



SCHOOL OF COMPUTING
COLLEGE OF ARTS AND SCIENCES
SKIH2103 DATA ANALYTICS

ASSIGNMENT 1

NAME	MATRIC NO
NURIN ANDRIANA BINTI MOHAMAD SUBRI	287957
NURIN IZZAH BINTI ISHAK	288063
AINUR HANIM BINTI ABDUL HALIM	288091

1. Briefly describe the dataset you worked on, including its source and any interesting characteristics.

The Home Loan Dataset is a comprehensive collection of information pertaining to the status of home loan applications. It encompasses various data points related to the loan application process, borrower details, loan approval status, and documentation status. This dataset has been curated to provide insights into the factors that influence loan approvals and the challenges encountered during the application process. It serves as a valuable resource for researchers, analysts and financial institutions interested in studying and improving the efficiency and effectiveness of the home loan application process.

The dataset contains demographic information such as gender, marital status, dependents, education level, property area as well as financial information such as applicant income, co-applicant income, and loan amount where there is also loan amount term that the applicant need to alert. Also, the dataset contains a binary variable, which is the attribute for self-employed, credit history and loan status. Primarily, “Yes” or “1” indicates that we agree with the attribute statement, and “No” or “0” indicates that the statement is opposite to what we wanted. Additionally, each variable provides valuable insights into the Home Loan applicant’s situation, such as their eligibility to make a house loan based on their income and the co-applicant income.

The dataset includes:

- **Loan_ID:** It typically categorizes a unique identifier assigned to each individual loan application.
- **Gender:** Gender refers to the gender of the loan applicant. It is categorized as a binary variable, with “Male” and “Female” being the most common options.
- **Married:** Married attribute refers to the Marital Status of the loan applicant. It categorizes the applicant’s current marital situation, providing insights into their relationship status at the time of applying for the loan.
- **Dependents:** Dependents refers to the number of dependents that the loan applicant has. Dependents are individuals who rely on the loan applicant for financial support, such as children, elderly parents, or other family members.
- **Education:** Education refers to the highest level of education attained by the loan applicant. It categorizes the educational background of the applicant and provides insights into their level of academic achievement.
- **Self_Employed:** Self Employed refers to whether the borrower is self-employed or not. It is categorized as a binary variable, where “Yes” indicates that the borrower is self-employed, and “No” indicates that the borrower is not self-employed.
- **ApplicantIncome:** Applicant income typically refers to the income earned by the primary applicant of the loan. It represents the financial earnings of the individual who is applying for the loan.
- **CoapplicantIncome:** Co-applicant income attribute refers to the income earned by the co-applicant or secondary applicant of the loan. A co-applicant is a person who applies

for the loan along with the primary applicant and shares the financial responsibility and liability of the loan.

- **LoanAmount:** Loan Amount attribute typically refers to the total amount of money being requested by the loan applicant. It represents the principal amount that the applicant is seeking to borrow from the lender.
- **Loan_Amount_Term:** Loan amount term refers to the duration or term of the loan. It represents the length of time over which the loan amount is expected to be repaid by the borrower to the lender.
- **Credit_History:** Credit history refers to the borrower's past credit behavior and repayment history. It provides information about the borrower's creditworthiness and their track record in meeting their financial obligations, including loans and credit card payments.
- **Property_Area:** Property Area refers to the location or area where the property being financed is situated. It provides information about the geographical location or classification of the property.
- **Loan_Status:** Loan Status refers to the current status of the loan application. It provides information about whether the loan has been approved or rejected by the lender.

There are several interesting characteristics of Home Loan Status Dataset that can be gained from analyzing the data:

- **Loan Approval Factors:** By examining the dataset, we can identify the key factors that significantly influence loan approvals. This could include attributes such as credit history, income level, employment type, loan amount, loan term, property area, and more. Understanding these factors can help lenders make more informed decisions and applicants can focus on strengthening their loan application.
- **Demographic Patterns:** Analyzing the dataset can reveal interesting demographic patterns in loan applications and approvals. This could include trends in loan approval rates based on gender, marital status, education level or age group. Identifying any disparities or biases in loan approvals can lead to fairer lending practices.
- **Default Risk Analysis:** By examining historical loan data, it becomes possible to assess the risk of loan defaults. We can analyze the relationship between various attributes and the likelihood of borrowers defaulting on their loans. This analysis can aid lenders in making more accurate risk assessments and implementing appropriate risk management strategies.
- **Process Efficiency:** Examining the loan application process can highlight obstructions, delay times, or other inefficiencies that can be fixed. Lenders can improve client experience and speed up loan approvals by finding areas where the process might be streamlined or automated.

2. What data problems did you identify, and how did you solve them? What method did you use to handle missing data, outliers, and inconsistencies? Why did you choose these methods?

Our group was able to identify three problems from this dataset including:

a) Missing data

Refers to the lack of values or observations in the given dataset. This can occur due to various reasons such as incomplete surveys or data collection errors. Once missing values have been detected, they need to be handled properly. There are several ways to handle missing values with mean, median or mode, and using advanced imputation techniques such as multiple imputation. The choice of handling missing values depends on the specific situation and the goals of the analysis. We will explain in more detail in the next point.

b) Outliers

Refers to data points that are significantly different from the rest of the data. They are often abnormal observations that skew the data distribution. Outliers can be caused by measurement errors or true anomalies in the data. Once outliers have been detected, they need to be handled properly. There are several ways to handle outliers, including removing them from the dataset, transforming the data using a logarithmic or other transformation, and using robust statistical methods that are less affected by outliers. The choice of handling outliers depends on the specific situation and the goals of the analysis. We will explain in more detail in the next point.

c) Inconsistent data

Refers to data that has inconsistent values or formatting. There can also be points where a combination of feature values appear that violate the patterns generally observed in the training data. This can occur due to data entry errors or data integration issues. Once inconsistent data has been detected, it needs to be handled properly. This may involve correcting errors in the data, removing problematic observations or variables, or using advanced methods such as imputation to fill in missing values. The choice of how to handle inconsistent data depends on the specific situation and the goals of the analysis.

Attribute	Data Problem
Gender	Missing Value
Married	Missing Value
Dependents	Missing Value
Self_Employed	Missing Value
ApplicantIncome	Outlier
CoapplicantIncome	Outlier/ Inconsistent
LoanAmount	Outlier
Loan_Amount_Term	Outlier
Credit_History	Missing Value

To solve the identified problems

1) Handling Missing Data

- Gender


For Gender data, I choose mode to solve all the missing values because this data is categorical.

A	B	C	D	E
Loan_ID	Gender	Married	Dependent	Educational
'001050		Yes	2	Not Graduate
'001448		Yes	3+	Graduate
'001585		Yes	3+	Graduate
'001644		Yes	0	Graduate
'002024		Yes	0	Graduate
'002103		Yes	1	Graduate
'002478		Yes	0	Graduate
'002501		Yes	0	Graduate
'002530		Yes	2	Graduate
'002625		No	0	Graduate
'002872		Yes	0	Graduate
'002925		No	0	Graduate
'002933		No	3+	Graduate

Gender		
Total Male		489
Total Female		111
Mode		Male

- Married

For Married data, I choose **mode** to solve the missing values because all the data are also categorical.

			Married	
	B	C	D	E
	Gender	Married	Dependent	Educational Attainment
	Female			Graduate
	Male			Graduate
	Male			Graduate

Married		
Total Yes		398
Total No		212
Mode		Yes

- Dependents

For the dependents data, i also use **mode** to solve the missing value. This is because mode can be done with numerical and categorical data.

A	B	C	D	E	F
loan_ID	Gender	Married	Dependent	Education	Self_Employed
P001945	Female	No		Graduate	No
P002144	Female	No		Graduate	No
P002393	Female			Graduate	No
P001350	Male	Yes		Graduate	No
P001357	Male			Graduate	No
P001426	Male	Yes		Graduate	No
P001754	Male	Yes		Not Graduate	Yes
P001760	Male			Graduate	No
P001972	Male	Yes		Not Graduate	No
P002100	Male	No		Graduate	No
P002106	Male	Yes		Graduate	Yes
P002130	Male	Yes		Not Graduate	No
P002682	Male	Yes		Not Graduate	No
P002847	Male	Yes		Graduate	No
P002943	Male	No		Graduate	No

Dependents			
Total 0		345	
Total 1		103	
Total 2		100	
Total 3		50	
Total 4		1	
Mode		0	

- **Self_Employed**

For the Self_Employed data, i also use the mode values to solve the missing values

	A	B	C	D	E	F	G	H	I	J
1	Loan_ID	Gender	Married	Depende	Education	Self_Em	Applica	Coapplicant	LoanAm	Loan_Ar
584	LP001087	Female	No		2 Graduate		3750	2083	120	360
585	LP001387	Female	Yes		0 Graduate		2929	2333	139	360
586	LP001883	Female	No		0 Graduate		3418	0	135	360
587	LP002209	Female	No		0 Graduate		2764	1459	110	360
588	LP002489	Female	No		1 Not Graduate		5191	0	132	360
589	LP002502	Female	Yes		2 Not Graduate		210	2917	98	360
590	LP002753	Female	No		1 Graduate		3652	0	95	360
591	LP002949	Female	No	3+	Graduate		416	41667	350	180
592	LP001027	Male	Yes		2 Graduate		2500	1840	109	360
593	LP001041	Male	Yes		0 Graduate		2600	3500	115	
594	LP001052	Male	Yes		1 Graduate		3717	2925	151	360
595	LP001091	Male	Yes		1 Graduate		4166	3369	201	360
596	LP001326	Male	No		0 Graduate		6782	0		360
597	LP001370	Male	No		0 Not Graduate		7333	0	120	360
598	LP001398	Male	No		0 Graduate		5050	0	118	360
599	LP001546	Male	No		0 Graduate		2980	2083	120	360
600	LP001581	Male	Yes		0 Not Graduate		1820	1769	95	360
601	LP001732	Male	Yes		2 Graduate		5000	0	72	360
602	LP001768	Male	Yes		0 Graduate		3716	0	42	180
603	LP001786	Male	Yes		0 Graduate		5746	0	255	360
604	LP001949	Male	Yes	3+	Graduate		4416	1250	110	360
605	LP002101	Male	Yes		0 Graduate		63337	0	490	180
606	LP002110	Male	Yes		1 Graduate		5250	688	160	360
607	LP002128	Male	Yes		2 Graduate		2583	2330	125	360
608	LP002226	Male	Yes		0 Graduate		3333	2500	128	360
609	LP002237	Male	No		1 Graduate		3667	0	113	180
610	LP002319	Male	Yes		0 Graduate		6256	0	160	360
611	LP002386	Male	No		0 Graduate		12876	0	405	360
612	LP002435	Male	Yes		0 Graduate		3539	1376	55	360
613	LP002732	Male	No		0 Not Graduate		2550	2042	126	360
614	LP002888	Male	No		0 Graduate		3182	2917	161	360
615	LP002950	Male	Yes		0 Not Graduate		2894	2792	155	360

647	Self_Employed		
648	Total No		498
649	Total Yes		81
650	Mode		No
651			

- LoanAmount

To solve the LoanAmount data, I choose by using the mean method. This is because the values in this data are numerical. So I just replace the missing values with the mean answer.

H	I	J
Coapplicant	LoanAmount	Loan_Amount
0		360

630			
631	LoanAmount		
632	Mean		146
633			

- **Loan_Amount_Term**

For this data, I choose mean values to solve the missing values.

	H	I	J	K	L
intIn	CoapplicantIn	LoanAmount	Loan_Amount_Term	Credit_Hist	Property_Type
410	0	88		1	Urban
907	2365	120		1	Urban
720	0	80		0	Urban
828	1330	100		0	Urban
755	0	95		0	Semiurban
572	4114	152		0	Rural
707	3166	182		1	Rural
578	1010	175		1	Semiurban
189	2598	120		1	Rural
124	0	124		0	Rural
250	1667	110		0	Urban
695	0	96		1	Urban
503	4490	70		1	Semiurban
600	3500	115		1	Urban

627				
628	Loan_Amount_Term			
629	Mean		342	
630				

- **Credit History**

For this data, I chose mode values to solve the missing values.

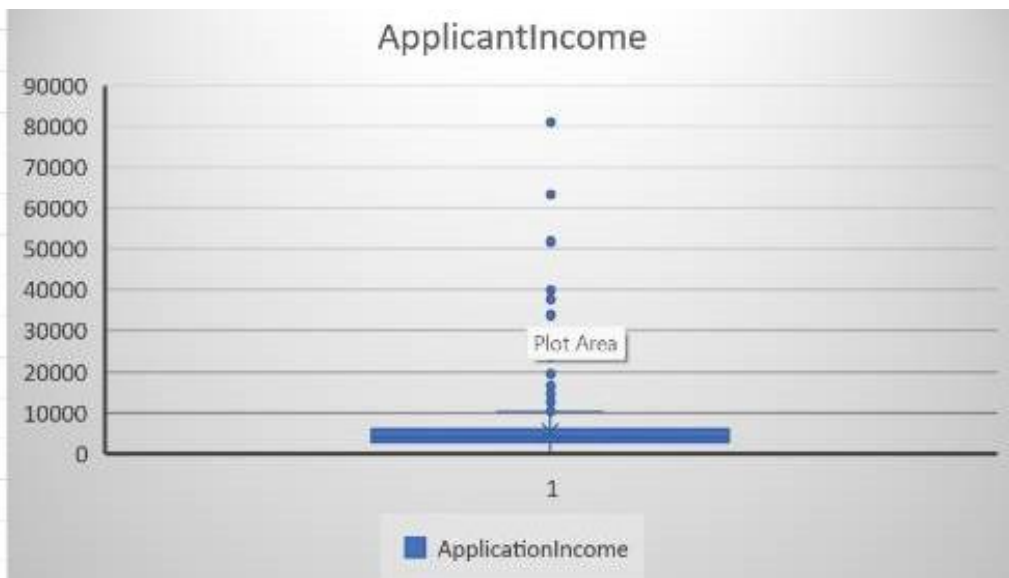
H	I	J	K	L	M	N	O
ApplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status		
0	93	360		Rural	Y		
2816	113	360		Semiurban	Y		
2383	127	360		Semiurban	Y		
0	124	360		Rural	Y		
0	112	360		Semiurban	Y		
2985	132	360		Rural	Y		
663	102	360		Semiurban	Y		
1625	84	360		Urban	Y		
0	100	240		Urban	Y		
0	75	360		Urban	Y		
2250	265	360		Semiurban	N		
2000	99	360		Semiurban	Y		
1398	85	360		Urban	Y		
2569	182	360		Rural	N		
0	160	360		Rural	Y		
5063	67	360		Rural	N		
2138	58	360		Rural	Y		
0	128	360		Semiurban	N		
4250	330	360		Urban	Y		
0	185	360		Rural	Y		
2134	88	360		Urban	Y		
4583	259	360		Semiurban	Y		
2222	85	360		Urban	Y		

633				
634	Credit History			
635	Total 0		89	
636	Total 1		474	
637	Mode		1	
638				

2) Handling Outlier

To handle the outlier, firstly, I visualize the data with boxplot. All the columns that had the outlier, I had visualized it with the box plot graph. Then, I also code by using python method to count the total number of the outlier which is in the data. Then, I solve the outlier by choosing the transformation method which logs all the values for each data.

- **ApplicantIncome**



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

data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

plt.boxplot(data['ApplicantIncome'])
plt.xlabel('ApplicantIncome')

def find_outlier_IQR(df):
    q1 = df.quantile(0.25)
    q3 = df.quantile(0.75)
    IQR = q3 - q1
    outliers = df[((df < (q1 - 1.5 * IQR)) | (df > (q3 + 1.5 * IQR)))]
    return outliers

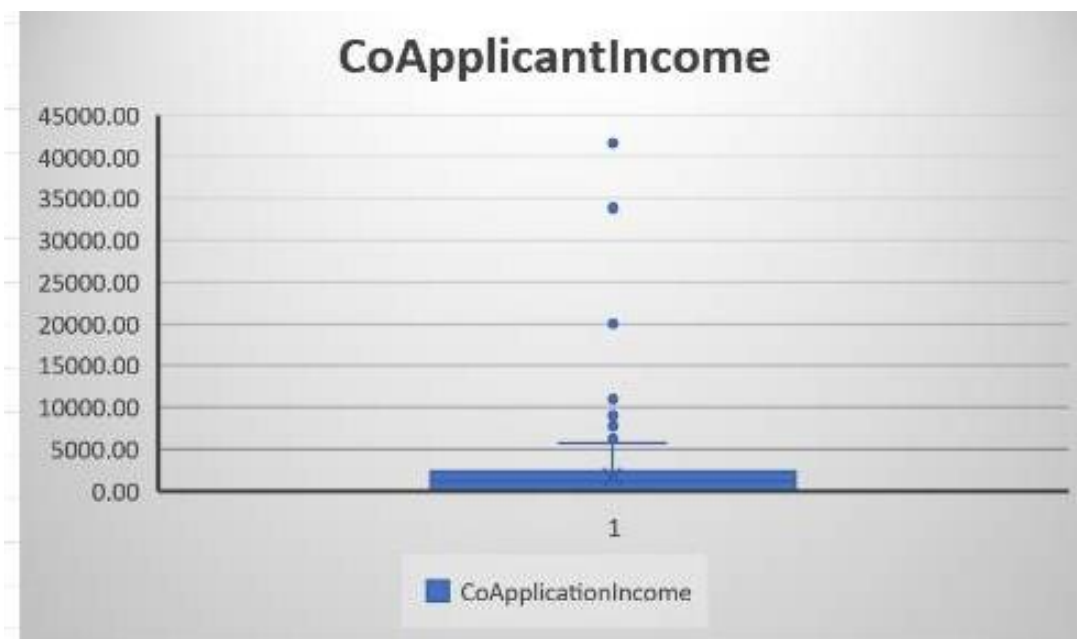
outliers = find_outlier_IQR(data["ApplicantIncome"])

print("Number of outliers: " + str(len(outliers)))
print("Max outlier value: " + str(outliers.max()))
print("Min outlier value: " + str(outliers.min()))
print(outliers)
```

```
Number of outliers: 50
Max outlier value: 81000
Min outlier value: 10408
9      12841
34     12500
54     11500
67     10750
```

s	Log ApplicantIncome	L
	5.017279837	
	5.351858133	
	6.033086222	
	6.470799504	
	6.514712691	
	6.908754779	
	6.933423026	
	6.933423026	
	7.170119543	
	7.229113878	
	7.274479559	
	7.313886832	
	7.338888134	
	7.378383713	
	7.39387829	
	7.419979924	
	7.473069088	
	7.486052618	
	7.496097345	
	7.496097345	
	7.501082124	
	7.502186487	

- CoApplicantIncome



```

import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

plt.boxplot(data['CoapplicantIncome'])
plt.xlabel('CoapplicantIncome')

def find_outlier_IQR(df):
    q1 = df.quantile(0.25)
    q3 = df.quantile(0.75)
    IQR = q3 - q1
    outliers = df[((df < (q1 - 1.5 * IQR)) | (df > (q3 + 1.5 * IQR)))]
    return outliers

outliers = find_outlier_IQR(data["CoapplicantIncome"])

print("Number of outliers: " + str(len(outliers)))
print("Max outlier value: " + str(outliers.max()))
print("Min outlier value: " + str(outliers.min()))
print(outliers)

```

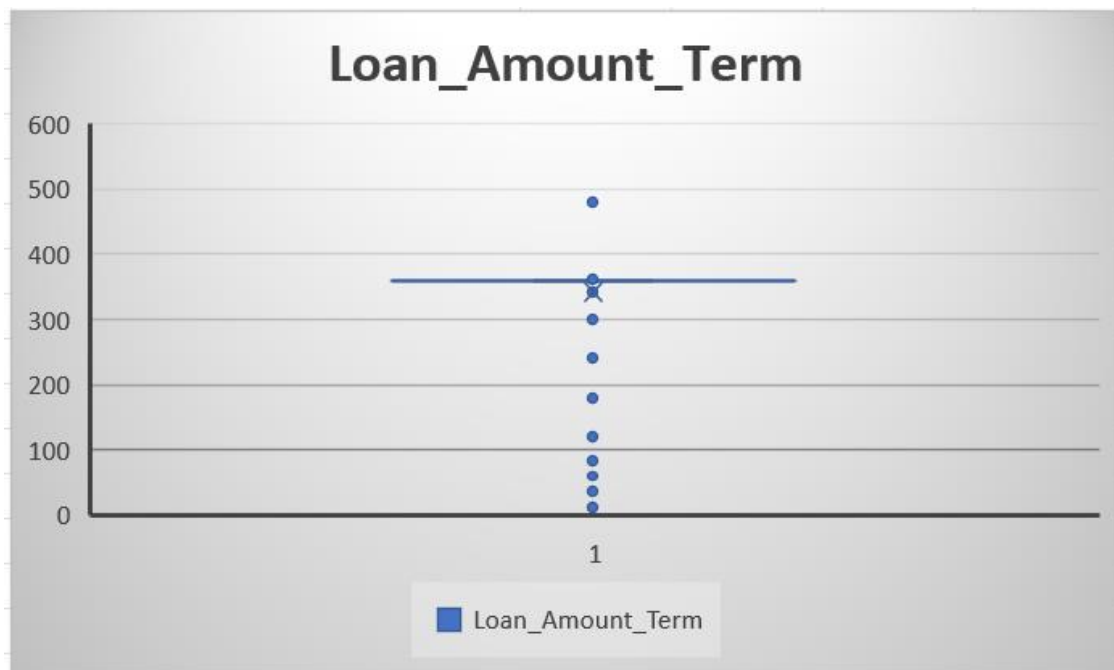
```

Number of outliers: 18
Max outlier value: 41667.0
Min outlier value: 6250.0

```

K	
Log CoapplicantIncome	
7.496097345	
7.978653729	
10.63748873	
8.211754397	
8.574895902	
8.014004995	
7.928045601	
8.612685173	
6.991176887	
7.54009032	
0	
7.496097345	
7.262628601	
9.903537551	
7.497761701	
8.266421473	
8.172446818	
7.711101252	
7.101675972	
7.984462732	
7.533158807	
7.418780883	
7.45667857	

- Loan_Amount_Term



```
plt.boxplot(data['Loan_Amount_Term'])
plt.xlabel('Loan_Amount_Term')

def find_outlier_IQR(df):
    q1 = df.quantile(0.25)
    q3 = df.quantile(0.75)
    IQR = q3 - q1
    outliers = df[((df < (q1 - 1.5 * IQR)) | (df > (q3 + 1.5 * IQR)))]
    return outliers

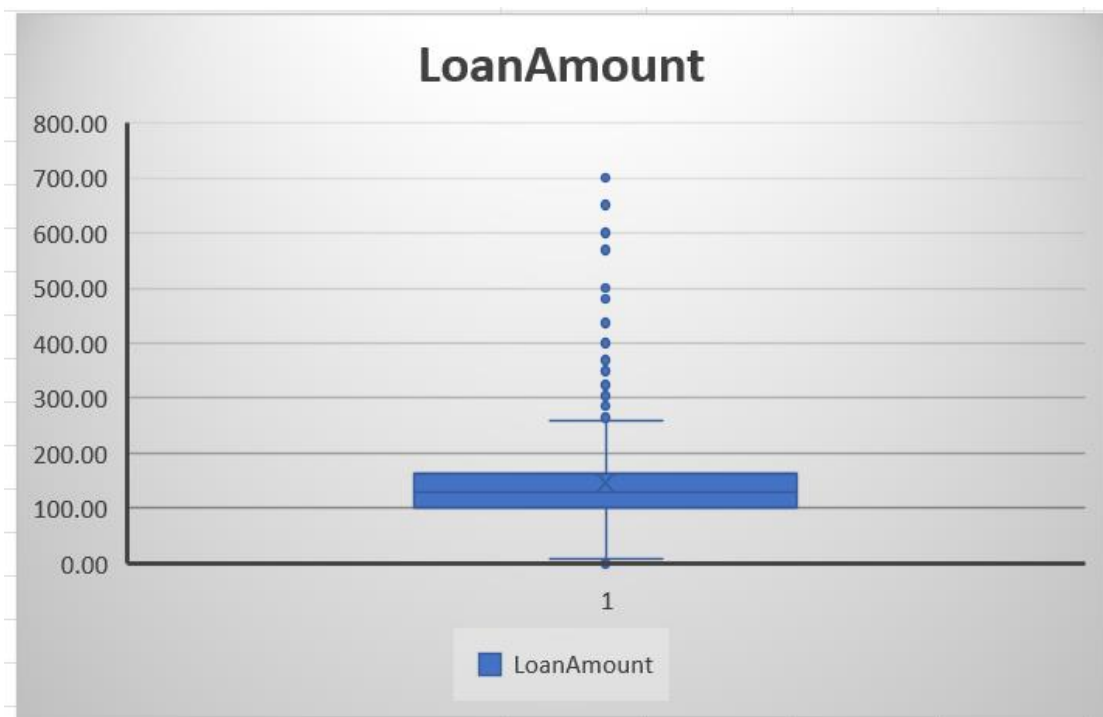
outliers = find_outlier_IQR(data["Loan_Amount_Term"])

print("Number of outliers: " + str(len(outliers)))
print("Max outlier value: " + str(outliers.max()))
print("Min outlier value: " + str(outliers.min()))
print(outliers)
```

```
Number of outliers: 88
Max outlier value: 480.0
Min outlier value: 12.0
```

M
Log Loan_Amount_Term
5.888877958
4.795790546
4.795790546
4.795790546
5.888877958
5.888877958
5.888877958
5.888877958
5.888877958
5.198497031
5.888877958
5.198497031
5.888877958
5.888877958
5.198497031
5.888877958
5.888877958
5.888877958
5.888877958
5.888877958
5.484796933
5.888877958
5.888877958

- LoanAmount



```
data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

plt.boxplot(data['LoanAmount'])
plt.xlabel('LoanAmount')

def find_outlier_IQR(df):
    q1 = df.quantile(0.25)
    q3 = df.quantile(0.75)
    IQR = q3 - q1
    outliers = df[((df < (q1 - 1.5 * IQR)) | (df > (q3 + 1.5 * IQR)))]
    return outliers

outliers = find_outlier_IQR(data["LoanAmount"])

print("Number of outliers: " + str(len(outliers)))
print("Max outlier value: " + str(outliers.max()))
print("Min outlier value: " + str(outliers.min()))
print(outliers)
```

Number of outliers: 39

Max outlier value: 700.0

Min outlier value: 275.0

Log LoanAmount
2.302585093
2.890371758
3.258096538
3.258096538
3.295836866
3.433987204
3.433987204
3.583518938
3.610917913
3.713572067
3.713572067
3.761200116
3.80666249
3.80666249
3.828641396
3.828641396
3.850147602
3.871201011
3.891820298
3.931825633
3.931825633
3.931825633
3.931825633

3) Inconsistent Data

For the Inconsistent data, we only found one column that has 2 inconsistent values. The values are in decimal numbers. So, I choose to convert the decimal into a whole number so that it will be parallel to other data.

id	CoapplicantIncome	LoanAmount
301	985.7999878	
320	16.12000084	

655			
656	Inconsistent Data		
657	coApplication income		
658	16.12000084	to	16
659	985.799878	to	986
660			
661			

- Describe the distribution of each attribute. Did you identify any patterns or trends in the data? If so, what were they?

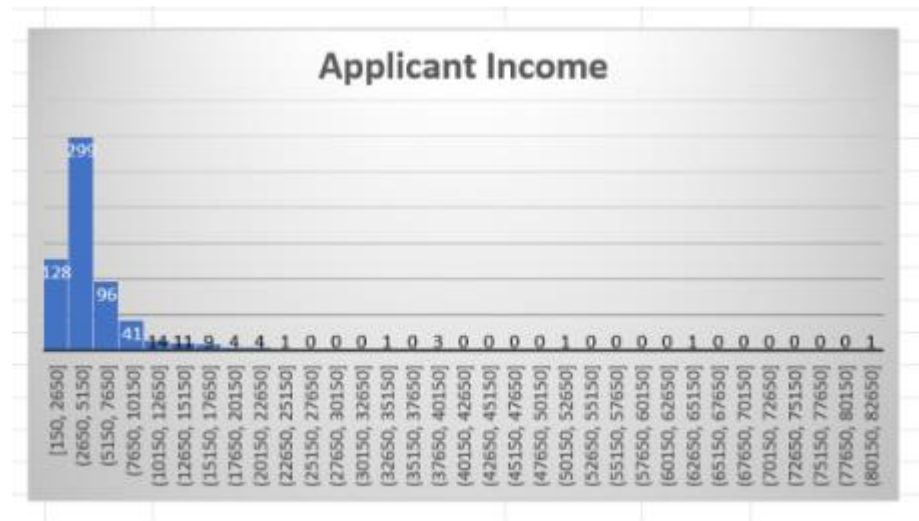
Attribute ApplicantIncome

Mean of Applicant Income	5403.459
Mode	2500
Variance	37320390
Standard Deviation	6109.042
Q1	2875.75
Median	3812.5
Q3	5803.75

This ApplicantIncome attribute has a mean value of 5403.459, which is equal to the average income of the applicants that is \$5403. The mode for this attribute is 2500 that indicates that this value appears in the data the most frequently. While the median of these attributes is

3812.5. When the data are organized from lowest to highest, this median explains the midway value of the data. The standard deviation for these attributes is 6109.042. It shows how much the data varies or spread out around the mean. Data points are further dispersed from the mean when the standard deviation is higher.

Here are the histograms that we had visualized from the data that we got from the table above.



From the data, we can first look at the measures of central tendency and dispersion to see if there are any patterns or trends in the data. The variance and standard deviation are indicators of dispersion, while the mean, mode and median are indicators of central tendency. So based on the given measures, we can observe the following patterns and trends.

The observation that can be made is the data's mode is noticeably lower than its mean and median, indicating that lower values are more prevalent in the data than higher ones. The values in the data set are significantly dispersed from the mean, as shown by the comparatively large variance and standard deviation. According to the quartiles, the distribution of the data is not symmetrical, with more data points lying below the median (3812) than above it. There might be some outliers or extreme values in the lower range of the data set since the gap between the Q3 and the mean is smaller than the difference between the Q1 and the mean. Overall, these patterns and trends indicate that the data may have a skewed distribution with some probable outliers or extreme values in the lower range and a higher proportion of lower values.

Attribute CoApplicantIncome

Mean of Co Applicant Income	1621.246
Mode	0
Variance	8562930
Standard Deviation	2926.248
Q1	0
Median	1188.5
Q3	2303

For the CoApplicantIncome, it means that is 1621.246 which is equal to the average that is \$1621. The mode for this CoApplicantIncome is 0 because the majority of data is 0. Its median is 1188.5. Then the variance and standard deviation for these attributes is 8562930 and 2926.248.

Here is the histograms that we had visualize from the data that we got from the table above



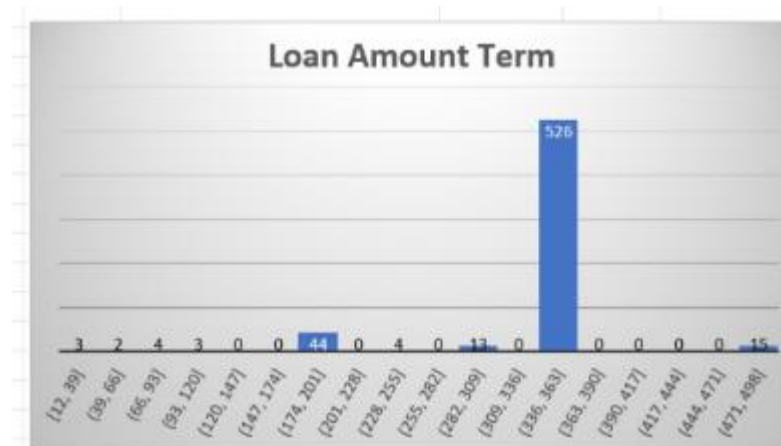
The following patterns or trends can be seen based on the provided measurements. The data's mode is noticeably lower than its mean and median, indicating that lower values are more prevalent in the data than higher ones. The variance and standard deviation are relatively high, indicating that the values in the data set are widely spread out from the mean. The quartiles suggest that the data is not symmetrically distributed, with a larger proportion of data points falling below the median (3812) than above it. There might be some outliers or extreme values in the lower range of the data set since the gap between the third quartile (q3) and the mean is smaller than the difference between the first quartile (q1) and the mean. Overall, these patterns and trends indicate that the data may have a skewed

distribution with some probable outliers or extreme values in the lower range and a higher proportion of lower values.

Attribute Loan Amount Term

Mean of Loan Amount Term	342
Mode	360
Variance	4143.817
Standard Deviation	64.37249
Q1	360
Median	360
Q3	360

Next is for Loan Amount Term attributes, it has a mean of 342 that is same with the average of \$342. It also has a mode that is 360. The variance and standard deviation for this attribute is 4143.817 and 64.37249. While the median for this attribute is 360.



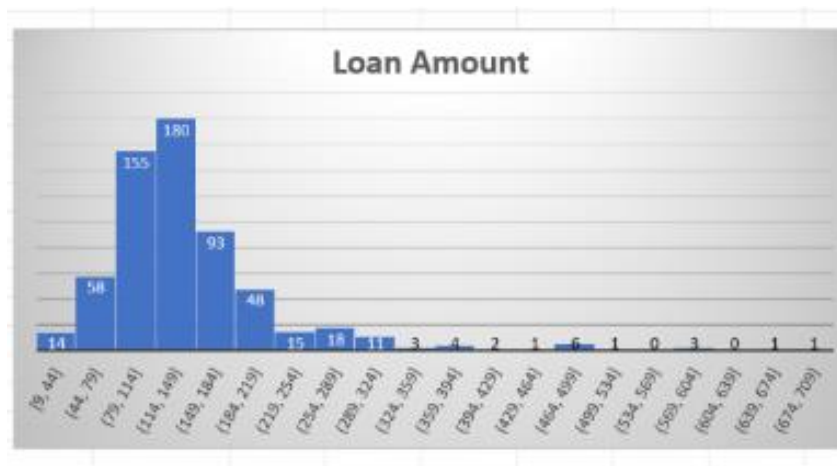
From the measures above, we can identify that the data's mode indicates a skewed distribution towards lower values because it is much lower than the mean and median. The values in the data set are significantly dispersed from the mean, as shown by the comparatively large variance and standard deviation. According to the quartiles, the distribution of the data is not symmetrical, with more data points lying below the median (3812) than above it. There might be some outliers or extreme values in the lower range of the data set since the gap between the third quartile (q3) and the mean is smaller than the difference between the first quartile (q1) and the mean. Overall, these patterns and trends indicate that the data may have a skewed distribution with some probable outliers or extreme values in the lower range and a higher proportion of lower values.

Attribute Loan Amount

Mean of Loan Amout	146.397394
Mode	146
Variance	7062.30185
Standard Deviation	84.0375027
Q1	100
Median	129
Q3	165

This Loan Amount attribute has a mean of 146.397394 that is equal to the average of \$146. It has a mode of 146 and variance that is 7062.30185. Its standard deviation is 84.0375027. The median for this attribute is 129.

Here is the graph that we had visualized from the data above.



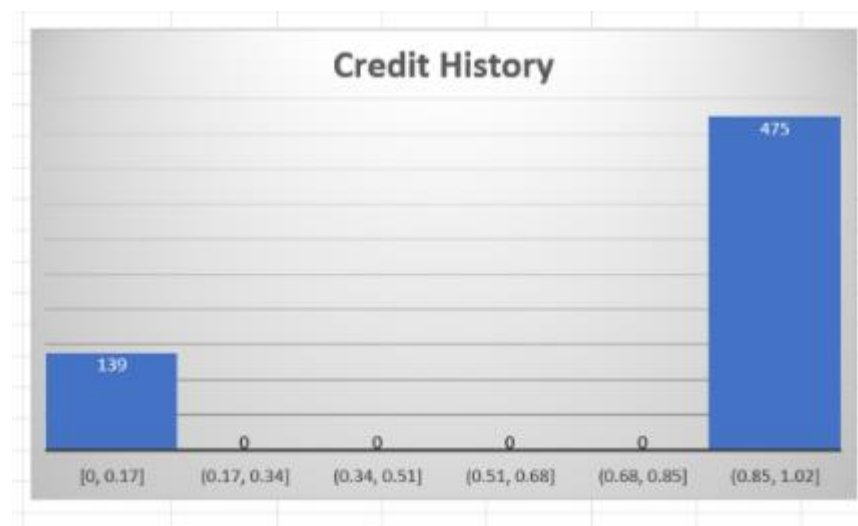
The following patterns or trends can be seen based on the provided measurements. The data's mode is noticeably lower than its mean and median, indicating that lower values are more prevalent in the data than higher ones. The values in the data set are significantly dispersed from the mean, as shown by the comparatively large variance and standard deviation. According to the quartiles, the distribution of the data is not symmetrical, with more data points lying below the median (3812) than above it. There might be some outliers or extreme values in the lower range of the data set since the gap between the third quartile (q3) and the mean is smaller than the difference between the first quartile (q1) and the mean. Overall, these patterns and trends indicate that the data may have a skewed distribution with some probable outliers or extreme values in the lower range and a higher proportion of lower values.

Attribute Credit History

Mean of Credit History	0.77361564
Mode	1
Variance	0.17542018
Standard Deviation	0.41883193
Q1	1
Median	1
Q3	1

For the Credit History attribute, it means that is 0.77361564 which is equal to the average of \$1. The mode for this attribute is 1 and the variance for this attribute is 0.17542018. The standard deviation is 0.41883193 and the median is 1.

Here is the graph that we had visualized from the data.



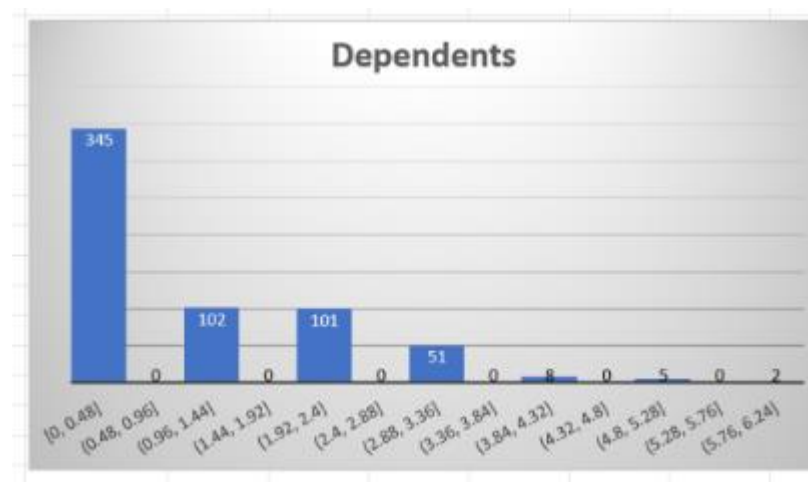
These measurements allow us to draw the conclusion that the data is largely centred around the number 1. The high mean value and the fact that the median and all quartiles are also 1 demonstrate this. The data appear to be tightly grouped around the mean and have a low variation and standard deviation. In conclusion, the presented data's pattern and trend indicate that there is a significant concentration of data values near the value of 1, with minimal variation.

Attribute Dependents

Mean of Dependants	0.85667752
Mode	0
Variance	1.36931097
Standard Deviation	1.17017561
Q1	0
Median	0
Q3	2

This attribute has mean that 0.85667752 which is equal to the average of \$1. Then, its mode is 0 because the majority of the data is 0. For the variance and standard deviation, it is 1.36931097 and 1.17017561. Then the median is 0.

Here is the histogram graph that we had visualized from the data that we had calculated.



We can spot the following patterns and trends in the data using the provided measures: The values in the data set are slightly biased towards higher values, as indicated by the data's mean of 0.85667752. The data's mode is 0, which denotes that 0 is the value that appears the most frequently throughout the dataset. The data's variance, which is measured in terms of the mean and mode, is 1.36931097, which is quite high. The values in the data set appear to be significantly dispersed from the mean, according to this indication. The data's standard deviation, which is likewise quite high at 1.17017561, shows that there is a significant degree of variance in the data. The data's quartiles are $q1=0$, median=0, and $q3=2$, corresponding to 25%, 50%, and 75% of the data falling below 0, 0, and 2, respectively. This implies that the data's distribution is a little bit skewed to the right. Overall, the pattern and trend in the data indicate that the data set's values are very varied, with a propensity for higher values. The distribution of the data is also right skewed, with the majority of values concentrated at the lower end of the scale and some extreme values at the higher end.

4. How did you normalize? Did these techniques improve the quality of the data? Why or why not?

We used Z-Score (Standardization) to normalize the data. We need to subtract the data set's mean from each data and divide it by the standard deviation. The Z-score technique improves the quality of the dataset because the normalized data has a mean close to zero and a standard deviation close to 1 compared to the original data, in which both mean and standard deviation are not close to zero and one.

Normalized data:

	Dependents	Credit_History	...	LoanAmount	Loan_Amount_Term
0	-0.732690	-2.428760	...	-1.073390	0.239847
1	-0.732690	0.411733	...	-0.778392	0.100465
2	-0.732690	0.411733	...	0.215783	0.239847
3	-0.732690	0.411733	...	-0.152769	0.239847
4	-0.732690	0.411733	...	-0.292094	0.239847
..
609	-0.732690	0.411733	...	-1.722028	0.239847
610	-0.732690	0.411733	...	0.364721	0.239847
611	2.688387	0.411733	...	0.243685	-0.801846
612	3.543657	0.411733	...	0.428980	0.239847
613	4.398926	0.411733	...	0.403519	0.979685

Mean and Standard Deviation:

Mean:	
Dependents	0.00
Credit_History	-0.00
ApplicantIncome	0.00
CoapplicantIncome	-0.00
LoanAmount	-0.00
Loan_Amount_Term	-0.00
dtype: object	
Standard Deviation:	
Dependents	1.00
Credit_History	1.00
ApplicantIncome	1.00
CoapplicantIncome	1.00
LoanAmount	1.00
Loan_Amount_Term	1.00

Coding for normalization using Z-score.

```
import pandas as pd
import numpy as np

data = pd.read_excel(r"C:\Users\myopc\Desktop\DatasetNew.xlsx")

# usage
def normalize(col):
    mean = np.mean(col)
    std = np.std(col)
    normalized_data = (col - mean) / std
    return normalized_data

dataset = data[data.select_dtypes(include=['number']).columns] = data.select_dtypes(include=['number']).apply(normalize)
print(dataset)

mean_values = dataset.mean()
formatted_mean_values = mean_values.apply(lambda x: '{:.2f}'.format(x))
std_values = dataset.std()
formatted_std_values = std_values.apply(lambda x: '{:.2f}'.format(x))

# Print the mean and standard deviation values
print("Mean:")
print(formatted_mean_values)
print("\nStandard Deviation:")
print(formatted_std_values)
```

We also used Mix-Max scaling which does not improve our data. Usually, this technique uses a scale of 0 and 1. We only need to subtract the minimum value of the data from each data and divide it by the range (maximum value minus minimum value).

Normalized data using Min-Max Scaling:

	Dependents	Credit_History	...	LoanAmount	Loan_Amount_Term
0	0.000000	0.0	...	0.480296	0.953531
1	0.000000	1.0	...	0.514374	0.944776
2	0.000000	1.0	...	0.629223	0.953531
3	0.000000	1.0	...	0.586647	0.953531
4	0.000000	1.0	...	0.570552	0.953531
..
609	0.000000	1.0	...	0.405364	0.953531
610	0.000000	1.0	...	0.646428	0.953531
611	0.666667	1.0	...	0.632446	0.888101
612	0.833333	1.0	...	0.653852	0.953531
613	1.000000	1.0	...	0.650910	1.000000

After we normalized the data using Mix-Max scaling, we can see that the range of the min and max is between 0 to 1

	Dependents	Credit_History	...	LoanAmount	Loan_Amount_Term
count	614.000000	614.000000	...	614.000000	614.000000
mean	0.142780	0.855049	...	0.604295	0.938466
std	0.195029	0.352339	...	0.115616	0.062861
min	0.000000	0.000000	...	0.000000	0.000000
25%	0.000000	1.000000	...	0.544715	0.953531
50%	0.000000	1.000000	...	0.603528	0.953531
75%	0.333333	1.000000	...	0.660693	0.953531
max	1.000000	1.000000	...	1.000000	1.000000

Coding normalization using Min-Max scaling:

```
import pandas as pd

# Read the dataset into a pandas dataframe
df = pd.read_excel(r"C:\Users\my pc\Desktop\DatasetNew.xlsx")

# Define a function to perform min-max scaling
!usage
def min_max_scale(col):
    return (col - col.min()) / (col.max() - col.min())

# Apply the min-max scaling function to the numerical columns in the dataframe
dataset = df[df.select_dtypes(include=['number']).columns] = df.select_dtypes(include=['number']).apply(min_max_scale)

print(dataset)
|

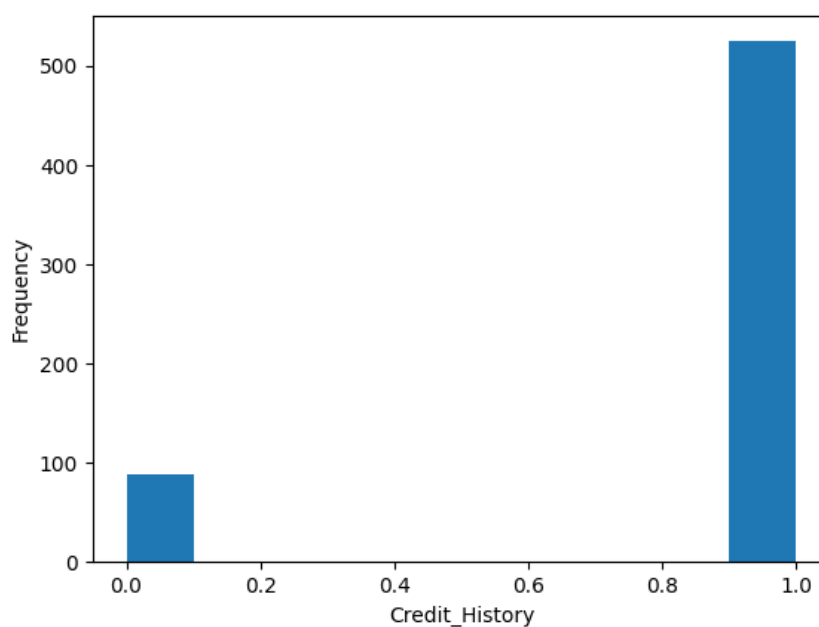
data = df.copy()
for col in ['LoanAmount', 'Loan_Amount_Term', 'CoapplicantIncome', 'ApplicantIncome', 'Dependents',
            'Credit_History']:
    data[col] = (data[col] - data[col].min()) / (data[col].max() - data[col].min())
print(data.describe())
```

5. Which attribute has been chosen from the feature selection? Why?

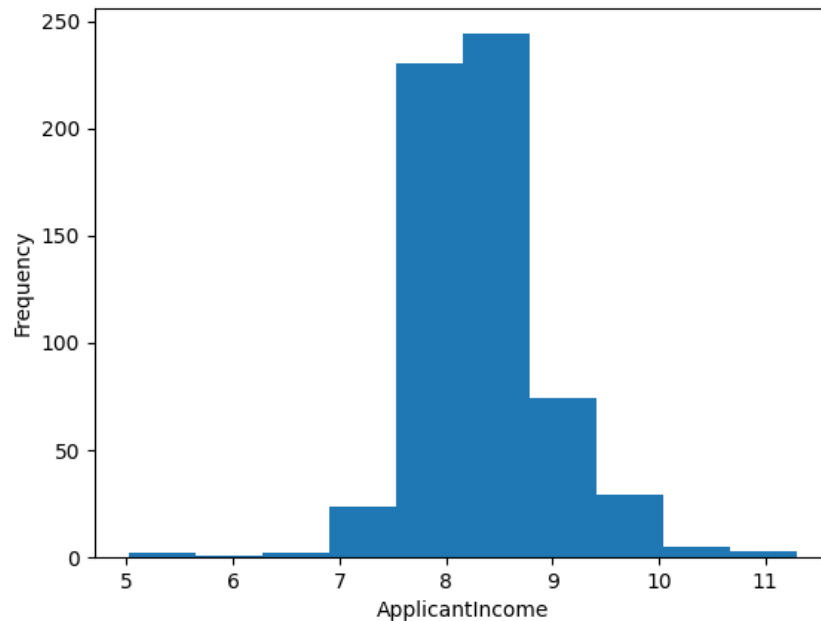
The chosen attributes are credit History, ApplicantIncome, LoanAmount and CoapplicantIncome. Credit History has the highest importance value among all the features, which is 0.290338. it means that credit history has a strong relationship to the target variable, loan status.

	feature	Importance
5	Credit_History	0.290338
4	ApplicantIncome	0.207800
2	LoanAmount	0.181096
1	CoapplicantIncome	0.112298
3	Loan_Amount_Term	0.044112
0	Dependents	0.037853
15	Property_Area_Semiurban	0.019193
14	Property_Area_Rural	0.016425
9	Married_Yes	0.013019
16	Property_Area_Urban	0.011230
8	Married_No	0.011170
11	Education_Not_Graduate	0.011150
10	Education_Graduate	0.010494
7	Gender_Male	0.009009
6	Gender_Female	0.008534
13	Self_Employed_Yes	0.008321
12	Self_Employed_No	0.007958

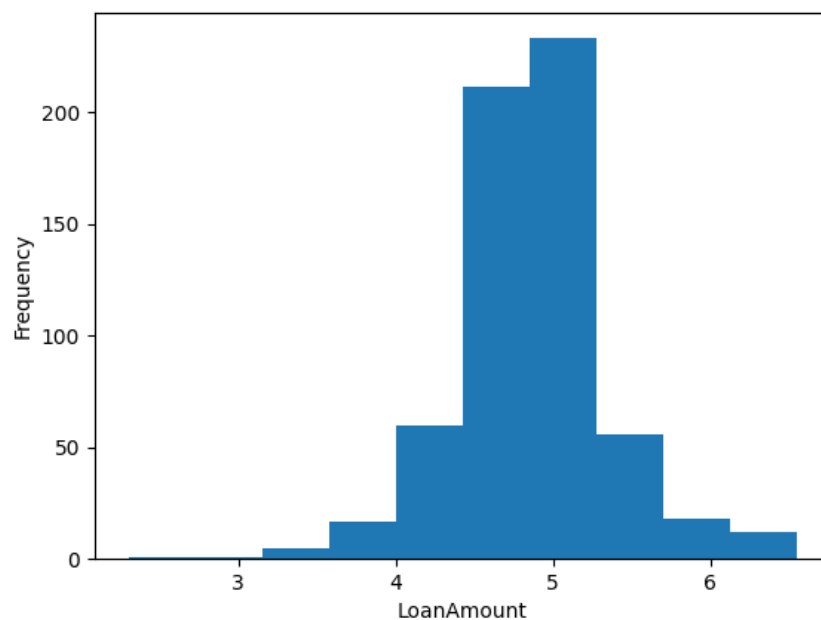
We can see that there has a distinct peak for different credit history categories.



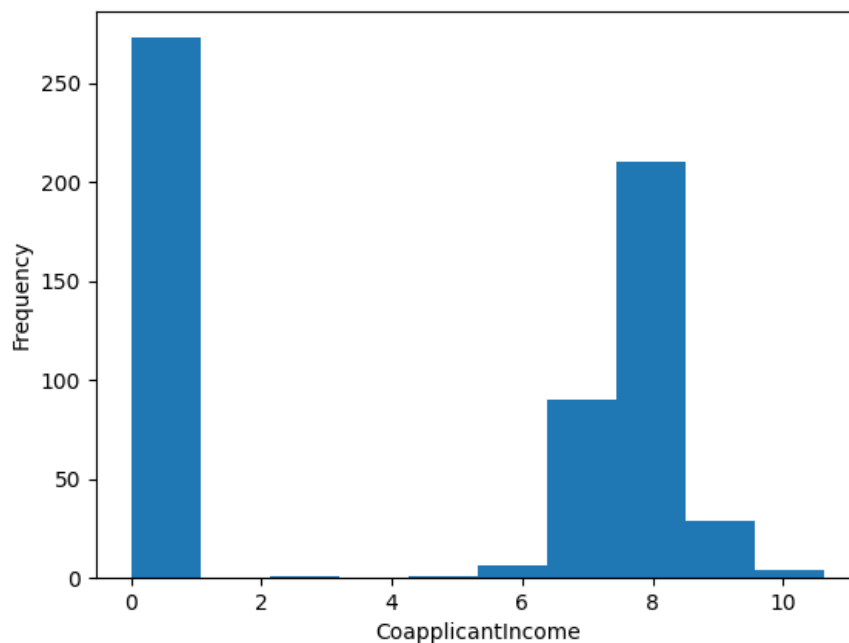
ApplicantIncome also has a high feature importance value (0.207800). It indicates that this feature strongly influences the target variable (Loan Status). We can observe that the income has varying frequencies and it seems like it has a normal distribution



The loan amount has the third largest feature importance value (0.181096). It suggests that the LoanAmount amount has a lower relationship than Credit_History and ApplicantIncome. We can see that the distribution is normal distribution, indicating common loan ranges that appear more frequently



The last feature is CoapplicantIncome which has the lowest importance value compared to Credit_History, ApplicantIncome, and LoanAmount. It has a lower relationship to the target variable (Loan_Status).



After we have made the feature selection, we set a target: if the importance value is below 0.1, the feature is not essential. So, we will not include those features. So right now, we have only 4 important features which are Credit_History, CoapplicantIncome, ApplicantIncome, and LoanAmount.

This is the coding for feature selection

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt
import seaborn as sns

# Read the Excel file into a pandas DataFrame
data = pd.read_excel(r"C:\Users\mypc\Desktop\DatasetNew.xlsx")

X = data[['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'CoapplicantIncome',
          'LoanAmount', 'Loan_Amount_Term', 'ApplicantIncome', 'Credit_History', 'Property_Area']]

Y = data['Loan_Status']

X = pd.get_dummies(X)
Y = pd.get_dummies(Y)
rf = RandomForestRegressor()
rf.fit(X, Y)
importances = pd.DataFrame({'feature': X.columns, 'Importance': rf.feature_importances_})
importances = importances.sort_values('Importance', ascending=False)

print(importances)

selected_features = ['Credit_History', 'LoanAmount', 'ApplicantIncome', 'CoapplicantIncome']
selected_df = data[selected_features]

for feature in selected_features:
    plt.hist(selected_df[feature])
    plt.xlabel(feature)
    plt.ylabel('Frequency')
    plt.show()
```

6. Based on your analysis, what insights or conclusions can you draw from the data? How could this information be used in practice?

Based on the analysis, the conclusion that can be made is that all the data required in the process of obtaining this loan has been successfully completed. All missing values in each customer detail column have been filled in using several methods. The average missing values problem encountered is in terms of categorical and numerical. Therefore, we can identify some suitable methods to solve this missing values problem. Among other things, we use methods such as mode values, mean values, transformation and also converting inconsistent data into consistent data such as decimal numbers to whole numbers.

Furthermore, we also calculate the mean of the data, mode, variance, standard deviation, quartile 1, median and quartile 3 for all the data. This calculation is shown in the summary statistic. The purpose of doing summary statistics is because A dataset is subjected to summary statistics in order to present a clear and insightful overview of the data. They aid in comprehending the dataset's variables' distribution, central tendency, variability, and other important properties. Then, we also visualize the data by using the histogram graph.

Then, we used random forest as a technique to select the most relevant features from the dataset. During the training process, Random Forest assigns an important score to each feature based on how much each feature contributes to the model's overall performance. Feature importance values can be used to rank features in order of importance. Less important features can be removed from the dataset. This reduces the dimensionality of the data and improves model performance. For our dataset, the most important features are LoanAmount, Credit_History, CoapplicantIncome, and ApplicantIncome. The important score that is below 0.01 will be removed from the dataset because it shows less contribution towards the dataset.

Lastly, we use Z-score technique(standardization) and Min-Max Scaling technique to normalize the dataset. For the Min-Max Scaling Technique, reducing the values of a dataset to a defined range, usually 0 to 1, is beneficial. By doing this, it guarantees that every value is proportionally changed to fit within this range. A dataset's values can be transformed to have a mean of 0 and a standard deviation of 1 for the Z-score technique. By dividing the standard deviation by the mean, it standardizes the variable.

7. Your report should answer all those questions. Please attach all charts, and data before and after pre-processing in the report.

a) Charts and data before the pre-processing

- Categorical Data

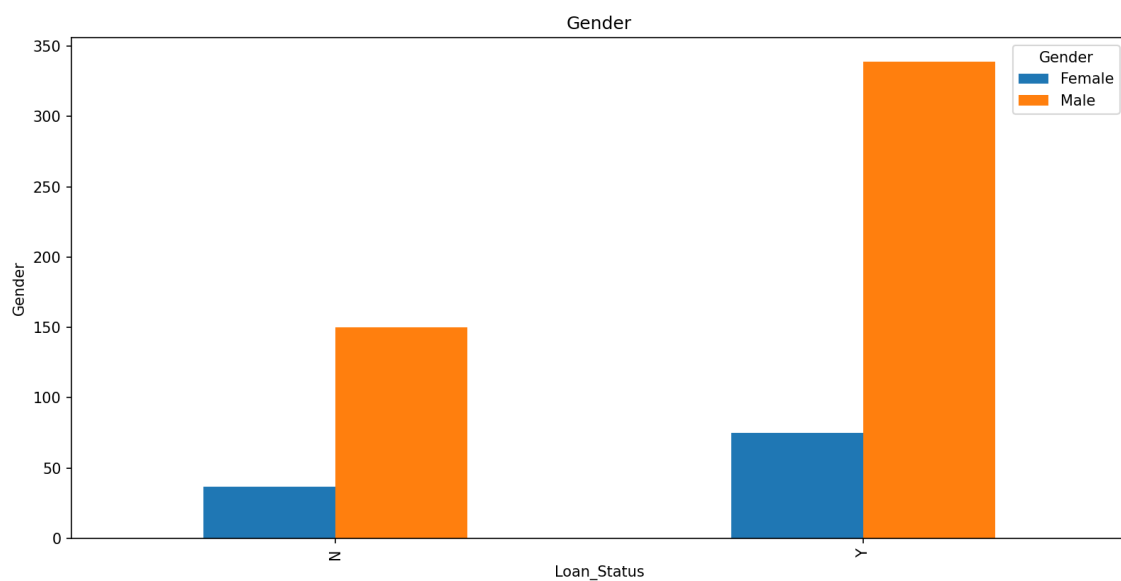
We use bar graphs for displaying the categorical data before the pre-processing.

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

data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

print(data.head())

grouped_data = data.groupby(['Loan_Status', 'Gender']).size().unstack()
grouped_data.plot(kind='bar')
plt.xlabel('Loan_Status')
plt.ylabel('Gender')
plt.title('Gender')
plt.show()
```



```

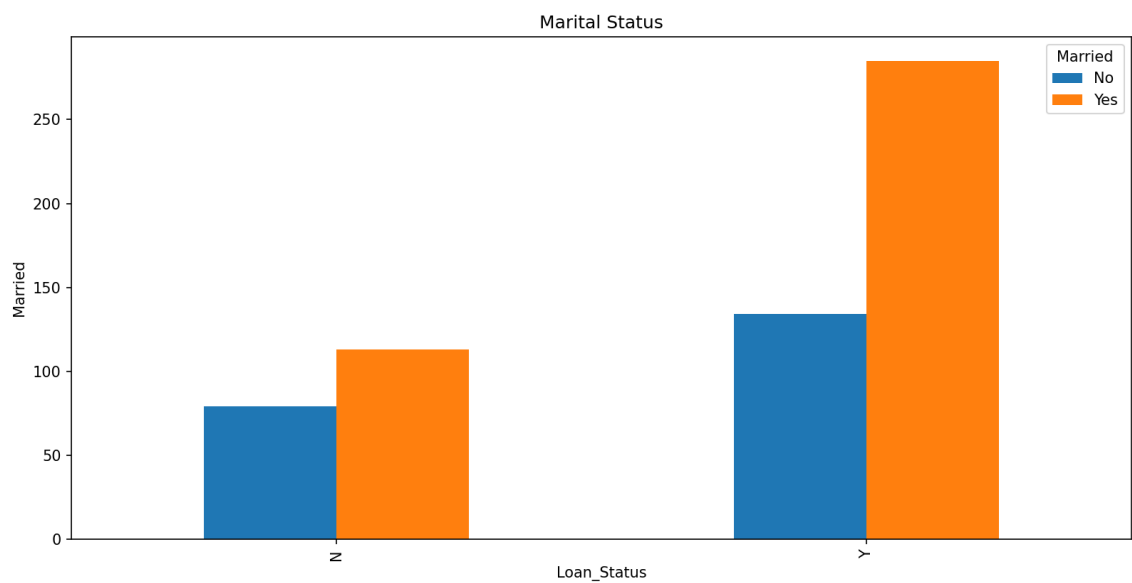
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

print(data.head())

grouped_data = data.groupby(['Loan_Status', 'Married']).size().unstack()
grouped_data.plot(kind='bar')
plt.xlabel('Loan_Status')
plt.ylabel('Married')
plt.title('Marital Status')
plt.show()

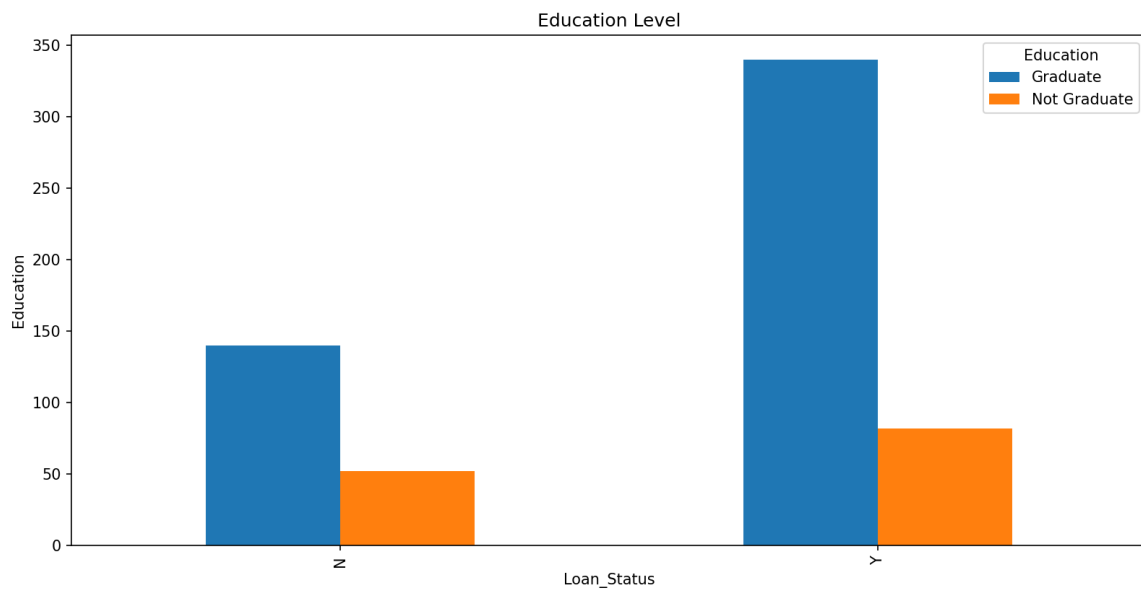
```



```

1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")
5
6 print(data.head())
7
8 grouped_data = data.groupby(['Loan_Status', 'Education']).size().unstack()
9 grouped_data.plot(kind='bar')
10 plt.xlabel('Loan_Status')
11 plt.ylabel('Education')
12 plt.title('Education Level')
13 plt.show()
14

```



```

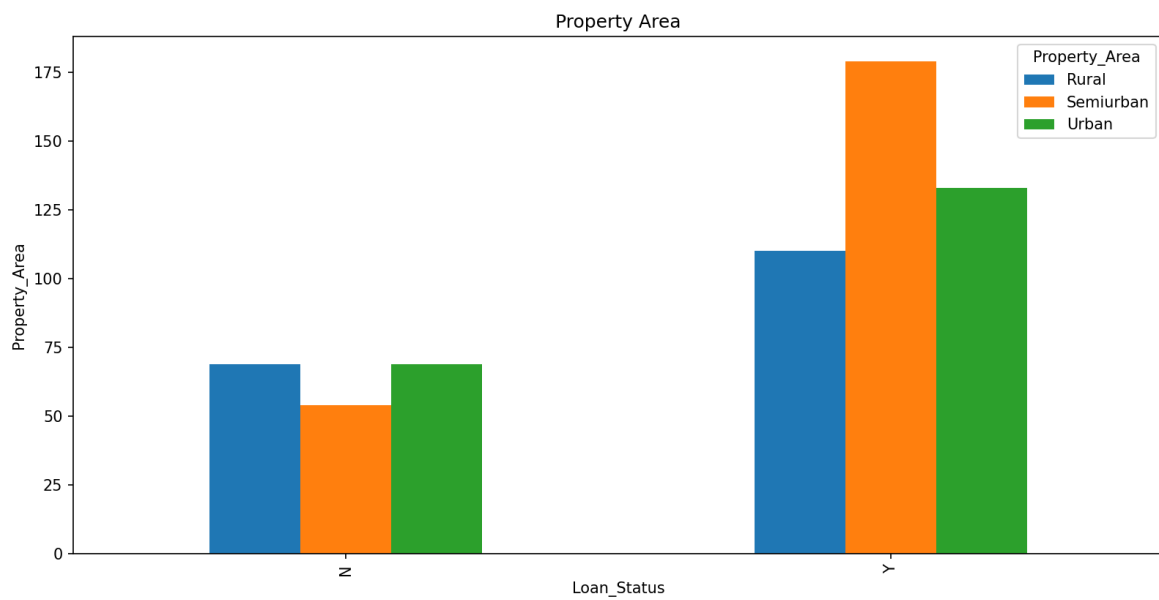
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

print(data.head())

grouped_data = data.groupby(['Loan_Status', 'Property_Area']).size().unstack()
grouped_data.plot(kind='bar')
plt.xlabel('Loan_Status')
plt.ylabel('Property_Area')
plt.title('Property Area')
plt.show()

```



- Numerical Data

We use histogram to displays the numerical data before the pre-processing

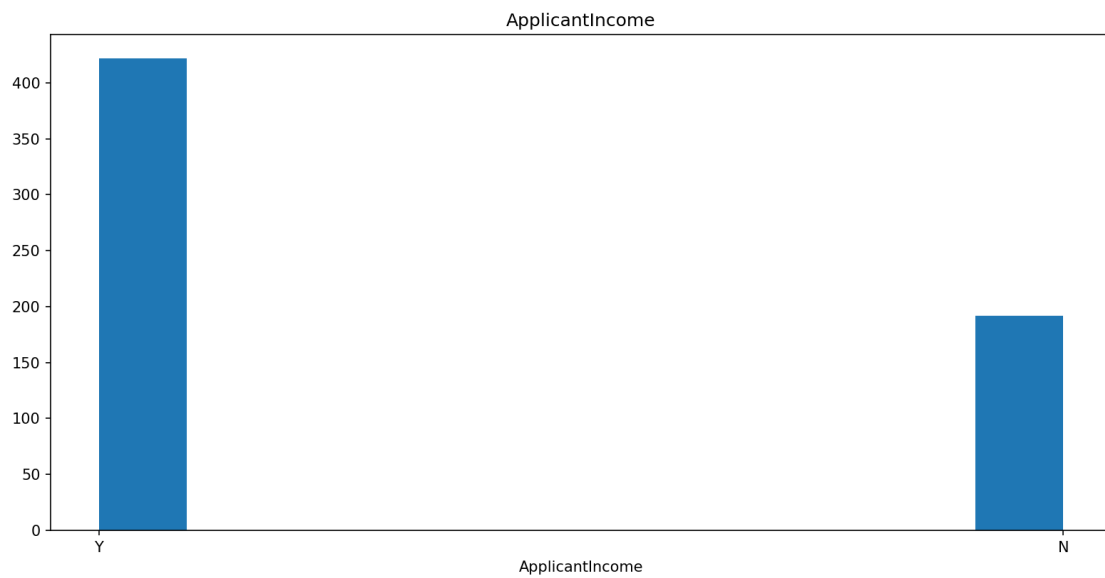
```
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

print(data.head())

plt.hist(data['Loan_Status'], bins='auto')

plt.xlabel('ApplicantIncome')
plt.title('ApplicantIncome')
plt.show()
```



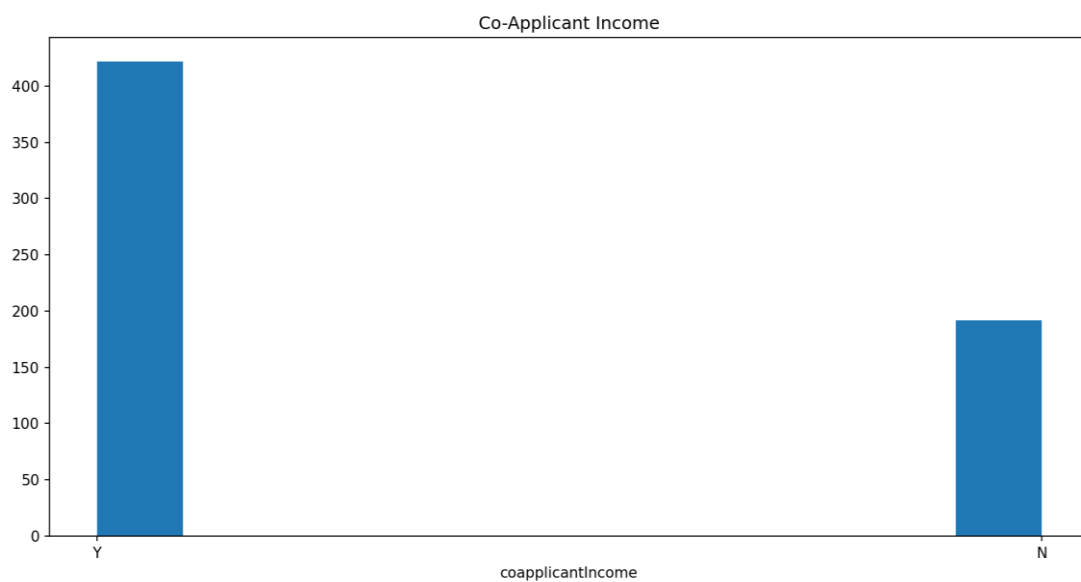
```
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

print(data.head())

plt.hist(data['Loan_Status'], bins='auto')

plt.xlabel('coapplicantIncome')
plt.title('Co-Applicant Income')
plt.show()
```



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

data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

print(data.head())

plt.hist(data['Loan_Status'], bins='auto')

plt.xlabel('LoanAmount')
plt.title('Loan Amount')
plt.show()
```



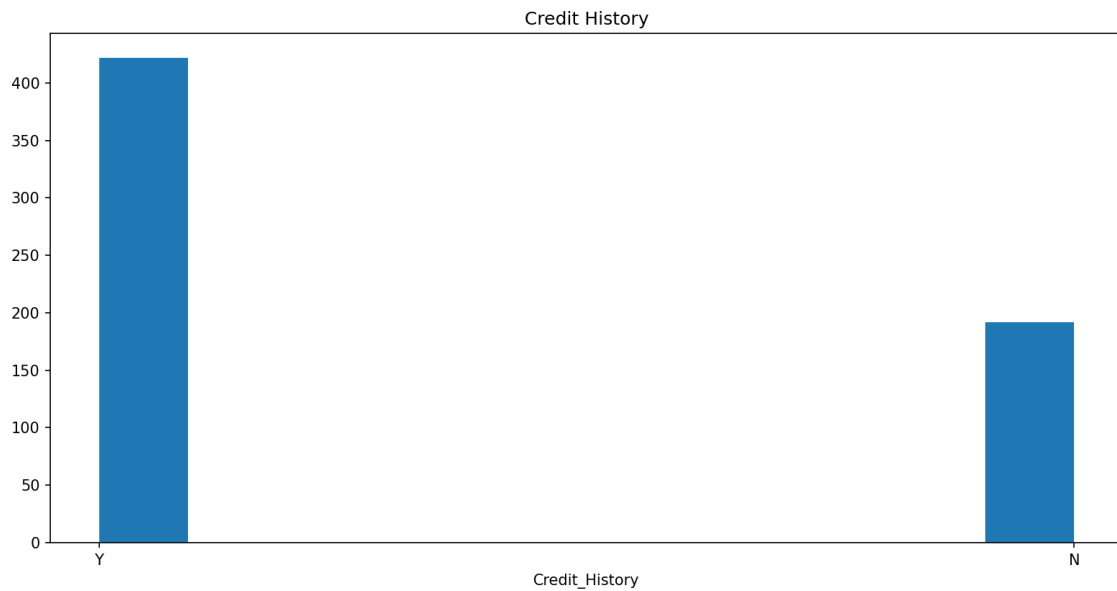
```
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv(r"C:\Users\ishak\Desktop\House Loan Dataset 615.csv")

print(data.head())

plt.hist(data['Loan_Status'], bins='auto')

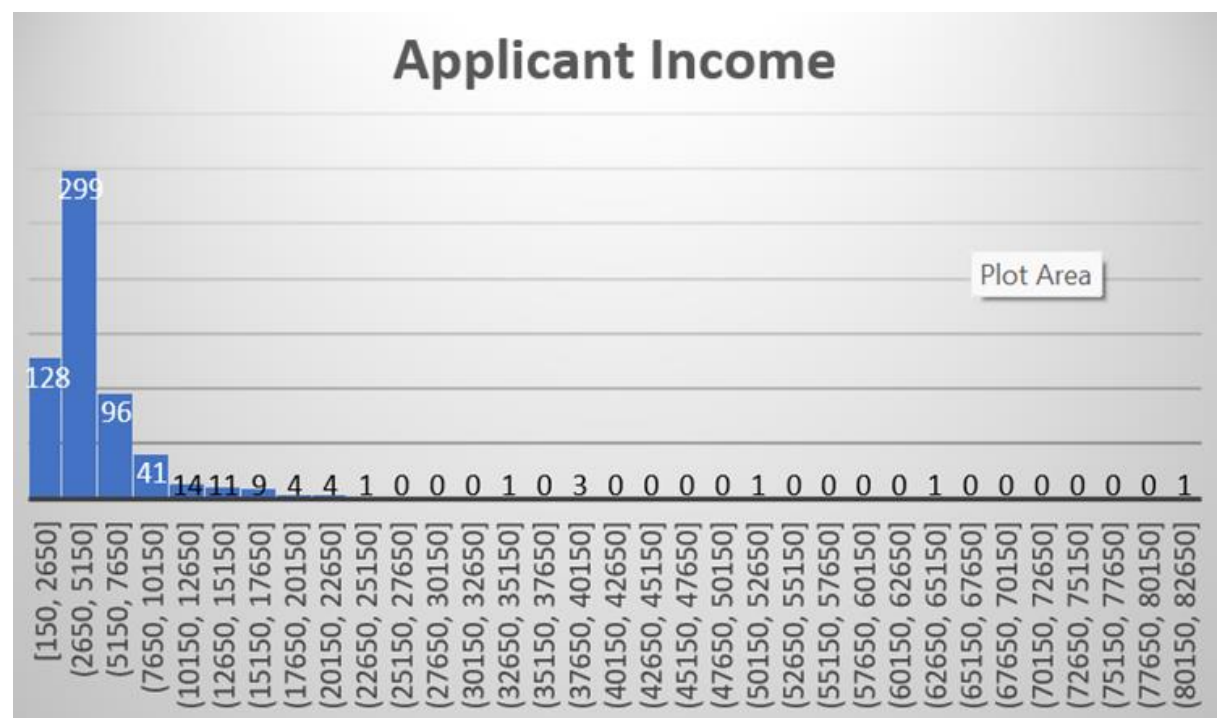
plt.xlabel('Credit_History')
plt.title('Credit History')
plt.show()
```



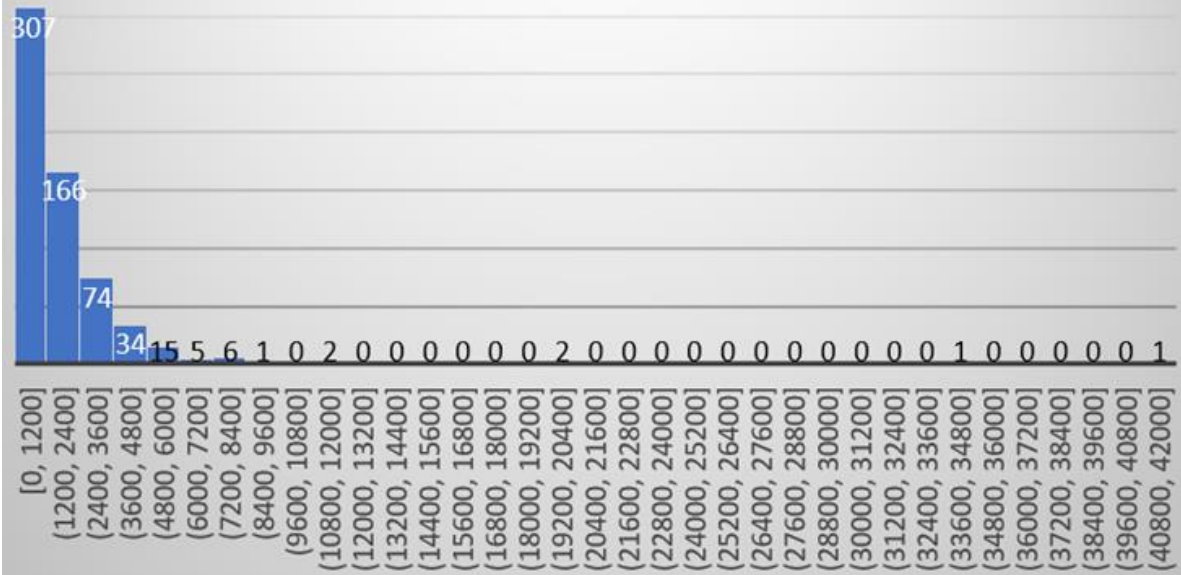
b) Charts and data after the pre-processing

We use histogram, scatter plot and boxplot to visualize the data.

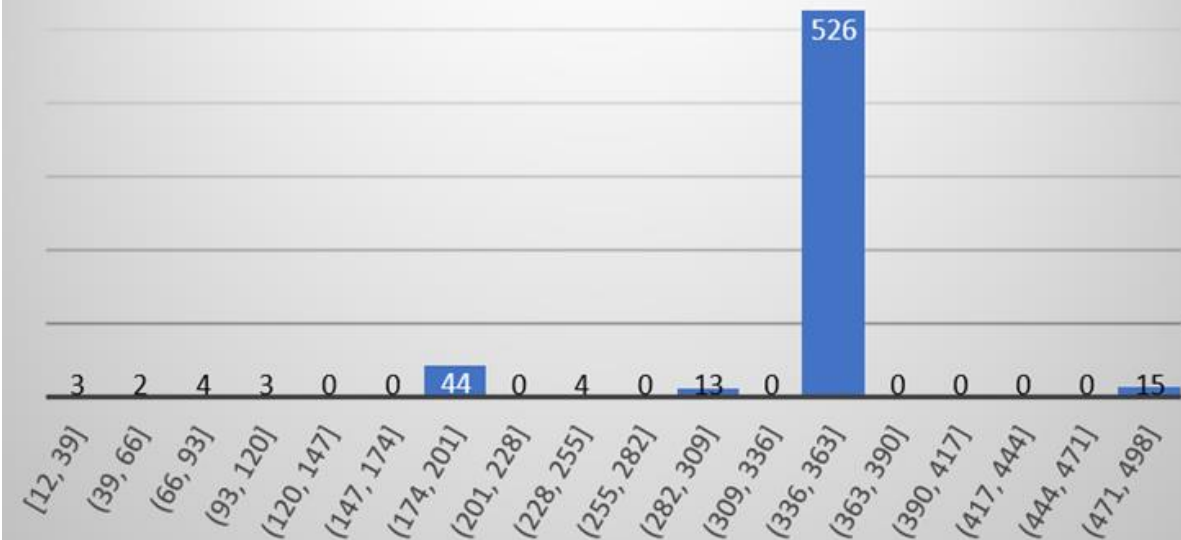
- Histogram



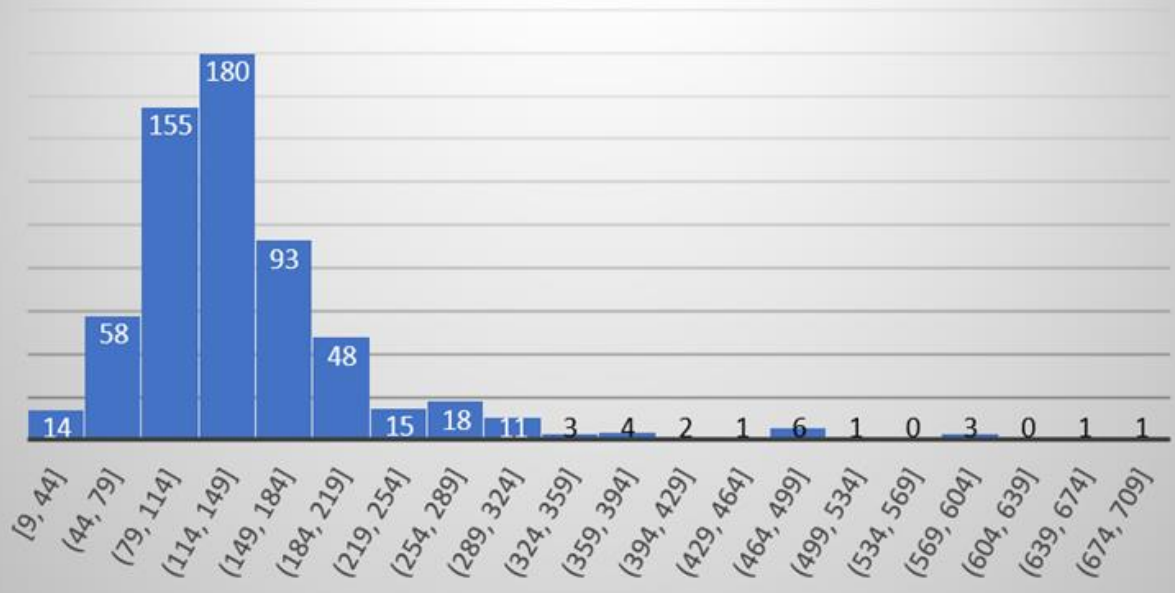
Co-Applicant Income



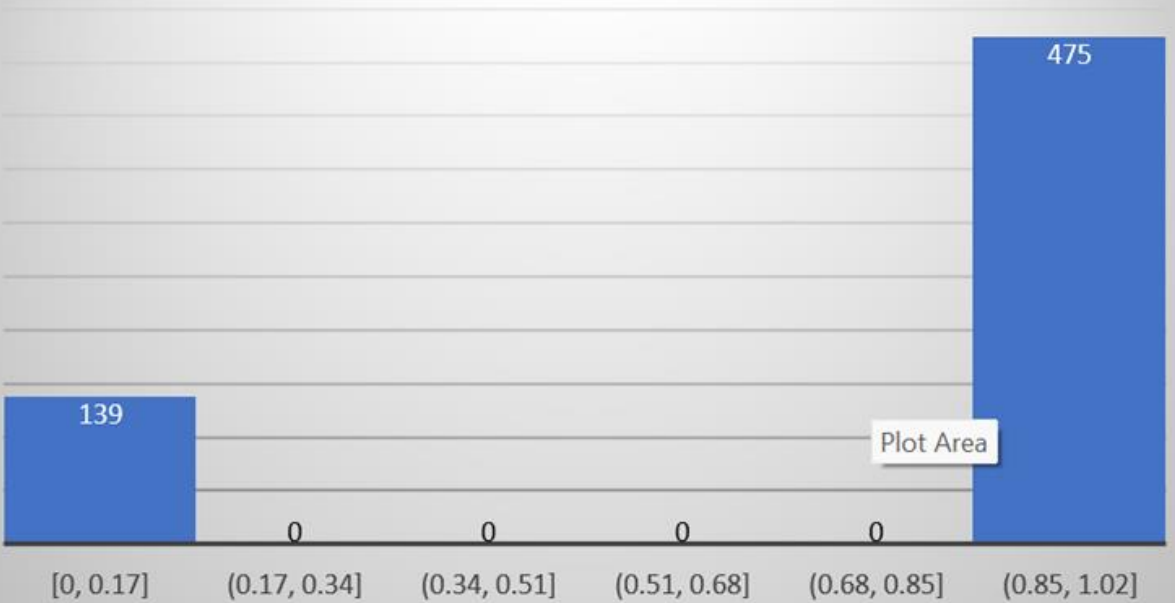
Loan Amount Term

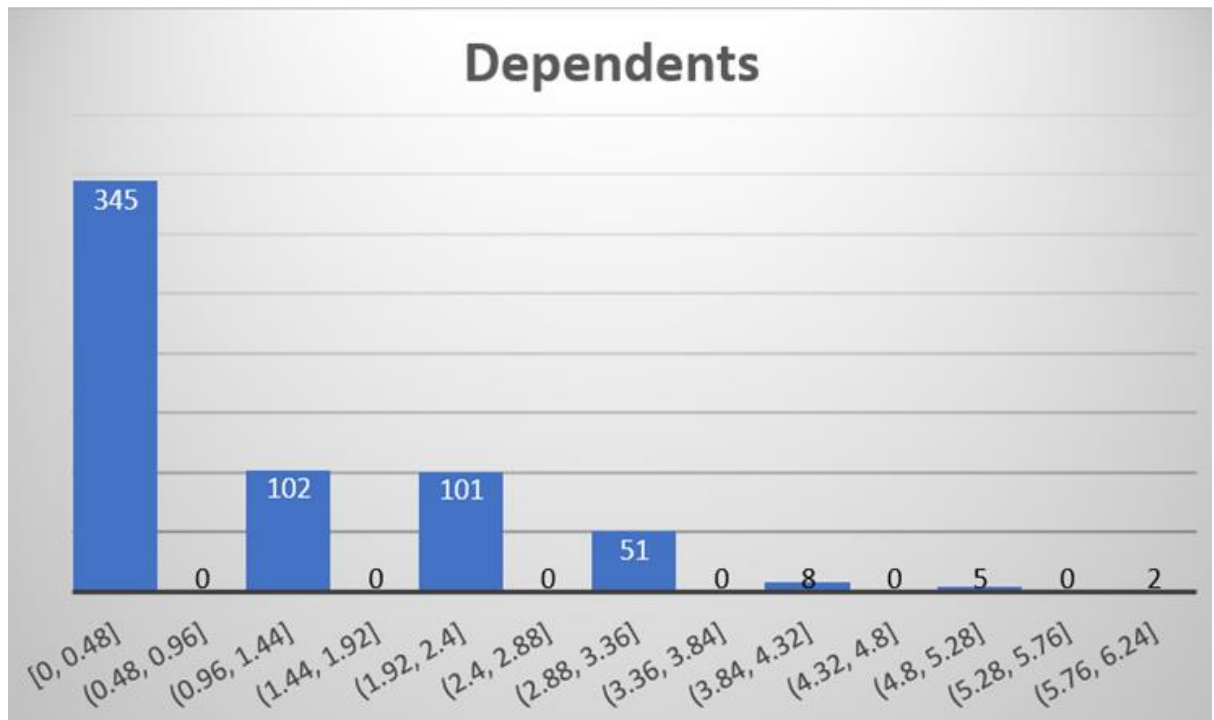


Loan Amount

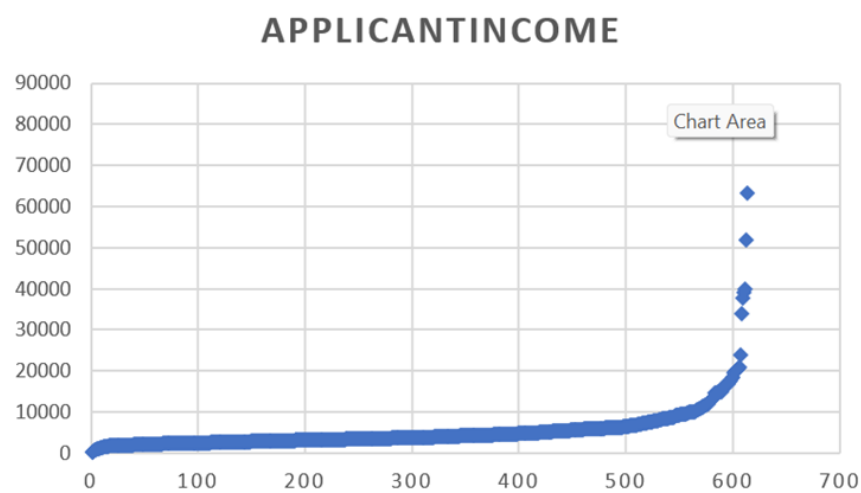


Credit History

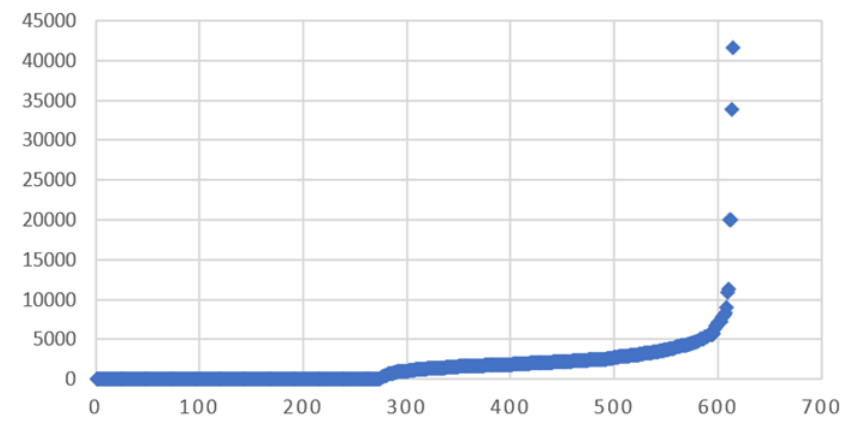




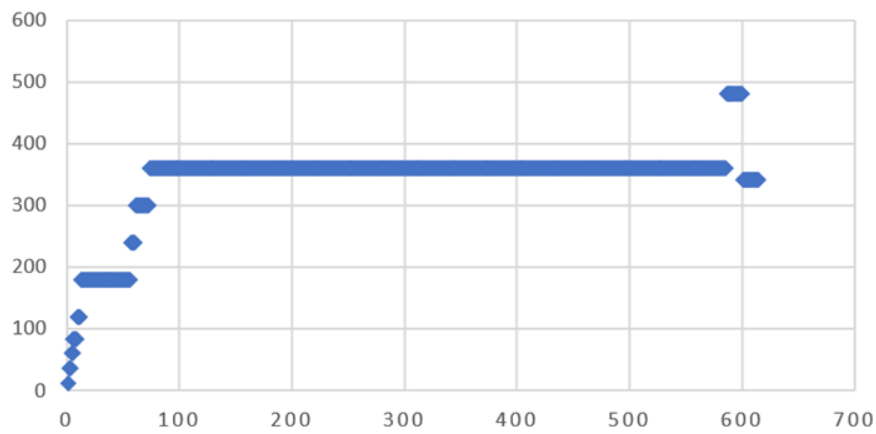
- Scatter Plot



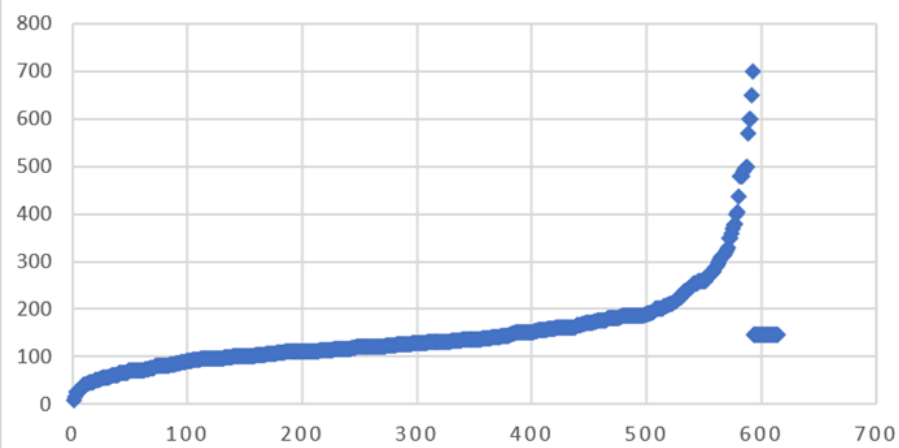
COAPPLICANTINCOME

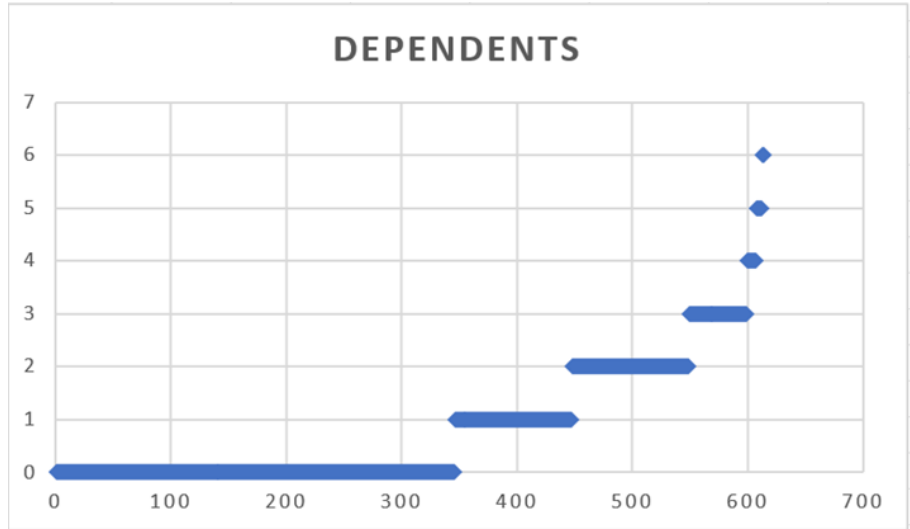
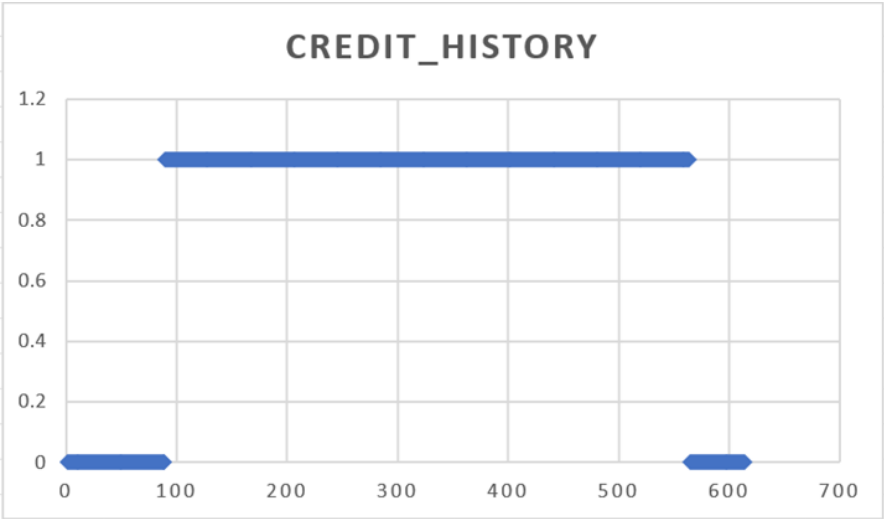


LOAN_AMOUNT_TERM



LOANAMOUNT





- **Box Plot**

