Loan Prediction

Group no: 8

Shashwath Kumar - PES2201800623
Manas V Shetty - PES2201800670
Yogadisha S - PES2201800037

Objectives

- Cleaning the dataset
- Analysing trend of Loan Amount drawn depending on Property
- Analysing Loan Approval depending on Education
- Analysing Loan Approval depending on Credit History
- Normalizing data
- Checking if data is normalized
- Finding Correlation between variables
- Plotting a Simple Linear Regression between Income and Loan
- Hypothesis Testing

Dataset

- Title : Loan Prediction
- Source : https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/
- Description : Analysing Loan Data
- No. of rows : 613
- No. of variables : 6 Numerical, 6 Categorical

Variable Description

Gender - Male/Female	13 Null
Married – Yes/No	3 Null
Dependents – Number of people dependent on Applicant	15 Null
Education – Graduate/Non-Graduate	0 Null
Self-Employed- Yes/No	32 Null
Applicant Income	0 Null
Co applicant Income	0 Null
Loan Amount	22 Null
Loan Amount Term – Time to pay off loan	14 Null
Credit History – If they have paid previous loans	50 Null
Property Area – Urban/Semi-urban/rural	0 Null
Loan Status – Y/N	0 Null

No. of columns: 12

In [604]:
```
data.head(7)
```

Out[604]:

Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area
Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban
Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural
Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban
Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban
Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban
Male	Yes	2	Graduate	Yes	5417	4196.0	267.0	360.0	1.0	Urban
Male	Yes	0	Not Graduate	No	2333	1516.0	95.0	360.0	1.0	Urban

Data Cleaning

- Gender, Married, Credit_History Null values are replaced with ‘Unknown’
- Dependents, Self_Employed, Loan_Amount_Term null values are replaced with mode
- Loan Amount null values replaced with mean
- Remove column Loan_ID
- Changing categorical data in Loan Status to 1s and 0s
- Adding an additional variable – Total Income

```
In [613]: replace_dict = {'Y':1, 'N':0}
          data.Loan_Status = data.Loan_Status.replace(replace_dict)

In [614]: data['TotalIncome'] = data['ApplicantIncome'] + data['CoapplicantIncome']

In [611]: data =data.drop('Loan_ID',axis =1)

In [189]: data['Gender'].fillna('Unknown',inplace =True)
          data['Married'].fillna('Unknown',inplace = True)
          data['Dependents'].fillna(data['Dependents'].mode()[0],inplace = True)
          data['Self_Employed'].fillna(data['Self_Employed'].mode()[0],inplace = True)
          data['Credit_History'].fillna(0.5,inplace = True)
          data['LoanAmount'].fillna(data.LoanAmount.mean(),inplace = True)          #data.LoanAmount.mean() = 128
          data['Loan_Amount_Term'].fillna(360,inplace = True)
```

Out[607]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
23	LP001050	NaN	Yes	2	Not Graduate	No	3365	1917.0	112.0	360.0	0.0
24	LP001052	Male	Yes	1	Graduate	NaN	3717	2925.0	151.0	360.0	NaN
25	LP001066	Male	Yes	0	Graduate	Yes	9560	0.0	191.0	360.0	1.0
26	LP001068	Male	Yes	0	Graduate	No	2799	2253.0	122.0	360.0	1.0
27	LP001073	Male	Yes	2	Not Graduate	No	4226	1040.0	110.0	360.0	1.0
28	LP001086	Male	No	0	Not Graduate	No	1442	0.0	35.0	360.0	1.0
29	LP001087	Female	No	2	Graduate	NaN	3750	2083.0	120.0	360.0	1.0
30	LP001091	Male	Yes	1	Graduate	NaN	4166	3369.0	201.0	360.0	NaN

Comparison

Out[612]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property
23	Unknown	Yes	2	Not Graduate	No	3365	1917.0	112.0	360.0	0	
24	Male	Yes	1	Graduate	No	3717	2925.0	151.0	360.0	Unknown	Sem
25	Male	Yes	0	Graduate	Yes	9560	0.0	191.0	360.0	1	Sem
26	Male	Yes	0	Graduate	No	2799	2253.0	122.0	360.0	1	Sem
27	Male	Yes	2	Not Graduate	No	4226	1040.0	110.0	360.0	1	
28	Male	No	0	Not Graduate	No	1442	0.0	35.0	360.0	1	
29	Female	No	2	Graduate	No	3750	2083.0	120.0	360.0	1	Sem
30	Male	Yes	1	Graduate	No	4166	3369.0	201.0	360.0	Unknown	

Data Normalization and Standardization

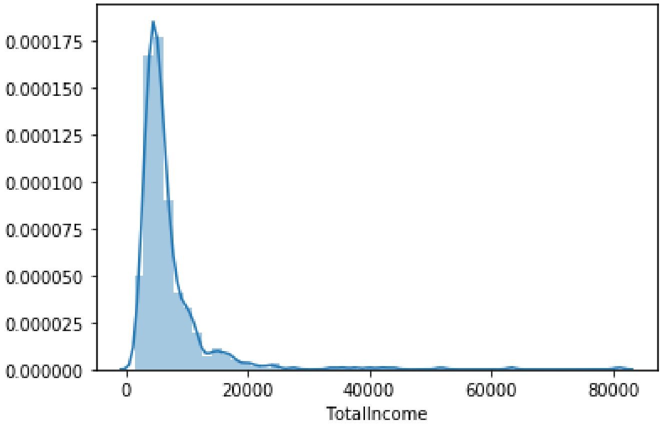
- **Normalization** is the process of organizing a database to reduce redundancy and improve data integrity.
- The goal of **normalization** is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.
- **Data standardization** is this process of making sure that your **data** set can be compared to other **data** sets.
- In our data set we normalize Loan Amount and Total Income so that they can be compared with each other

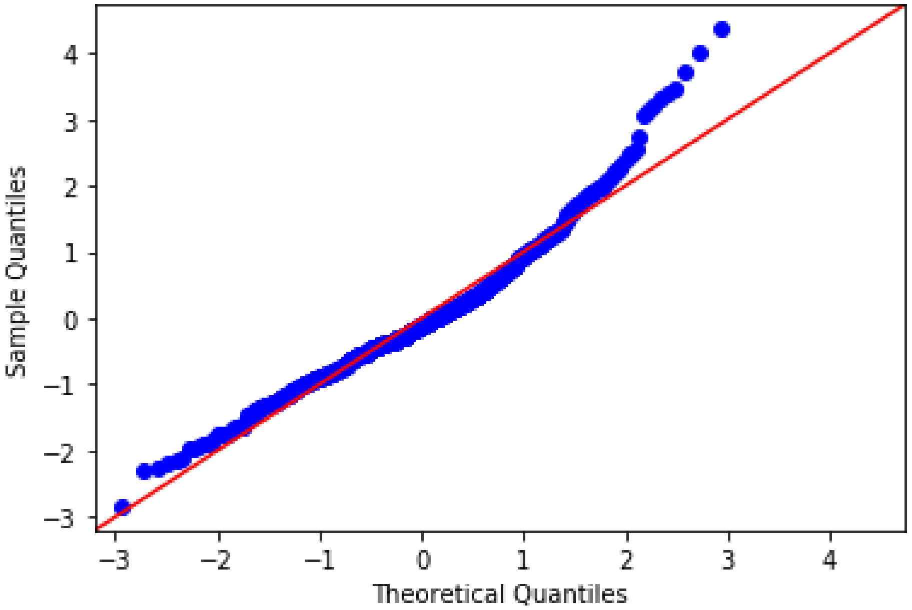
Checking with Q-Q plot

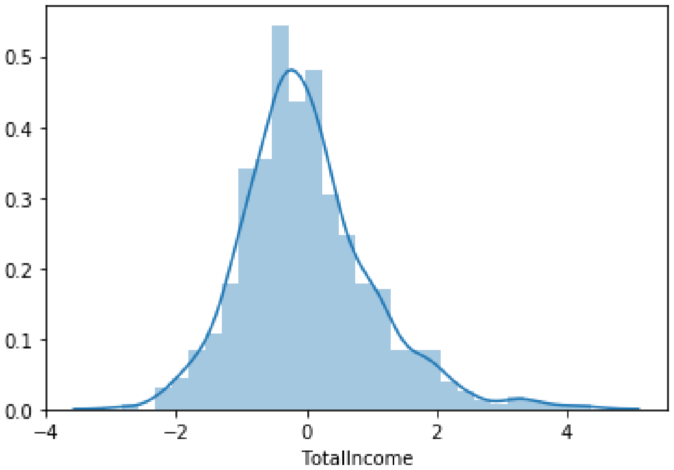
```
In [617]: data.LoanAmount = preprocessing.scale(np.log(data.LoanAmount))
```

```
In [618]: data['TotalIncome'] = preprocessing.scale(np.log(np.log(data.TotalIncome)))
```

- Log is applied to remove the right skewness of the graph

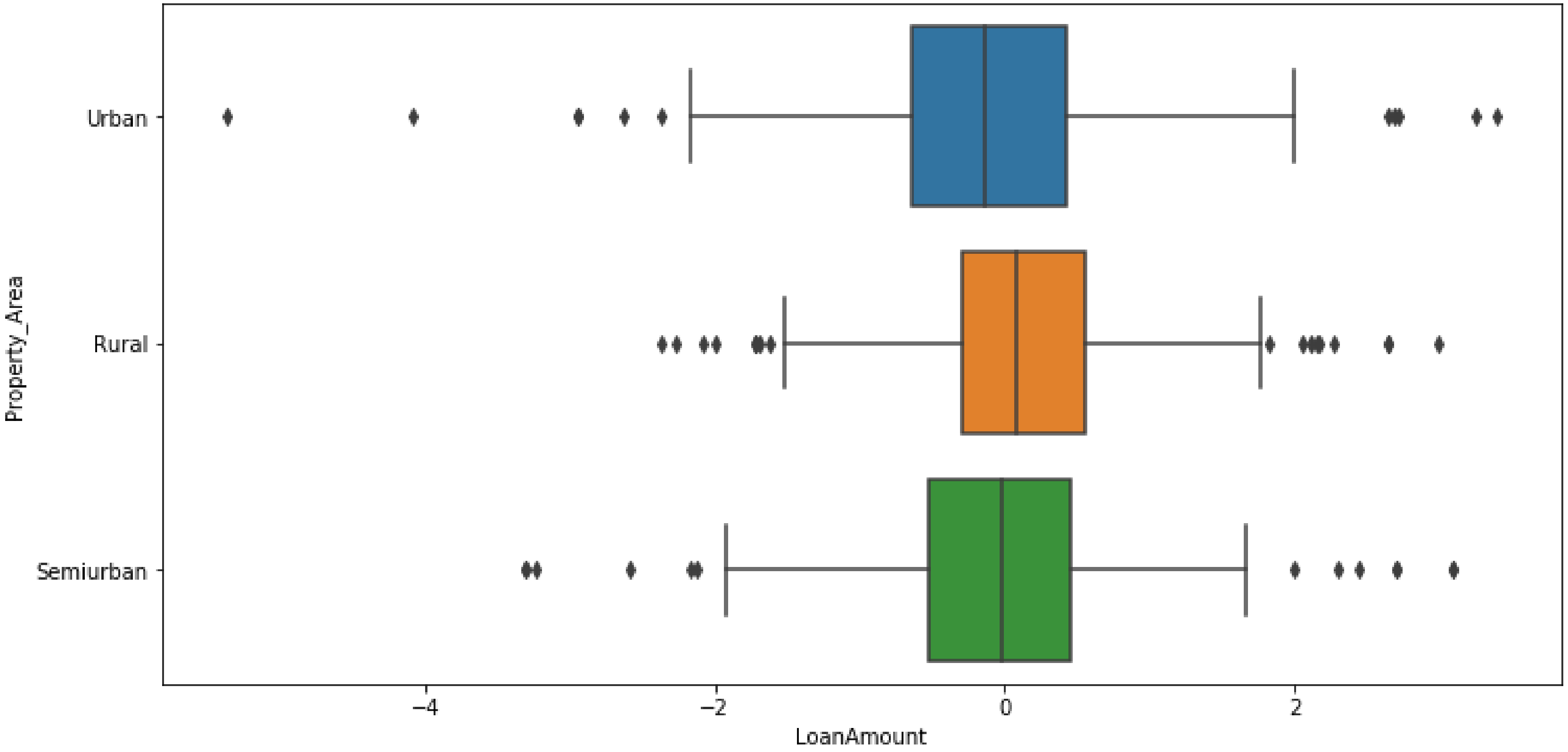






Data Visualization

Trends between Area and Loan taken

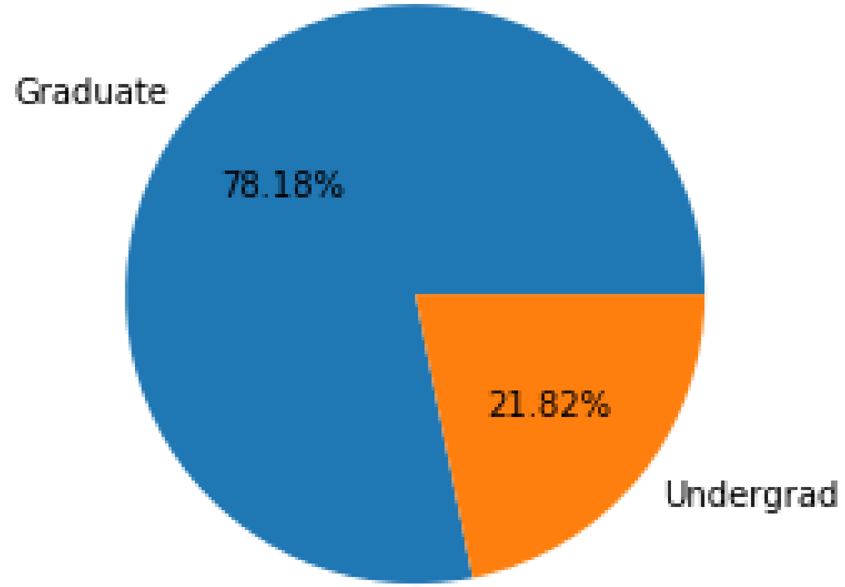


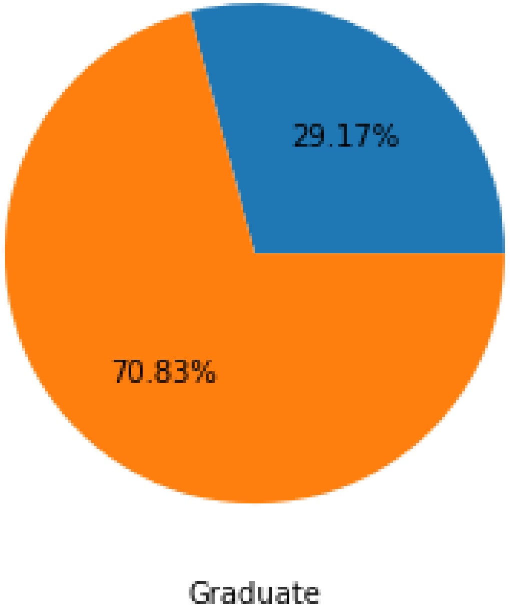
sns.boxplot(x = 'Property_Area', y = 'LoanAmount', data = data, orient = 'v')

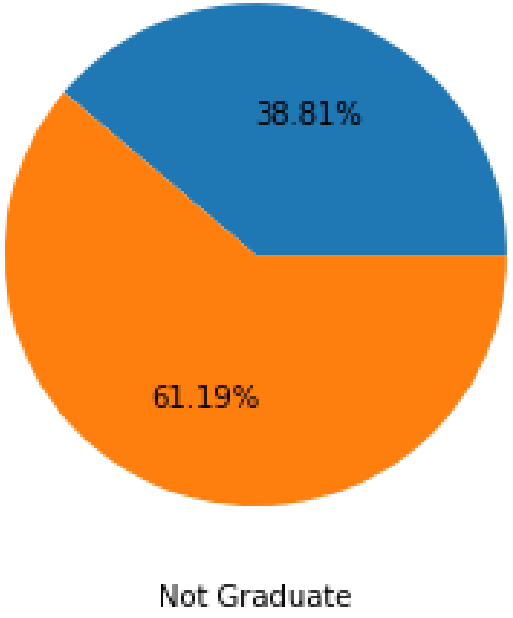
Education and Loan Approval

```
plt.pie(data.Education.value_counts(),labels=['Graduate','Undergrad'],autopct='%1.2f%%')

gbs = data.groupby(by=["Education", "Loan_Status"]).size()

plt.figure(0)
plt.pie([gbs[0],gbs[1]],autopct='%1.2f%%')
plt.xlabel('Graduate')
plt.figure(1)
plt.pie([gbs[2],gbs[3]],autopct='%1.2f%%')
plt.xlabel('Not Graduate')
```

Graduates have a higher rate of Loan Approval



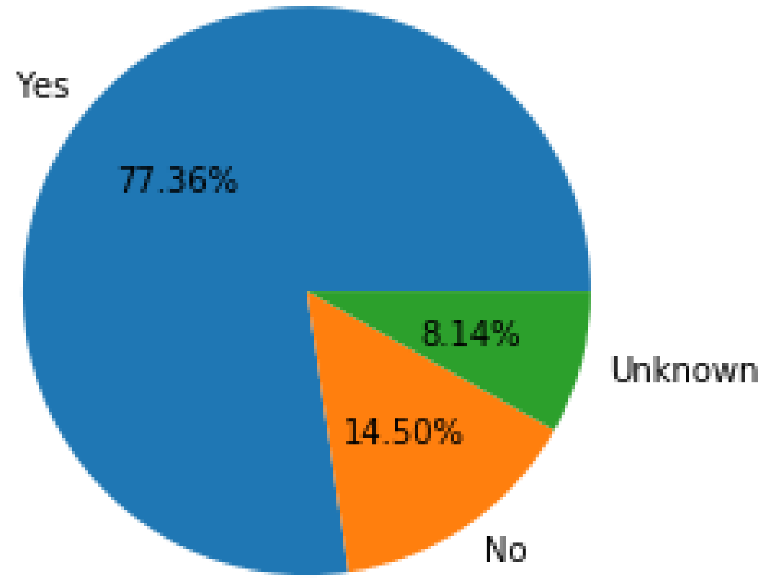


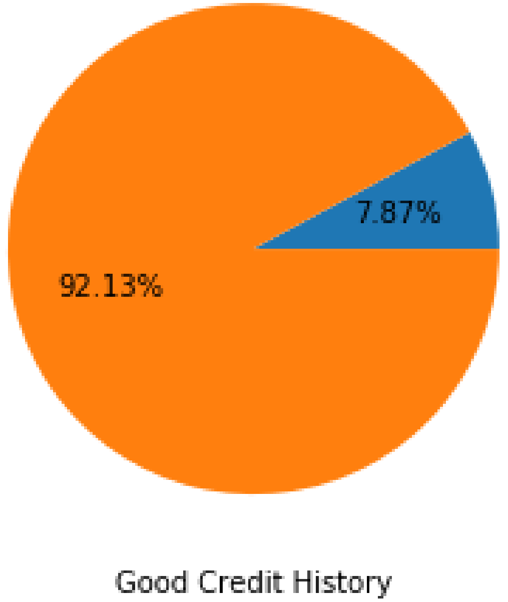


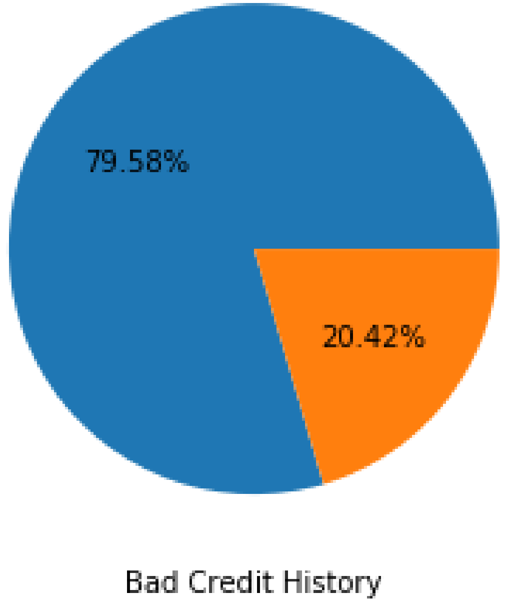
Credit History and Loan Approval

```
plt.pie(data.Credit_History.value_counts(),labels=['Yes','No','Unknown'],autopct='%1.2f%%')

gbs = data.groupby(by=["Credit_History", "Loan_Status"]).tolist().size()

plt.figure(0)
plt.pie([gbs[1],gbs[0]],autopct='%1.2f%%')
plt.xlabel('Good Credit History')
plt.figure(1)
plt.pie([gbs[3],gbs[2]],autopct='%1.2f%%')
plt.xlabel('Bad Credit History')
```

Loan Approved 92% of the time for people with Good credit history







Exploratory Analysis

- All three property areas have similar means and interquartile ranges.
- However, we can see that urban properties have a much wider range.

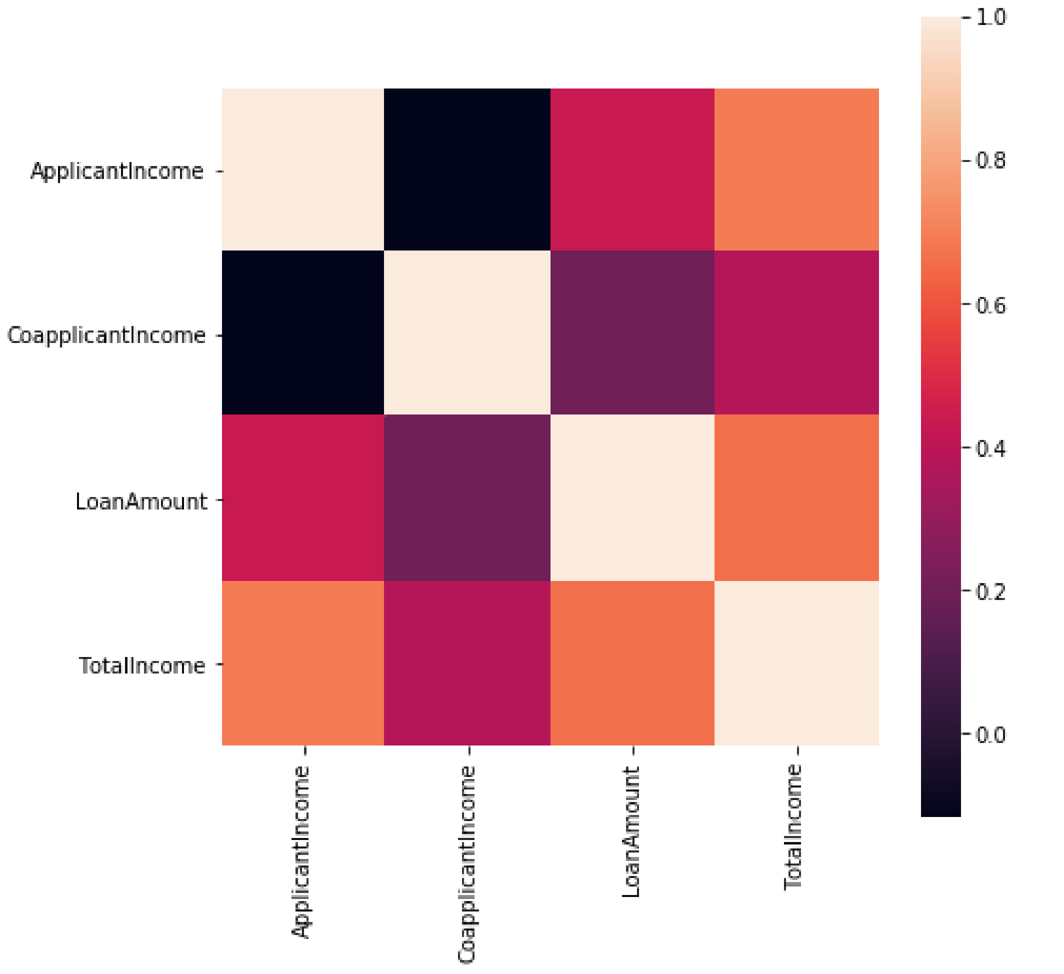
- Graduates make up the vast majority of applicants for loans.
- Graduates also have a higher loan approval rate compared to non graduates

- Only 20% of applicants were given approval with a bad credit history
- But, 92% of applicants were approved with a good credit history

Correlation and Regression

Correlation

statistical technique that can show whether and how strongly pairs of variables are related.




```
In [622]: corr = data.select_dtypes(include = ['float64', 'int64']).corr()
          plt.figure(figsize=(7, 7))
          sns.heatmap(corr, vmax=1, square=True)
          plt.show()
```

Out[623]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Loan_Status	TotalIncome
ApplicantIncome	1.000000	-0.116605	0.434849	-0.046531	-0.004710	0.691849
CoapplicantIncome	-0.116605	1.000000	0.204179	-0.059383	-0.059187	0.379764
LoanAmount	0.434849	0.204179	1.000000	0.084616	-0.041874	0.662631
Loan_Amount_Term	-0.046531	-0.059383	0.084616	1.000000	-0.022549	-0.054548
Loan_Status	-0.004710	-0.059187	-0.041874	-0.022549	1.000000	0.012783
TotalIncome	0.691849	0.379764	0.662631	-0.054548	0.012783	1.000000

From the graph we can see that Loan Amount and Total Income are correlated with r=0.66

Regression

a statistical approach to find the relationship between variables

- In our problem, we are going to use Simple Linear Regression.

  Y=a + bx
- Plotting the regression curve between Loan Amount and Total Income.

```
In [633]: b = estimate_coef(data.TotalIncome,data.LoanAmount)

          plot_regression_line(data.TotalIncome,data.LoanAmount,b)
```

- Applying the formula, we get the coefficients to be:

  a = -7.591214173554817e-16

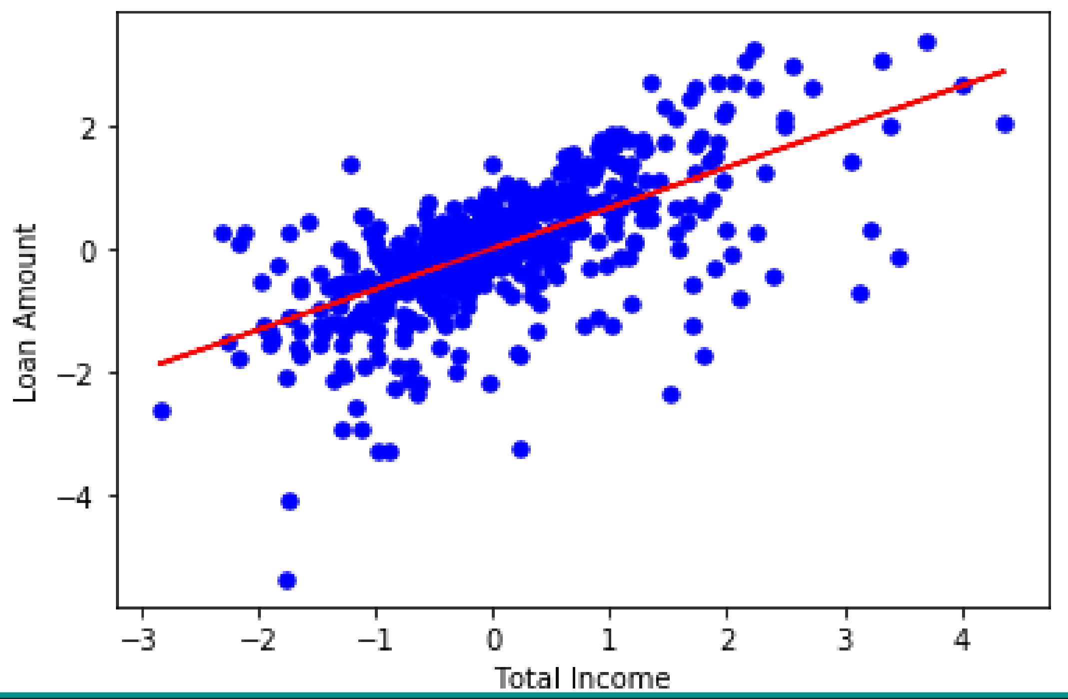
  b = 0.6626313083498345

Regression Formula: $\mathbf{Y} = \mathbf{a} + \mathbf{bX} + \boldsymbol{\epsilon}$

$Y = a + bX$

$$a = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$b = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$



```python
def estimate_coef(x, y):
    n = np.size(x)

    m_x, m_y = np.mean(x), np.mean(y)

    numa = np.sum(y*x) - n*m_y*m_x
    numb = np.sum(x*x) - n*m_x*m_x

    b_1 = numa / numb
    b_0 = m_y - b_1*m_x

    return(b_0, b_1)
```

```python
def plot_regression_line(x, y, b):
    plt.scatter(x, y, color = "b",
                marker = "o", s = 30)

    y_pred = b[0] + b[1]*x

    plt.plot(x, y_pred, color = "r")
    plt.xlabel('Total Income')
    plt.ylabel('Loan Amount')

    plt.show()
```

Hypothesis Testing

H0 : Loan Amount and Co applicant Income are Independent

Using Pearson's correlation test

In [637]:
```
from scipy.stats import pearsonr
data1 = data.CoapplicantIncome
data2 = data.LoanAmount
stat, p = pearsonr(data1, data2)
print('stat=',stat,', p=',p)
if p > 0.05:
    print('Failed to reject H0')
else:
    print('Reject H0')
```

```
stat= 0.20417874816300638 , p= 3.3450057734449784e-07
Reject H0
```

We find that the p-value < 0.05, So we can reject the hypothesis.

Co applicant Income and Loan Amount are correlated

H0 : Loan Amount and Loan Status are Independent

Using Pearson's correlation test

```
In [639]: from scipy.stats import pearsonr
          data1 = data.Loan_Status
          data2 = data.LoanAmount
          stat, p = pearsonr(data1, data2)
          print('stat=',stat,', p=',p)
          if p > 0.05:
              print('Failed to reject H0')
          else:
              print('Reject H0')

          stat= -0.04187358290777825 , p= 0.3002364101035521
          Failed to reject H0
```

We find that the p-value > 0.05, So we cannot reject the hypothesis.

Loan Status and Loan Amount are probably independent

H0 : Credit History and Loan Status are Independent

Using Pearson's correlation test

```
In [253]: from scipy.stats import pearsonr
          data1 = data.Credit_History
          data2 = data.Loan_Status
          stat, p = pearsonr(data1, data2)
          print('stat=',stat,', p=',p)
          if p > 0.05:
              print('Failed to reject H0')
          else:
              print('Reject H0')

          stat= 0.5133194232478169 , p= 1.4173527081623127e-42
          Reject H0
```

We find that the p-value << 0.05, So we cannot reject the hypothesis.

Credit History and Loan Status are dependent.

Conclusion

- Loan Amount and Total Income are linearly dependent. R=0.66
- Applicant Income also forms a linear relationship with Loan Amount. R=0.43
- Co applicant Income also forms a linear relationship with Loan Amount. R=0.20

- Credit History affects Loan status.
- Loan Status is independent of Loan Amount, so credibility does not correlate to loan taken.

THANK YOU