In [3]:
```
# import the packages
# read the data

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

path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan=pd.read_csv(path)
loan
```

Out[3]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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

head-tail

In [5]:
```
loan.head(5)
```

Out[5]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantI
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	

In [6]:
```
loan.tail()
```

Out[6]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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	

In [8]:
```
loan.shape
```

Out[8]:
```
(614, 13)
```

In [9]:
```
print("The number of rows:",loan.shape[0])
print("The number of columns:",loan.shape[1])
```

```
The number of rows: 614
The number of columns: 13
```

In [10]:
```
loan.size
```

Out[10]:
```
7982
```

columns-dtypes

In [11]:
```
loan.columns
```

Out[11]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [12]:
```
loan.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
```

extract Numerical and Categorical sepearetly by using dtypes output

In [14]:
```
d1=dict(loan.dtypes)

cat=[i for i in d1 if d1[i]=='object']
num=[i for i in d1 if d1[i]!='object']
```

In [15]:
```
cat
```

Out[15]:
```
['Loan_ID',
 'Gender',
 'Married',
 'Dependents',
 'Education',
 'Self_Employed',
 'Property_Area',
 'Loan_Status']
```

In [16]:
```
loan.select_dtypes(include='object').columns
```

Out[16]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [17]:
```
loan.select_dtypes(exclude='object').columns
```

Out[17]:
```
Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History'],
      dtype='object')
```

isnull-dropduplicates-info

In [18]:
```
loan.isnull()
```

Out[18]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplicant
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 [19]:
```
loan.isnull().sum()
```

Out[19]:
```
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 [22]:
```
loan.drop_duplicates()
```

Out[22]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [23]:
```
loan.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
```

take-loc-iloc

In [25]:
```
loan.take((5,8,11))
```

Out[25]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplicant
5	LP001011	Male	Yes	2	Graduate	Yes	5417	
8	LP001018	Male	Yes	2	Graduate	No	4006	
11	LP001027	Male	Yes	2	Graduate	NaN	2500	

In [27]:
```
loan.take([5,8,12],axis=1)
```

Out[27]:

	Self_Employed	LoanAmount	Loan_Status
0	No	NaN	Y
1	No	128.0	N
2	Yes	66.0	Y
3	No	120.0	Y
4	No	141.0	Y
...	...	...	...
609	No	71.0	Y
610	No	40.0	Y
611	No	253.0	Y
612	No	187.0	Y
613	Yes	133.0	N

614 rows × 3 columns

In [28]:
```
loan.take([150,300,500])
```

Out[28]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
150	LP001528	Male	No	0	Graduate	No	6277	
300	LP001964	Male	Yes	0	Not Graduate	No	1800	
500	LP002603	Female	No	0	Graduate	No	645	

In [30]:
```
loan.iloc[1:8]
```

Out[30]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantI
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	
5	LP001011	Male	Yes	2	Graduate	Yes	5417	
6	LP001013	Male	Yes	0	Not Graduate	No	2333	
7	LP001014	Male	Yes	3+	Graduate	No	3036	

In [31]:
```
loan.iloc[11:20,5:9]
```

Out[31]:

	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount
11	NaN	2500	1840.0	109.0
12	No	3073	8106.0	200.0
13	No	1853	2840.0	114.0
14	No	1299	1086.0	17.0
15	No	4950	0.0	125.0
16	No	3596	0.0	100.0
17	No	3510	0.0	76.0
18	No	4887	0.0	133.0
19	NaN	2600	3500.0	115.0

In [32]:
```
loan.iloc[[20,30,40],[3,6]]
```

Out[32]:

	Dependents	ApplicantIncome
20	0	7660
30	1	4166
40	0	3600

In [35]:
```
loan.loc[[1,2,3,4,5,6,7,8,9,10],['ApplicantIncome']]
```

Out[35]:

	ApplicantIncome
1	4583
2	3000
3	2583
4	6000
5	5417
6	2333
7	3036
8	4006
9	12841
10	3200

In [37]:
```
loan.loc[:,['ApplicantIncome','Loan_Amount_Term']]
```

Out[37]:

	ApplicantIncome	Loan_Amount_Term
0	5849	360.0
1	4583	360.0
2	3000	360.0
3	2583	360.0
4	6000	360.0
...	...	...
609	2900	360.0
610	4106	180.0
611	8072	360.0
612	7583	360.0
613	4583	360.0

614 rows × 2 columns

In [38]:
```
loan.loc[[10,20,30],:]
```

Out[38]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplicant
10	LP001024	Male	Yes	2	Graduate	No	3200	
20	LP001043	Male	Yes	0	Not Graduate	No	7660	
30	LP001091	Male	Yes	1	Graduate	NaN	4166	

CategoricalAnalysis

In [39]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)
loan_df
```

Out[39]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [41]:
```
loan_df[['Education','Loan_Amount_Term']]
```

Out[41]:

	Education	Loan_Amount_Term
0	Graduate	360.0
1	Graduate	360.0
2	Graduate	360.0
3	Not Graduate	360.0
4	Graduate	360.0
...	...	...
609	Graduate	360.0
610	Graduate	180.0
611	Graduate	360.0
612	Graduate	360.0
613	Graduate	360.0

614 rows × 2 columns

In [42]:
```
cols=['Education','Loan_Amount_Term','ApplicantIncome']
loan_df[cols]
```

Out[42]:

	Education	Loan_Amount_Term	ApplicantIncome
0	Graduate	360.0	5849
1	Graduate	360.0	4583
2	Graduate	360.0	3000
3	Not Graduate	360.0	2583
4	Graduate	360.0	6000
...	...	...	...
609	Graduate	360.0	2900
610	Graduate	180.0	4106
611	Graduate	360.0	8072
612	Graduate	360.0	7583
613	Graduate	360.0	4583

614 rows × 3 columns

In [43]:
```
loan_df.values
```

Out[43]:
```
array([['LP001002', 'Male', 'No', ..., 1.0, 'Urban', 'Y'],
       ['LP001003', 'Male', 'Yes', ..., 1.0, 'Rural', 'N'],
       ['LP001005', 'Male', 'Yes', ..., 1.0, 'Urban', 'Y'],
       ...,
       ['LP002983', 'Male', 'Yes', ..., 1.0, 'Urban', 'Y'],
       ['LP002984', 'Male', 'Yes', ..., 1.0, 'Urban', 'Y'],
       ['LP002990', 'Female', 'No', ..., 0.0, 'Semiurban', 'N']],
      dtype=object)
```

In [44]:
```
loan_df['Education'].unique()
```

Out[44]:
```
array(['Graduate', 'Not Graduate'], dtype=object)
```

In [45]:
```
loan_df['Education'].nunique()
```

Out[45]:
```
2
```

In [47]:
```
con=loan_df['Education']=='Not Graduate'
loan_df[con]
```

Out[47]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
6	LP001013	Male	Yes	0	Not Graduate	No	2333	
16	LP001034	Male	No	1	Not Graduate	No	3596	
18	LP001038	Male	Yes	0	Not Graduate	No	4887	
20	LP001043	Male	Yes	0	Not Graduate	No	7660	
...	...	...	...	...	...	...	...	
595	LP002940	Male	No	0	Not Graduate	No	3833	
596	LP002941	Male	Yes	2	Not Graduate	Yes	6383	
601	LP002950	Male	Yes	0	Not Graduate	NaN	2894	
605	LP002960	Male	Yes	0	Not Graduate	No	2400	
607	LP002964	Male	Yes	2	Not Graduate	No	3987	

134 rows × 13 columns

In [49]:
```
con=loan_df['Education']=='Graduate'
len(loan_df[con])
```

Out[49]:
```
480
```

In [51]:
```
unique_labels=loan_df['Property_Area'].unique()
count=[]
for i in unique_labels:
    con=loan_df['Property_Area']==i
    count.append(len(loan_df[con]))

Property_Area_df=pd.DataFrame(zip(unique_labels,count),
                              columns=['Property_Area','count'])
```

In [52]:
```
Property_Area_df
```

Out[52]:

	Property_Area	count
0	Urban	202
1	Rural	179
2	Semiurban	233

In [54]:
```
Property_Area_vc=loan_df['Property_Area'].value_counts()
Property_Area_vc
```

Out[54]:
```
Property_Area
Semiurban    233
Urban        202
Rural        179
Name: count, dtype: int64
```

In [56]:
```
Property_Area_vc.keys()
```

Out[56]:
```
Index(['Semiurban', 'Urban', 'Rural'], dtype='object', name='Property_Area')
```

In [57]:
```
Property_Area_vc.values
```

Out[57]:
```
array([233, 202, 179], dtype=int64)
```

In [58]:
```
Property_Area_vc=loan_df['Property_Area'].value_counts()
l1=Property_Area_vc.keys()
l2=Property_Area_vc.values
Property_Area_vc_df=pd.DataFrame(zip(l1,l2),
                                 columns=['Property_Area','count'])
Property_Area_vc_df
```

Out[58]:

	Property_Area	count
0	Semiurban	233
1	Urban	202
2	Rural	179

Bar-chart

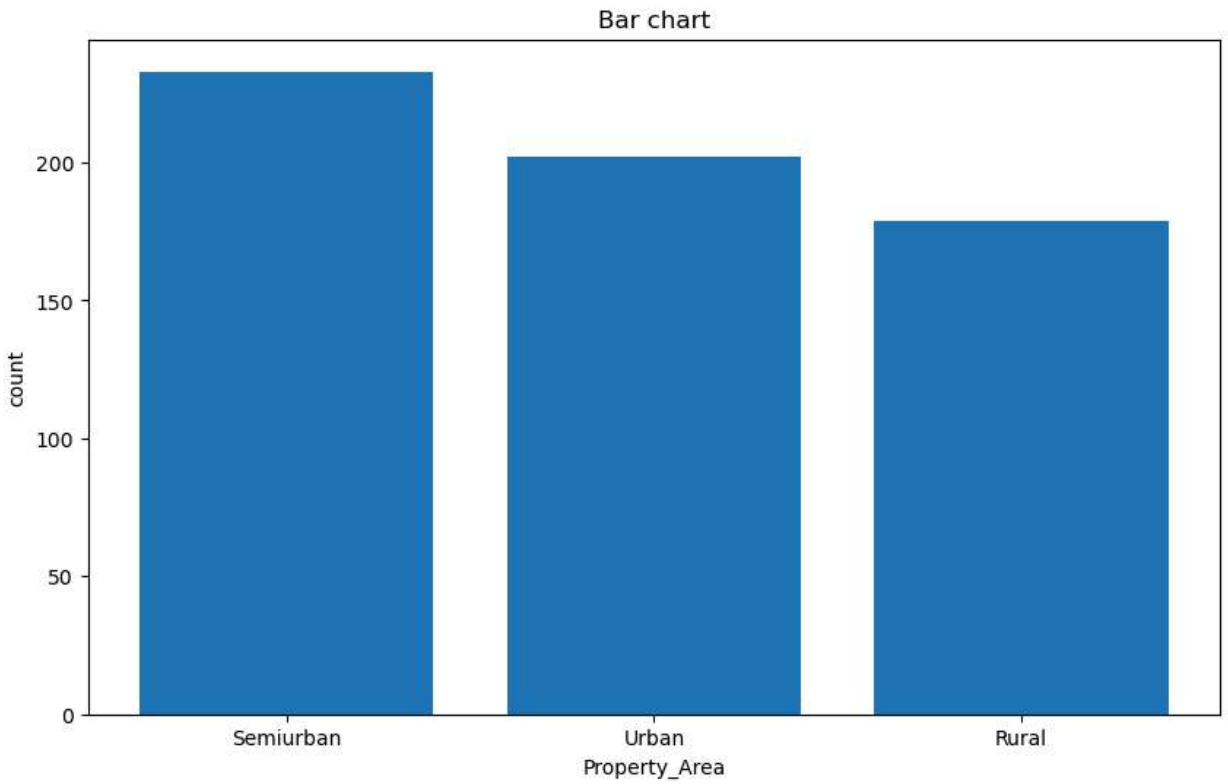
In [59]:
```
Property_Area_vc_df
```

Out[59]:

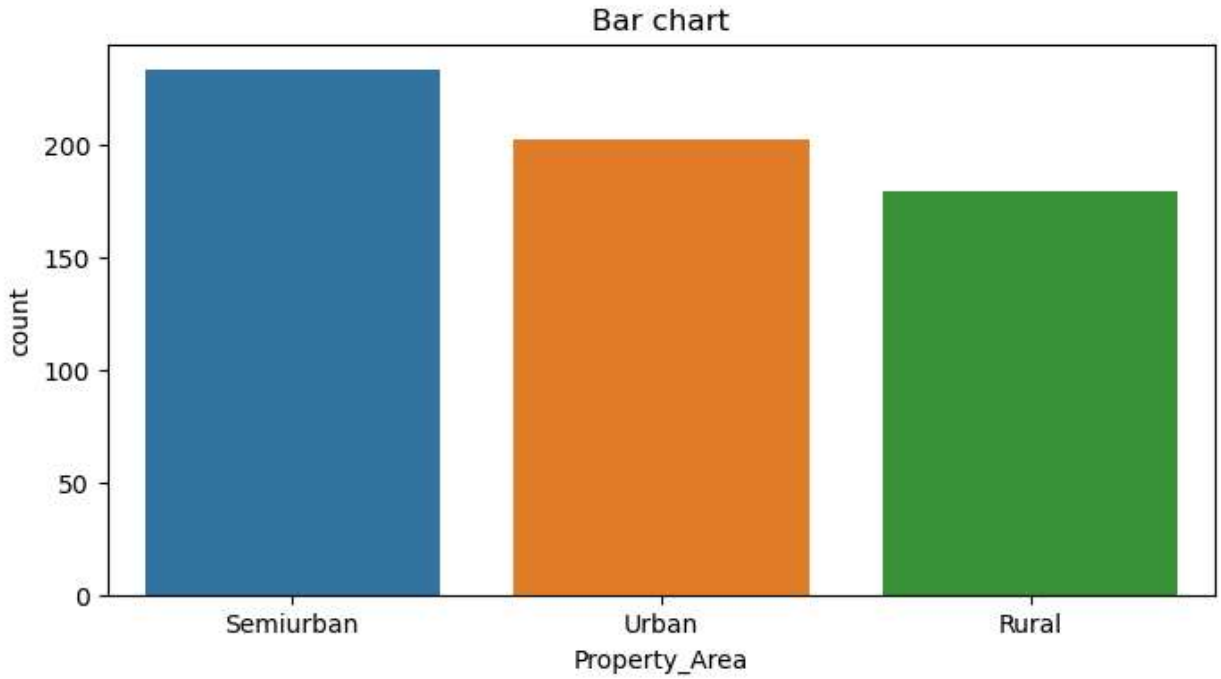
	Property_Area	count
0	Semiurban	233
1	Urban	202
2	Rural	179

In [61]:
```
plt.figure(figsize=(10,6)) # to incease the plot size
plt.bar('Property_Area',
        'count',
        data=Property_Area_vc_df)
plt.xlabel("Property_Area") # x-axis name
```

```
plt.ylabel('count') # y-axis name
plt.title("Bar chart") # title of the chart
plt.show()
```



In [67]:
```
plt.figure(figsize=(8,4))
l=loan_df['Property_Area'].value_counts().keys() # order provide automatically
sns.countplot(data=loan_df,
              x='Property_Area',
             order=l)
plt.xlabel("Property_Area") # x-axis name
plt.ylabel('count') # y-axis name
plt.title("Bar chart") # title of the chart
plt.show()
```

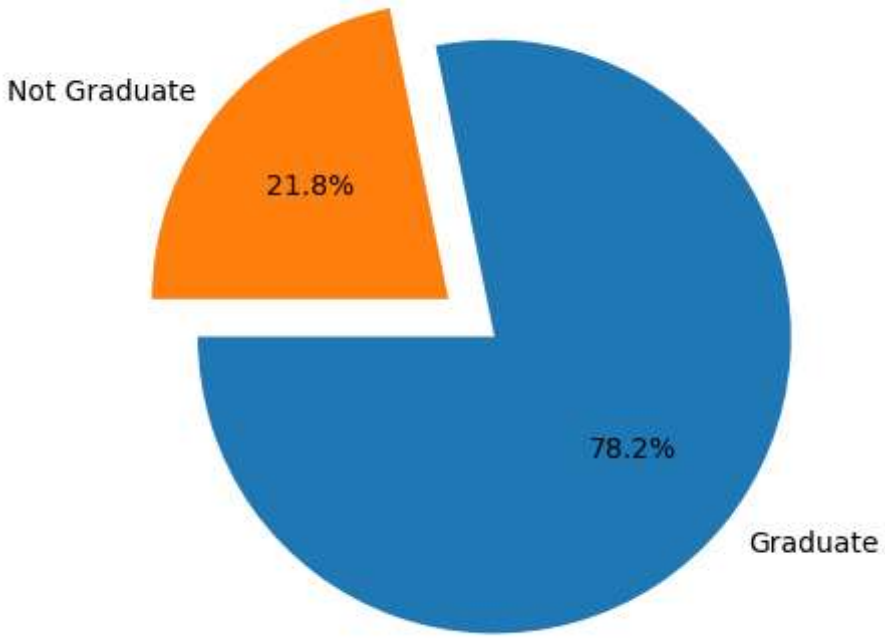


Pie-chart

In [69]:
```
keys=loan_df['Education'].value_counts().keys()
values=loan_df['Education'].value_counts().values
values
```

Out[69]:
```
array([480, 134], dtype=int64)
```

In [73]:
```
plt.pie(values,
        labels=keys,
       autopct="%0.1f%%",
       explode=[0.1,0.1],
       startangle=180,
       radius=1) # rotation
plt.show()
```



NumericalAnalysis

In [74]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)
loan_df
```

Out[74]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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

mean-median

In [77]:
```
loan_df['ApplicantIncome'].mean()
```

Out[77]:
```
5403.459283387622
```

In [78]:
```
loan_df['ApplicantIncome'].median()
```

Out[78]:
```
3812.5
```

max-min-std

In [79]:
```
loan_df['ApplicantIncome'].max()
```

Out[79]:
```
81000
```

In [80]:
```
loan_df['ApplicantIncome'].min()
```

Out[80]:
```
150
```

In [81]:
```
loan_df['ApplicantIncome'].std()
```

Out[81]:
```
6109.041673387174
```

In [83]:
```
loan_df.columns
```

Out[83]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [84]:
```
loan_df.select_dtypes(exclude='object').columns
```

Out[84]:
```
Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History'],
      dtype='object')
```

In [86]:
```
wage_count=round(loan_df['ApplicantIncome'].count(),2)
wage_max=round(loan_df['ApplicantIncome'].max(),2)
wage_min=round(loan_df['ApplicantIncome'].min(),2)
wage_mean=round(loan_df['ApplicantIncome'].mean(),2)
wage_median=round(loan_df['ApplicantIncome'].median(),2)
wage_std=round(loan_df['ApplicantIncome'].std(),2)

l=[wage_count,wage_max,wage_min,wage_mean,wage_median,wage_std]
cols=['ApplicantIncome']
index=['count','max','min','mean','median','std']
pd.DataFrame(l,columns=cols,index=index)
```

Out[86]:

	ApplicantIncome
count	614.00
max	81000.00
min	150.00
mean	5403.46
median	3812.50
std	6109.04

percentile-quantile

In [87]:
```
np.percentile(loan_df['ApplicantIncome'],25)
```

Out[87]:
```
2877.5
```

In [88]:
```
np.quantile(loan_df['ApplicantIncome'],0.25)
```

Out[88]:
```
2877.5
```

In [90]:
```
wage_50=np.percentile(loan_df['ApplicantIncome'],50)
con=loan_df['ApplicantIncome']<wage_50
len(loan_df[con])
```

Out[90]:
```
307
```

In [91]:
```
wage_count=round(loan_df['ApplicantIncome'].count(),2)
wage_max=round(loan_df['ApplicantIncome'].max(),2)
wage_min=round(loan_df['ApplicantIncome'].min(),2)
wage_mean=round(loan_df['ApplicantIncome'].mean(),2)
wage_median=round(loan_df['ApplicantIncome'].median(),2)
wage_std=round(loan_df['ApplicantIncome'].std(),2)
wage_25=np.percentile(loan_df['ApplicantIncome'],25)
wage_50=np.percentile(loan_df['ApplicantIncome'],50)
wage_75=np.percentile(loan_df['ApplicantIncome'],75)
```

```
l=[wage_count,wage_max,wage_min,
   wage_mean,wage_median,wage_std,
  wage_25,wage_50,wage_75]
cols=['ApplicantIncome']
index=['count','max','min',
       'mean','median','std',
      '25%','50%','75%']
pd.DataFrame(l,columns=cols,index=index)
```

Out[91]:

	ApplicantIncome
count	614.00
max	81000.00
min	150.00
mean	5403.46
median	3812.50
std	6109.04
25%	2877.50
50%	3812.50
75%	5795.00

In [92]:
```
loan_df.describe()
```

Out[92]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

In [95]:
```
cols=loan_df.select_dtypes(exclude='object').columns
l=[]
for i in cols:
    count=round(loan_df[i].count(),2)
    maxx=round(loan_df[i].max(),2)
    minn=round(loan_df[i].min(),2)
    mean=round(loan_df[i].mean(),2)
    median=round(loan_df[i].median(),2)
    std=round(loan_df[i].std(),2)
    p_25=np.percentile(loan_df[i],25)
    p_50=np.percentile(loan_df[i],50)
    p_75=np.percentile(loan_df[i],75)
```

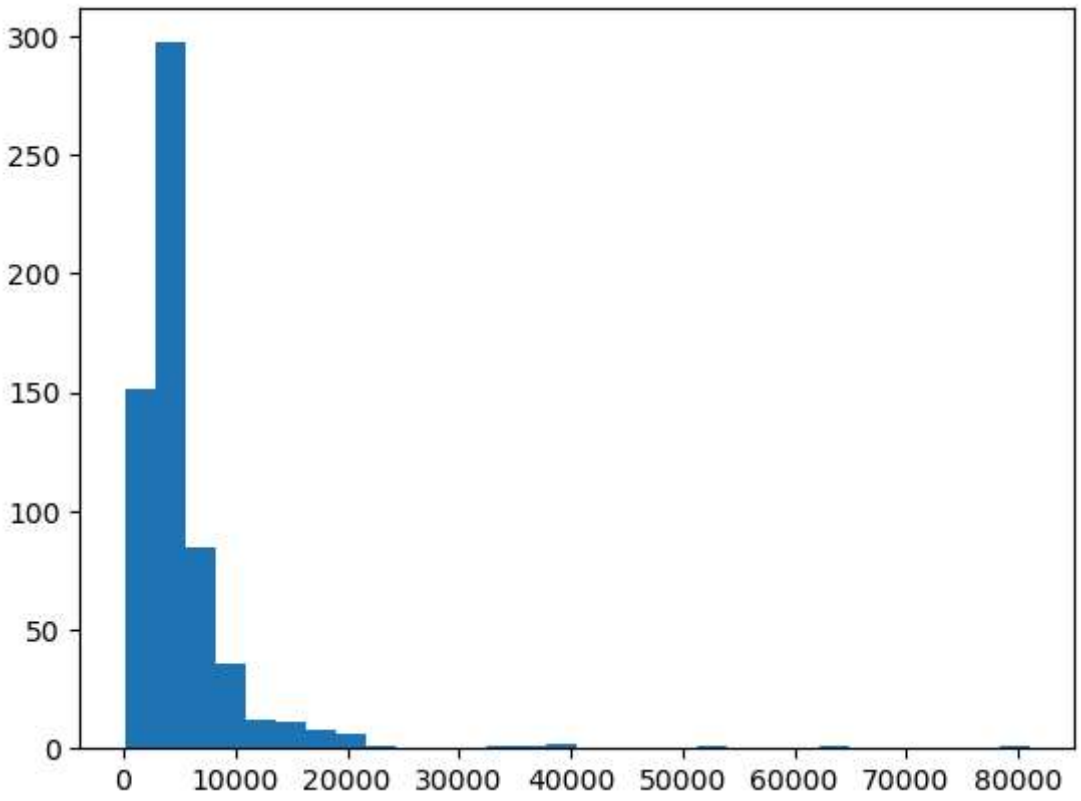
```
    l.append([count,maxx,minn,mean,median,std,
              p_25,p_50,p_75])

index=['count','max','min',
       'mean','median','std',
      '25%','50%','75%']
pd.DataFrame(zip(l[0],l[1],l[2],l[3],l[4]),columns=cols,index=index)
```

Out[95]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.00	614.00	592.00	600.00	564.00
max	81000.00	41667.00	700.00	480.00	1.00
min	150.00	0.00	9.00	12.00	0.00
mean	5403.46	1621.25	146.41	342.00	0.84
median	3812.50	1188.50	128.00	360.00	1.00
std	6109.04	2926.25	85.59	65.12	0.36
25%	2877.50	0.00	NaN	NaN	NaN
50%	3812.50	1188.50	NaN	NaN	NaN
75%	5795.00	2297.25	NaN	NaN	NaN

histogram

In [99]:
```
f,i,n=plt.hist(loan_df['ApplicantIncome'],
               bins=30)
```



In [100…
```
len(f),len(i),len(n)
```

Out[100]:
```
(30, 31, 30)
```

In [101…
```
i
```

Out[101]:
```
array([  150.,  2845.,  5540.,  8235., 10930., 13625., 16320., 19015.,
       21710., 24405., 27100., 29795., 32490., 35185., 37880., 40575.,
       43270., 45965., 48660., 51355., 54050., 56745., 59440., 62135.,
       64830., 67525., 70220., 72915., 75610., 78305., 81000.])
```

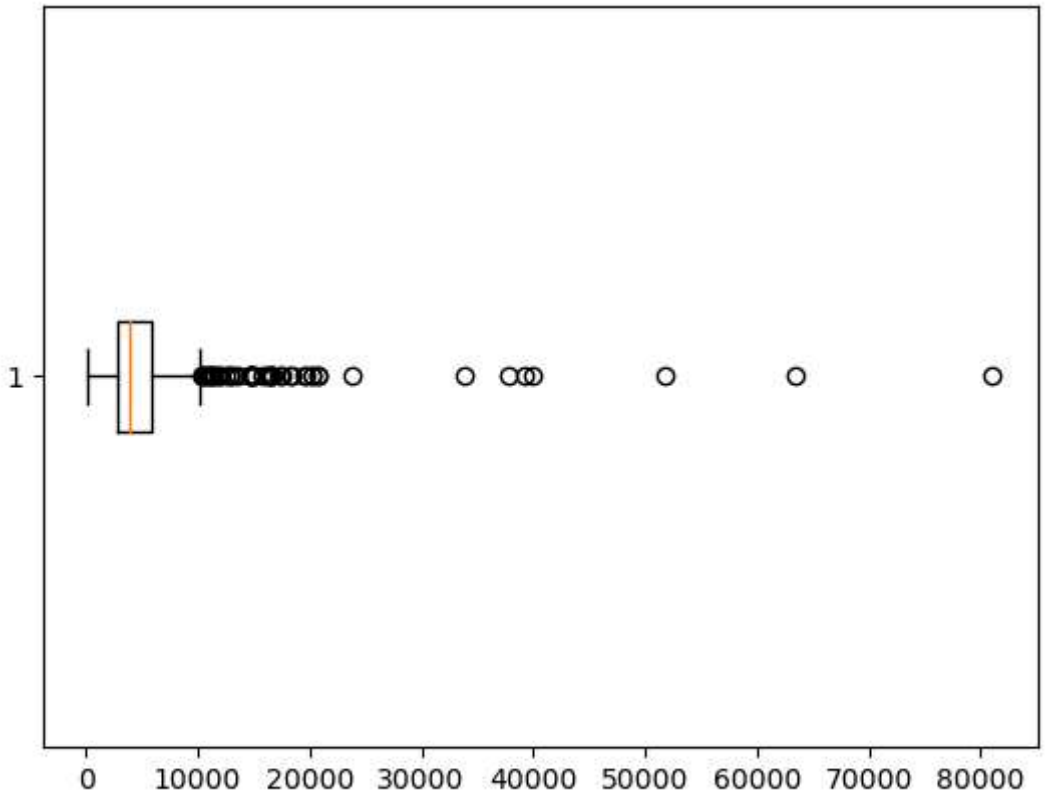
In [102…
```
def frequency(l,u):
    c1=loan_df['ApplicantIncome']>=1
    c2=loan_df['ApplicantIncome']<u
    c=c1&c2
    print(len(loan_df[c]))
frequency(150,2845)
```

```
151
```

In [ ]:

Boxplot

In [104…
```
plt.boxplot(loan_df['ApplicantIncome'],
            vert=False)
plt.show()
```



In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
```

```
loan_df=pd.read_csv(path)
loan_df
```

Out[2]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [3]:

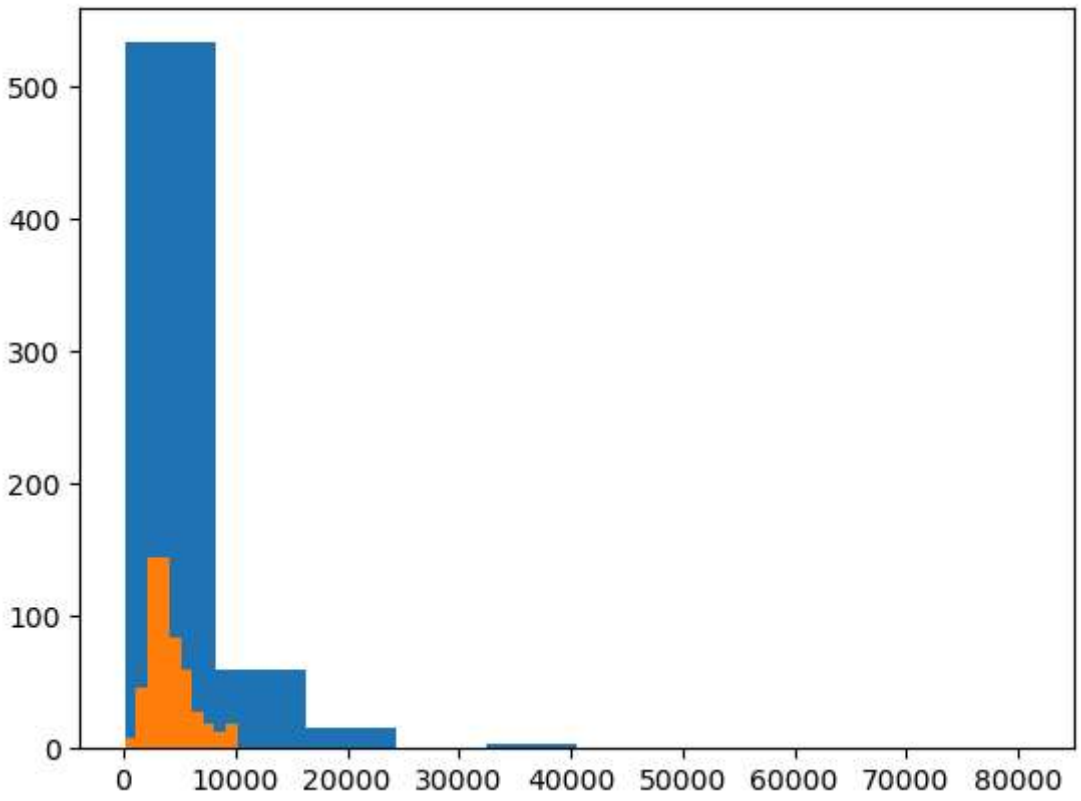
```
Q1=np.percentile(loan_df['ApplicantIncome'],25)
Q2=np.percentile(loan_df['ApplicantIncome'],50)
Q3=np.percentile(loan_df['ApplicantIncome'],75)

IQR=Q3-Q1

lb=Q1-1*IQR
ub=Q3+1.5*IQR

c1=loan_df['ApplicantIncome']<lb
c2=loan_df['ApplicantIncome']>ub
con=c1|c2

outliers_df=loan_df[con]
outliers_df

c1=loan_df['ApplicantIncome']>lb
c2=loan_df['ApplicantIncome']<ub
con=c1&c2
non_outliers_df=loan_df[c1&c2]
non_outliers_df
```

Out[3]:

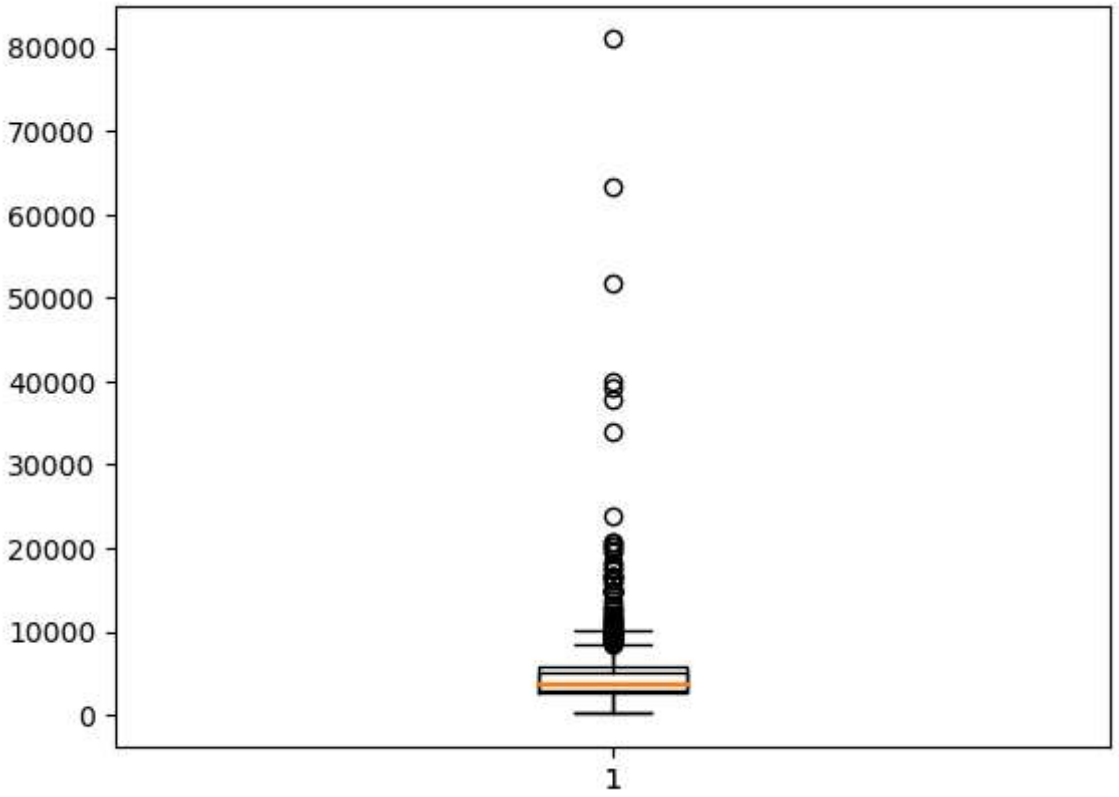
	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
...	...	...	...	...	...	...	...	
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	

564 rows × 13 columns

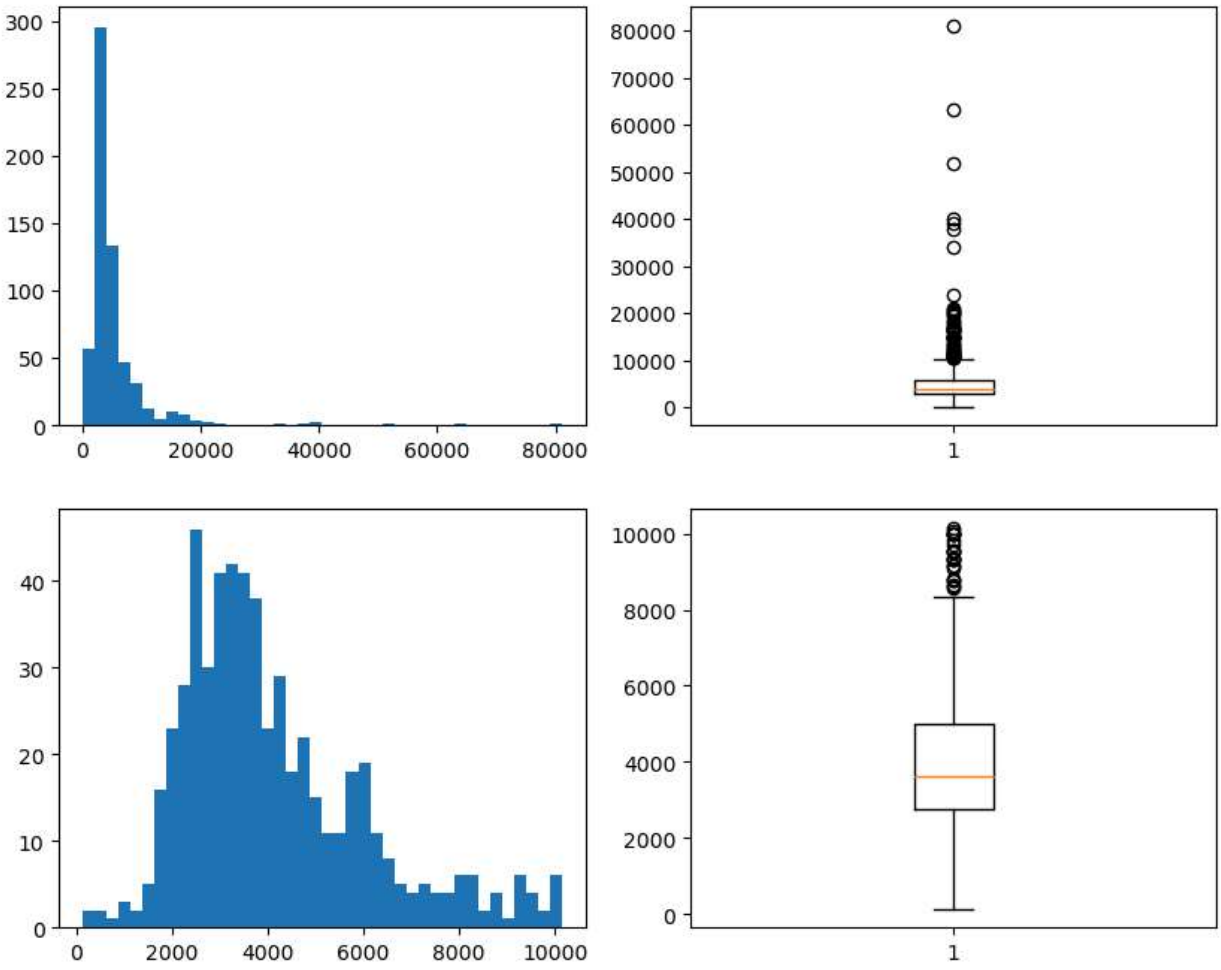
In [4]:
```
plt.hist(loan_df['ApplicantIncome'])
plt.hist(non_outliers_df['ApplicantIncome'])
plt.show()
```



In [5]:
```
plt.boxplot(loan_df['ApplicantIncome'])
plt.boxplot(non_outliers_df['ApplicantIncome'])
plt.show()
```



In [6]:
```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1).hist(loan_df['ApplicantIncome'],bins=40)
plt.subplot(2,2,2).boxplot(loan_df['ApplicantIncome'])
plt.subplot(2,2,3).hist(non_outliers_df['ApplicantIncome'],bins=40)
plt.subplot(2,2,4).boxplot(non_outliers_df['ApplicantIncome'])
plt.show()
```



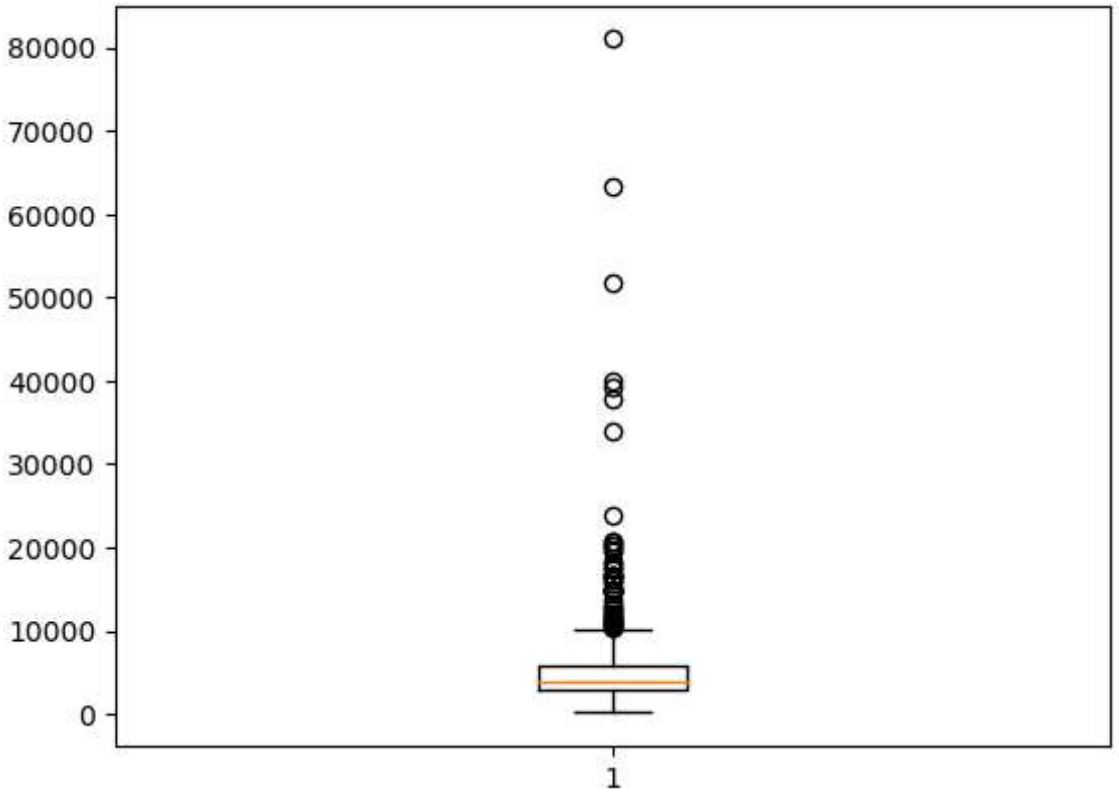
In [7]:
```
median=loan_df['ApplicantIncome'].median()

list1=[]
for i in loan_df['ApplicantIncome']:
    if i<lb or i>ub:
        list1.append(median)
    else:
        list1.append(i)

len(list1)
```

Out[7]: 614

In [9]:
```
Q1=np.percentile(loan_df['ApplicantIncome'],25)
Q2=np.percentile(loan_df['ApplicantIncome'],50)
Q3=np.percentile(loan_df['ApplicantIncome'],75)


IQR=Q3-Q1

lb=Q1-1*IQR
ub=Q3+1.5*IQR

median=loan_df['ApplicantIncome'].median()

list1=[]
for i in loan_df['ApplicantIncome']:
    if i<lb or i>ub:
```

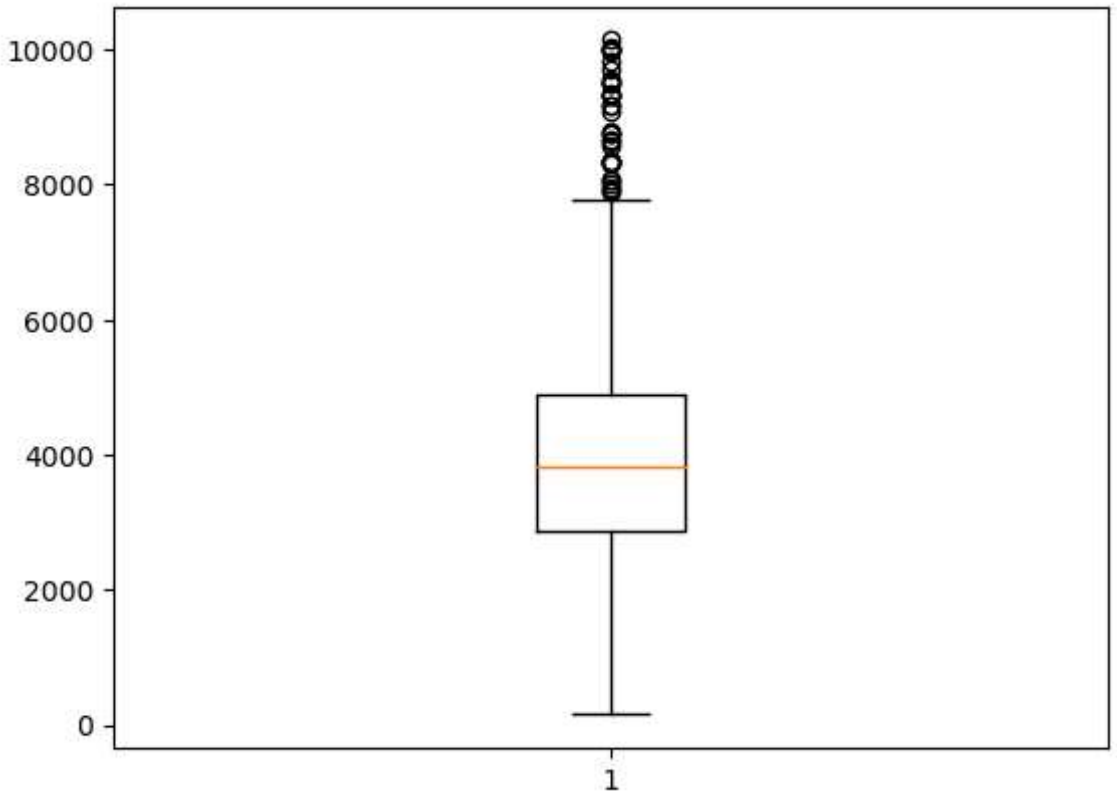
```
        list1.append(median)
    else:
        list1.append(i)

loan_df['ApplicantIncome_new']=list1
```

In [10]:
```
plt.boxplot(loan_df['ApplicantIncome'])
plt.show()
```



*np.where

In [11]:
```
c1=loan_df['ApplicantIncome']<lb
c2=loan_df['ApplicantIncome']>ub
loan_df['ApplicantIncome']=np.where(c1|c2,median,loan_df['ApplicantIncome'])
plt.boxplot(loan_df['ApplicantIncome'])
plt.show()
```



Categorical-Categorical

In [18]:
```
c1=loan_df['Property_Area']=='Rural'
c2=loan_df['Education']=='Graduate'
c3=loan_df['Education']=='NotGraduate'

cert_con=c1&c2
den_con=c1&c3

Graduate_count=len(loan_df[cert_con])
Not_Graduate_count=len(loan_df[den_con])
print(f"there are {Graduate_count} got Graduate from Rural")
print(f"there are {Not_Graduate_count} got NotGraduate from Rural")
```

```
there are 131 got Graduate from Rural
there are 0 got NotGraduate from Rural
```

In [20]:
```
labels=loan_df['Property_Area'].unique()
Graduate_count=[]
Not_Graduate_count=[]
for i in labels:
    c1=loan_df['Property_Area']==i
    c2=loan_df['Education']=='Graduate'
    c3=loan_df['Education']=='NotGraduate'
    Edt_con=c1&c2
    Non_Edt_con=c1&c3
    Graduate_count.append(len(loan_df[Edt_con]))
    Not_Graduate_count.append(len(loan_df[Non_Edt_con]))

cols=['Property_Area','Graduate','NotGradute']
d1=pd.DataFrame(zip(labels,
                    Graduate_count,
```

```
                      Not_Graduate_count), columns=cols)
d1.set_index('Property_Area')
```

Out[20]:

Property_Area	Graduate	NotGradute
Urban	162	0
Rural	131	0
Semiurban	187	0

In [22]:
```
col1=[loan_df['Property_Area']]
col2=loan_df['Education']
result1=pd.crosstab(col1,col2)
result1
```

Out[22]:

Property_Area	Education	Graduate	Not Graduate
Rural		131	48
Semiurban		187	46
Urban		162	40

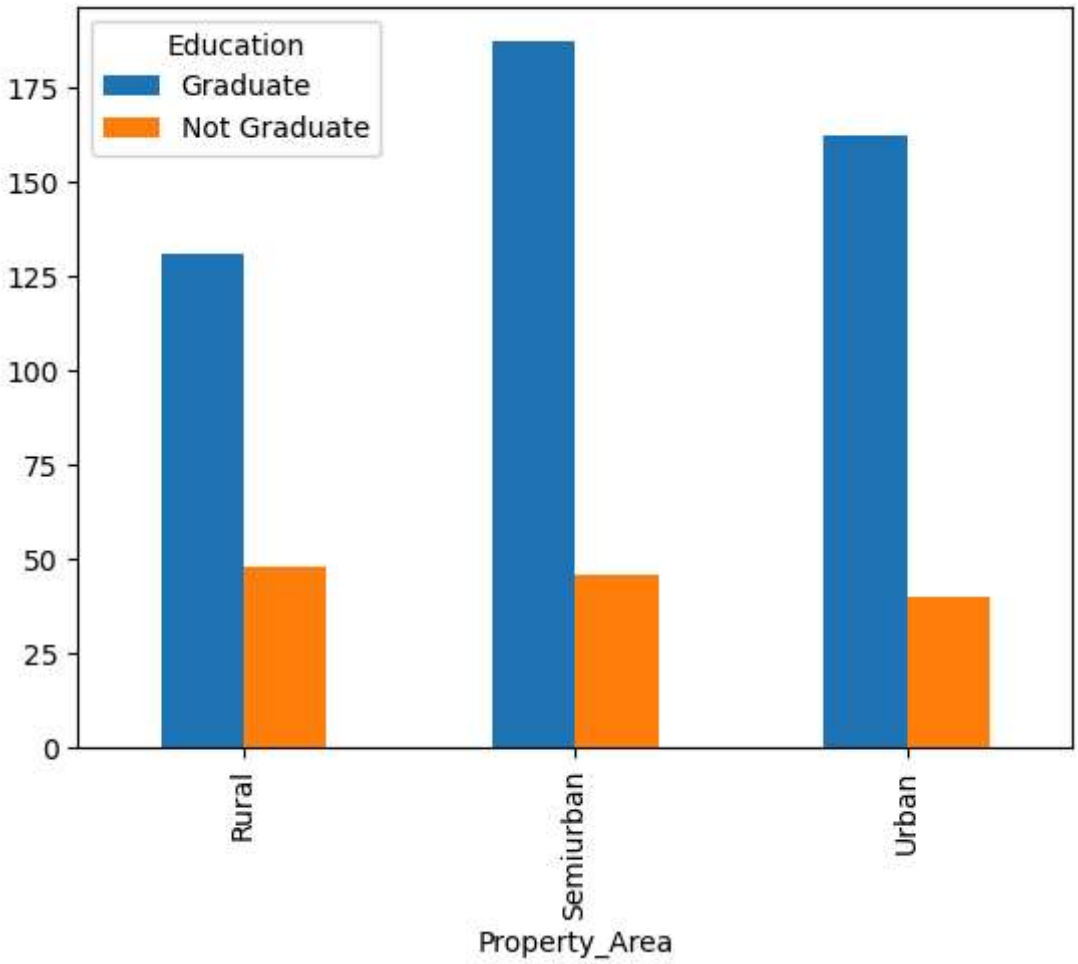
In [25]:
```
col1=[loan_df['Property_Area'],
     loan_df['Loan_Status']]
col2=loan_df['Education']
result2=pd.crosstab(col1,col2)
result2
```

Out[25]:

Property_Area	Loan_Status	Education	Graduate	Not Graduate
Rural	**N**		47	22
	Y		84	26
Semiurban	**N**		43	11
	Y		144	35
Urban	**N**		50	19
	Y		112	21

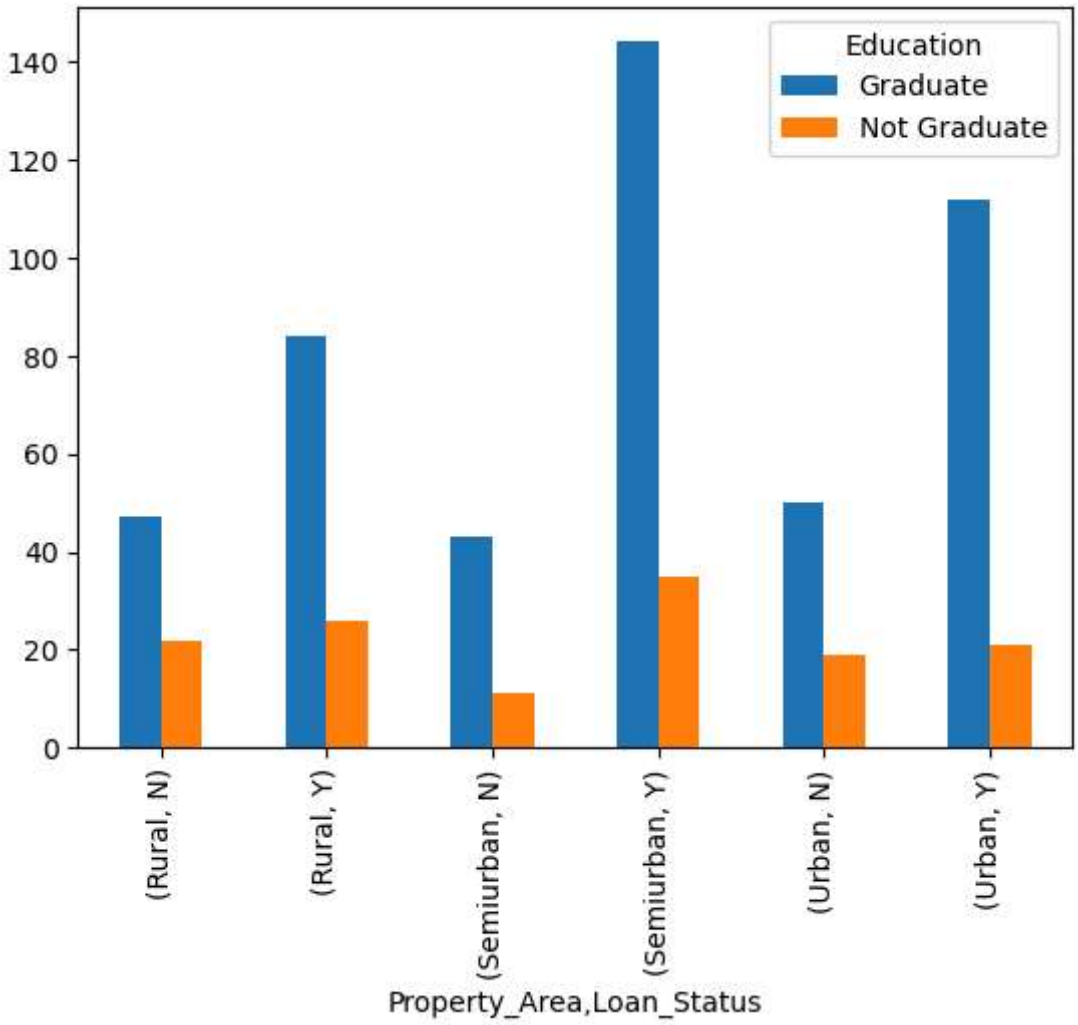
In [26]:
```
result1.plot(kind='bar')
```

Out[26]:
```
<Axes: xlabel='Property_Area'>
```



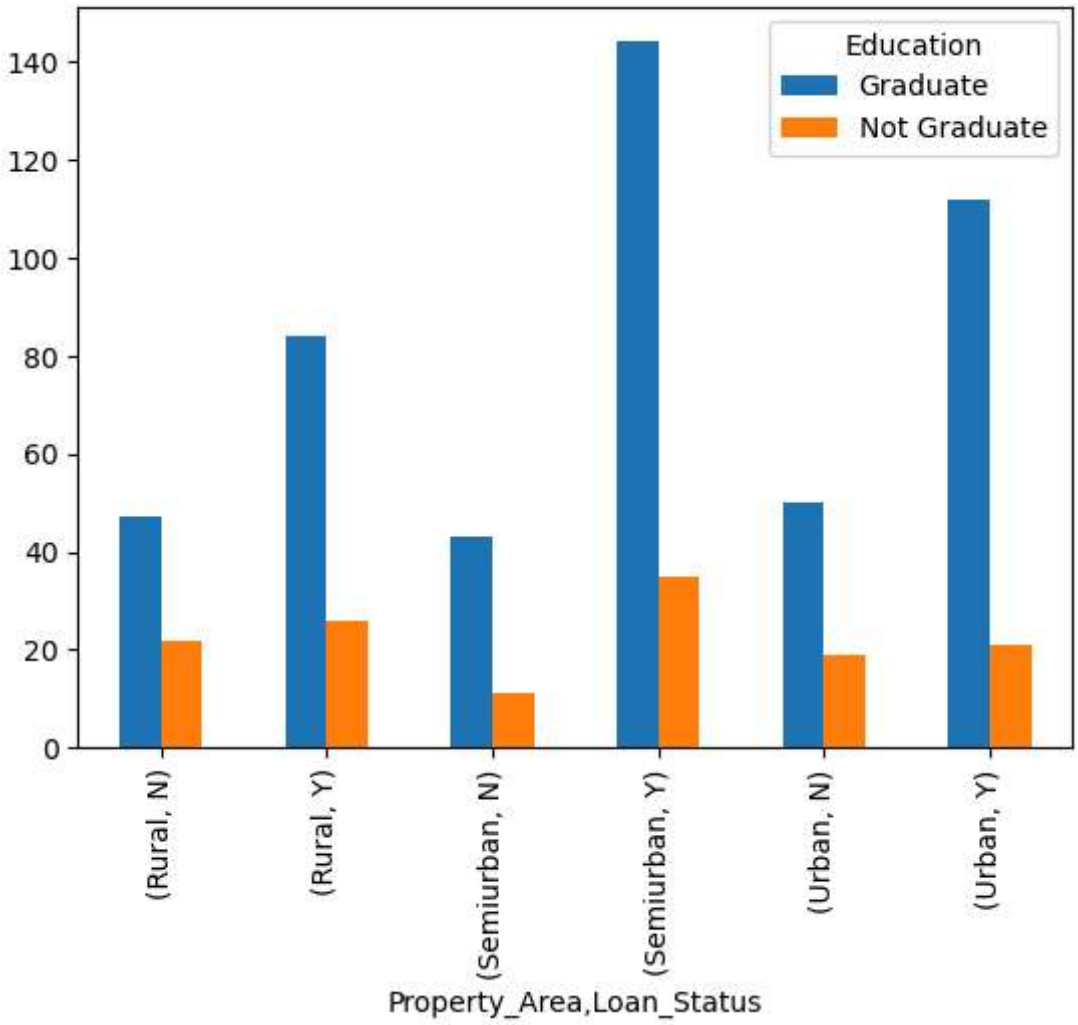
In [27]:
```
result2.plot(kind='bar')
```

Out[27]:
```
<Axes: xlabel='Property_Area,Loan_Status'>
```



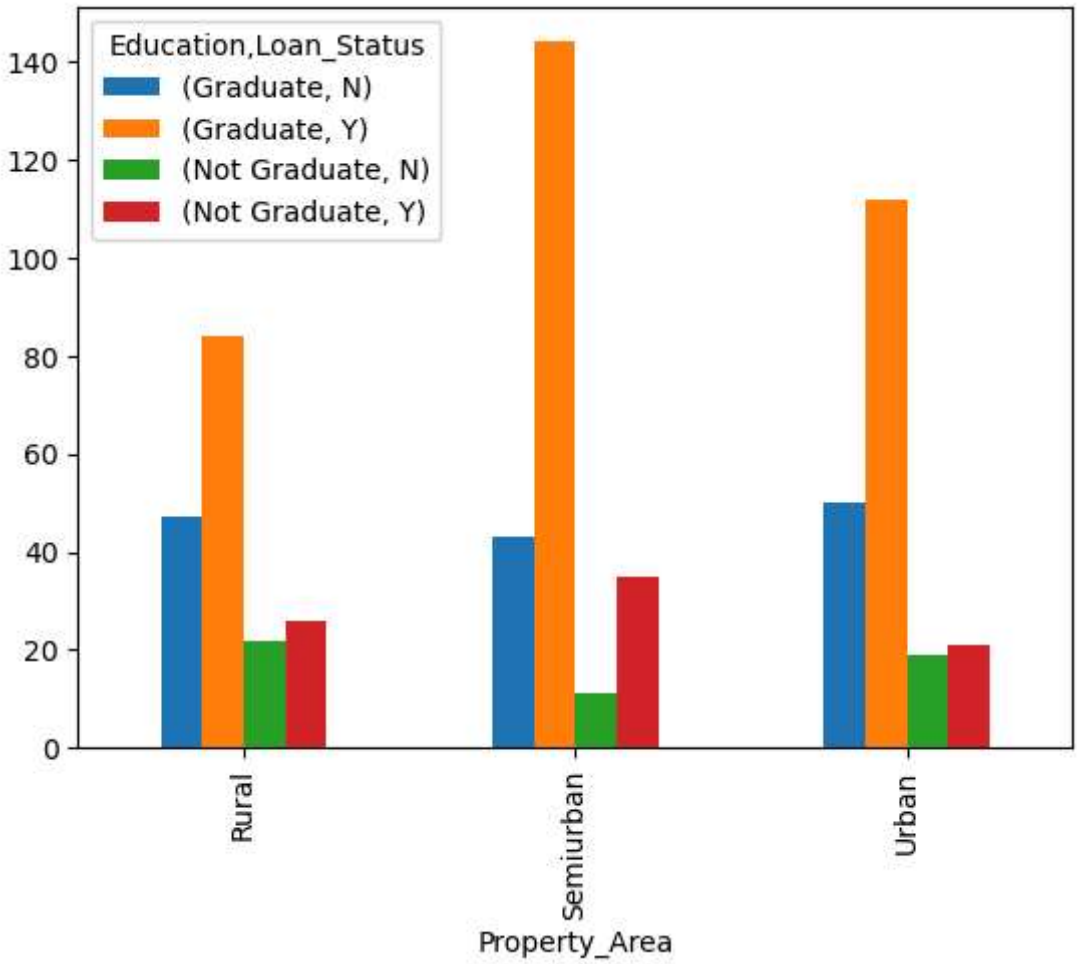
In [28]:
```
col1=[loan_df['Property_Area'],
     loan_df['Loan_Status']]
col2=loan_df['Education']
result2=pd.crosstab(col1,col2)
result2.plot(kind='bar')
```

Out[28]:
```
<Axes: xlabel='Property_Area,Loan_Status'>
```



In [31]:
```
col1=loan_df['Property_Area']
col2=loan_df['Education']
col3=loan_df['Loan_Status']
r1=pd.crosstab(col1,[col2,col3])
r1.plot(kind='bar')
```

Out[31]:
```
<Axes: xlabel='Property_Area'>
```

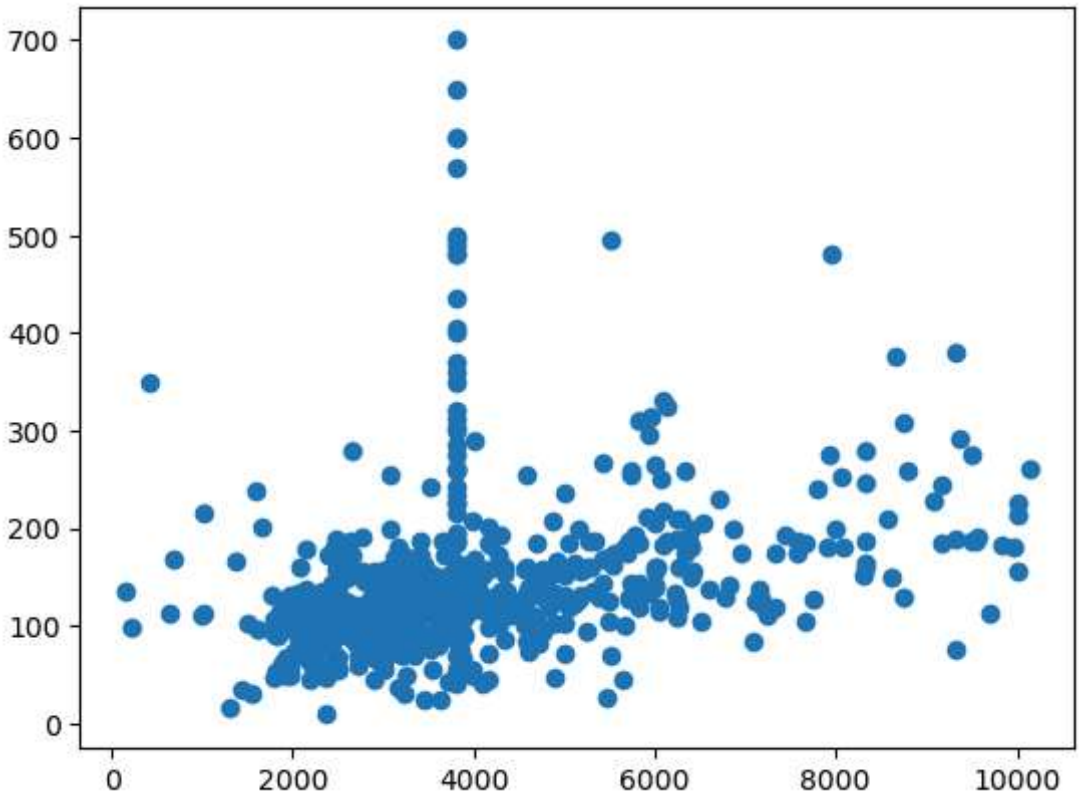


NumericalvsNumerical

In [34]:
```
num_cols=loan_df.select_dtypes(exclude='object')
num_cols.columns
```

Out[34]:
```
Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'ApplicantIncome_new'],
      dtype='object')
```

In [35]:
```
col1=loan_df['ApplicantIncome']
col2=loan_df['LoanAmount']
plt.scatter(col1,col2)
plt.show()
```



In [36]:
```
loan_df.corr(numeric_only=True)
```

Out[36]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_
ApplicantIncome	1.000000	-0.171521	0.301133	-0.036034	0
CoapplicantIncome	-0.171521	1.000000	0.188619	-0.059878	-0
LoanAmount	0.301133	0.188619	1.000000	0.039447	-0
Loan_Amount_Term	-0.036034	-0.059878	0.039447	1.000000	0
Credit_History	0.049876	-0.002056	-0.008433	0.001470	1
ApplicantIncome_new	1.000000	-0.171521	0.301133	-0.036034	0

In [38]:
```
corr_loan=loan_df.corr(numeric_only=True)
plt.figure(figsize=(10,8))
sns.heatmap(corr_loan,annot=True)
plt.show()
```

Convert Categorical to Numerical

In [40]:
```
path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)
d={'Y':0,'N':1}
loan_df['Loan_Status']=loan_df['Loan_Status'].map(d)
loan_df
```

Out[40]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [41]:

```
path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)
cat_cols=loan_df.select_dtypes(include='object').columns

d={}
for j in cat_cols[1:]:
    labels=loan_df[j].unique()
    for i in range(len(labels)):
        d[labels[i]]=i
    loan_df[j]=loan_df[j].map(d)

loan_df
```

Out[41]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
0	LP001002	0	0	0	0	0	5849	
1	LP001003	0	1	1	0	0	4583	
2	LP001005	0	1	0	0	1	3000	
3	LP001006	0	1	0	1	0	2583	
4	LP001008	0	0	0	0	0	6000	
...	...	...	...	...	...	...	...	
609	LP002978	1	0	0	0	0	2900	
610	LP002979	0	1	3	0	0	4106	
611	LP002983	0	1	1	0	0	8072	
612	LP002984	0	1	2	0	0	7583	
613	LP002990	1	0	0	0	1	4583	

614 rows × 13 columns

In [43]:
```
cat_cols=loan_df.select_dtypes(include='object').columns
cat_cols[1:]
```

Out[43]:
```
Index([], dtype='object')
```

LabelEncoder

In [45]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
loan_df['Loan_Status']=le.fit_transform(loan_df['Loan_Status'])
loan_df
```

Out[45]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
0	LP001002	0	0	0	0	0	5849	
1	LP001003	0	1	1	0	0	4583	
2	LP001005	0	1	0	0	1	3000	
3	LP001006	0	1	0	1	0	2583	
4	LP001008	0	0	0	0	0	6000	
...	...	...	...	...	...	...	...	
609	LP002978	1	0	0	0	0	2900	
610	LP002979	0	1	3	0	0	4106	
611	LP002983	0	1	1	0	0	8072	
612	LP002984	0	1	2	0	0	7583	
613	LP002990	1	0	0	0	1	4583	

614 rows × 13 columns

In [46]:
```
path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)
cat_cols=loan_df.select_dtypes(include='object').columns
cat_cols

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for i in cat_cols:
    loan_df[i]:le.fit_transform(loan_df[i])
```

In [47]:
```
loan_df
```

Out[47]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [48]:

```
path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)
cat_cols=loan_df.select_dtypes(include='object').columns
cat_cols

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
loan_df['Property_Area']=le.fit_transform(loan_df['Property_Area'])
loan_df
```

Out[48]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [49]:
```
le.inverse_transform(loan_df['Property_Area'])
```

Out[49]:
```
array(['Urban', 'Rural', 'Urban', 'Urban', 'Urban', 'Urban', 'Urban',
       'Semiurban', 'Urban', 'Semiurban', 'Urban', 'Urban', 'Urban',
       'Rural', 'Urban', 'Urban', 'Urban', 'Urban', 'Rural', 'Urban',
       'Urban', 'Urban', 'Semiurban', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Urban', 'Semiurban', 'Urban', 'Urban',
       'Rural', 'Semiurban', 'Rural', 'Urban', 'Urban', 'Semiurban',
       'Urban', 'Semiurban', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Urban', 'Urban', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Urban', 'Urban',
       'Semiurban', 'Semiurban', 'Rural', 'Urban', 'Urban', 'Urban',
       'Urban', 'Rural', 'Rural', 'Semiurban', 'Semiurban', 'Urban',
       'Urban', 'Urban', 'Semiurban', 'Urban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban',
       'Urban', 'Semiurban', 'Semiurban', 'Semiurban', 'Urban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban',
       'Semiurban', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Semiurban', 'Urban', 'Semiurban', 'Urban',
       'Urban', 'Urban', 'Rural', 'Urban', 'Semiurban', 'Urban',
       'Semiurban', 'Rural', 'Semiurban', 'Semiurban', 'Rural',
       'Semiurban', 'Urban', 'Rural', 'Urban', 'Rural', 'Semiurban',
       'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Rural', 'Rural',
       'Urban', 'Rural', 'Urban', 'Urban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Rural',
       'Rural', 'Urban', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban',
       'Urban', 'Urban', 'Rural', 'Semiurban', 'Rural', 'Rural', 'Urban',
       'Rural', 'Urban', 'Semiurban', 'Rural', 'Urban', 'Rural',
       'Semiurban', 'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Urban',
       'Rural', 'Rural', 'Rural', 'Semiurban', 'Semiurban', 'Rural',
       'Urban', 'Rural', 'Semiurban', 'Semiurban', 'Rural', 'Rural',
       'Semiurban', 'Semiurban', 'Urban', 'Urban', 'Rural', 'Semiurban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Rural',
       'Rural', 'Rural', 'Semiurban', 'Urban', 'Semiurban', 'Rural',
       'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Urban', 'Semiurban',
       'Semiurban', 'Urban', 'Urban', 'Semiurban', 'Semiurban', 'Urban',
       'Rural', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban', 'Urban',
       'Rural', 'Urban', 'Semiurban', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Semiurban', 'Rural',
       'Semiurban', 'Rural', 'Rural', 'Semiurban', 'Semiurban', 'Rural',
       'Urban', 'Urban', 'Rural', 'Semiurban', 'Rural', 'Urban', 'Urban',
       'Rural', 'Semiurban', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Urban', 'Semiurban', 'Urban', 'Rural', 'Semiurban', 'Urban',
       'Rural', 'Rural', 'Urban', 'Rural', 'Semiurban', 'Urban',
       'Semiurban', 'Semiurban', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Urban', 'Rural', 'Urban', 'Urban', 'Urban', 'Rural',
       'Semiurban', 'Semiurban', 'Semiurban', 'Semiurban', 'Urban',
       'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Urban', 'Urban',
       'Rural', 'Rural', 'Semiurban', 'Rural', 'Semiurban', 'Rural',
       'Rural', 'Semiurban', 'Urban', 'Urban', 'Semiurban', 'Urban',
       'Semiurban', 'Urban', 'Rural', 'Urban', 'Urban', 'Semiurban',
       'Rural', 'Urban', 'Rural', 'Urban', 'Rural', 'Urban', 'Rural',
       'Rural', 'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Rural',
       'Urban', 'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Semiurban',
       'Semiurban', 'Rural', 'Rural', 'Rural', 'Rural', 'Rural',
       'Semiurban', 'Urban', 'Urban', 'Urban', 'Semiurban', 'Urban',
       'Urban', 'Urban', 'Semiurban', 'Rural', 'Rural', 'Urban',
       'Semiurban', 'Rural', 'Rural', 'Urban', 'Semiurban', 'Rural',
       'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Rural', 'Semiurban',
       'Rural', 'Rural', 'Rural', 'Semiurban', 'Urban', 'Rural', 'Urban',
```

```
       'Urban', 'Semiurban', 'Semiurban', 'Semiurban', 'Urban', 'Urban',
       'Rural', 'Rural', 'Semiurban', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Semiurban', 'Rural', 'Urban', 'Urban', 'Urban',
       'Rural', 'Semiurban', 'Urban', 'Urban', 'Semiurban', 'Semiurban',
       'Urban', 'Urban', 'Urban', 'Urban', 'Semiurban', 'Urban', 'Urban',
       'Rural', 'Rural', 'Rural', 'Urban', 'Semiurban', 'Urban',
       'Semiurban', 'Urban', 'Urban', 'Rural', 'Semiurban', 'Urban',
       'Rural', 'Semiurban', 'Semiurban', 'Rural', 'Semiurban', 'Urban',
       'Rural', 'Semiurban', 'Rural', 'Semiurban', 'Urban', 'Rural',
       'Rural', 'Semiurban', 'Urban', 'Urban', 'Urban', 'Rural',
       'Semiurban', 'Rural', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Rural', 'Rural', 'Semiurban', 'Rural', 'Urban', 'Semiurban',
       'Rural', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban',
       'Semiurban', 'Urban', 'Urban', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Rural', 'Rural',
       'Rural', 'Semiurban', 'Rural', 'Urban', 'Rural', 'Rural',
       'Semiurban', 'Semiurban', 'Urban', 'Urban', 'Rural', 'Semiurban',
       'Semiurban', 'Urban', 'Rural', 'Semiurban', 'Semiurban', 'Rural',
       'Urban', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban', 'Rural',
       'Semiurban', 'Urban', 'Rural', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Rural', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Rural', 'Rural', 'Urban', 'Semiurban', 'Urban',
       'Semiurban', 'Urban', 'Rural', 'Semiurban', 'Rural', 'Semiurban',
       'Rural', 'Urban', 'Semiurban', 'Rural', 'Rural', 'Semiurban',
       'Semiurban', 'Rural', 'Semiurban', 'Rural', 'Urban', 'Urban',
       'Urban', 'Urban', 'Rural', 'Semiurban', 'Urban', 'Semiurban',
       'Rural', 'Rural', 'Semiurban', 'Semiurban', 'Semiurban', 'Rural',
       'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Semiurban', 'Semiurban', 'Semiurban', 'Rural',
       'Urban', 'Rural', 'Semiurban', 'Urban', 'Rural', 'Urban',
       'Semiurban', 'Rural', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Rural', 'Rural', 'Rural', 'Urban', 'Urban', 'Rural',
       'Semiurban', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Rural', 'Semiurban', 'Urban', 'Rural',
       'Urban', 'Rural', 'Urban', 'Urban', 'Urban', 'Urban', 'Urban',
       'Semiurban', 'Semiurban', 'Urban', 'Semiurban', 'Urban', 'Rural',
       'Urban', 'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Rural',
       'Rural', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Rural', 'Urban', 'Rural',
       'Rural', 'Semiurban', 'Rural', 'Urban', 'Urban', 'Rural', 'Urban',
       'Rural', 'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Rural',
       'Rural', 'Rural', 'Urban', 'Urban', 'Semiurban'], dtype=object)
```

In [50]:
```
path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
loan_df['Property_Area']=le.fit_transform(loan_df['Property_Area'])
```

In [51]:
```
le.inverse_transform(loan_df['Property_Area'])
```

Out[51]:
```
array(['Urban', 'Rural', 'Urban', 'Urban', 'Urban', 'Urban', 'Urban',
       'Semiurban', 'Urban', 'Semiurban', 'Urban', 'Urban', 'Urban',
       'Rural', 'Urban', 'Urban', 'Urban', 'Urban', 'Rural', 'Urban',
       'Urban', 'Urban', 'Semiurban', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Urban', 'Semiurban', 'Urban', 'Urban',
       'Rural', 'Semiurban', 'Rural', 'Urban', 'Urban', 'Semiurban',
       'Urban', 'Semiurban', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Urban', 'Urban', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Urban', 'Urban',
       'Semiurban', 'Semiurban', 'Rural', 'Urban', 'Urban', 'Urban',
       'Urban', 'Rural', 'Rural', 'Semiurban', 'Semiurban', 'Urban',
       'Urban', 'Urban', 'Semiurban', 'Urban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban',
       'Urban', 'Semiurban', 'Semiurban', 'Semiurban', 'Urban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban',
       'Semiurban', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Semiurban', 'Urban', 'Semiurban', 'Urban',
       'Urban', 'Urban', 'Rural', 'Urban', 'Semiurban', 'Urban',
       'Semiurban', 'Rural', 'Semiurban', 'Semiurban', 'Rural',
       'Semiurban', 'Urban', 'Rural', 'Urban', 'Rural', 'Semiurban',
       'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Rural', 'Rural',
       'Urban', 'Rural', 'Urban', 'Urban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Rural',
       'Rural', 'Urban', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban',
       'Urban', 'Urban', 'Rural', 'Semiurban', 'Rural', 'Rural', 'Urban',
       'Rural', 'Urban', 'Semiurban', 'Rural', 'Urban', 'Rural',
       'Semiurban', 'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Urban',
       'Rural', 'Rural', 'Rural', 'Semiurban', 'Semiurban', 'Rural',
       'Urban', 'Rural', 'Semiurban', 'Semiurban', 'Rural', 'Rural',
       'Semiurban', 'Semiurban', 'Urban', 'Urban', 'Rural', 'Semiurban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Rural',
       'Rural', 'Rural', 'Semiurban', 'Urban', 'Semiurban', 'Rural',
       'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Urban', 'Semiurban',
       'Semiurban', 'Urban', 'Urban', 'Semiurban', 'Semiurban', 'Urban',
       'Rural', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban', 'Urban',
       'Rural', 'Urban', 'Semiurban', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Semiurban', 'Rural',
       'Semiurban', 'Rural', 'Rural', 'Semiurban', 'Semiurban', 'Rural',
       'Urban', 'Urban', 'Rural', 'Semiurban', 'Rural', 'Urban', 'Urban',
       'Rural', 'Semiurban', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Urban', 'Semiurban', 'Urban', 'Rural', 'Semiurban', 'Urban',
       'Rural', 'Rural', 'Urban', 'Rural', 'Semiurban', 'Urban',
       'Semiurban', 'Semiurban', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Urban', 'Rural', 'Urban', 'Urban', 'Urban', 'Rural',
       'Semiurban', 'Semiurban', 'Semiurban', 'Semiurban', 'Urban',
       'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Urban', 'Urban',
       'Rural', 'Rural', 'Semiurban', 'Rural', 'Semiurban', 'Rural',
       'Rural', 'Semiurban', 'Urban', 'Urban', 'Semiurban', 'Urban',
       'Semiurban', 'Urban', 'Rural', 'Urban', 'Urban', 'Semiurban',
       'Rural', 'Urban', 'Rural', 'Urban', 'Rural', 'Urban', 'Rural',
       'Rural', 'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Rural',
       'Urban', 'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Semiurban',
       'Semiurban', 'Rural', 'Rural', 'Rural', 'Rural', 'Rural',
       'Semiurban', 'Urban', 'Urban', 'Urban', 'Semiurban', 'Urban',
       'Urban', 'Urban', 'Semiurban', 'Rural', 'Rural', 'Urban',
       'Semiurban', 'Rural', 'Rural', 'Urban', 'Semiurban', 'Rural',
       'Semiurban', 'Rural', 'Urban', 'Semiurban', 'Rural', 'Semiurban',
       'Rural', 'Rural', 'Rural', 'Semiurban', 'Urban', 'Rural', 'Urban',
```

```
       'Urban', 'Semiurban', 'Semiurban', 'Semiurban', 'Urban', 'Urban',
       'Rural', 'Rural', 'Semiurban', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Semiurban', 'Rural', 'Urban', 'Urban', 'Urban',
       'Rural', 'Semiurban', 'Urban', 'Urban', 'Semiurban', 'Semiurban',
       'Urban', 'Urban', 'Urban', 'Urban', 'Semiurban', 'Urban', 'Urban',
       'Rural', 'Rural', 'Rural', 'Urban', 'Semiurban', 'Urban',
       'Semiurban', 'Urban', 'Urban', 'Rural', 'Semiurban', 'Urban',
       'Rural', 'Semiurban', 'Semiurban', 'Rural', 'Semiurban', 'Urban',
       'Rural', 'Semiurban', 'Rural', 'Semiurban', 'Urban', 'Rural',
       'Rural', 'Semiurban', 'Urban', 'Urban', 'Urban', 'Rural',
       'Semiurban', 'Rural', 'Urban', 'Urban', 'Urban', 'Semiurban',
       'Rural', 'Rural', 'Semiurban', 'Rural', 'Urban', 'Semiurban',
       'Rural', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban',
       'Semiurban', 'Urban', 'Urban', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Rural', 'Rural',
       'Rural', 'Semiurban', 'Rural', 'Urban', 'Rural', 'Rural',
       'Semiurban', 'Semiurban', 'Urban', 'Urban', 'Rural', 'Semiurban',
       'Semiurban', 'Urban', 'Rural', 'Semiurban', 'Semiurban', 'Rural',
       'Urban', 'Semiurban', 'Semiurban', 'Urban', 'Semiurban', 'Rural',
       'Semiurban', 'Urban', 'Rural', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Rural', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Rural', 'Rural', 'Urban', 'Semiurban', 'Urban',
       'Semiurban', 'Urban', 'Rural', 'Semiurban', 'Rural', 'Semiurban',
       'Rural', 'Urban', 'Semiurban', 'Rural', 'Rural', 'Semiurban',
       'Semiurban', 'Rural', 'Semiurban', 'Rural', 'Urban', 'Urban',
       'Urban', 'Urban', 'Rural', 'Semiurban', 'Urban', 'Semiurban',
       'Rural', 'Rural', 'Semiurban', 'Semiurban', 'Semiurban', 'Rural',
       'Semiurban', 'Semiurban', 'Rural', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Semiurban', 'Semiurban', 'Semiurban', 'Rural',
       'Urban', 'Rural', 'Semiurban', 'Urban', 'Rural', 'Urban',
       'Semiurban', 'Rural', 'Rural', 'Semiurban', 'Semiurban',
       'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Semiurban', 'Rural',
       'Semiurban', 'Rural', 'Rural', 'Rural', 'Urban', 'Urban', 'Rural',
       'Semiurban', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Rural', 'Semiurban', 'Urban', 'Rural',
       'Urban', 'Rural', 'Urban', 'Urban', 'Urban', 'Urban', 'Urban',
       'Semiurban', 'Semiurban', 'Urban', 'Semiurban', 'Urban', 'Rural',
       'Urban', 'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Rural',
       'Rural', 'Urban', 'Semiurban', 'Semiurban', 'Semiurban',
       'Semiurban', 'Semiurban', 'Semiurban', 'Rural', 'Urban', 'Rural',
       'Rural', 'Semiurban', 'Rural', 'Urban', 'Urban', 'Rural', 'Urban',
       'Rural', 'Semiurban', 'Urban', 'Semiurban', 'Rural', 'Rural',
       'Rural', 'Rural', 'Urban', 'Urban', 'Semiurban'], dtype=object)
```

np.where

In [52]:
```
con=loan_df['Loan_Status']=='Y'
loan_df['Loan_Status']=np.where(con,0,1)
loan_df
```

Out[52]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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

pd.get_dummies

In [56]:
```
path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)
pd.get_dummies(loan_df,
               columns=['Self_Employed','Loan_Status'],
               dtype='int')
```

Out[56]:

	Loan_ID	Gender	Married	Dependents	Education	ApplicantIncome	CoapplicantIncome	LoanA
0	LP001002	Male	No	0	Graduate	5849	0.0	
1	LP001003	Male	Yes	1	Graduate	4583	1508.0	
2	LP001005	Male	Yes	0	Graduate	3000	0.0	
3	LP001006	Male	Yes	0	Not Graduate	2583	2358.0	
4	LP001008	Male	No	0	Graduate	6000	0.0	
...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	2900	0.0	
610	LP002979	Male	Yes	3+	Graduate	4106	0.0	
611	LP002983	Male	Yes	1	Graduate	8072	240.0	
612	LP002984	Male	Yes	2	Graduate	7583	0.0	
613	LP002990	Female	No	0	Graduate	4583	0.0	

614 rows × 15 columns

In [57]:
```
path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)
loan_df.drop('Loan_Status',axis=1,inplace=True)
pd.get_dummies(loan_df,dtype='int')
```

Out[57]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Loan_ID_
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 × 634 columns

DataStandardization

In [58]:
```
lwage=loan_df['ApplicantIncome']
lwage_mean=loan_df['ApplicantIncome'].mean()
lwage_std=loan_df['ApplicantIncome'].std()
Nr=lwage-lwage_mean
loan_df['ApplicantIncome_wage']=Nr/lwage_std
```

In [60]:
```
loan_df[['ApplicantIncome','ApplicantIncome_wage']]
```

Out[60]:

	ApplicantIncome	ApplicantIncome_wage
0	5849	0.072931
1	4583	-0.134302
2	3000	-0.393427
3	2583	-0.461686
4	6000	0.097649
...	...	...
609	2900	-0.409796
610	4106	-0.212383
611	8072	0.436818
612	7583	0.356773
613	4583	-0.134302

614 rows × 2 columns

In [61]:
```
loan_df['ApplicantIncome'].idxmax()
```

Out[61]:
```
409
```

In [62]:
```
loan_df['ApplicantIncome'].idxmin()
```

Out[62]:
```
216
```

In [63]:
```
loan_df.iloc[[409,216]]
```

Out[63]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
409	LP002317	Male	Yes	3+	Graduate	No	81000	
216	LP001722	Male	Yes	0	Graduate	No	150	

In [64]:
```
cols=['ApplicantIncome','ApplicantIncome_wage']
ids=[409,216]
loan_df[['ApplicantIncome','ApplicantIncome_wage']].iloc[[409,216]]
loan_df[cols].iloc[ids]
```

Out[64]:

	ApplicantIncome	ApplicantIncome_wage
409	81000	12.374533
216	150	-0.859948

StandardScalar

In [68]:
```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
loan_df['ApplicantIncome_wage_ss']=ss.fit_transform(loan_df[['ApplicantIncome']])
```

In [69]:
```
cols=['ApplicantIncome','ApplicantIncome_wage','ApplicantIncome_wage_ss']
loan_df[cols]
```

Out[69]:

	ApplicantIncome	ApplicantIncome_wage	ApplicantIncome_wage_ss
0	5849	0.072931	0.072991
1	4583	-0.134302	-0.134412
2	3000	-0.393427	-0.393747
3	2583	-0.461686	-0.462062
4	6000	0.097649	0.097728
...	...	...	...
609	2900	-0.409796	-0.410130
610	4106	-0.212383	-0.212557
611	8072	0.436818	0.437174
612	7583	0.356773	0.357064
613	4583	-0.134302	-0.134412

614 rows × 3 columns

Normalization

minmaxscalar

In [73]:
```
lwage=loan_df['ApplicantIncome']
lwage_min=loan_df['ApplicantIncome'].min()
lwage_max=loan_df['ApplicantIncome'].max()
nr=lwage-lwage_min
dr=lwage_max-lwage_min
loan_df['ApplicantIncome_norm']=nr/dr
```

In [74]:
```
loan_df['ApplicantIncome_norm'].min(),loan_df['ApplicantIncome_norm'].max()
```

Out[74]:
```
(0.0, 1.0)
```

In [75]:
```
from sklearn.preprocessing import MinMaxScaler
mms=MinMaxScaler()
```

```
loan_df['ApplicantIncome_mms']=mms.fit_transform(loan_df[['ApplicantIncome']])
```

In [76]:
```
cols=['ApplicantIncome','ApplicantIncome_wage','ApplicantIncome_wage_ss','ApplicantInc
loan_df[cols]
```

Out[76]:

	ApplicantIncome	ApplicantIncome_wage	ApplicantIncome_wage_ss	ApplicantIncome_norm
0	5849	0.070489	0.072991	0.070489
1	4583	0.054830	-0.134412	0.054830
2	3000	0.035250	-0.393747	0.035250
3	2583	0.030093	-0.462062	0.030093
4	6000	0.072356	0.097728	0.072356
...	...	...	...	...
609	2900	0.034014	-0.410130	0.034014
610	4106	0.048930	-0.212557	0.048930
611	8072	0.097984	0.437174	0.097984
612	7583	0.091936	0.357064	0.091936
613	4583	0.054830	-0.134412	0.054830

614 rows × 4 columns

Missingvalueanalysis

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path=r"C:\Users\jagad\OneDrive\Documents\DATA SCIENCE\analyticsvidhya\train_ctrUa4K.cs
loan_df=pd.read_csv(path)
loan_df
```

Out[2]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [3]:
```
loan_df.isnull()
```

Out[3]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplicant
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 [4]:
```
loan_df.isnull().sum()
```

Out[4]:
```
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
```

Method-2

In [5]:
```
loan_df.fillna(180)
```

Out[5]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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

Method-2

- Fill the missing values with random number on specific column

In [6]:
```
loan_df['Gender'].fillna('Male',inplace=True)
loan_df
```

Out[6]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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

Method-3

- bfill
- ffill
- pad
- backfill

In [8]:
```
loan_df.fillna(method='backfill')
```

Out[8]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [9]:
```
loan_df.fillna(method='ffill')
```

Out[9]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [10]:
```
loan_df.fillna(method='pad')
```

Out[10]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 [11]:
```
loan_df.fillna(method='backfill')
```

Out[11]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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

Mean-Median-Mode

In [19]:
```
LoanAmount_mean=loan_df['LoanAmount'].mean()
LoanAmount_mean
```

Out[19]:
```
146.41216216216216
```

In [21]:
```
loan_df['LoanAmount'].fillna(LoanAmount_mean)
```

Out[21]:
```
0      146.412162
1      128.000000
2       66.000000
3      120.000000
4      141.000000
          ...
609     71.000000
610     40.000000
611    253.000000
612    187.000000
613    133.000000
Name: LoanAmount, Length: 614, dtype: float64
```

In [17]:
```
LoanAmount_median=loan_df['LoanAmount'].median()
LoanAmount_median
```

Out[17]:
```
128.0
```

In [22]:
```
loan_df['LoanAmount'].fillna(LoanAmount_median)
```

Out[22]:
```
0      128.0
1      128.0
2       66.0
3      120.0
4      141.0
       ...
609     71.0
610     40.0
611    253.0
612    187.0
613    133.0
Name: LoanAmount, Length: 614, dtype: float64
```

In [25]:
```
LoanAmount_mode=loan_df['LoanAmount'].mode()
LoanAmount_mode
```

Out[25]:
```
0    120.0
Name: LoanAmount, dtype: float64
```

In [26]:
```
loan_df['LoanAmount'].fillna(LoanAmount_mode)
```

Out[26]:
```
0      120.0
1      128.0
2       66.0
3      120.0
4      141.0
       ...
609     71.0
610     40.0
611    253.0
612    187.0
613    133.0
Name: LoanAmount, Length: 614, dtype: float64
```

Method-5:KNN imputer

In [23]:
```
from sklearn.impute import KNNImputer
knn=KNNImputer(n_neighbors=2)
knn.fit_transform(loan_df[['LoanAmount']])
```

Out[23]:
```
array([[146.41216216],
       [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.        ],
       [120.        ],
       [201.        ],
       [ 74.        ],
       [106.        ],
       [114.        ],
       [320.        ],
       [146.41216216],
       [100.        ],
       [144.        ],
       [184.        ],
       [110.        ],
       [ 80.        ],
       [ 47.        ],
       [ 75.        ],
       [134.        ],
       [ 96.        ],
       [ 88.        ],
       [ 44.        ],
       [144.        ],
       [120.        ],
       [144.        ],
       [100.        ],
       [120.        ],
       [112.        ],
       [134.        ],
       [286.        ],
       [ 97.        ],
       [ 96.        ],
       [135.        ],
       [180.        ],
       [144.        ],
```

```
       [120.        ],
       [ 99.        ],
       [165.        ],
       [146.41216216],
       [116.        ],
       [258.        ],
       [126.        ],
       [312.        ],
       [125.        ],
       [136.        ],
       [172.        ],
       [ 97.        ],
       [ 81.        ],
       [ 95.        ],
       [187.        ],
       [113.        ],
       [176.        ],
       [110.        ],
       [180.        ],
       [130.        ],
       [111.        ],
       [146.41216216],
       [167.        ],
       [265.        ],
       [ 50.        ],
       [136.        ],
       [ 99.        ],
       [104.        ],
       [210.        ],
       [175.        ],
       [131.        ],
       [188.        ],
       [ 81.        ],
       [122.        ],
       [ 25.        ],
       [146.41216216],
       [137.        ],
       [ 50.        ],
       [115.        ],
       [131.        ],
       [133.        ],
       [151.        ],
       [146.41216216],
       [146.41216216],
       [160.        ],
       [100.        ],
       [225.        ],
       [120.        ],
       [216.        ],
       [ 94.        ],
       [136.        ],
       [139.        ],
       [152.        ],
       [146.41216216],
       [118.        ],
       [185.        ],
       [154.        ],
       [ 85.        ],
       [175.        ],
       [259.        ],
```

```
[180.        ],
[ 44.        ],
[137.        ],
[ 81.        ],
[194.        ],
[ 93.        ],
[370.        ],
[146.41216216],
[160.        ],
[182.        ],
[650.        ],
[ 74.        ],
[ 70.        ],
[ 25.        ],
[102.        ],
[290.        ],
[ 84.        ],
[ 88.        ],
[242.        ],
[129.        ],
[185.        ],
[168.        ],
[175.        ],
[122.        ],
[187.        ],
[100.        ],
[ 70.        ],
[ 30.        ],
[225.        ],
[125.        ],
[118.        ],
[152.        ],
[244.        ],
[113.        ],
[ 50.        ],
[600.        ],
[160.        ],
[187.        ],
[120.        ],
[255.        ],
[ 98.        ],
[275.        ],
[121.        ],
[158.        ],
[ 75.        ],
[182.        ],
[112.        ],
[129.        ],
[ 63.        ],
[200.        ],
[ 95.        ],
[700.        ],
[ 81.        ],
[187.        ],
[ 87.        ],
[116.        ],
[101.        ],
[495.        ],
[116.        ],
[102.        ],
```

```
[180.        ],
[ 67.        ],
[ 73.        ],
[260.        ],
[108.        ],
[120.        ],
[ 66.        ],
[ 58.        ],
[168.        ],
[188.        ],
[ 48.        ],
[164.        ],
[160.        ],
[ 76.        ],
[120.        ],
[170.        ],
[187.        ],
[120.        ],
[113.        ],
[ 83.        ],
[ 90.        ],
[166.        ],
[146.41216216],
[135.        ],
[124.        ],
[120.        ],
[ 80.        ],
[ 55.        ],
[ 59.        ],
[127.        ],
[214.        ],
[128.        ],
[240.        ],
[130.        ],
[137.        ],
[100.        ],
[135.        ],
[131.        ],
[ 72.        ],
[127.        ],
[ 60.        ],
[116.        ],
[144.        ],
[175.        ],
[128.        ],
[170.        ],
[138.        ],
[210.        ],
[158.        ],
[200.        ],
[104.        ],
[ 42.        ],
[120.        ],
[280.        ],
[140.        ],
[170.        ],
[255.        ],
[122.        ],
[112.        ],
[ 96.        ],
```

```
       [120.        ],
       [140.        ],
       [155.        ],
       [108.        ],
       [123.        ],
       [120.        ],
       [112.        ],
       [137.        ],
       [123.        ],
       [ 90.        ],
       [201.        ],
       [138.        ],
       [104.        ],
       [279.        ],
       [192.        ],
       [255.        ],
       [115.        ],
       [ 94.        ],
       [304.        ],
       [128.        ],
       [330.        ],
       [134.        ],
       [155.        ],
       [120.        ],
       [128.        ],
       [151.        ],
       [150.        ],
       [160.        ],
       [135.        ],
       [ 90.        ],
       [ 30.        ],
       [136.        ],
       [126.        ],
       [150.        ],
       [ 90.        ],
       [115.        ],
       [207.        ],
       [ 80.        ],
       [436.        ],
       [124.        ],
       [158.        ],
       [112.        ],
       [ 78.        ],
       [ 54.        ],
       [146.41216216],
       [ 89.        ],
       [ 99.        ],
       [120.        ],
       [115.        ],
       [187.        ],
       [139.        ],
       [127.        ],
       [134.        ],
       [143.        ],
       [172.        ],
       [110.        ],
       [200.        ],
       [135.        ],
       [151.        ],
       [113.        ],
```

```
       [ 93.        ],
       [105.        ],
       [132.        ],
       [ 96.        ],
       [140.        ],
       [146.41216216],
       [135.        ],
       [104.        ],
       [480.        ],
       [185.        ],
       [ 84.        ],
       [111.        ],
       [ 56.        ],
       [144.        ],
       [159.        ],
       [111.        ],
       [120.        ],
       [ 88.        ],
       [112.        ],
       [155.        ],
       [115.        ],
       [124.        ],
       [146.41216216],
       [132.        ],
       [300.        ],
       [376.        ],
       [130.        ],
       [184.        ],
       [110.        ],
       [ 67.        ],
       [117.        ],
       [ 98.        ],
       [ 71.        ],
       [490.        ],
       [182.        ],
       [ 70.        ],
       [160.        ],
       [176.        ],
       [146.41216216],
       [ 71.        ],
       [173.        ],
       [ 46.        ],
       [158.        ],
       [ 74.        ],
       [125.        ],
       [160.        ],
       [152.        ],
       [126.        ],
       [259.        ],
       [187.        ],
       [228.        ],
       [308.        ],
       [ 95.        ],
       [105.        ],
       [130.        ],
       [116.        ],
       [165.        ],
       [ 67.        ],
       [100.        ],
       [200.        ],
```

```
[ 81.        ],
[236.        ],
[130.        ],
[ 95.        ],
[141.        ],
[133.        ],
[ 96.        ],
[124.        ],
[175.        ],
[570.        ],
[ 55.        ],
[155.        ],
[380.        ],
[111.        ],
[110.        ],
[120.        ],
[130.        ],
[130.        ],
[ 71.        ],
[130.        ],
[128.        ],
[296.        ],
[156.        ],
[128.        ],
[100.        ],
[113.        ],
[132.        ],
[146.41216216],
[136.        ],
[125.        ],
[185.        ],
[275.        ],
[120.        ],
[113.        ],
[113.        ],
[135.        ],
[ 71.        ],
[ 95.        ],
[109.        ],
[103.        ],
[ 45.        ],
[ 65.        ],
[103.        ],
[ 53.        ],
[194.        ],
[115.        ],
[115.        ],
[ 66.        ],
[152.        ],
[360.        ],
[ 62.        ],
[160.        ],
[218.        ],
[110.        ],
[178.        ],
[ 60.        ],
[160.        ],
[239.        ],
[112.        ],
[138.        ],
```

```
       [138.        ],
       [ 80.        ],
       [100.        ],
       [110.        ],
       [ 96.        ],
       [121.        ],
       [ 81.        ],
       [133.        ],
       [ 87.        ],
       [ 60.        ],
       [150.        ],
       [105.        ],
       [405.        ],
       [143.        ],
       [100.        ],
       [146.41216216],
       [ 50.        ],
       [146.41216216],
       [187.        ],
       [138.        ],
       [187.        ],
       [180.        ],
       [148.        ],
       [152.        ],
       [175.        ],
       [130.        ],
       [110.        ],
       [ 55.        ],
       [150.        ],
       [190.        ],
       [125.        ],
       [ 60.        ],
       [149.        ],
       [ 90.        ],
       [ 84.        ],
       [ 96.        ],
       [118.        ],
       [173.        ],
       [136.        ],
       [160.        ],
       [160.        ],
       [128.        ],
       [153.        ],
       [132.        ],
       [ 98.        ],
       [140.        ],
       [ 70.        ],
       [110.        ],
       [ 98.        ],
       [110.        ],
       [162.        ],
       [113.        ],
       [100.        ],
       [ 93.        ],
       [162.        ],
       [150.        ],
       [230.        ],
       [132.        ],
       [ 86.        ],
       [146.41216216],
```

```
       [154.        ],
       [113.        ],
       [128.        ],
       [234.        ],
       [246.        ],
       [131.        ],
       [ 80.        ],
       [500.        ],
       [160.        ],
       [ 75.        ],
       [ 96.        ],
       [186.        ],
       [110.        ],
       [225.        ],
       [119.        ],
       [105.        ],
       [107.        ],
       [111.        ],
       [ 95.        ],
       [209.        ],
       [113.        ],
       [100.        ],
       [208.        ],
       [138.        ],
       [124.        ],
       [243.        ],
       [480.        ],
       [ 96.        ],
       [188.        ],
       [ 40.        ],
       [100.        ],
       [250.        ],
       [148.        ],
       [ 70.        ],
       [311.        ],
       [150.        ],
       [113.        ],
       [123.        ],
       [185.        ],
       [ 95.        ],
       [ 45.        ],
       [ 55.        ],
       [100.        ],
       [480.        ],
       [146.41216216],
       [400.        ],
       [110.        ],
       [161.        ],
       [ 94.        ],
       [130.        ],
       [216.        ],
       [100.        ],
       [110.        ],
       [196.        ],
       [125.        ],
       [126.        ],
       [324.        ],
       [107.        ],
       [ 66.        ],
       [157.        ],
```

```
       [140.        ],
       [ 99.        ],
       [ 95.        ],
       [128.        ],
       [102.        ],
       [155.        ],
       [ 80.        ],
       [145.        ],
       [103.        ],
       [110.        ],
       [146.41216216],
       [146.41216216],
       [158.        ],
       [181.        ],
       [132.        ],
       [ 26.        ],
       [ 84.        ],
       [260.        ],
       [162.        ],
       [182.        ],
       [108.        ],
       [600.        ],
       [211.        ],
       [132.        ],
       [258.        ],
       [120.        ],
       [ 70.        ],
       [123.        ],
       [  9.        ],
       [104.        ],
       [186.        ],
       [165.        ],
       [275.        ],
       [187.        ],
       [150.        ],
       [108.        ],
       [136.        ],
       [110.        ],
       [107.        ],
       [161.        ],
       [205.        ],
       [ 90.        ],
       [ 36.        ],
       [ 61.        ],
       [146.        ],
       [172.        ],
       [104.        ],
       [ 70.        ],
       [ 94.        ],
       [106.        ],
       [ 56.        ],
       [205.        ],
       [292.        ],
       [142.        ],
       [260.        ],
       [110.        ],
       [187.        ],
       [ 88.        ],
       [180.        ],
       [192.        ],
```

```
       [350.        ],
       [155.        ],
       [128.        ],
       [172.        ],
       [496.        ],
       [146.41216216],
       [173.        ],
       [157.        ],
       [108.        ],
       [ 71.        ],
       [ 40.        ],
       [253.        ],
       [187.        ],
       [133.        ]])
```

In [24]:
```
loan_df
```

Out[24]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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

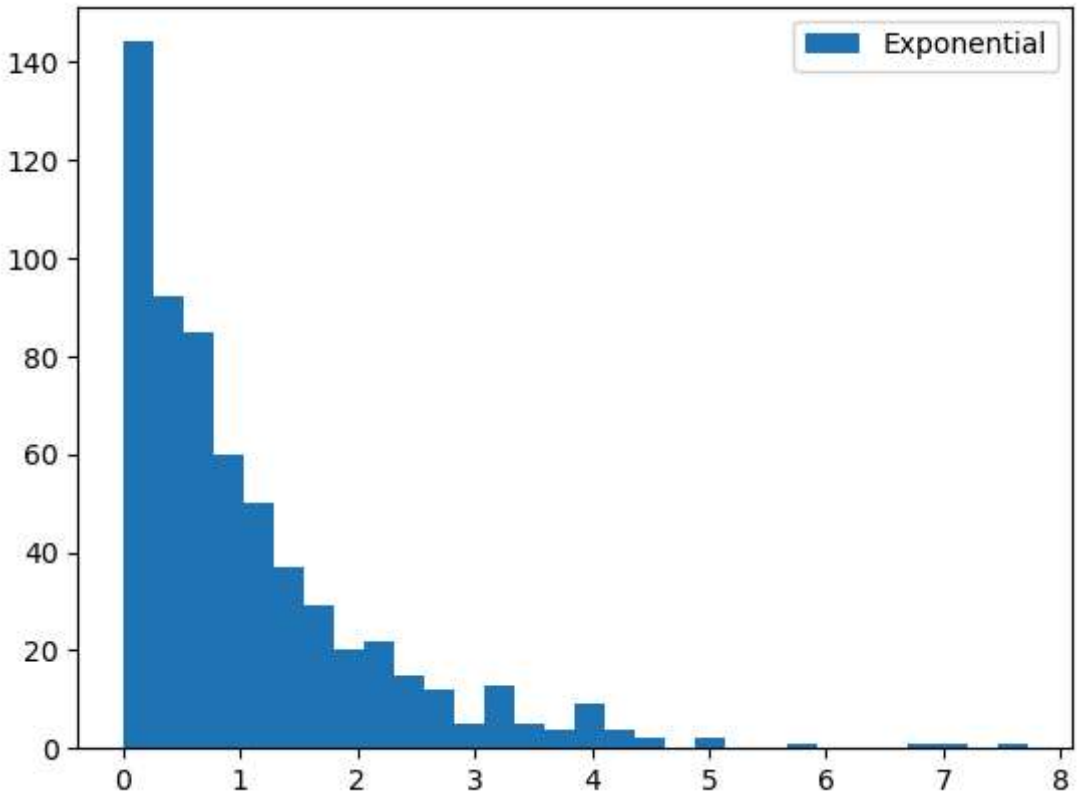
Data Transformation Techniques

Exponential-data

In [27]:
```
exp_loan=np.random.exponential(size=614)
exp_loan[:20]
```

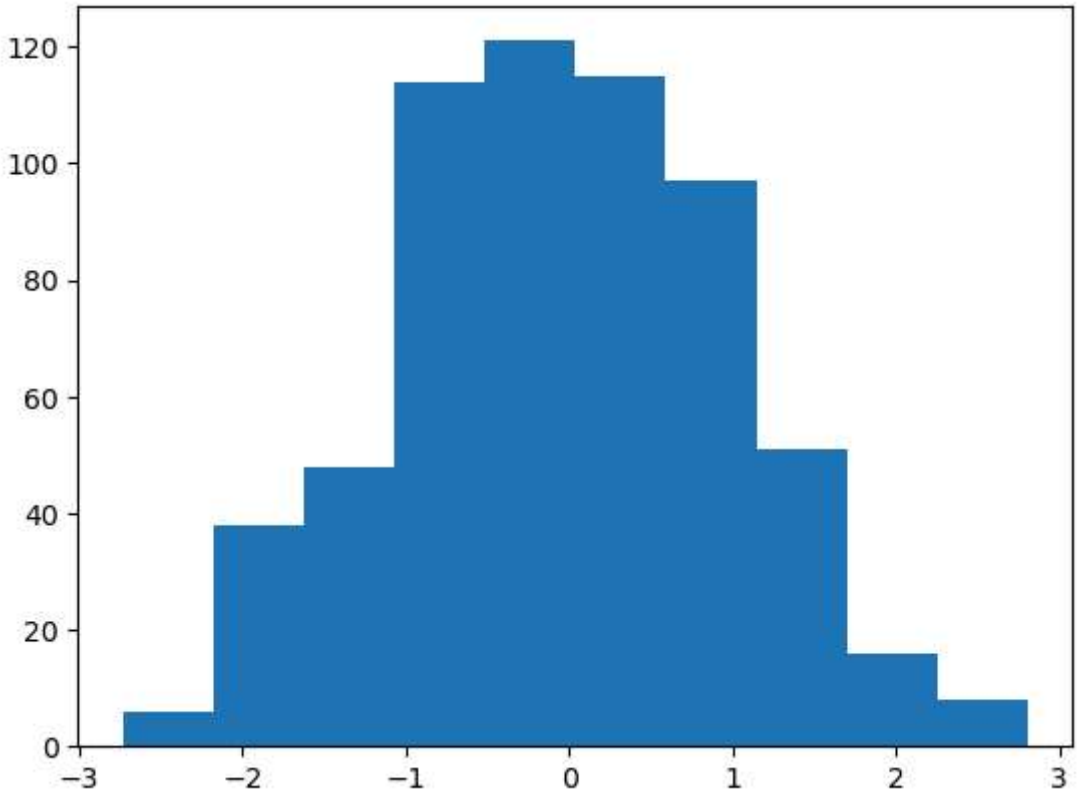
Out[27]:
```
array([3.60523959, 0.82645378, 0.55097796, 1.79338761, 0.05728885,
       3.01653891, 0.35461946, 0.09290766, 2.75157543, 0.42351153,
       0.44161361, 0.26070121, 0.3715359 , 0.81661513, 0.21807939,
       0.80920791, 1.06115294, 0.48160385, 0.52425165, 1.79869136])
```

In [29]:
```
plt.hist(exp_loan,bins=30,label='Exponential')
plt.legend()
plt.show()
```



Norm-data

In [31]:
```
norm_loan=np.random.normal(size=614)
plt.hist(norm_loan)
```



Log Transformation

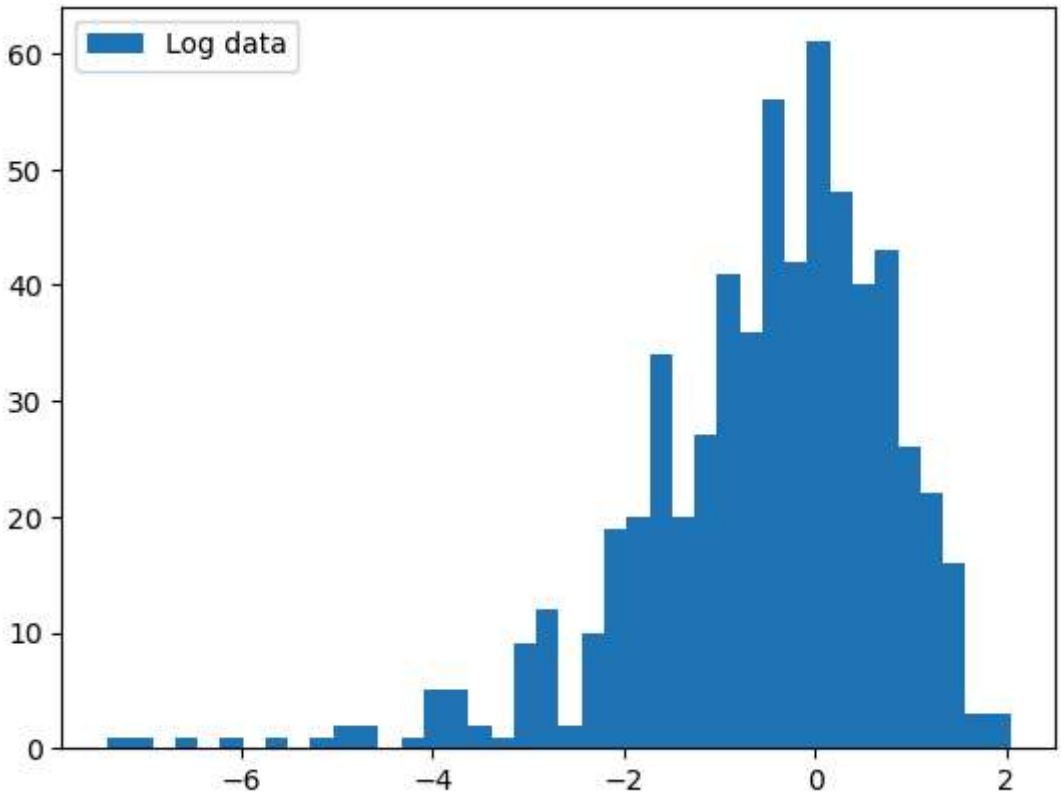
In [33]:
```
log_loan=np.log(exp_loan)
log_loan[:30]
```

Out[33]:
```
array([ 1.28238823, -0.19061129, -0.59606046,  0.58410635, -2.85964921,
        1.10411012, -1.03671   , -2.37614923,  1.01217363, -0.85917453,
       -0.81731996, -1.3443803 , -0.99010979, -0.20258737, -1.5228961 ,
       -0.21169939,  0.059356  , -0.73063339, -0.64578346,  0.58705938,
        0.14632918,  0.93868218,  0.78511583,  0.68069599, -2.00125465,
       -1.00080273, -6.98489074, -0.03183859, -0.32585585, -1.58938623])
```

In [34]:
```
exp_loan[:30]
```

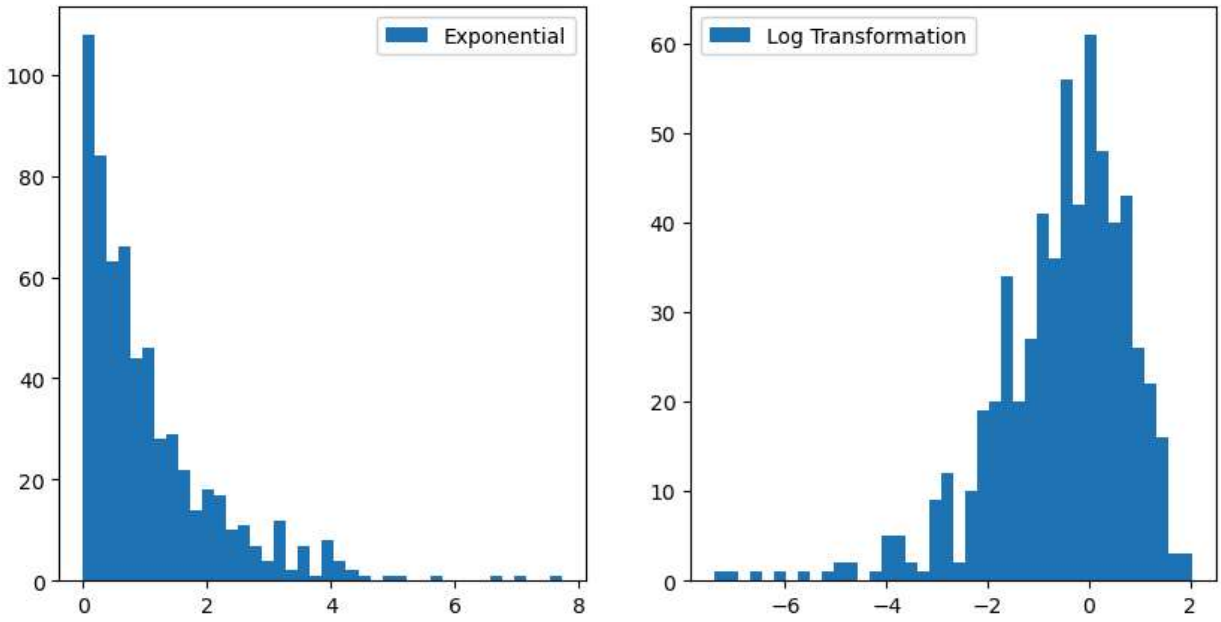
Out[34]:
```
array([3.60523959e+00, 8.26453775e-01, 5.50977964e-01, 1.79338761e+00,
       5.72888531e-02, 3.01653891e+00, 3.54619462e-01, 9.29076555e-02,
       2.75157543e+00, 4.23511532e-01, 4.41613612e-01, 2.60701215e-01,
       3.71535898e-01, 8.16615133e-01, 2.18079391e-01, 8.09207914e-01,
       1.06115294e+00, 4.81603852e-01, 5.24251649e-01, 1.79869136e+00,
       1.15757718e+00, 2.55661004e+00, 2.19266091e+00, 1.97525202e+00,
       1.35165592e-01, 3.67584252e-01, 9.25764439e-04, 9.68662925e-01,
       7.21909243e-01, 2.04050814e-01])
```

In [35]:
```
plt.hist(log_loan,bins=40,label='Log data')
plt.legend()
plt.show()
```



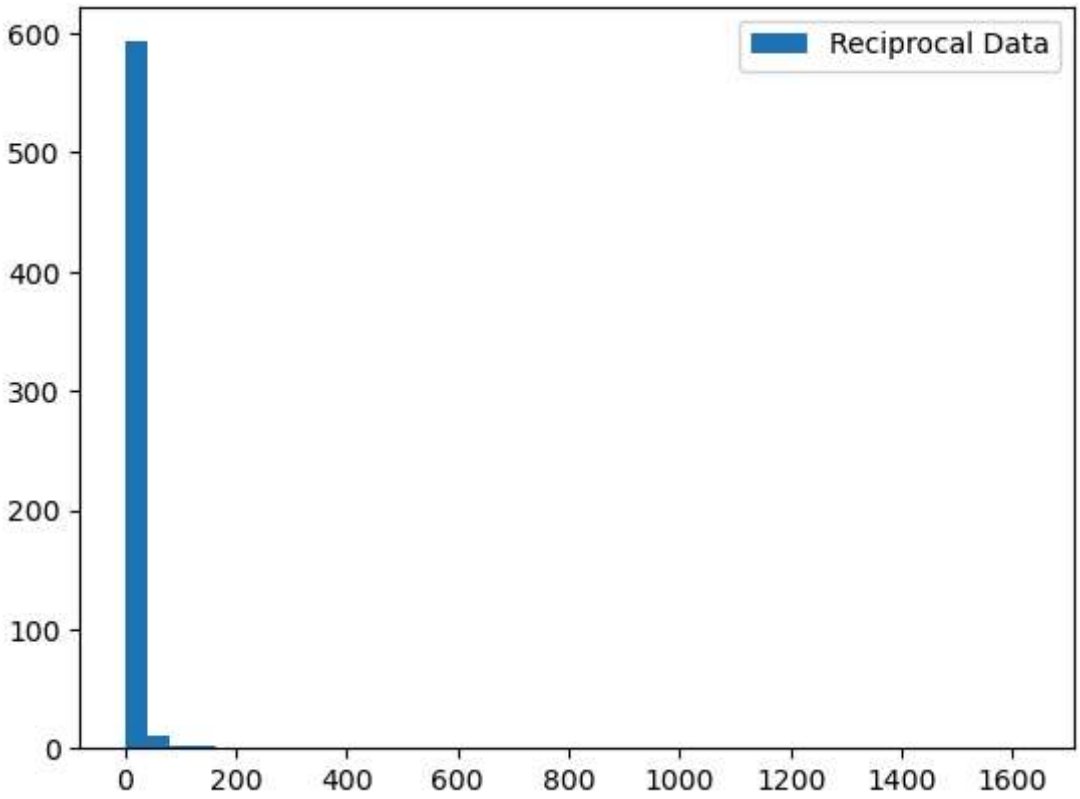
In [36]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1).hist(exp_loan,
bins=40,

label='Exponential')

plt.legend()
plt.subplot(1,2,2).hist(log_loan,
```

```
bins=40,

label='Log Transformation')

plt.legend()
plt.show()
```



Reciprocal transformation

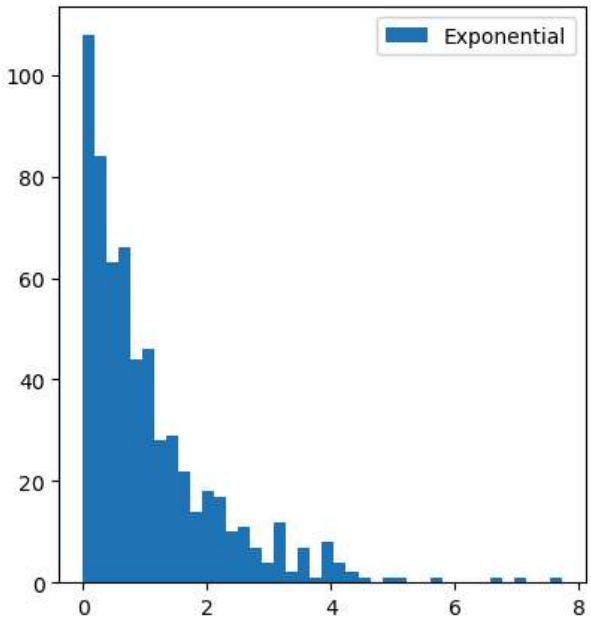
In [39]:
```
rec_loan=np.reciprocal(exp_loan)
plt.hist(rec_loan,bins=40,label='Reciprocal Data')
plt.legend()
plt.show()
```

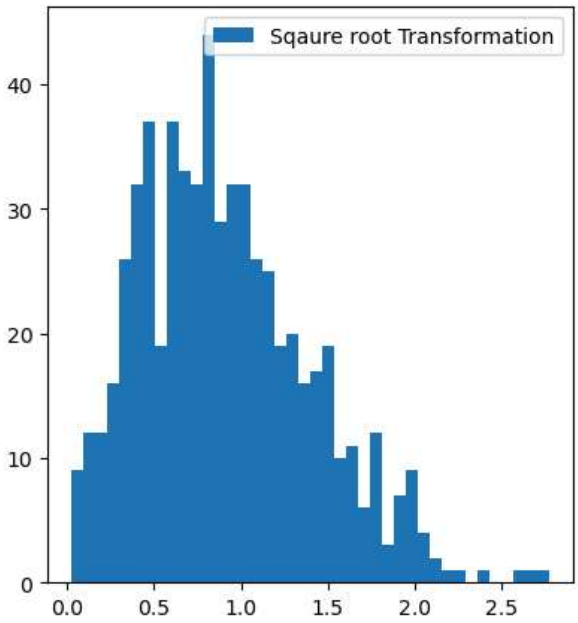


Squarerootransformation

In [40]:
```
sqrt_loan=np.sqrt(exp_loan)
```

In [41]:
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1).hist(exp_loan,
bins=40,

label='Exponential')

plt.legend()
plt.subplot(1,2,2).hist(sqrt_loan,
bins=40,

label='Sqaure root Transformation')

plt.legend()
plt.show()
```





Power Transformer

In [43]:
```
from sklearn.preprocessing import PowerTransformer
pt=PowerTransformer()
pt_loan=pt.fit([exp_loan]).transform([exp_loan])
```

In [44]:
```
pt_loan
```

Out[44]:
```
array([[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
        -2.22044605e-16,  8.32667268e-17,  0.00000000e+00,
         5.55111512e-17,  1.38777878e-17,  0.00000000e+00,
        -1.11022302e-16,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00, -1.11022302e-16, -8.32667268e-17,
         0.00000000e+00,  0.00000000e+00,  5.55111512e-17,
         0.00000000e+00,  0.00000000e+00, -2.22044605e-16,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
        -1.11022302e-16,  0.00000000e+00, -4.46691295e-17,
         1.11022302e-16,  0.00000000e+00, -8.32667268e-17,
         2.22044605e-16,  2.22044605e-16,  0.00000000e+00,
        -1.11022302e-16,  2.22044605e-16, -8.32667268e-17,
         1.11022302e-16, -1.11022302e-16,  0.00000000e+00,
         0.00000000e+00,  8.32667268e-17,  1.11022302e-16,
        -8.32667268e-17, -2.77555756e-17, -1.11022302e-16,
         0.00000000e+00,  1.11022302e-16,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  8.32667268e-17,
         0.00000000e+00,  0.00000000e+00,  8.32667268e-17,
         0.00000000e+00,  1.11022302e-16,  5.55111512e-17,
         1.11022302e-16,  0.00000000e+00,  5.55111512e-17,
        -2.77555756e-17,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
        -2.22044605e-16, -1.11022302e-16,  0.00000000e+00,
         2.22044605e-16, -2.22044605e-16,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00, -2.77555756e-17,
         8.32667268e-17,  2.22044605e-16,  2.22044605e-16,
         0.00000000e+00,  2.22044605e-16,  0.00000000e+00,
         0.00000000e+00,  5.55111512e-17,  0.00000000e+00,
         1.11022302e-16,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00, -2.22044605e-16,  4.44089210e-16,
         0.00000000e+00,  1.11022302e-16, -9.71445147e-17,
        -1.11022302e-16,  1.38777878e-17, -1.11022302e-16,
         0.00000000e+00,  0.00000000e+00, -8.32667268e-17,
        -5.55111512e-17, -2.22044605e-16,  0.00000000e+00,
         2.22044605e-16,  0.00000000e+00, -1.11022302e-16,
         1.11022302e-16,  1.11022302e-16,  0.00000000e+00,
        -5.55111512e-17,  0.00000000e+00,  0.00000000e+00,
        -1.11022302e-16, -4.44089210e-16,  2.22044605e-16,
         0.00000000e+00, -1.11022302e-16,  2.22044605e-16,
         0.00000000e+00, -9.02056208e-17,  0.00000000e+00,
         0.00000000e+00, -4.44089210e-16,  4.85722573e-17,
         0.00000000e+00,  0.00000000e+00, -1.11022302e-16,
        -5.55111512e-17, -4.16333634e-17,  0.00000000e+00,
         1.11022302e-16, -1.11022302e-16,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  4.44089210e-16,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         4.44089210e-16,  2.22044605e-16, -2.22044605e-16,
        -2.22044605e-16,  2.77555756e-17,  9.36750677e-17,
        -5.55111512e-17,  0.00000000e+00,  0.00000000e+00,
         8.32667268e-17,  8.32667268e-17,  0.00000000e+00,
         0.00000000e+00,  2.77555756e-17,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  2.22044605e-16,  0.00000000e+00,
         1.11022302e-16,  1.11022302e-16,  1.11022302e-16,
        -4.16333634e-17,  2.77555756e-17,  2.22044605e-16,
        -5.55111512e-17,  0.00000000e+00, -1.11022302e-16,
        -2.22044605e-16,  0.00000000e+00,  1.11022302e-16,
         0.00000000e+00,  1.11022302e-16, -5.55111512e-17,
         2.22044605e-16,  0.00000000e+00, -1.11022302e-16,
        -8.32667268e-17, -4.44089210e-16,  0.00000000e+00,
```

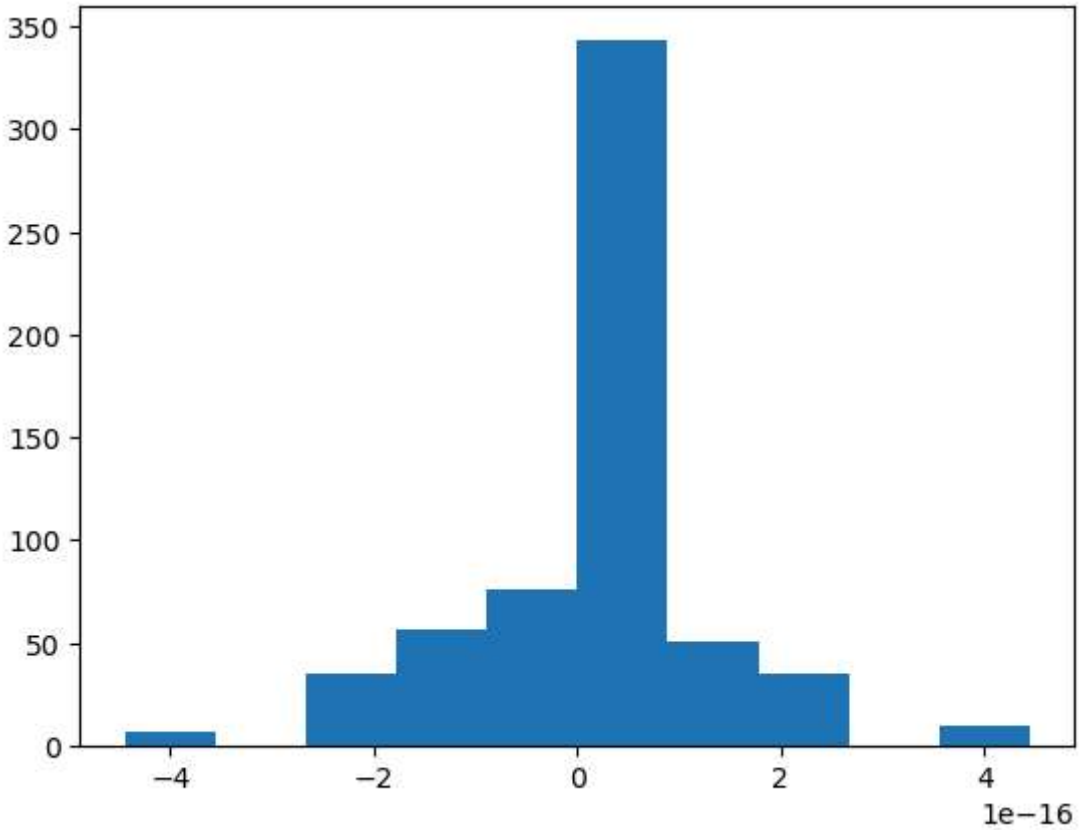
```
 0.00000000e+00,  0.00000000e+00,  1.11022302e-16,
 0.00000000e+00,  1.11022302e-16,  6.93889390e-18,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  1.11022302e-16,
 5.55111512e-17,  0.00000000e+00, -5.55111512e-17,
 0.00000000e+00,  2.77555756e-17,  0.00000000e+00,
 0.00000000e+00, -8.32667268e-17, -2.22044605e-16,
-2.22044605e-16,  8.67361738e-17,  0.00000000e+00,
 2.22044605e-16, -9.71445147e-17,  0.00000000e+00,
-1.01915004e-16,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  5.55111512e-17,
 2.77555756e-17, -1.11022302e-16,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  2.77555756e-17,
 1.11022302e-16,  0.00000000e+00, -2.22044605e-16,
 0.00000000e+00,  0.00000000e+00,  2.22044605e-16,
 5.55111512e-17, -8.32667268e-17,  0.00000000e+00,
 5.55111512e-17, -8.32667268e-17,  0.00000000e+00,
 5.55111512e-17,  4.44089210e-16,  0.00000000e+00,
-2.22044605e-16,  0.00000000e+00,  0.00000000e+00,
-1.11022302e-16,  0.00000000e+00, -4.85722573e-17,
 1.38777878e-17,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00, -2.22044605e-16,  0.00000000e+00,
-4.16333634e-17,  0.00000000e+00,  8.32667268e-17,
 0.00000000e+00,  0.00000000e+00, -1.11022302e-16,
 0.00000000e+00,  0.00000000e+00, -1.11022302e-16,
-2.22044605e-16,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00, -5.55111512e-17,
 0.00000000e+00, -8.32667268e-17,  0.00000000e+00,
 0.00000000e+00,  6.93889390e-17,  5.55111512e-17,
 8.32667268e-17, -4.16333634e-17, -2.22044605e-16,
 0.00000000e+00,  5.55111512e-17,  1.11022302e-16,
-8.32667268e-17,  0.00000000e+00,  5.55111512e-17,
 1.11022302e-16,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  1.11022302e-16,  2.22044605e-16,
 0.00000000e+00,  4.44089210e-16,  0.00000000e+00,
-2.77555756e-17, -5.55111512e-17,  0.00000000e+00,
 0.00000000e+00, -4.44089210e-16,  8.32667268e-17,
-5.55111512e-17, -1.11022302e-16,  0.00000000e+00,
-2.22044605e-16, -2.22044605e-16,  8.32667268e-17,
-5.55111512e-17,  0.00000000e+00,  0.00000000e+00,
-5.55111512e-17,  1.11022302e-16,  2.77555756e-17,
-5.55111512e-17,  0.00000000e+00,  0.00000000e+00,
-1.11022302e-16,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00, -2.22044605e-16,
 0.00000000e+00,  0.00000000e+00,  1.11022302e-16,
 0.00000000e+00,  5.55111512e-17,  0.00000000e+00,
 0.00000000e+00, -1.11022302e-16,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00, -2.22044605e-16,
 0.00000000e+00,  8.67361738e-17,  5.55111512e-17,
 0.00000000e+00, -1.11022302e-16, -5.89805982e-17,
 0.00000000e+00,  0.00000000e+00, -9.71445147e-17,
-8.32667268e-17,  0.00000000e+00,  0.00000000e+00,
-5.55111512e-17, -1.11022302e-16,  2.22044605e-16,
-5.55111512e-17, -1.11022302e-16,  0.00000000e+00,
 1.10154941e-16,  0.00000000e+00, -2.22044605e-16,
 0.00000000e+00, -1.11022302e-16,  0.00000000e+00,
 0.00000000e+00,  1.11022302e-16, -5.55111512e-17,
 1.11022302e-16,  1.11022302e-16, -5.55111512e-17,
-1.11022302e-16,  0.00000000e+00, -9.02056208e-17,
-2.36356074e-17, -8.32667268e-17, -1.11022302e-16,
```

```
 0.00000000e+00, -1.11022302e-16,  0.00000000e+00,
-8.32667268e-17,  1.11022302e-16, -1.11022302e-16,
 0.00000000e+00, -2.22044605e-16,  5.55111512e-17,
 0.00000000e+00, -2.22044605e-16, -5.20417043e-17,
 0.00000000e+00,  8.32667268e-17,  1.11022302e-16,
 4.44089210e-16,  1.11022302e-16,  0.00000000e+00,
 5.55111512e-17,  2.22044605e-16,  2.22044605e-16,
 2.22044605e-16,  0.00000000e+00,  2.77555756e-17,
 0.00000000e+00,  1.11022302e-16, -2.22044605e-16,
-5.55111512e-17,  0.00000000e+00, -2.77555756e-17,
 0.00000000e+00, -1.11022302e-16,  1.38777878e-17,
 2.22044605e-16,  1.11022302e-16, -4.16333634e-17,
 2.22044605e-16,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  8.32667268e-17, -5.55111512e-17,
 0.00000000e+00, -1.11022302e-16,  1.11022302e-16,
 0.00000000e+00,  0.00000000e+00, -5.20417043e-17,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 1.01915004e-17,  4.44089210e-16, -2.22044605e-16,
 5.55111512e-17, -4.44089210e-16,  1.11022302e-16,
-2.22044605e-16,  1.11022302e-16,  0.00000000e+00,
 0.00000000e+00,  2.77555756e-17,  2.77555756e-17,
-7.28583860e-17,  8.32667268e-17,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 5.55111512e-17, -2.22044605e-16,  2.22044605e-16,
-1.11022302e-16, -1.11022302e-16,  0.00000000e+00,
 0.00000000e+00,  8.32667268e-17,  9.71445147e-17,
-1.38777878e-17,  0.00000000e+00, -2.22044605e-16,
 0.00000000e+00, -5.55111512e-17,  0.00000000e+00,
 0.00000000e+00, -1.11022302e-16,  0.00000000e+00,
-5.55111512e-17,  0.00000000e+00,  2.22044605e-16,
-1.11022302e-16,  0.00000000e+00,  1.11022302e-16,
 0.00000000e+00, -1.11022302e-16,  1.11022302e-16,
 2.77555756e-17, -1.11022302e-16, -1.11022302e-16,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
-2.22044605e-16,  2.22044605e-16, -2.77555756e-17,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  4.44089210e-16,
-5.55111512e-17, -8.32667268e-17,  0.00000000e+00,
 8.32667268e-17,  0.00000000e+00, -5.55111512e-17,
-1.11022302e-16,  0.00000000e+00, -2.22044605e-16,
 0.00000000e+00,  0.00000000e+00, -1.11022302e-16,
 2.77555756e-17,  0.00000000e+00, -1.11022302e-16,
 0.00000000e+00,  4.16333634e-17,  0.00000000e+00,
 1.11022302e-16, -2.22044605e-16,  0.00000000e+00,
 2.22044605e-16,  0.00000000e+00, -1.38777878e-17,
 0.00000000e+00,  0.00000000e+00, -5.55111512e-17,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 2.60208521e-17,  0.00000000e+00, -4.44089210e-16,
-1.11022302e-16,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00, -5.55111512e-17, -4.16333634e-17,
 1.11022302e-16,  2.22044605e-16,  5.55111512e-17,
 0.00000000e+00,  8.32667268e-17,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 3.12250226e-17,  0.00000000e+00,  1.38777878e-17,
 0.00000000e+00, -2.22044605e-16,  0.00000000e+00,
 2.22044605e-16,  1.11022302e-16,  1.04517089e-16,
 0.00000000e+00, -6.93889390e-17,  0.00000000e+00,
 0.00000000e+00,  6.93889390e-17, -2.22044605e-16,
 0.00000000e+00, -8.32667268e-17, -5.55111512e-17,
 0.00000000e+00,  0.00000000e+00,  7.97972799e-17,
```

```
 0.00000000e+00,  2.22044605e-16,  0.00000000e+00,
 0.00000000e+00,  1.11022302e-16,  2.22044605e-16,
-1.11022302e-16, -5.55111512e-17,  8.32667268e-17,
 0.00000000e+00, -1.11022302e-16,  0.00000000e+00,
-1.11022302e-16,  0.00000000e+00,  0.00000000e+00,
-2.77555756e-17,  4.44089210e-16,  2.22044605e-16,
-1.11022302e-16,  0.00000000e+00, -5.55111512e-17,
 0.00000000e+00, -1.11022302e-16,  0.00000000e+00,
 0.00000000e+00,  3.46944695e-18, -2.77555756e-17,
-5.55111512e-17,  5.55111512e-17,  5.55111512e-17,
 2.08166817e-17,  0.00000000e+00,  0.00000000e+00,
-5.55111512e-17, -5.55111512e-17, -2.77555756e-17,
-2.77555756e-17,  0.00000000e+00, -1.11022302e-16,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
-2.22044605e-16,  1.11022302e-16, -1.11022302e-16,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  5.55111512e-17,  0.00000000e+00,
-4.44089210e-16,  9.02056208e-17,  5.55111512e-17,
 0.00000000e+00,  4.44089210e-16,  5.55111512e-17,
-5.55111512e-17,  0.00000000e+00,  4.16333634e-17,
 2.22044605e-16,  1.11022302e-16,  2.22044605e-16,
 0.00000000e+00,  1.11022302e-16,  0.00000000e+00,
 0.00000000e+00, -2.22044605e-16,  0.00000000e+00,
 1.11022302e-16,  0.00000000e+00, -5.55111512e-17,
 2.22044605e-16, -5.55111512e-17]])
```

In [45]:
```
plt.hist(pt_loan[0])
plt.show
```

Out[45]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [ ]: