In [3]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

In [4]:
```
df=pd.read_csv("loan.csv") #read
print(df)
```

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

     ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  
0               5849                0.0         NaN             360.0   \
1               4583             1508.0       128.0             360.0   
2               3000                0.0        66.0             360.0   
3               2583             2358.0       120.0             360.0   
4               6000                0.0       141.0             360.0   
```

In [4]:
```
df.describe() #Display descriptive statistics on the dataset
```

Out[4]:

	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 [5]:
```
df.head() #returns the top five rows by default
```

Out[5]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coappli
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 [7]:
```
df[df.isna().any(axis=1)] #Check if any records in the data have any missing va
```

Out[7]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coap
0	LP001002	Male	No	0	Graduate	No	5849	
11	LP001027	Male	Yes	2	Graduate	NaN	2500	
16	LP001034	Male	No	1	Not Graduate	No	3596	
19	LP001041	Male	Yes	0	Graduate	NaN	2600	
23	LP001050	NaN	Yes	2	Not Graduate	No	3365	
...	...	...	...	...	...	...	...	
592	LP002933	NaN	No	3+	Graduate	Yes	9357	
597	LP002943	Male	No	NaN	Graduate	No	2987	
600	LP002949	Female	No	3+	Graduate	NaN	416	
601	LP002950	Male	Yes	0	Not Graduate	NaN	2894	
605	LP002960	Male	Yes	0	Not Graduate	No	2400	

134 rows × 13 columns

In [8]:
```
df.dropna(inplace=True)
```

In [9]:
```
df[df.isna().any(axis=1)]
```

Out[9]:

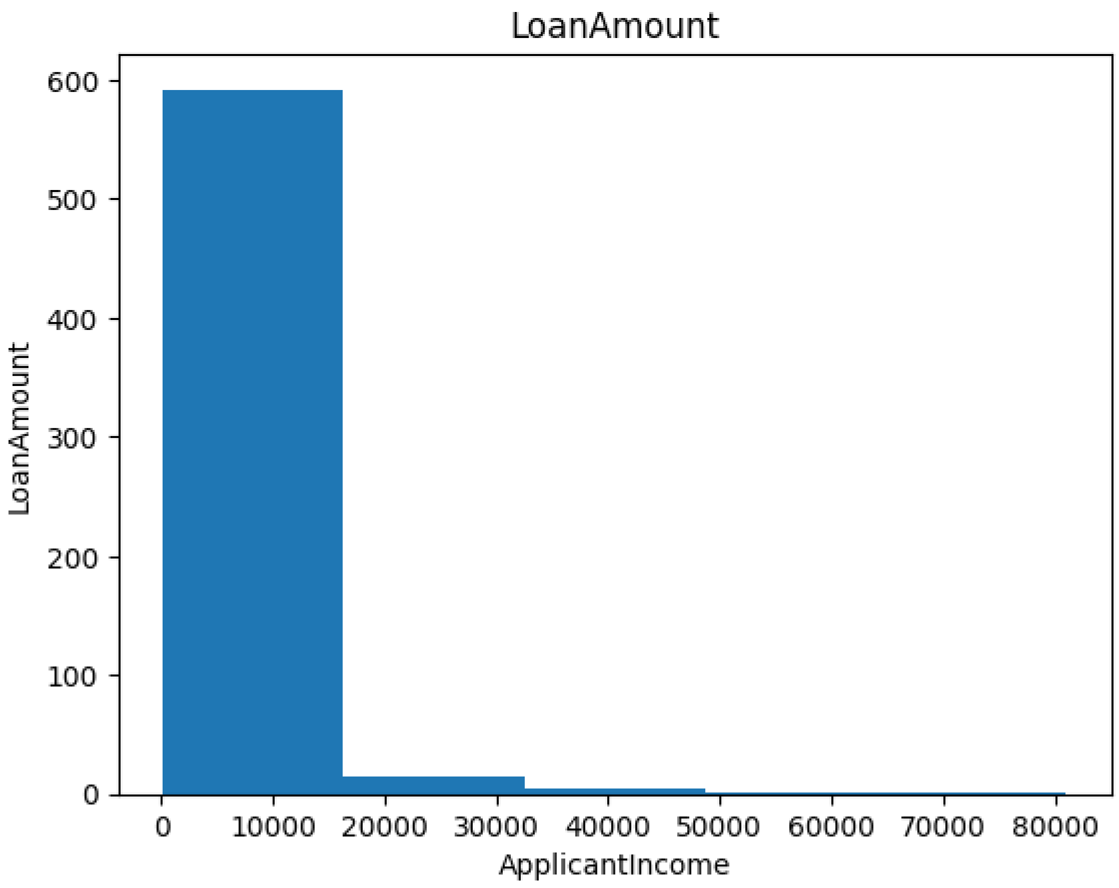
Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica

In [10]:
```
pip install matplotlib
```

```
Collecting matplotlibNote: you may need to restart the kernel to use updated
packages.

  Downloading matplotlib-3.7.1-cp311-cp311-win_amd64.whl (7.6 MB)
                                             0.0/7.6 MB ? eta -:--:--
                                             0.2/7.6 MB 5.3 MB/s eta 0:00:02
     --                                      0.4/7.6 MB 3.9 MB/s eta 0:00:02
     ---                                     0.7/7.6 MB 4.5 MB/s eta 0:00:02
     -----                                   1.0/7.6 MB 5.0 MB/s eta 0:00:02
     -------                                 1.4/7.6 MB 5.4 MB/s eta 0:00:02
     -------                                 1.5/7.6 MB 5.1 MB/s eta 0:00:02
     --------                                1.7/7.6 MB 4.9 MB/s eta 0:00:02
     --------                                1.7/7.6 MB 4.9 MB/s eta 0:00:02
     --------                                1.7/7.6 MB 4.9 MB/s eta 0:00:02
     --------                                1.7/7.6 MB 3.8 MB/s eta 0:00:02
     -----------                             2.2/7.6 MB 3.9 MB/s eta 0:00:02
     ------------                            2.4/7.6 MB 4.0 MB/s eta 0:00:02
     -------------                           2.5/7.6 MB 4.1 MB/s eta 0:00:02
     --------------                          2.7/7.6 MB 4.0 MB/s eta 0:00:02
     ---------------                         2.9/7.6 MB 4.0 MB/s eta 0:00:02
     ----------------                        3.1/7.6 MB 4.0 MB/s eta 0:00:02
     -----------------                       3.3/7.6 MB 4.0 MB/s eta 0:00:02
     ------------------                      3.5/7.6 MB 4.0 MB/s eta 0:00:02
     -------------------                     3.6/7.6 MB 3.9 MB/s eta 0:00:02
     --------------------                    3.8/7.6 MB 4.0 MB/s eta 0:00:01
     --------------------                    4.0/7.6 MB 3.9 MB/s eta 0:00:01
     ---------------------                   4.2/7.6 MB 4.0 MB/s eta 0:00:01
     ----------------------                  4.4/7.6 MB 4.0 MB/s eta 0:00:01
     -----------------------                 4.5/7.6 MB 4.0 MB/s eta 0:00:01
     -------------------------               4.8/7.6 MB 4.0 MB/s eta 0:00:01
     -------------------------               5.0/7.6 MB 4.0 MB/s eta 0:00:01
     --------------------------              5.1/7.6 MB 3.9 MB/s eta 0:00:01
     ---------------------------             5.3/7.6 MB 3.9 MB/s eta 0:00:01
     ----------------------------            5.5/7.6 MB 3.9 MB/s eta 0:00:01
     -----------------------------           5.7/7.6 MB 3.9 MB/s eta 0:00:01
     ------------------------------          5.9/7.6 MB 3.9 MB/s eta 0:00:01
     -------------------------------         6.0/7.6 MB 3.9 MB/s eta 0:00:01
     --------------------------------        6.2/7.6 MB 3.9 MB/s eta 0:00:01
     ---------------------------------       6.4/7.6 MB 3.9 MB/s eta 0:00:01
     ----------------------------------      6.6/7.6 MB 3.9 MB/s eta 0:00:01
     -----------------------------------     6.8/7.6 MB 3.9 MB/s eta 0:00:01
     ------------------------------------    7.0/7.6 MB 3.9 MB/s eta 0:00:01
     -------------------------------------   7.2/7.6 MB 3.9 MB/s eta 0:00:01
     --------------------------------------  7.4/7.6 MB 3.9 MB/s eta 0:00:01
     --------------------------------------  7.5/7.6 MB 3.9 MB/s eta 0:00:01
     --------------------------------------  7.6/7.6 MB 3.9 MB/s eta 0:00:01
     --------------------------------------  7.6/7.6 MB 3.9 MB/s eta 0:00:01
     --------------------------------------  7.6/7.6 MB 3.9 MB/s eta 0:00:01
     --------------------------------------  7.6/7.6 MB 3.9 MB/s eta 0:00:01
     --------------------------------------  7.6/7.6 MB 3.9 MB/s eta 0:00:01
     ---------------------------------------- 7.6/7.6 MB 3.5 MB/s eta 0:00:00
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.1.0-cp311-cp311-win_amd64.whl (470 kB)
                                             0.0/470.9 kB ? eta -:--:--
     -----------------------               297.0/470.9 kB 6.1 MB/s eta 0:00:
01
     -------------------------------------  460.8/470.9 kB 5.8 MB/s eta 0:00:
```

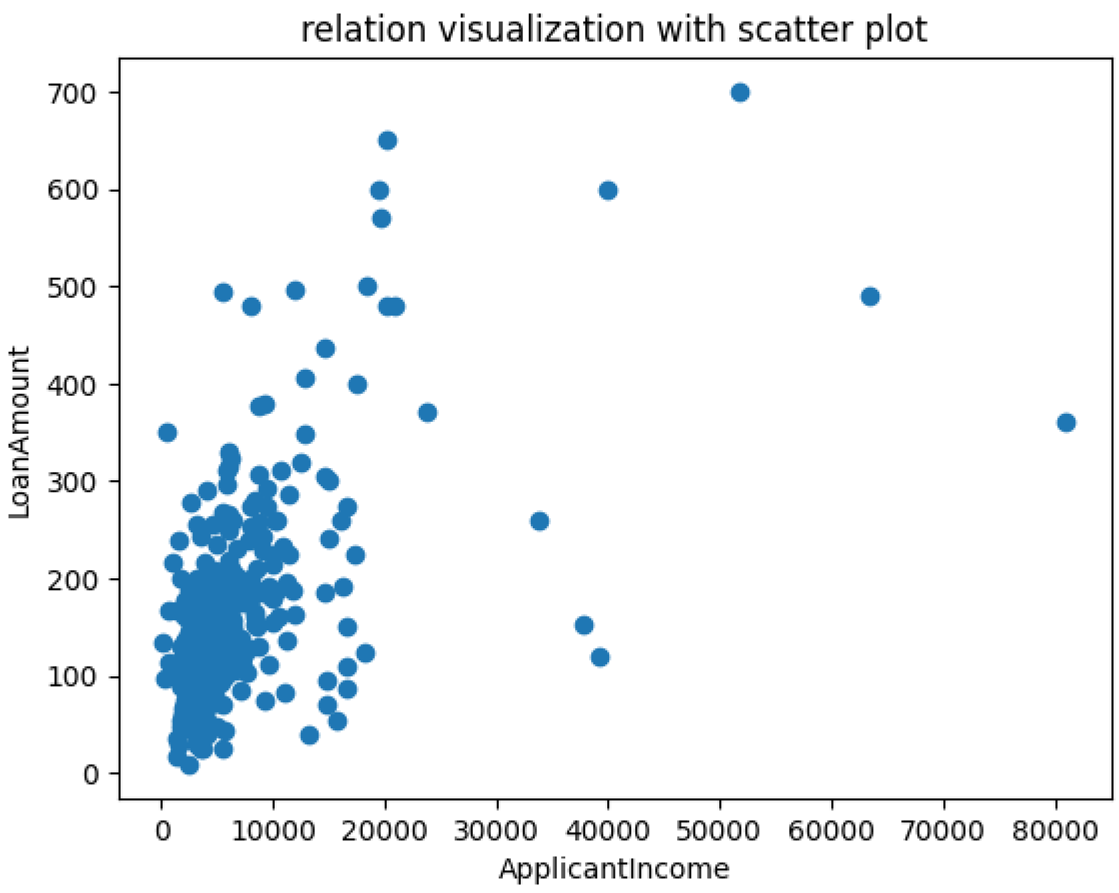
```
01
     -------------------------------------- 470.9/470.9 kB 3.7 MB/s eta 0:00:
00
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.11.0-py3-none-any.whl (6.4 kB)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\user\appdata\loc
al\programs\python\python311\lib\site-packages (from matplotlib) (4.39.4)
Collecting kiwisolver>=1.0.1 (from matplotlib)
  Downloading kiwisolver-1.4.4-cp311-cp311-win_amd64.whl (55 kB)
                                             0.0/55.4 kB ? eta -:--:--
     ------------------------------------     51.2/55.4 kB 2.6 MB/s eta 0:00:
01
     -------------------------------------- 55.4/55.4 kB 960.7 kB/s eta 0:00:
00
Requirement already satisfied: numpy>=1.20 in c:\users\user\appdata\local\pro
grams\python\python311\lib\site-packages (from matplotlib) (1.24.3)
Requirement already satisfied: packaging>=20.0 in c:\users\user\appdata\local
\programs\python\python311\lib\site-packages (from matplotlib) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\user\appdata\local\p
rograms\python\python311\lib\site-packages (from matplotlib) (9.5.0)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
                                             0.0/98.3 kB ? eta -:--:--
     -------------------------------------    92.2/98.3 kB 5.1 MB/s eta 0:00:
01
     ---------------------------------------- 98.3/98.3 kB 1.4 MB/s eta 0:00:
00
Requirement already satisfied: python-dateutil>=2.7 in c:\users\user\appdata
\local\programs\python\python311\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\user\appdata\local\progra
ms\python\python311\lib\site-packages (from python-dateutil>=2.7->matplotlib)
(1.16.0)
Installing collected packages: pyparsing, kiwisolver, cycler, contourpy, matp
lotlib
Successfully installed contourpy-1.1.0 cycler-0.11.0 kiwisolver-1.4.4 matplot
lib-3.7.1 pyparsing-3.0.9
```

In [13]:
```
import matplotlib.pyplot as plt #Build a graph visualizing the distribution of
import pandas as pd
df=pd.read_csv("loan.csv")
plt.hist(df['ApplicantIncome'], bins=5)
plt.xlabel('ApplicantIncome')
plt.ylabel('LoanAmount')
plt.title('LoanAmount')
plt.show()
```



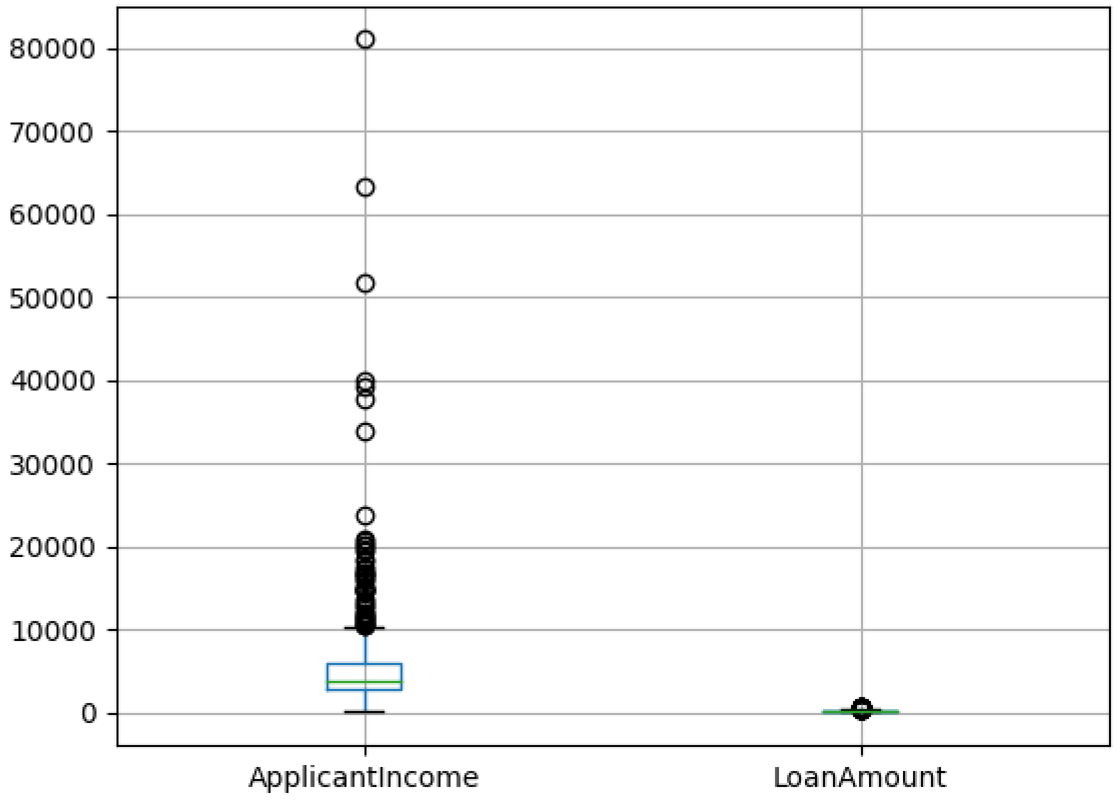
In [15]:
```
x=df["ApplicantIncome"].to_numpy()
y=df["LoanAmount"].to_numpy()
df1=plt.scatter(x,y)
plt.title("relation visualization with scatter plot")
plt.xlabel("ApplicantIncome")
plt.ylabel("LoanAmount")
```

Out[15]:
```
Text(0, 0.5, 'LoanAmount')
```

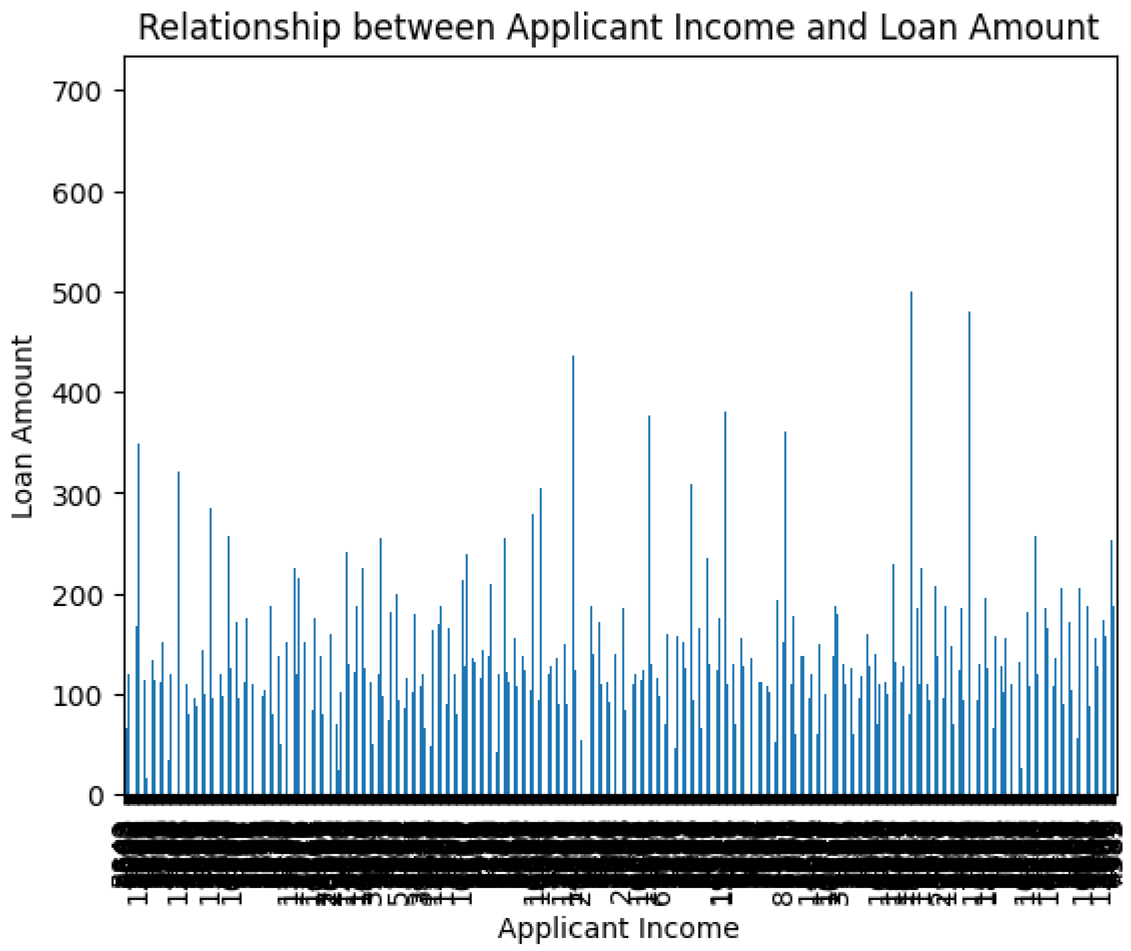


```
In [19]: df.boxplot(column=['ApplicantIncome', 'LoanAmount']) #Build a graph visualizin
```

```
Out[19]: <Axes: >
```



In [6]:
```
df.plot(kind='bar', x='ApplicantIncome', y='LoanAmount', legend=False) #withou
plt.xlabel('Applicant Income')
plt.ylabel('Loan Amount')
plt.title('Relationship between Applicant Income and Loan Amount')
plt.show()
```



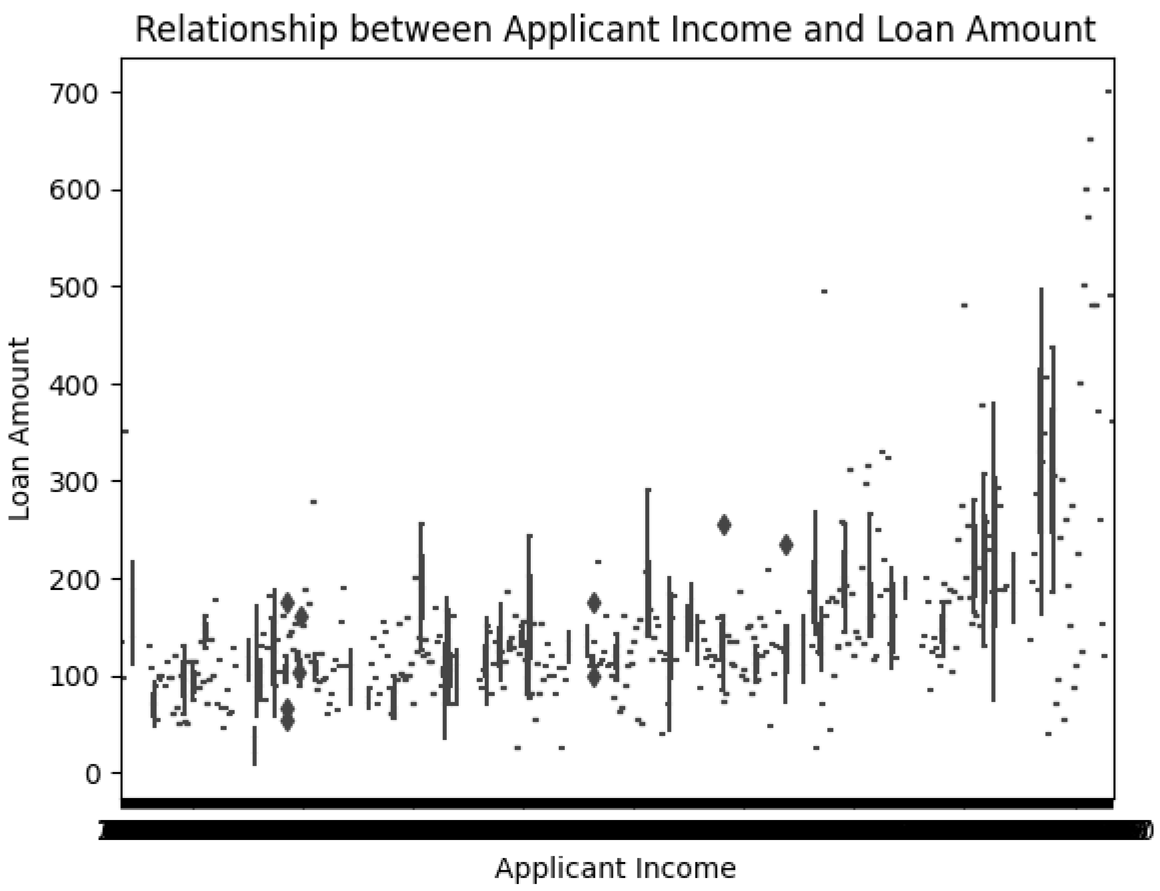
In [5]:
```
import seaborn as sns
```

In [16]:
```
pip install seaborn
```

```
Collecting seabornNote: you may need to restart the kernel to use updated pac
kages.

  Downloading seaborn-0.12.2-py3-none-any.whl (293 kB)
                                              0.0/293.3 kB ? eta -:--:--
     --------------------------            204.8/293.3 kB 6.3 MB/s eta 0:00:
01
     -------------------------------------  286.7/293.3 kB 3.5 MB/s eta 0:00:
01
     -------------------------------------- 293.3/293.3 kB 2.0 MB/s eta 0:00:
00
Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\users\user\appdata
\local\programs\python\python311\lib\site-packages (from seaborn) (1.24.3)
Requirement already satisfied: pandas>=0.25 in c:\users\user\appdata\local\pr
ograms\python\python311\lib\site-packages (from seaborn) (2.0.1)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\user\appda
ta\local\programs\python\python311\lib\site-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\user\appdata\loca
l\programs\python\python311\lib\site-packages (from matplotlib!=3.6.1,>=3.1->
seaborn) (1.1.0)
Requirement already satisfied: cycler>=0.10 in c:\users\user\appdata\local\pr
ograms\python\python311\lib\site-packages (from matplotlib!=3.6.1,>=3.1->seab
orn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\user\appdata\loc
al\programs\python\python311\lib\site-packages (from matplotlib!=3.6.1,>=3.1-
>seaborn) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\user\appdata\loc
al\programs\python\python311\lib\site-packages (from matplotlib!=3.6.1,>=3.1-
>seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\user\appdata\local
\programs\python\python311\lib\site-packages (from matplotlib!=3.6.1,>=3.1->s
eaborn) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\user\appdata\local\p
rograms\python\python311\lib\site-packages (from matplotlib!=3.6.1,>=3.1->sea
born) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\user\appdata\loca
l\programs\python\python311\lib\site-packages (from matplotlib!=3.6.1,>=3.1->
seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\user\appdata
\local\programs\python\python311\lib\site-packages (from matplotlib!=3.6.1,>=
3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\user\appdata\local\pr
ograms\python\python311\lib\site-packages (from pandas>=0.25->seaborn) (2023.
3)
Requirement already satisfied: tzdata>=2022.1 in c:\users\user\appdata\local
\programs\python\python311\lib\site-packages (from pandas>=0.25->seaborn) (20
23.3)
Requirement already satisfied: six>=1.5 in c:\users\user\appdata\local\progra
ms\python\python311\lib\site-packages (from python-dateutil>=2.7->matplotlib!
=3.6.1,>=3.1->seaborn) (1.16.0)
Installing collected packages: seaborn
Successfully installed seaborn-0.12.2
```

In [17]:
```
import seaborn as sns
```

In [18]:
```
sns.boxplot(x=df['ApplicantIncome'], y=df['LoanAmount']) #using seaborn
plt.xlabel('Applicant Income')
plt.ylabel('Loan Amount')
plt.title('Relationship between Applicant Income and Loan Amount')
plt.show()
```



In [19]:
```
unique_values = df['Property_Area'].unique() #Display unique values of a catego
print(unique_values)
```

```
['Urban' 'Rural' 'Semiurban']
```

In [20]:
```
unique_values = df['Education'].unique() #Display unique values of a categorica
print(unique_values)
```

```
['Graduate' 'Not Graduate']
```

In [24]:
```
pip install scipy
```

```
Collecting scipy
  Downloading scipy-1.10.1-cp311-cp311-win_amd64.whl (42.2 MB)
                                             0.0/42.2 MB ? eta -:--:--
                                             0.1/42.2 MB 3.2 MB/s eta 0:0
0:14
                                             0.2/42.2 MB 2.9 MB/s eta 0:0
0:15
                                             0.3/42.2 MB 2.6 MB/s eta 0:0
0:17
                                             0.3/42.2 MB 1.8 MB/s eta 0:0
0:23
                                             0.5/42.2 MB 2.2 MB/s eta 0:0
0:19
                                             0.6/42.2 MB 2.3 MB/s eta 0:0
0:19
                                             0.7/42.2 MB 2.1 MB/s eta 0:0
0:20
                                             0.7/42.2 MB 2.0 MB/s eta 0:0
0:21
```

In [22]:
```
import scipy.stats as stats
```

In [24]:
```
# Build a contingency table of two potentially related categorical variables. (
contingency_table = pd.crosstab(df['Property_Area'], df['Self_Employed'])
print(contingency_table)

# Perform the chi-square test of independence
chi2, p_value, _, _ = stats.chi2_contingency(contingency_table)
print('Chi-square statistic:', chi2)
print('p-value:', p_value)
```

```
Self_Employed   No  Yes
Property_Area
Rural          143   26
Semiurban      191   32
Urban          166   24
Chi-square statistic: 0.5803134707599138
p-value: 0.7481462973944264
```

In [25]:
```
# Retrieve one or more subset of rows based on two or more criteria and presen
subset1 = df[(df['Gender'] == 'Male') & (df['Married'] == 'Yes')]
subset2 = df[(df['Loan_Status'] == 'N') & (df['Property_Area'] == 'Rural')]

# Present descriptive statistics on the subset(s)
subset1_stats = subset1.describe()
subset2_stats = subset2.describe()

print("Subset 1 descriptive statistics:")
print(subset1_stats)

print("\nSubset 2 descriptive statistics:")
print(subset2_stats)
```

```
Subset 1 descriptive statistics:
       ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  
count       357.000000         357.000000  343.000000        348.000000   \
mean       5529.540616        1828.330308  154.011662        335.931034   
std        6743.209021        2096.367198   83.025254         67.342095   
min         150.000000           0.000000   17.000000         12.000000   
25%        2882.000000           0.000000  108.000000        360.000000   
50%        3875.000000        1619.000000  132.000000        360.000000   
75%        5829.000000        2500.000000  180.000000        360.000000   
max       81000.000000       20000.000000  600.000000        480.000000   

       Credit_History  
count      326.000000  
mean         0.846626  
std          0.360902  
min          0.000000  
25%          1.000000  
50%          1.000000  
75%          1.000000  
max          1.000000  

Subset 2 descriptive statistics:
       ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  
count        69.000000          69.000000   66.000000         67.000000   \
mean       6497.521739        1436.507246  158.742424        345.134328   
std        9949.733021        1618.225813   96.412305         56.751691   
min         150.000000           0.000000   46.000000         84.000000   
25%        3365.000000           0.000000  110.500000        360.000000   
50%        4283.000000        1287.000000  133.000000        360.000000   
75%        6216.000000        2200.000000  172.750000        360.000000   
max       81000.000000        5302.000000  570.000000        480.000000   

       Credit_History  
count        65.00000  
mean          0.60000  
std           0.49371  
min           0.00000  
25%           0.00000  
50%           1.00000  
75%           1.00000  
max           1.00000  
```

In [1]:
```
pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\user\appdata\local\pr
ograms\python\python311\lib\site-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in c:\users\user\appdata\local\p
rograms\python\python311\lib\site-packages (from scikit-learn) (1.24.3)
Requirement already satisfied: scipy>=1.3.2 in c:\users\user\appdata\local\pr
ograms\python\python311\lib\site-packages (from scikit-learn) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in c:\users\user\appdata\local\p
rograms\python\python311\lib\site-packages (from scikit-learn) (1.3.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\user\appdata
\local\programs\python\python311\lib\site-packages (from scikit-learn) (3.1.
0)
Note: you may need to restart the kernel to use updated packages.
```

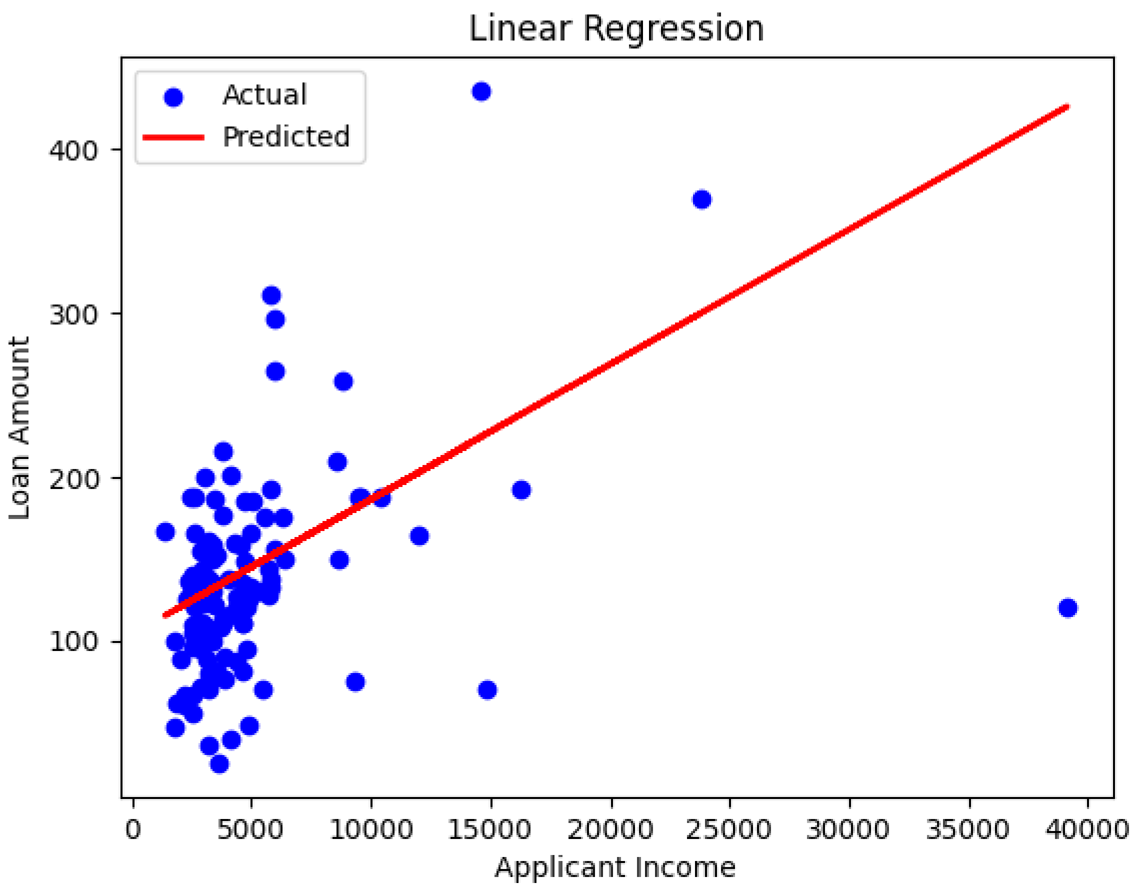
In [1]:
```
import numpy as np     #Conduct a statistical test of the significance of the
import scipy.stats as stats

# Define the two subsets of data
data_a = np.array([5849, 4583, 3000, 2583, 6000, 5417, 2333, 3036, 4006])
data_b = np.array([0, 128, 66, 120, 141, 267, 95, 158, 168])

# Conduct the independent t-test
t_statistic, p_value = stats.ttest_ind(data_a, data_b)

# Print the results
print("t-statistic:", t_statistic)
print("p-value:", p_value)
```

```
t-statistic: 8.289381635365402
p-value: 3.486682839706522e-07
```

In [9]:
```
#Create one or more tables that group the data by a certain categorical variab

# Group the data by 'Gender' and calculate the mean and sum for selected colum
grouped_gender = df.groupby('Gender').agg({'ApplicantIncome': 'mean', 'Coappli

# Group the data by 'Education' and calculate the mean and sum for selected co
grouped_education = df.groupby('Education').agg({'ApplicantIncome': 'mean', 'C

# Print the grouped data for 'Gender'
print("Grouped Data by Gender:")
print(grouped_gender)
```

```
Grouped Data by Gender:
        ApplicantIncome  CoapplicantIncome  LoanAmount
Gender
Female      4593.954128        1138.504587     13810.0
Male        5455.895745        1764.442383     70155.0
```

In [7]:
```
from sklearn.model_selection import train_test_split    #Implement a linear re
from sklearn.linear_model import LinearRegression
# Remove rows with missing values
df.dropna(subset=['LoanAmount'], inplace=True)

# Split the data into training and testing sets
X = df[['ApplicantIncome']]
y = df['LoanAmount']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rando

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Plot the actual vs. predicted values
plt.scatter(X_test, y_test, color='blue', label='Actual')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted')
plt.xlabel('Applicant Income')
plt.ylabel('Loan Amount')
plt.title('Linear Regression')
plt.legend()
plt.show()

# Print the model coefficients and intercept
print('Coefficients:', model.coef_)
print('Intercept:', model.intercept_)
```



```
Coefficients: [0.00823739]
Intercept: 103.70294803264798
```

```
In [ ]:
```