Desktop\Anudip\pythonProject'
folder='BAR CHART'
new_dir=os.path.join(dir,folder)
os.makedirs(new_dir)
for i in cat:
    keys=loan_df[i].value_counts().keys()
    values=loan_df[i].value_counts().values
    plt.bar(keys,values)
    plt.xlabel('Class')
    plt.ylabel('Class Frequency')
```

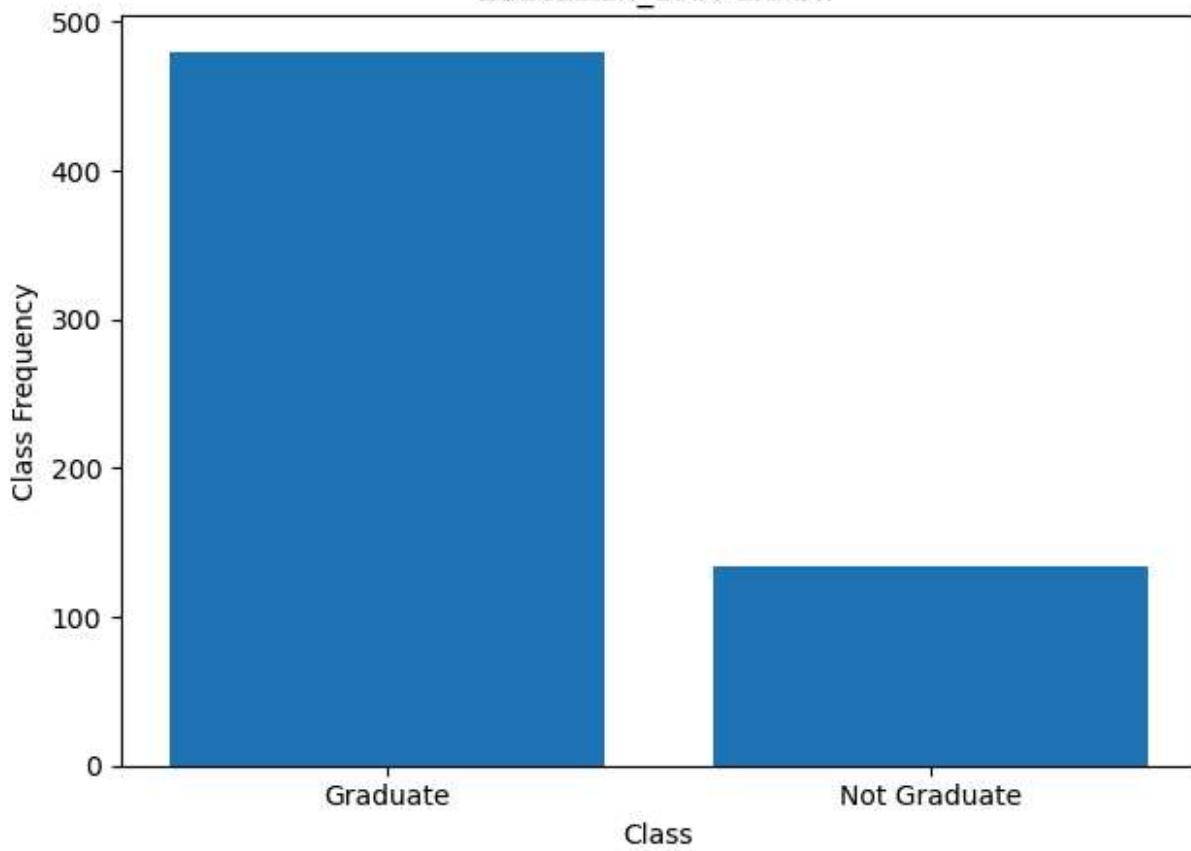
```
plt.title(f'{i}_BAR CHART')
plt.savefig(os.path.join(new_dir,f'{i}_bar chart.jpg'))
plt.tight_layout()
plt.show()
```



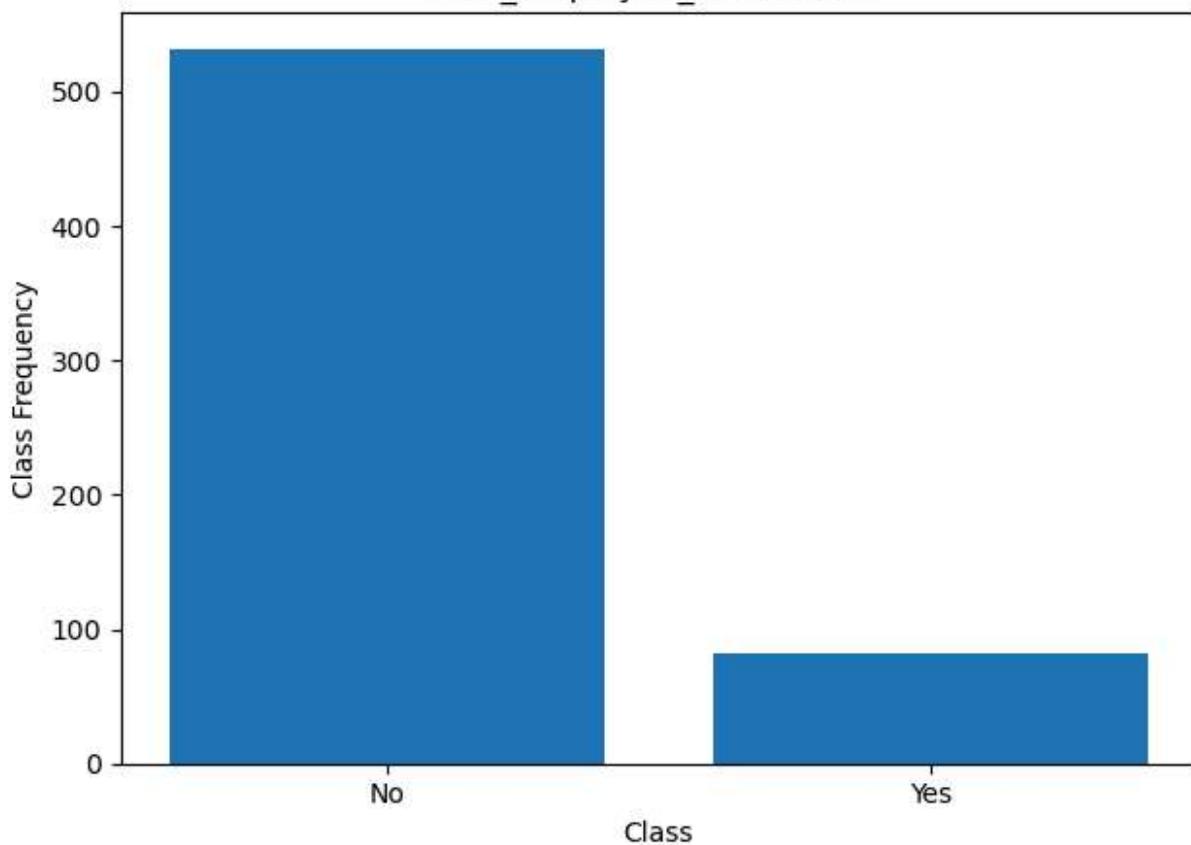
Married_BAR CHART



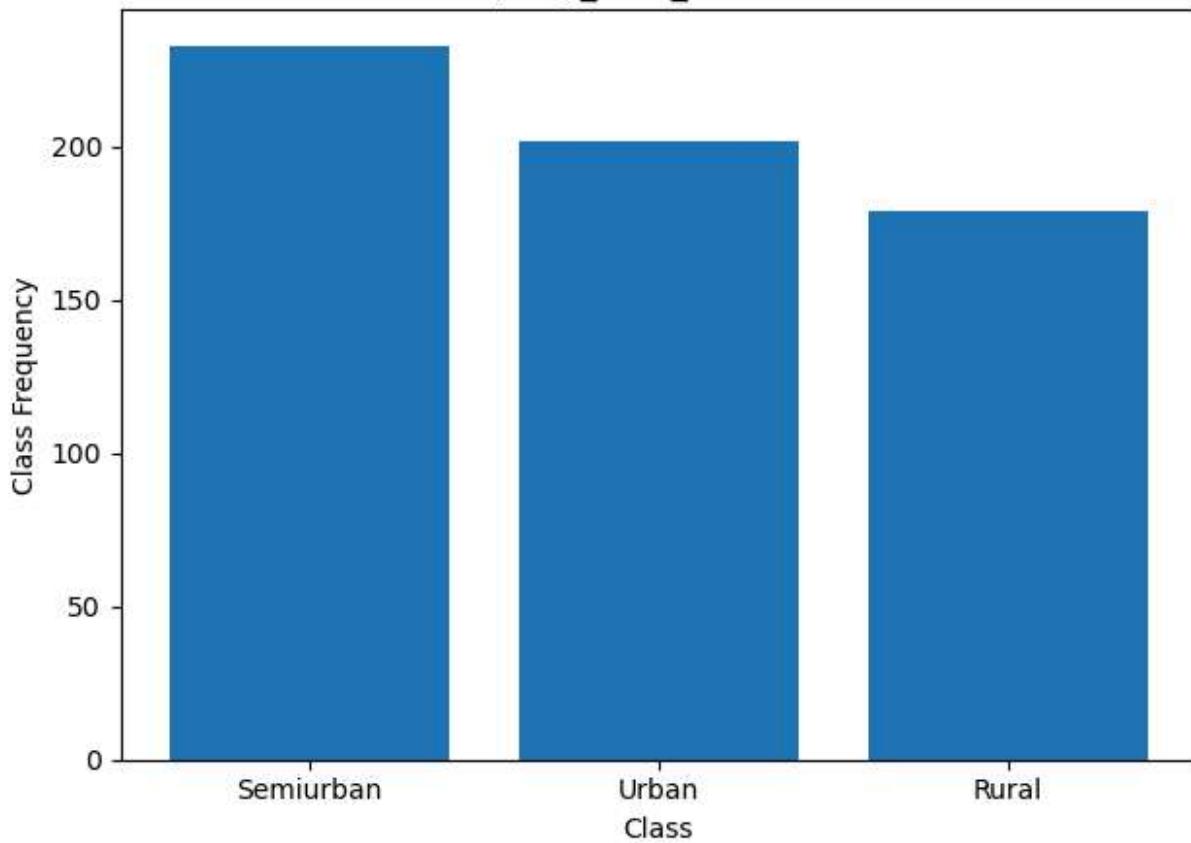
Education_BAR CHART

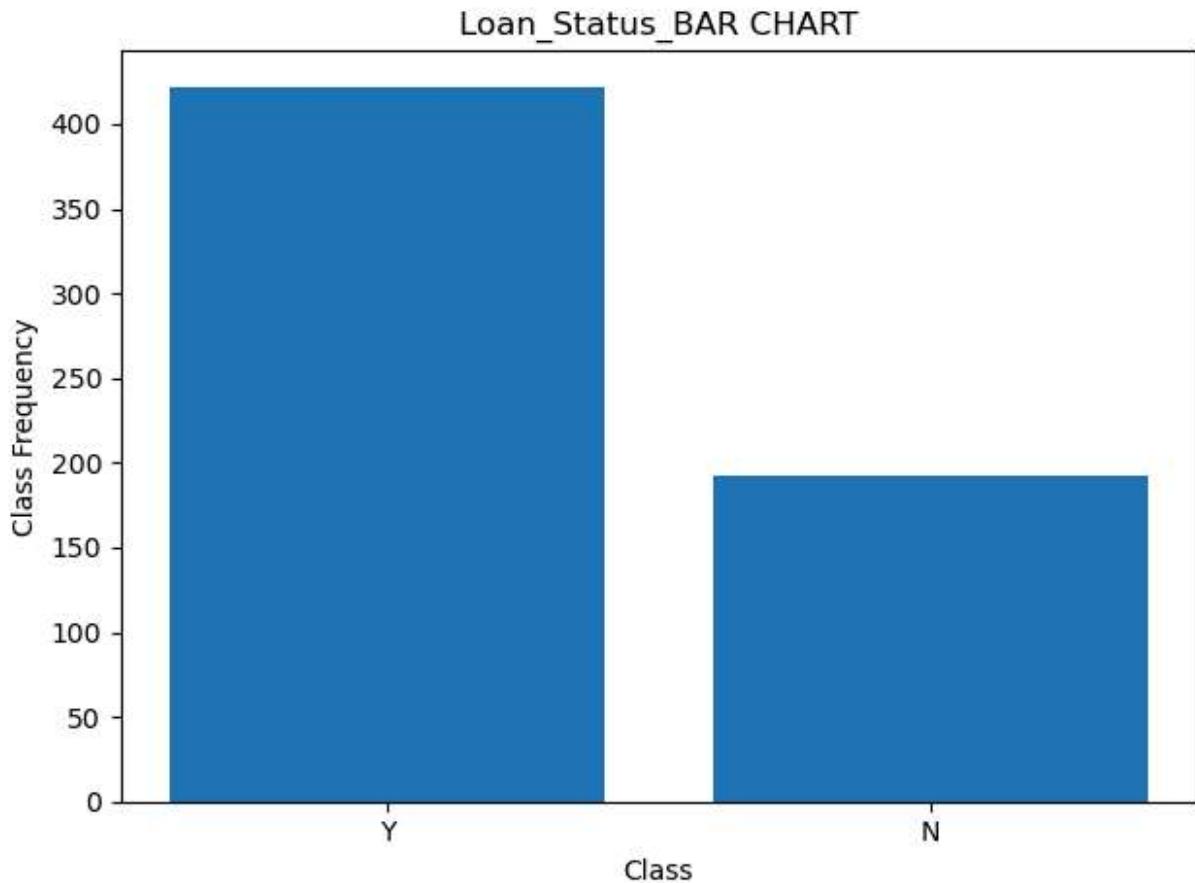


Self_Employed_BAR CHART



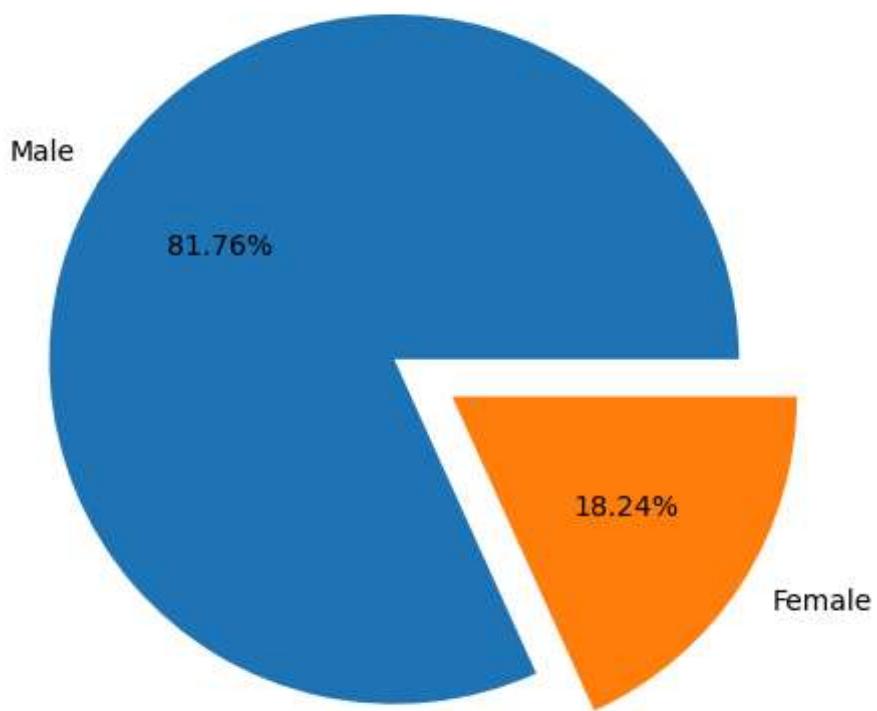
Property_Area_BAR CHART



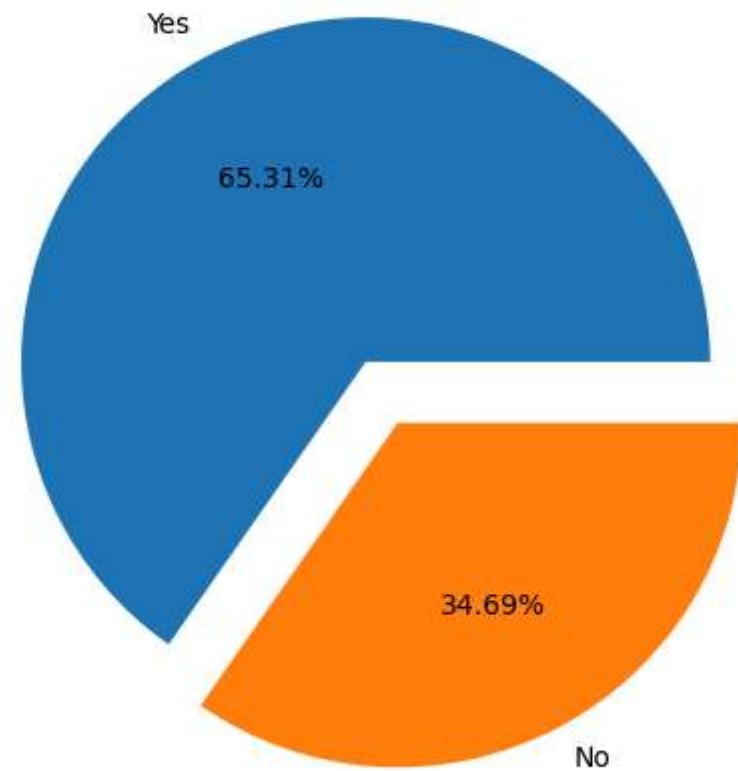


```
In [82]: import os
dir=r'C:\Users\kisha\OneDrive\Desktop\Anudip\pythonProject'
folder='PIE CHART'
new_dir=os.path.join(dir,folder)
os.makedirs(new_dir)
for i in cat:
    keys=loan_df[i].value_counts().keys()
    values=loan_df[i].value_counts().values
    explode=[0.1]*len(keys)
    plt.pie(values,labels=keys,autopct='%0.2f%%',explode=explode)
    plt.title(f'{i}_bar chart.jpg')
    plt.savefig(os.path.join(new_dir,f'{i}_pie chart.jpg'))
    plt.tight_layout()
    plt.show()
```

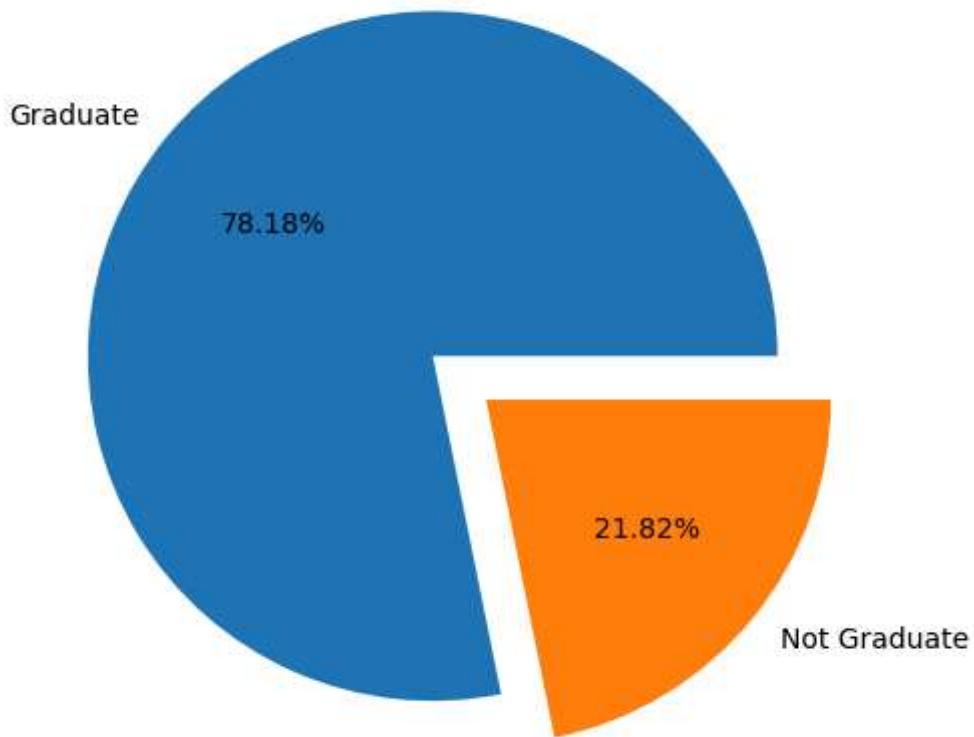
Gender_bar chart.jpg



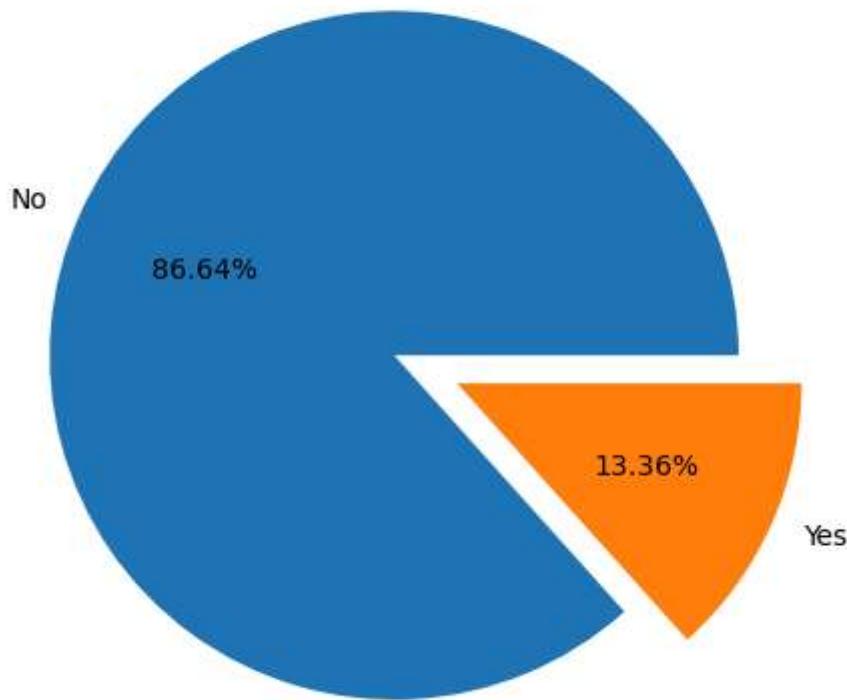
Married_bar chart.jpg



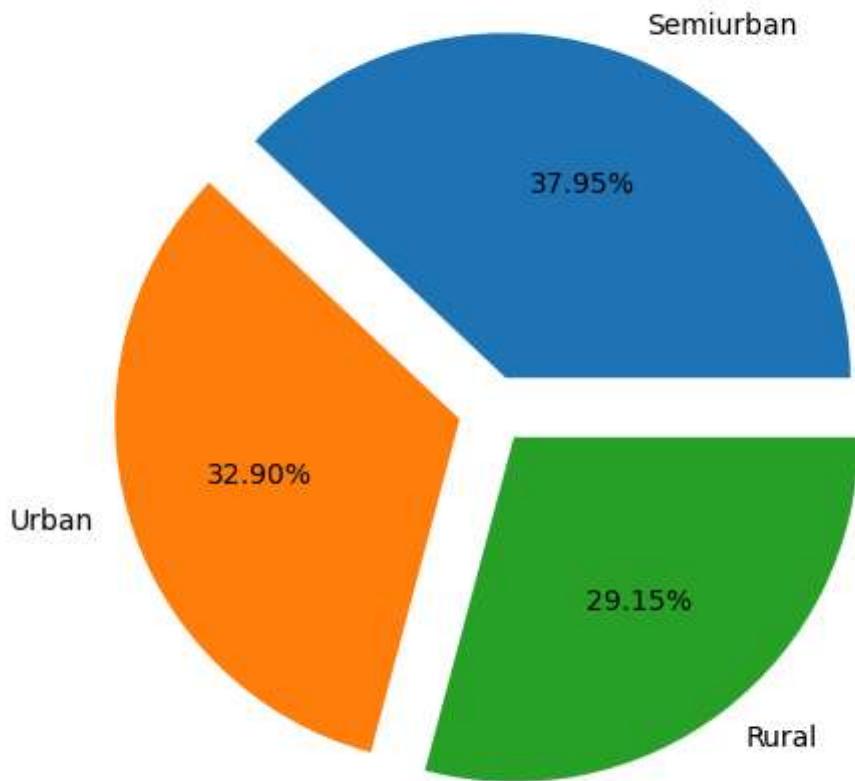
Education_bar chart.jpg



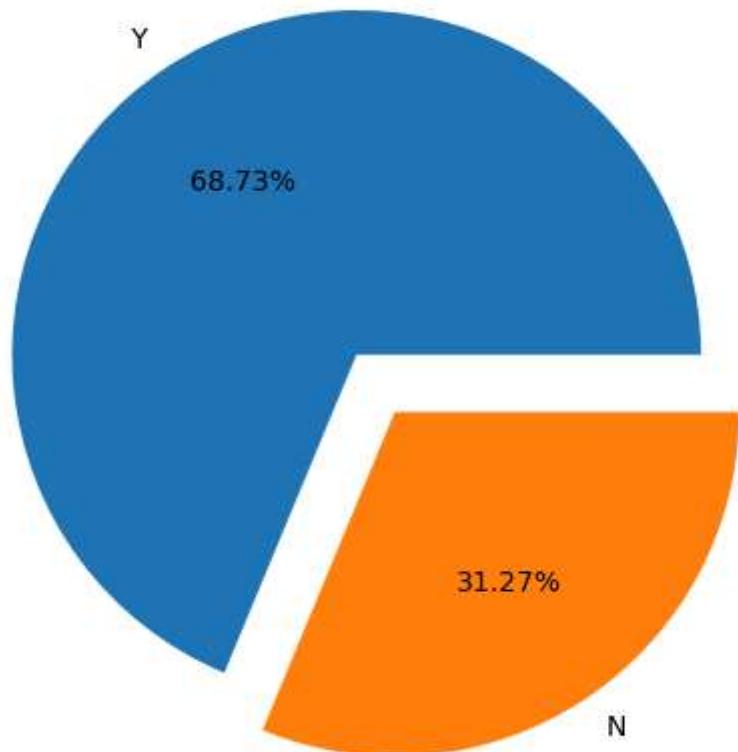
Self_Employed_bar chart.jpg



Property_Area_bar chart.jpg



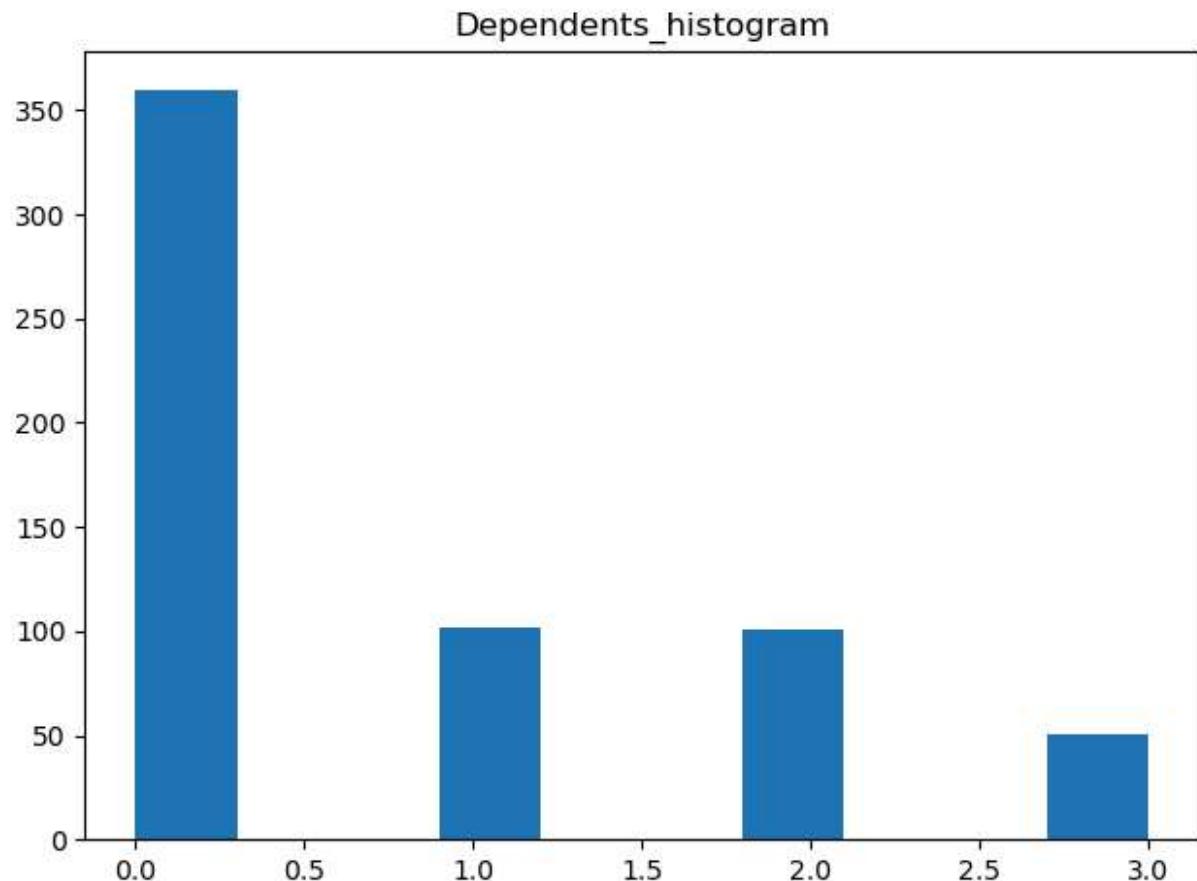
Loan_Status_bar chart.jpg



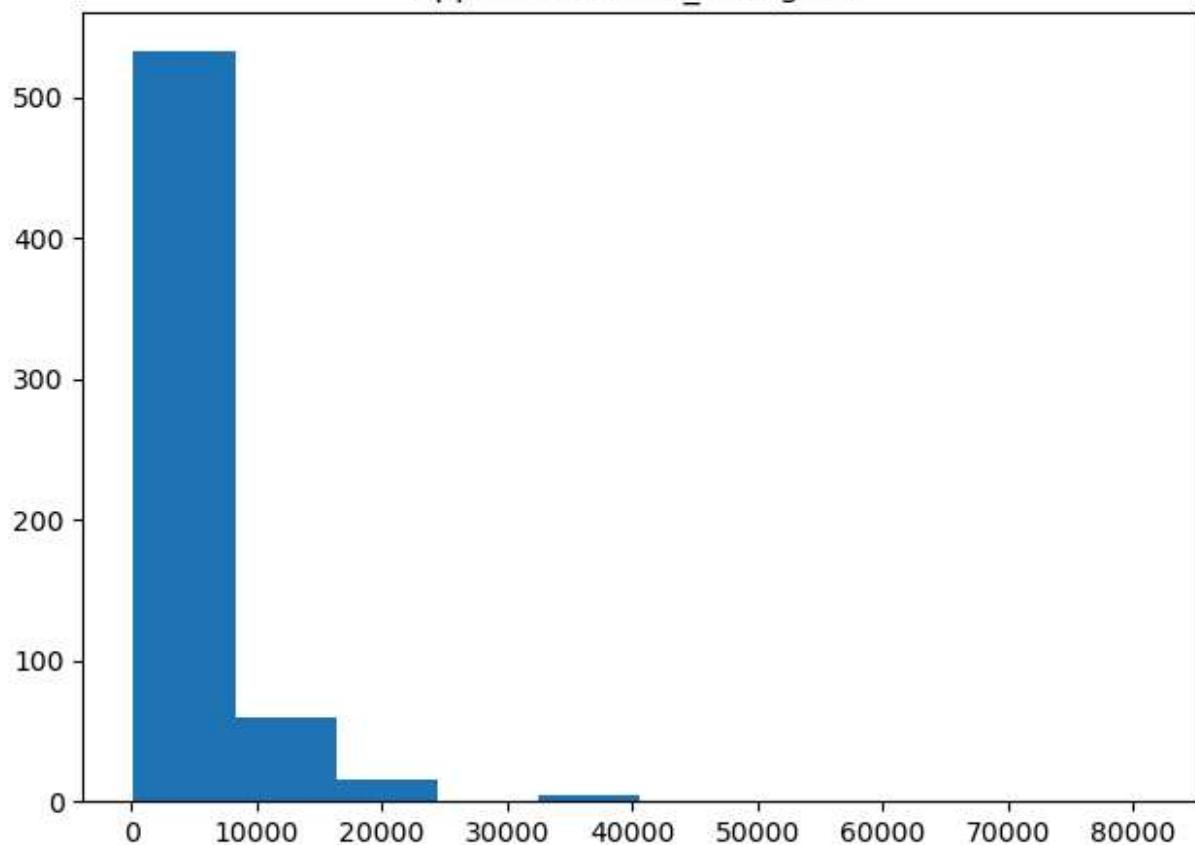
```
In [83]: num
```

```
Out[83]: Index(['Dependents', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History'],
      dtype='object')
```

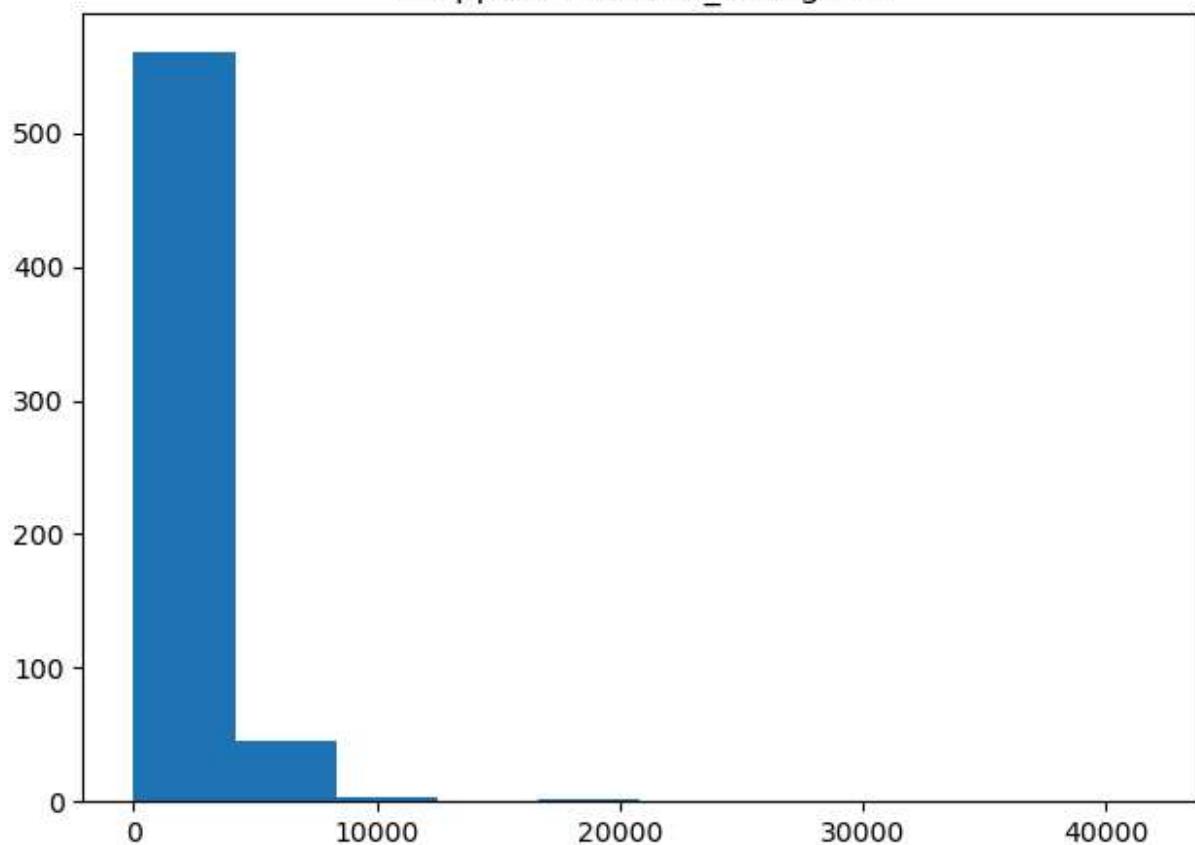
```
In [84]: for i in num:
    plt.hist(loan_df[i])
    plt.title(f'{i}_histogram')
    plt.tight_layout()
    plt.show()
```



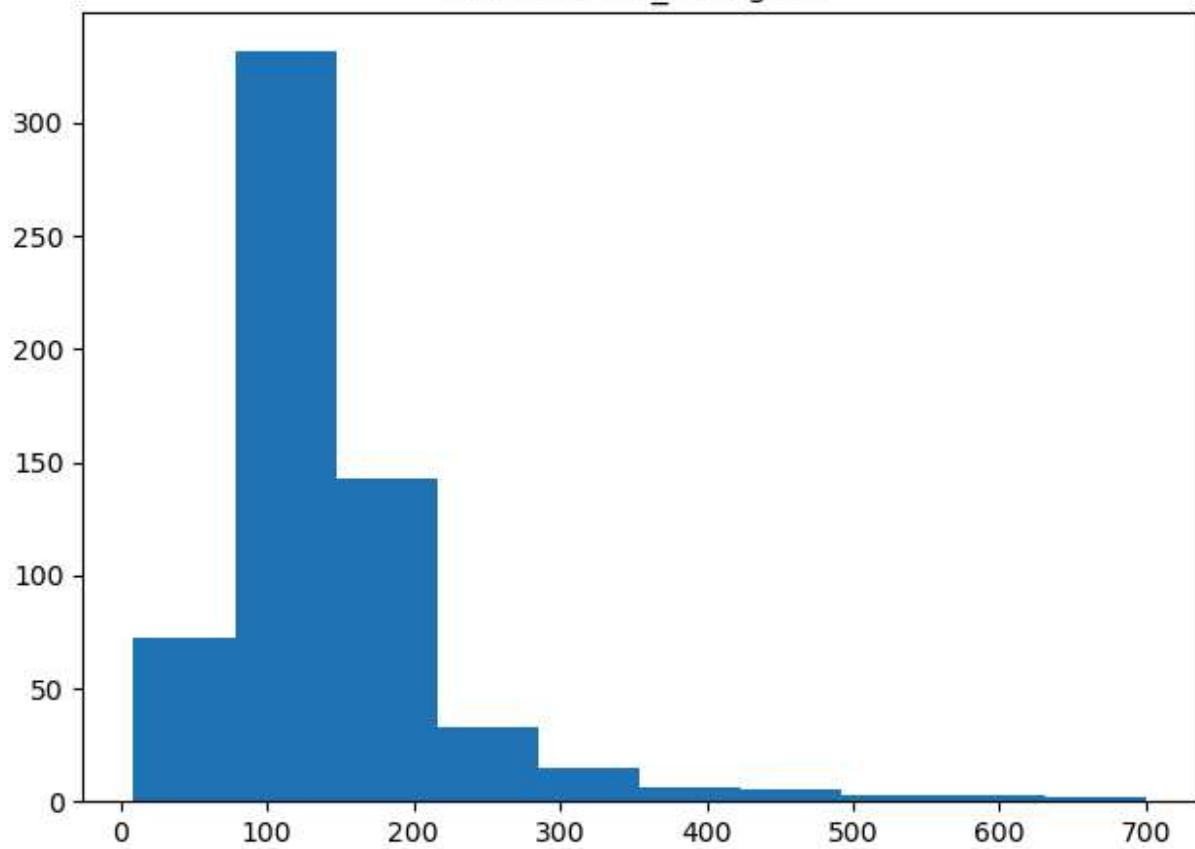
ApplicantIncome_histogram



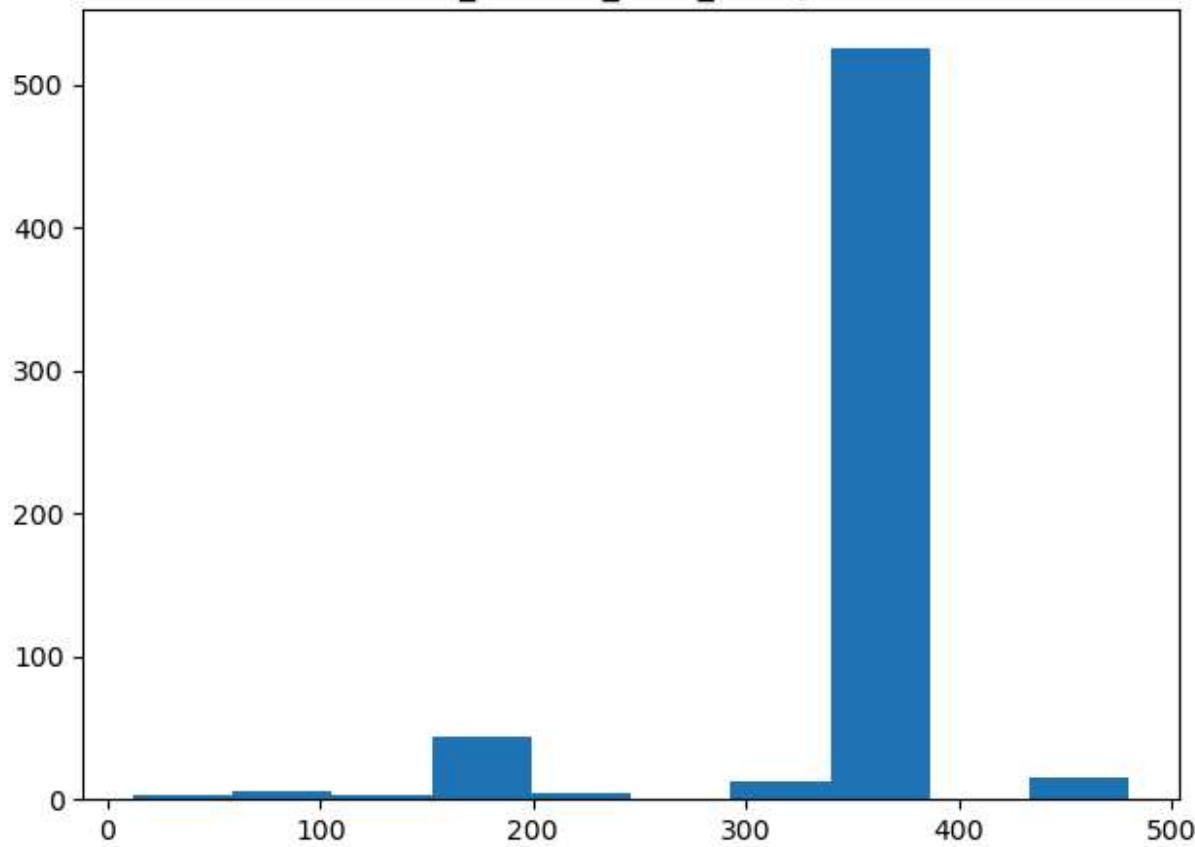
CoapplicantIncome_histogram

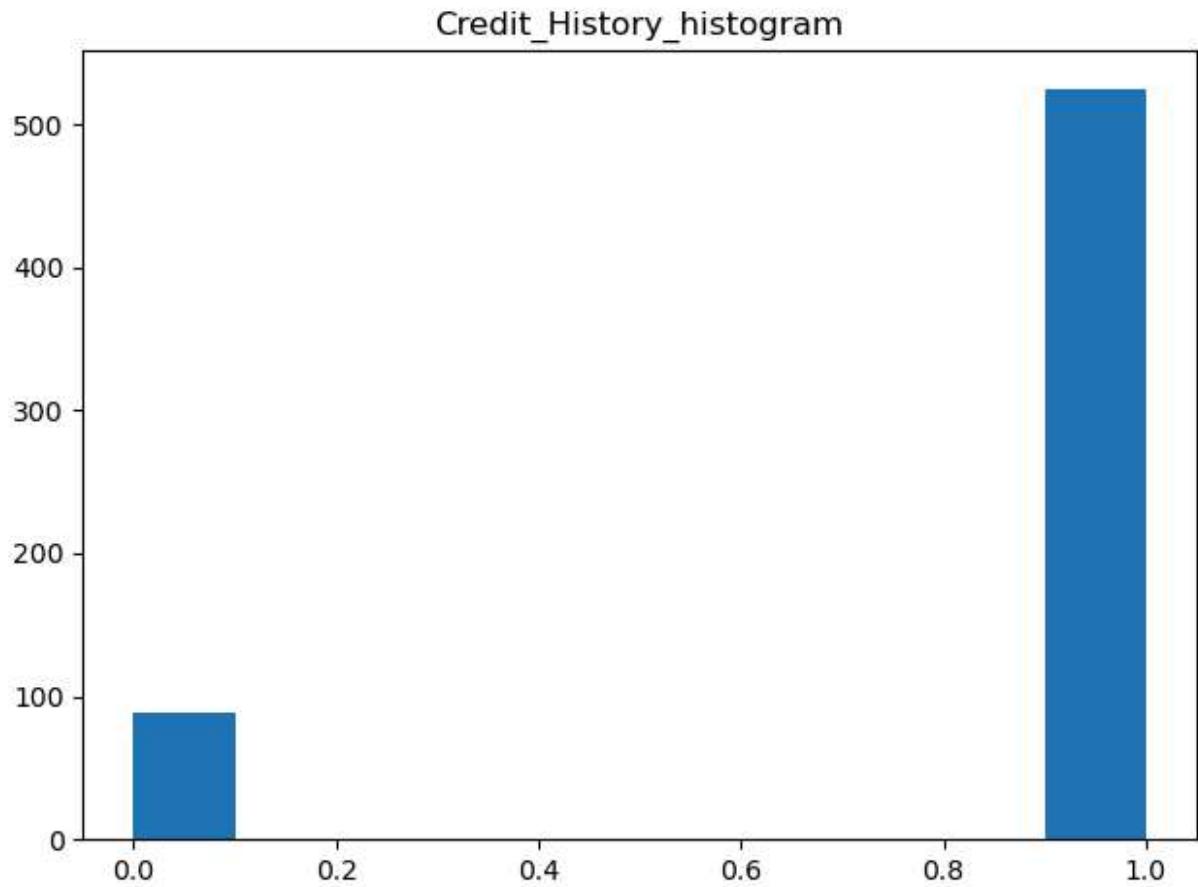


LoanAmount_histogram



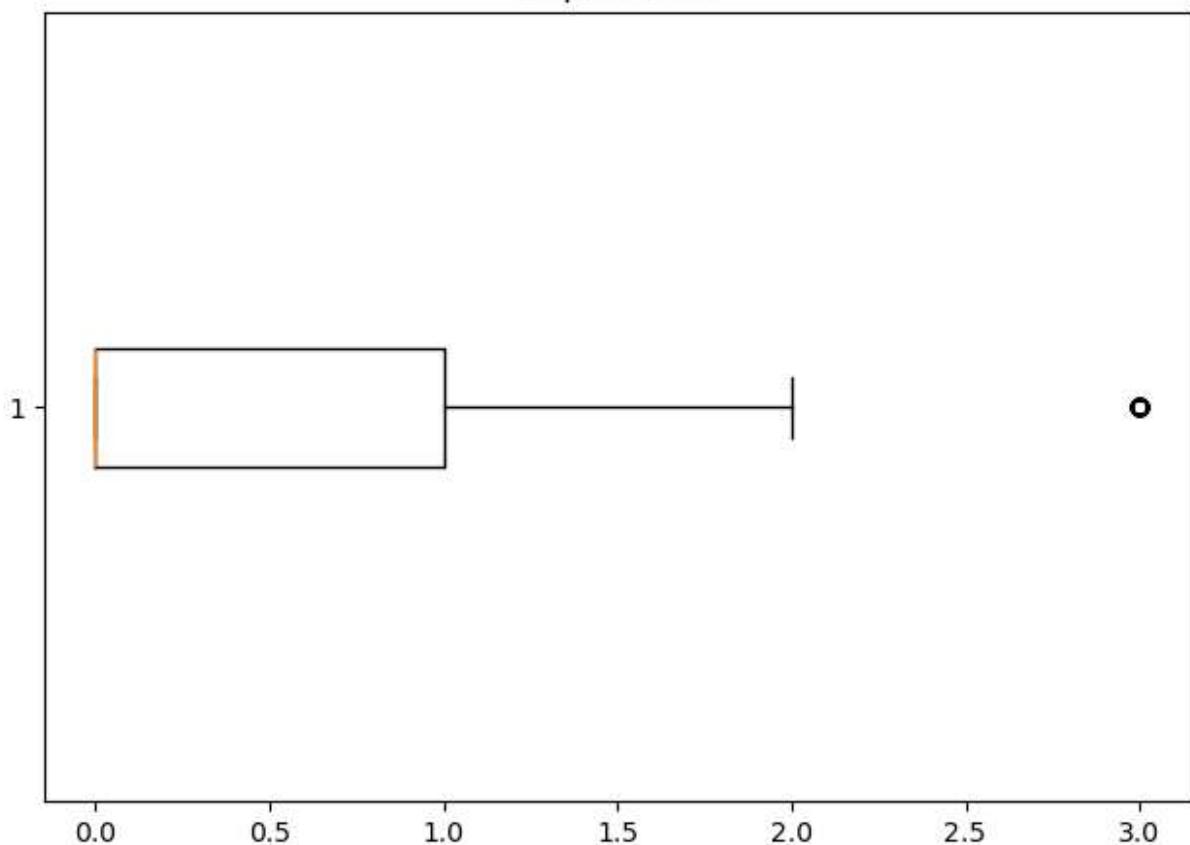
Loan_Amount_Term_histogram



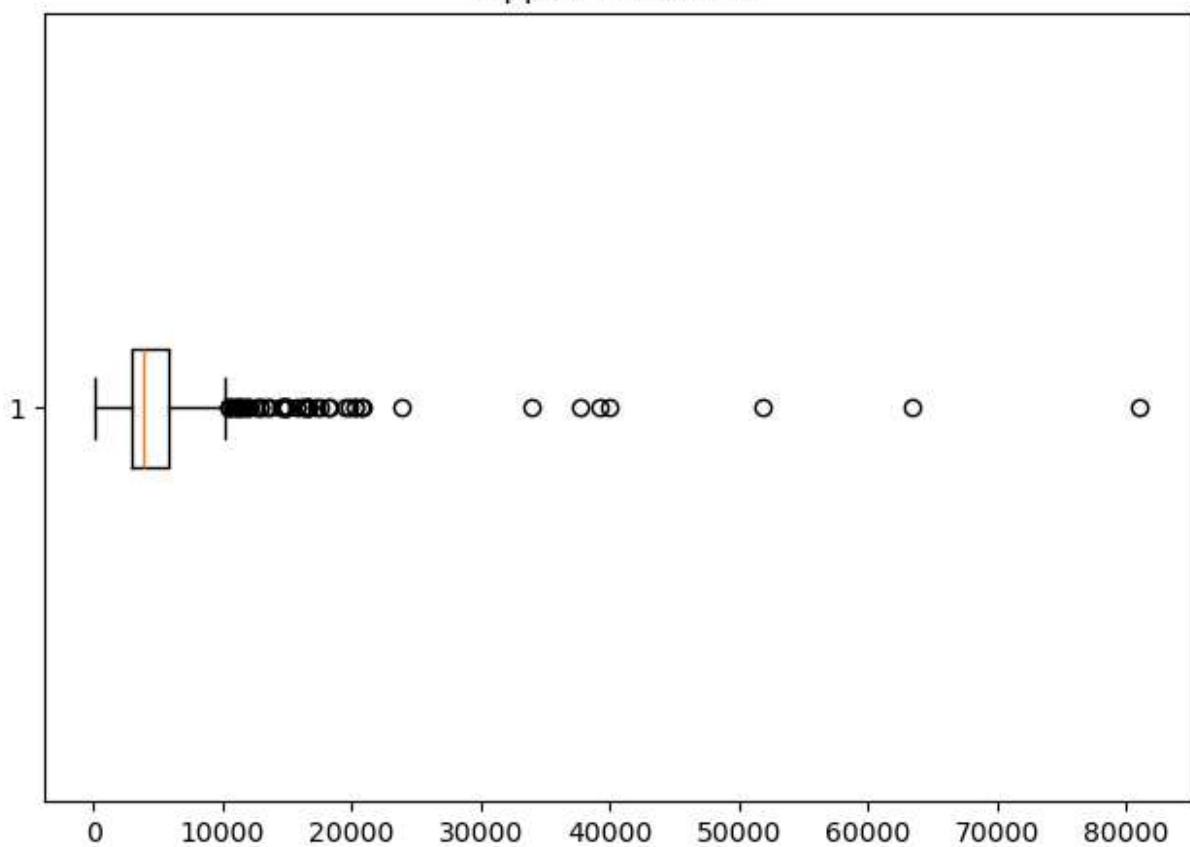


```
In [85]: for i in num:  
    plt.boxplot(loan_df[i],vert=False)  
    plt.title(f'{i}')  
    plt.tight_layout()  
    plt.show()
```

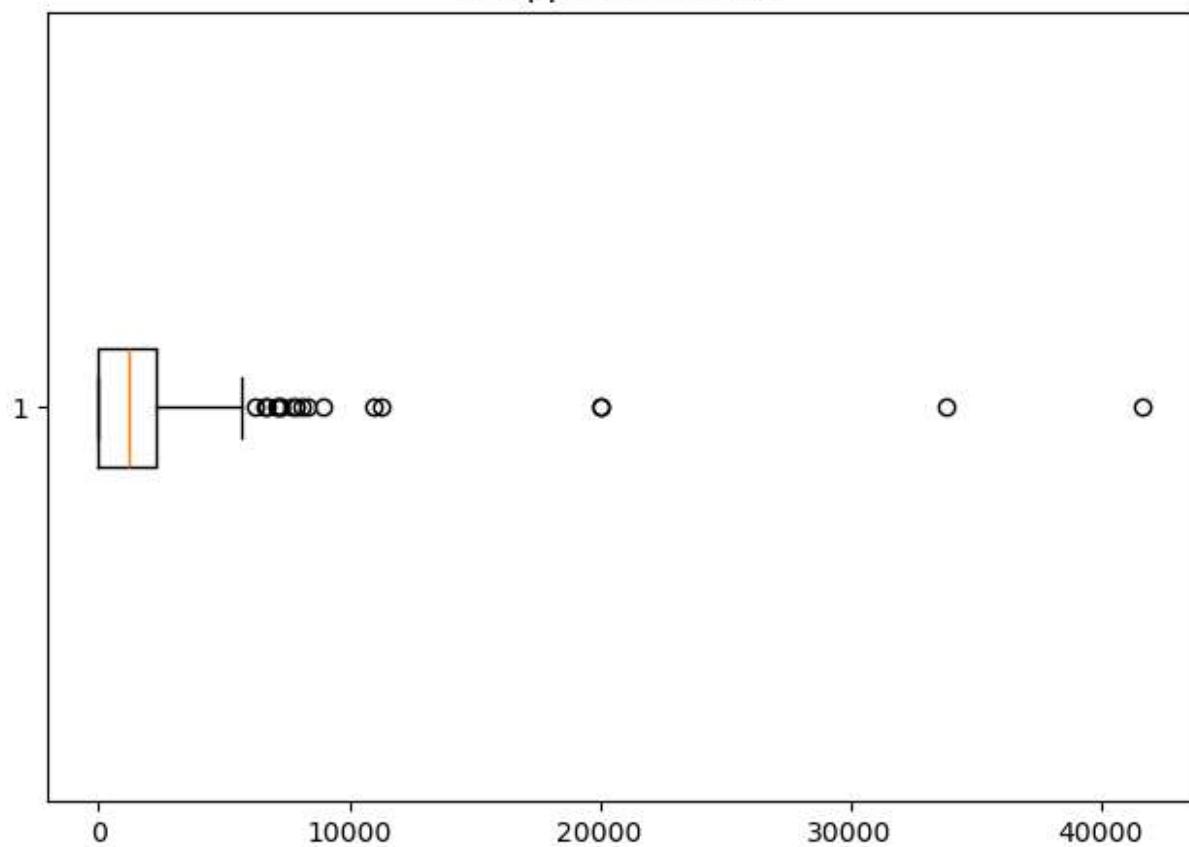
Dependents



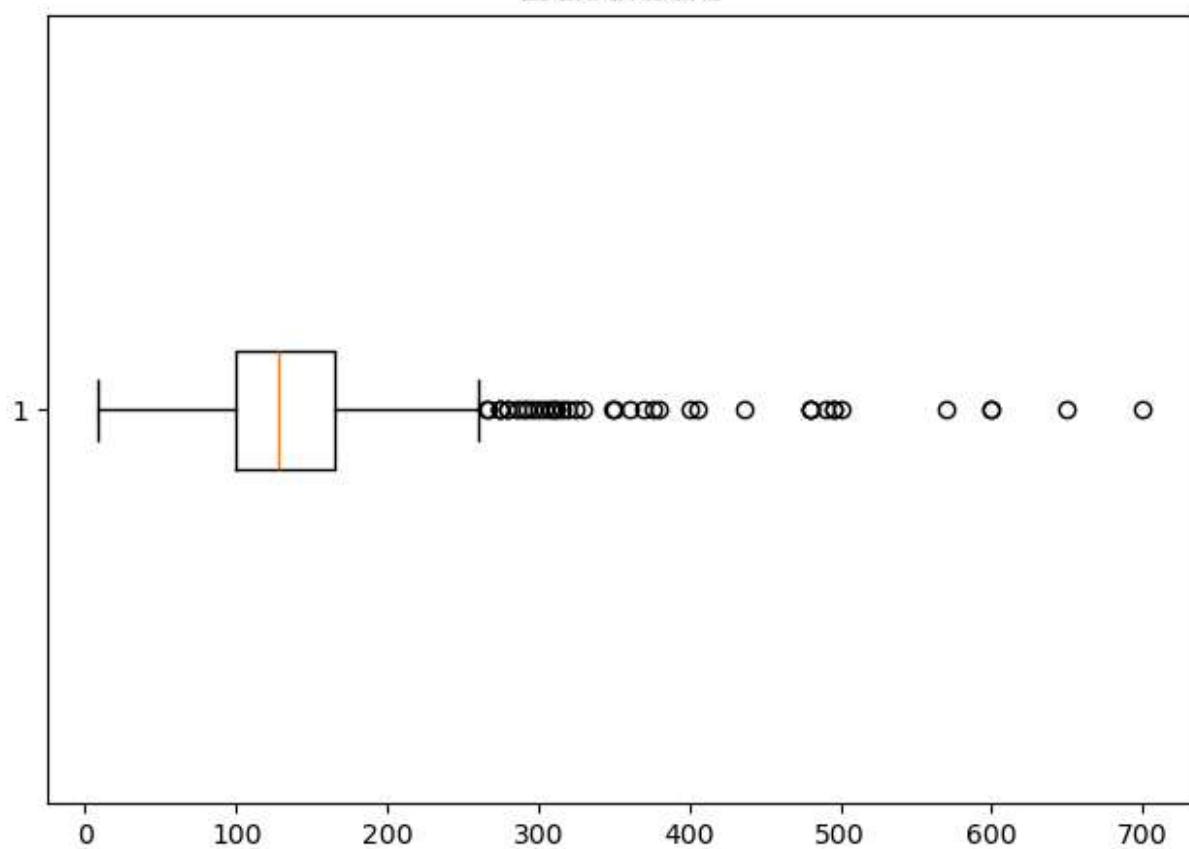
ApplicantIncome



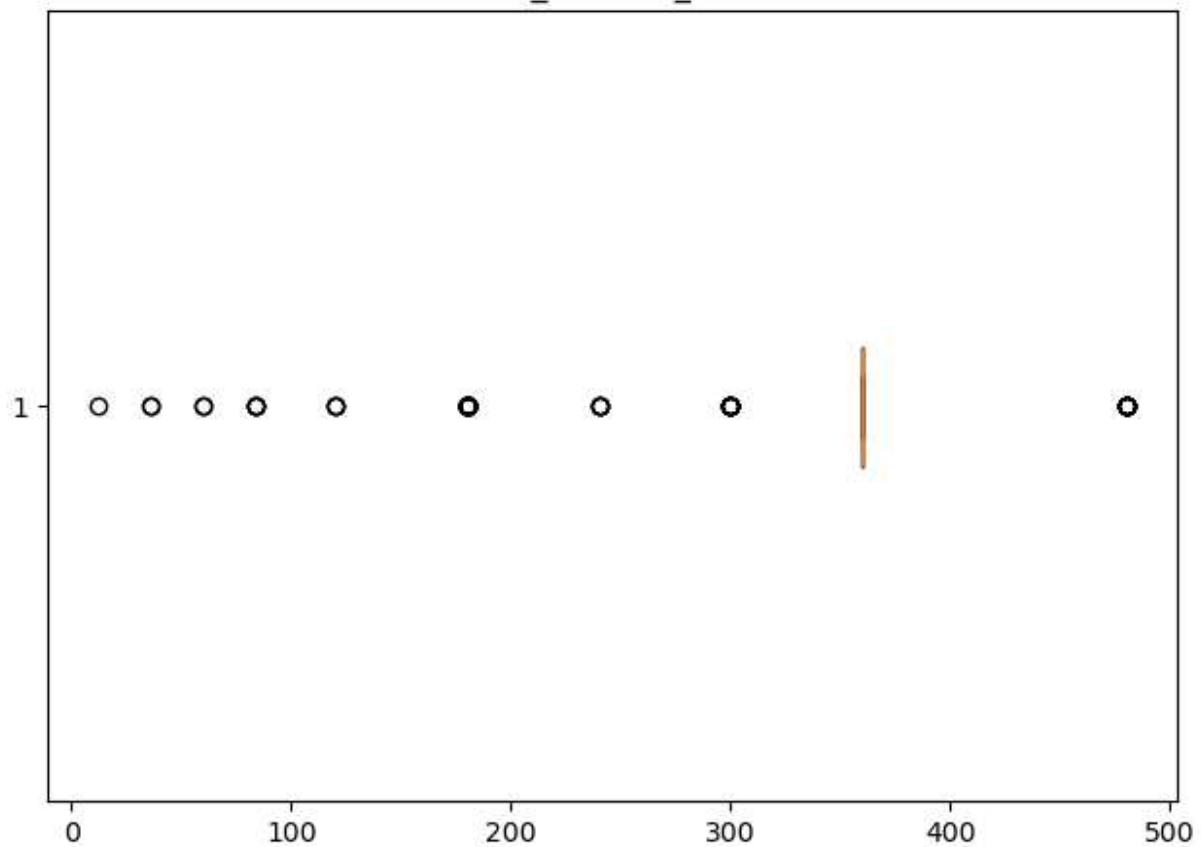
CoapplicantIncome



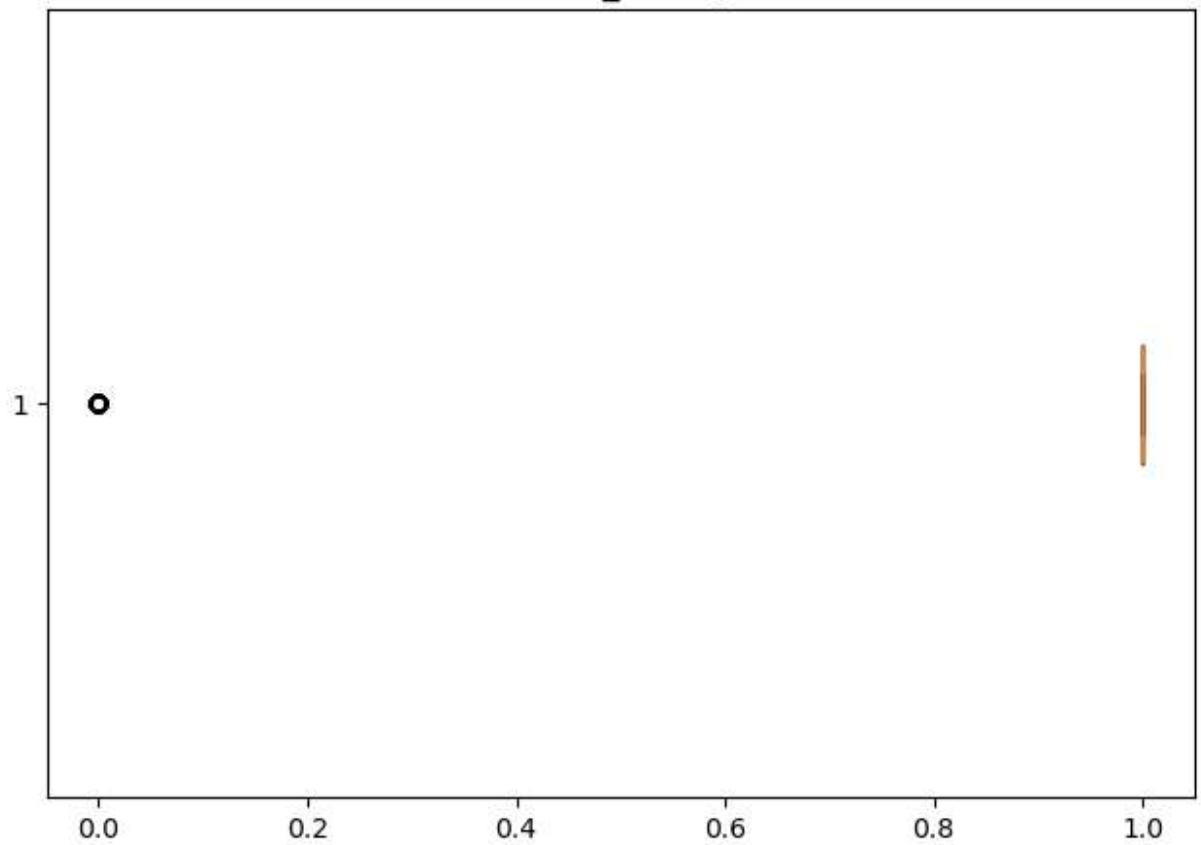
LoanAmount



Loan_Amount_Term



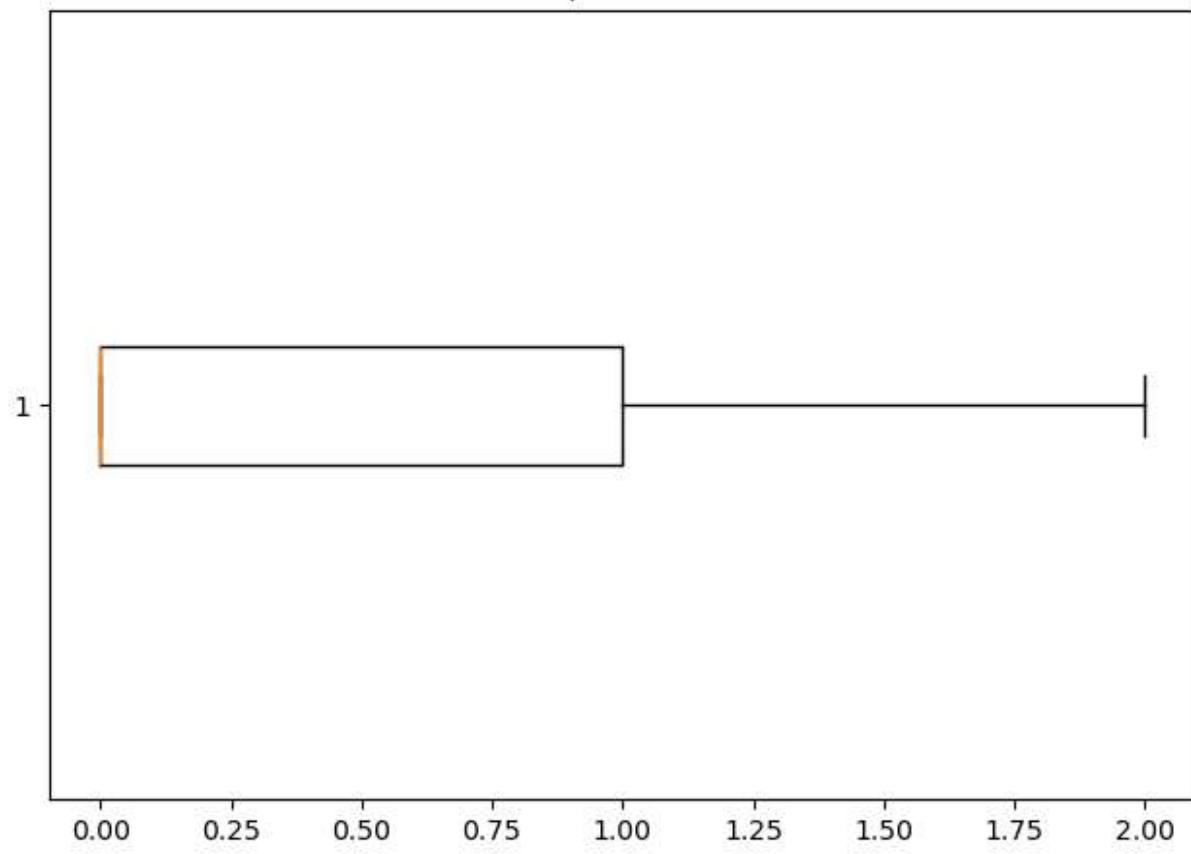
Credit_History



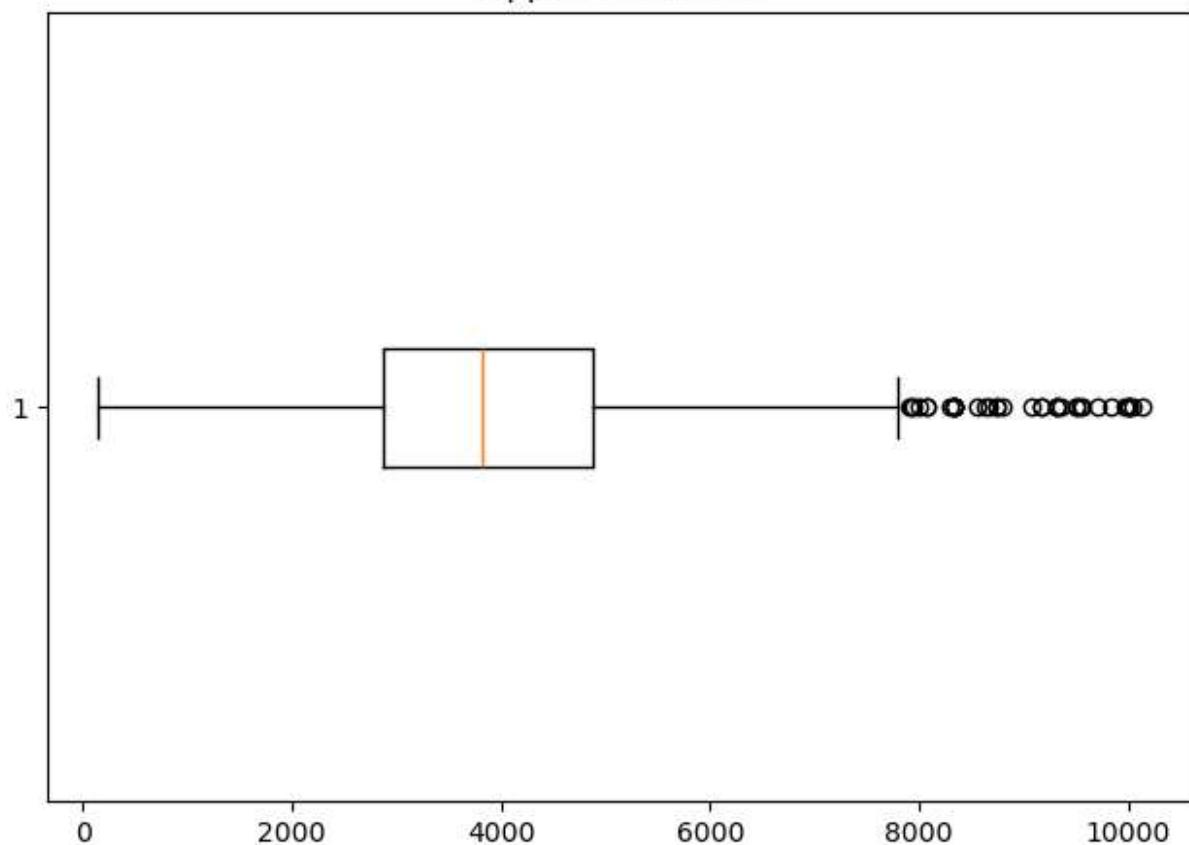
```
In [87]: for i in num:  
    q1=np.percentile(loan_df[i],25)  
    q3=np.percentile(loan_df[i],75)  
    iqr=q3-q1  
    lb=q1-1.5*iqr  
    ub=q3+1.5*iqr  
    new_data=[]  
    for j in loan_df[i]:  
        if j<lb or j>ub:  
            new_data.append(loan_df[i].median())  
        else:  
            new_data.append(j)  
    loan_df[f'{i}']=new_data
```

```
In [88]: for i in num:  
    plt.boxplot(loan_df[i],vert=False)  
    plt.title(f'{i}')  
    plt.tight_layout()  
    plt.show()
```

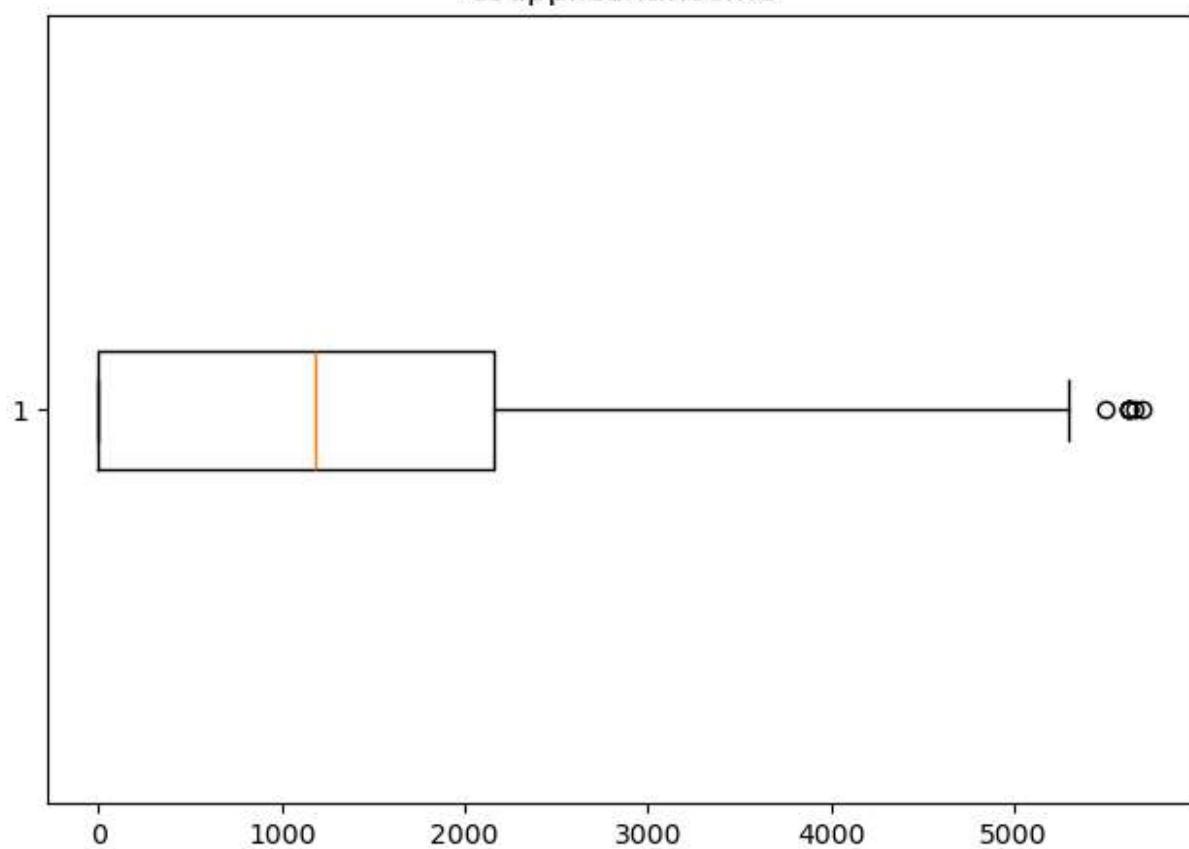
Dependents



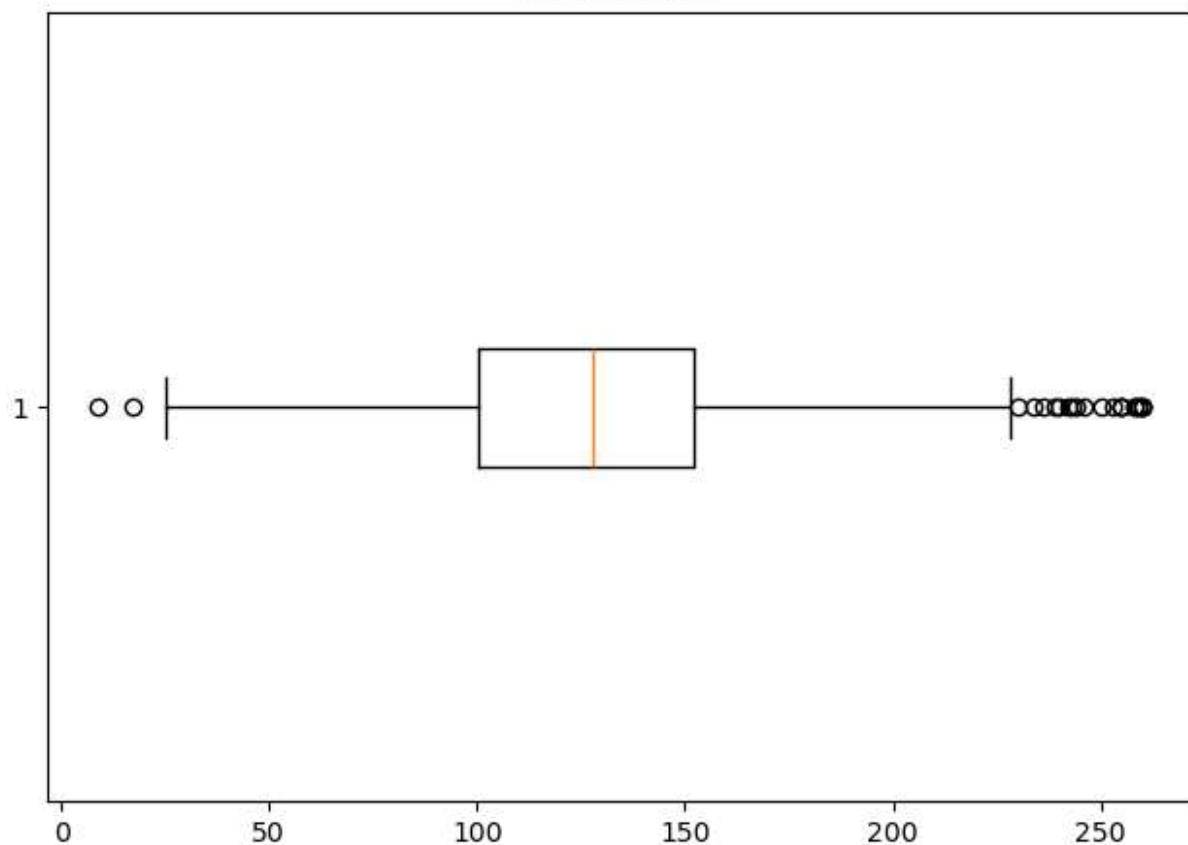
ApplicantIncome



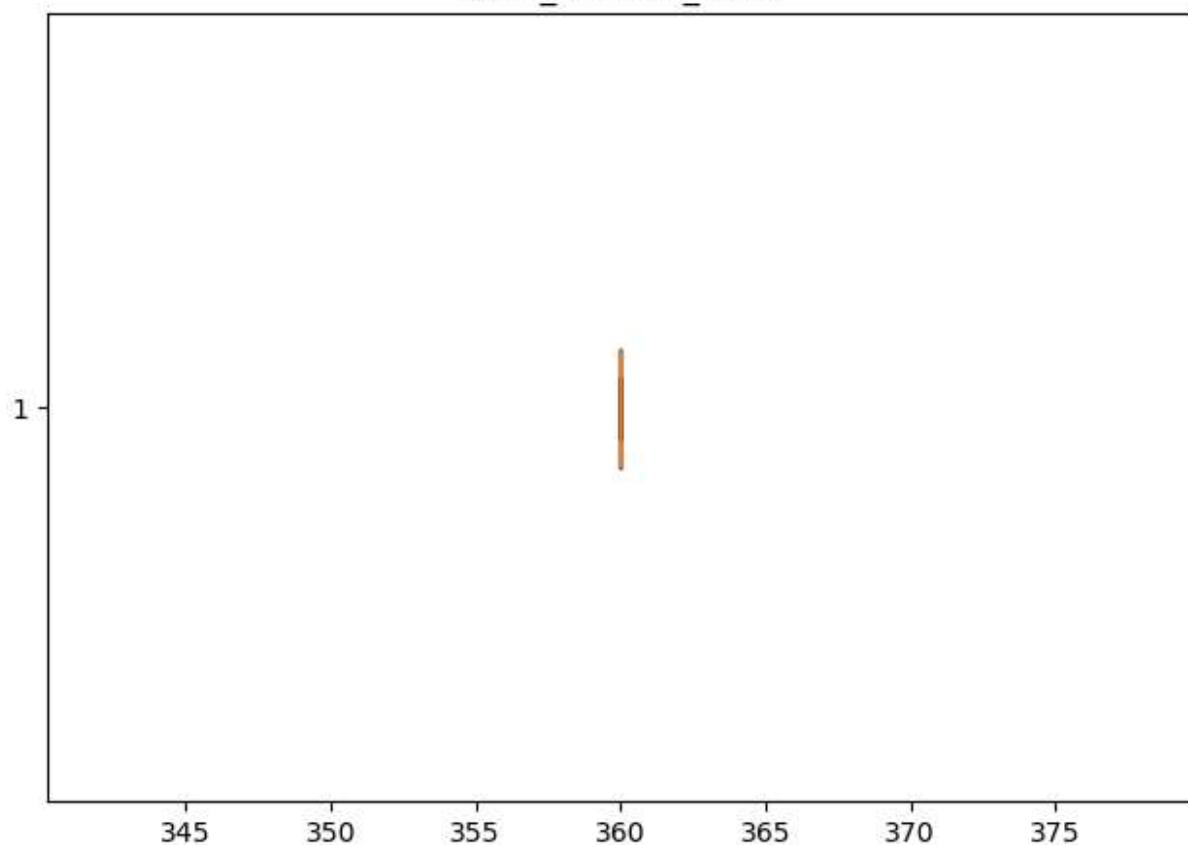
CoapplicantIncome



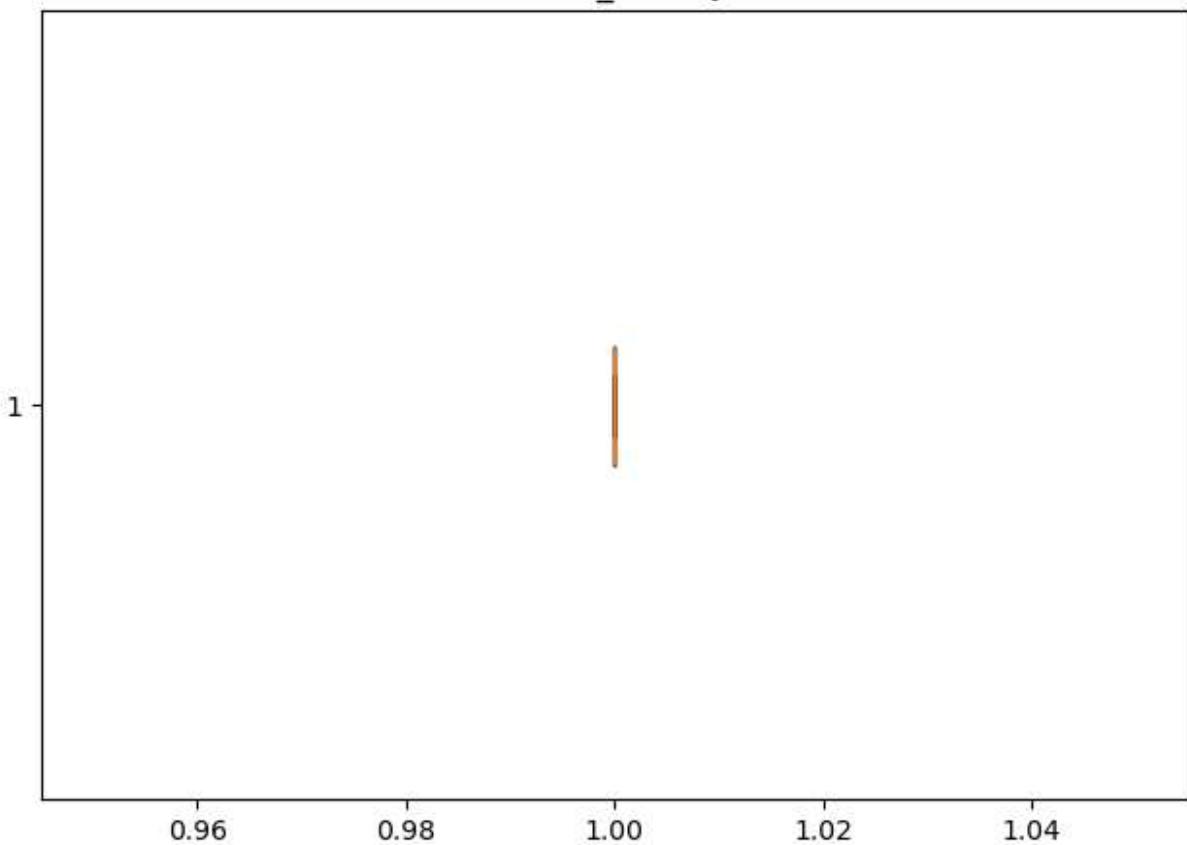
LoanAmount



Loan_Amount_Term



Credit_History



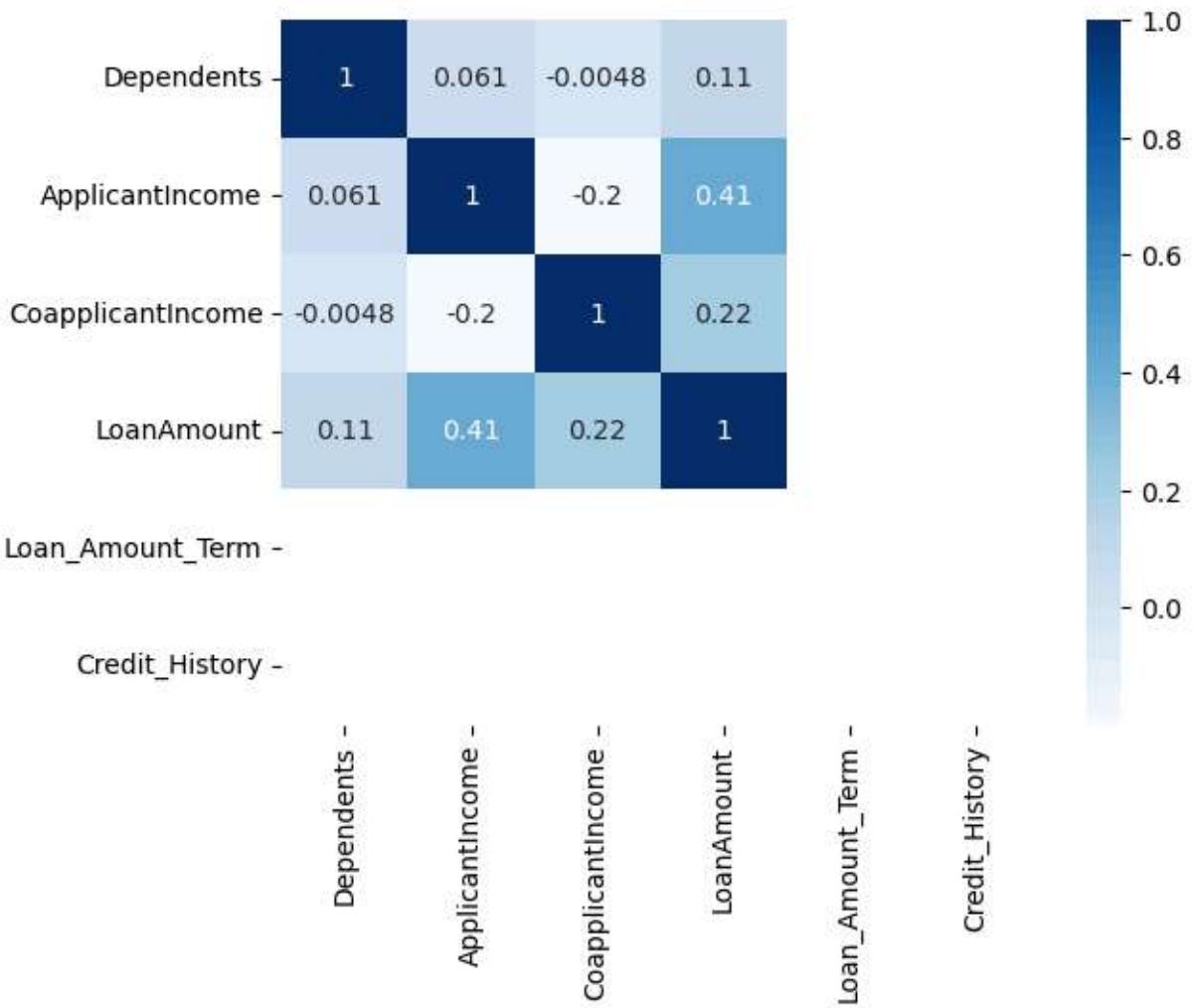
```
In [89]: loan_df.corr(numeric_only=True)
```

Out[89]:

	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
Dependents	1.000000	0.060577	-0.004811	0.111551		
ApplicantIncome	0.060577	1.000000	-0.198660	0.409572		
CoapplicantIncome	-0.004811	-0.198660	1.000000	0.220926		
LoanAmount	0.111551	0.409572	0.220926	1.000000		
Loan_Amount_Term		NaN	NaN	NaN	NaN	
Credit_History		NaN	NaN	NaN	NaN	



```
In [90]: cor=loan_df.corr(numeric_only=True)
sns.heatmap(cor,annot=True,cmap='Blues')
plt.show()
```



In [91]: `cat`

Out[91]: `Index(['Gender', 'Married', 'Education', 'Self_Employed', 'Property_Area', 'Loan_Status'],
 dtype='object')`

In [92]: `from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for i in cat:
 loan_df[i]=le.fit_transform(loan_df[i])
loan_df_num_format=loan_df
loan_df_num_format`

Out[92]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
0	1	0	0.0	0	0	5849.0	
1	1	1	1.0	0	0	4583.0	
2	1	1	0.0	0	1	3000.0	
3	1	1	0.0	1	0	2583.0	
4	1	0	0.0	0	0	6000.0	
...
609	0	0	0.0	0	0	2900.0	
610	1	1	0.0	0	0	4106.0	
611	1	1	1.0	0	0	8072.0	
612	1	1	2.0	0	0	7583.0	
613	0	0	0.0	0	1	4583.0	

614 rows × 12 columns



In [93]:

```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
for i in loan_df_num_format:
    loan_df_num_format[i]=ss.fit_transform(loan_df_num_format[[i]])
loan_df_Zstan=loan_df_num_format
loan_df_Zstan
```

Out[93]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	0.472343	-1.372089	-0.650697	-0.528362	-0.392601	0.947218	
1	0.472343	0.728816	0.663539	-0.528362	-0.392601	0.261857	
2	0.472343	0.728816	-0.650697	-0.528362	2.547117	-0.595116	
3	0.472343	0.728816	-0.650697	1.892641	-0.392601	-0.820863	
4	0.472343	-1.372089	-0.650697	-0.528362	-0.392601	1.028964	
...
609	-2.117107	-1.372089	-0.650697	-0.528362	-0.392601	-0.649252	
610	0.472343	0.728816	-0.650697	-0.528362	-0.392601	0.003628	
611	0.472343	0.728816	0.663539	-0.528362	-0.392601	2.150661	
612	0.472343	0.728816	1.977775	-0.528362	-0.392601	1.885936	
613	-2.117107	-1.372089	-0.650697	-0.528362	2.547117	0.261857	

614 rows × 12 columns



In [94]:

```
from sklearn.preprocessing import MinMaxScaler
mm=MinMaxScaler()
for i in loan_df_num_format:
    loan_df_num_format[i]=mm.fit_transform(loan_df_num_format[[i]])
loan_df_Minmax_Norm=loan_df_num_format
loan_df_Minmax_Norm
```

Out[94]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
0	1.0	0.0	0.0	0.0	0.0	0.570528	
1	1.0	1.0	0.5	0.0	0.0	0.443788	
2	1.0	1.0	0.0	0.0	1.0	0.285314	
3	1.0	1.0	0.0	1.0	0.0	0.243568	
4	1.0	0.0	0.0	0.0	0.0	0.585644	
...
609	0.0	0.0	0.0	0.0	0.0	0.275303	
610	1.0	1.0	0.0	0.0	0.0	0.396036	
611	1.0	1.0	0.5	0.0	0.0	0.793072	
612	1.0	1.0	1.0	0.0	0.0	0.744119	
613	0.0	0.0	0.0	0.0	1.0	0.443788	

614 rows × 12 columns



In [95]: loan_df

Out[95]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
0	1.0	0.0	0.0	0.0	0.0	0.570528	
1	1.0	1.0	0.5	0.0	0.0	0.443788	
2	1.0	1.0	0.0	0.0	1.0	0.285314	
3	1.0	1.0	0.0	1.0	0.0	0.243568	
4	1.0	0.0	0.0	0.0	0.0	0.585644	
...
609	0.0	0.0	0.0	0.0	0.0	0.275303	
610	1.0	1.0	0.0	0.0	0.0	0.396036	
611	1.0	1.0	0.5	0.0	0.0	0.793072	
612	1.0	1.0	1.0	0.0	0.0	0.744119	
613	0.0	0.0	0.0	0.0	1.0	0.443788	

614 rows × 12 columns



In []:

