Loan Approval : Data Analysis

In this project, we explored data from people who applied for loans. Our goal is to understand what kind of people usually get loans approved and what information banks look at before saying "yes" or "no." We used data analysis and visual tools to see patterns in the data. We'll cover initial data inspection, handling missing values, and various univariate, bivariate, and multivariate analysis to understand the data's characteristics and relationships.

Presented by: Abhishek Bhagat

Project Objective

- Data Exploration: Initial insights into the dataset.
- Dealing with Null Values: Cleaning the data for accuracy.
- Data Visualization: Understanding distributions and relationships.
- Univariate Analysis: Examining single variables.
- Bivariate Analysis: Exploring relationships between two variables.
- Multivariate Analysis: Discovering complex interactions.

We wanted to study what factors affect loan approvals. This includes income, education, property area, and more. First, we cleaned the data to remove problems. Then, we made charts to see trends. This helps us learn how banks might decide who gets a loan.

Dataset Overview

- Total loan applications: **367**
- Each application has **12 details** (like income, gender, etc.)
- Some data is missing in places.

Our dataset has 367 people who applied for loans. Each person has information like how much they earn, if they are married, if they are self-employed, and more. Some of this information was missing, which we needed to fix before doing any analysis.

```
df.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	LP001015	Male	Yes	0	Graduate	No	5720	0	110.0	360.0	1.0
1	LP001022	Male	Yes	1	Graduate	No	3076	1500	126.0	360.0	1.0
2	LP001031	Male	Yes	2	Graduate	No	5000	1800	208.0	360.0	1.0
3	LP001035	Male	Yes	2	Graduate	No	2340	2546	100.0	360.0	NaN
4	LP001051	Male	No	0	Not Graduate	No	3276	0	78.0	360.0	1.0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            367 non-null    object
 1   Gender             356 non-null    object
 2   Married            367 non-null    object
 3   Dependents         357 non-null    object
 4   Education          367 non-null    object
 5   Self_Employed      344 non-null    object
 6   ApplicantIncome    367 non-null    int64
 7   CoapplicantIncome  367 non-null    int64
 8   LoanAmount         362 non-null    float64
 9   Loan_Amount_Term   361 non-null    float64
 10  Credit_History     338 non-null    float64
 11  Property_Area      367 non-null    object
dtypes: float64(3), int64(2), object(7)
```

Data Cleaning

- Removed rows with missing or blank information.
- Final number of clean records: **289**
- Made sure no repeated (duplicate) rows exist.

Before we can study the data, we need to make sure it is complete. We removed rows where important details were missing — like income or credit history. After cleaning, we had 289 complete applications that we could trust for our analysis.

```
df.isnull().sum()

Loan_ID               0
Gender               11
Married               0
Dependents           10
Education             0
Self_Employed        23
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            5
Loan_Amount_Term      6
Credit_History       29
Property_Area         0
dtype: int64
```

```
df.dropna(inplace=True)
```

```
df.isnull().sum()

Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
dtype: int64
```

```
df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 289 entries, 0 to 366
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            289 non-null    object
 1   Gender             289 non-null    object
 2   Married            289 non-null    object
 3   Dependents         289 non-null    object
 4   Education          289 non-null    object
 5   Self_Employed      289 non-null    object
 6   ApplicantIncome    289 non-null    int64
 7   CoapplicantIncome  289 non-null    int64
 8   LoanAmount         289 non-null    float64
 9   Loan_Amount_Term   289 non-null    float64
 10  Credit_History     289 non-null    float64
 11  Property_Area      289 non-null    object
dtypes: float64(3), int64(2), object(7)
memory usage: 29.4+ KB
```

Summary of Numbers

- **Average applicant income**: 4,637
- **Highest income**: 72,529
- **Average loan amount**: 137,000
- Most people asked for 30-year loans (360 months)

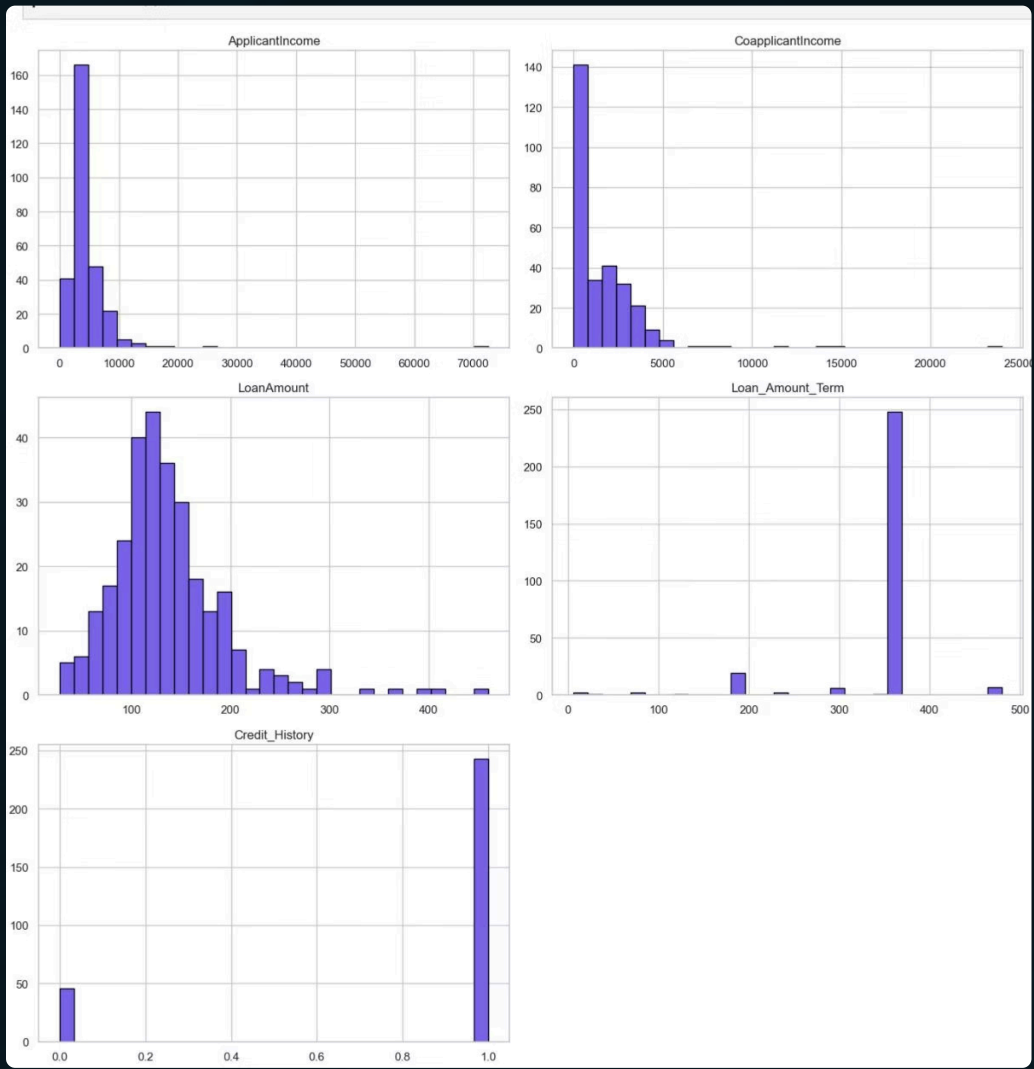
This slide gives a quick look at the numbers in our data. Most people earn around 4,000 to 5,000 per month, but some people earn much more. The average loan request was 137,000. Most people ask for long-term loans — usually for 30 years.

```
df.describe()
```

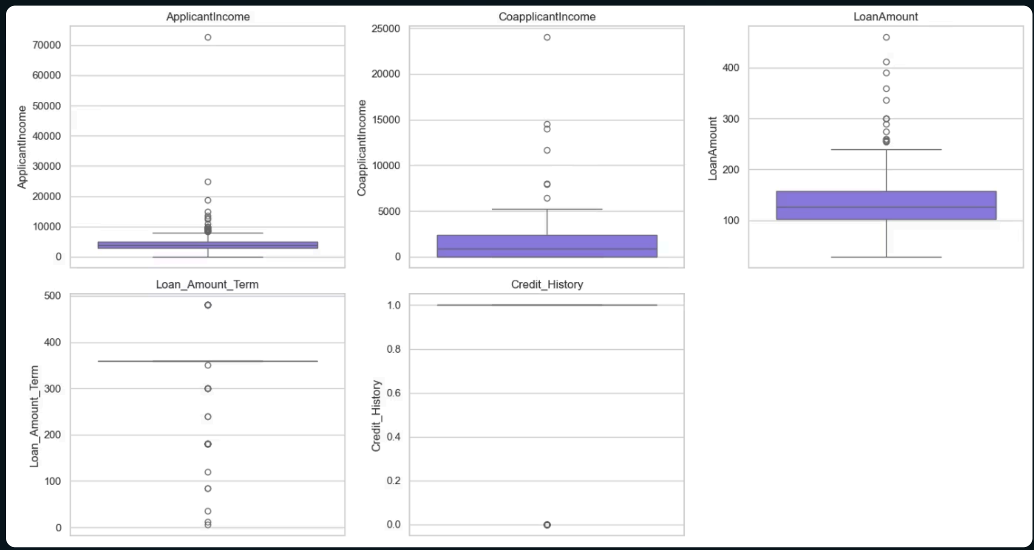
	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	289.000000	289.000000	289.000000	289.000000	289.000000
mean	4637.352941	1528.262976	136.792388	342.671280	0.840830
std	4790.683934	2377.599209	59.699582	65.655503	0.366469
min	0.000000	0.000000	28.000000	6.000000	0.000000
25%	2875.000000	0.000000	102.000000	360.000000	1.000000
50%	3833.000000	879.000000	126.000000	360.000000	1.000000
75%	5000.000000	2400.000000	158.000000	360.000000	1.000000
max	72529.000000	24000.000000	460.000000	480.000000	1.000000

Univariate Analysis: Numeric Variables

We made simple charts to see how income and loan amounts are spread. Most people earn moderate incomes, but a few earn very high amounts — these are called "outliers." These charts help us understand the range of our data and spot any unusual values.



Histograms show how many people fall into each income or loan range.

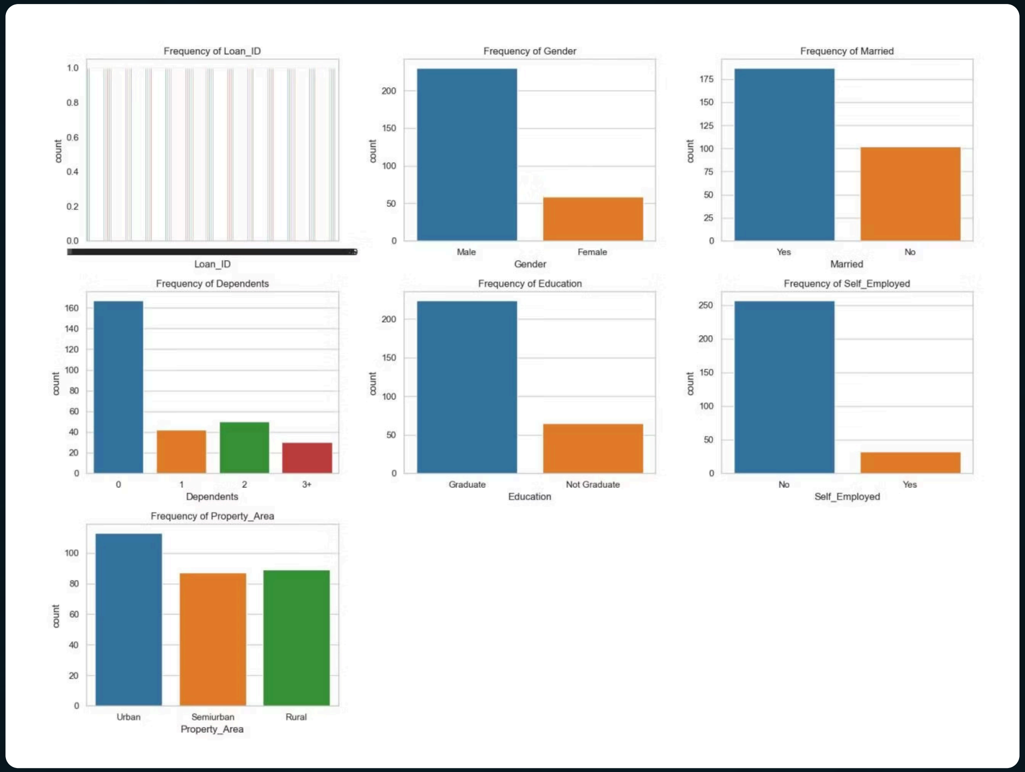


Boxplots help spot outliers (people with very high or low values)

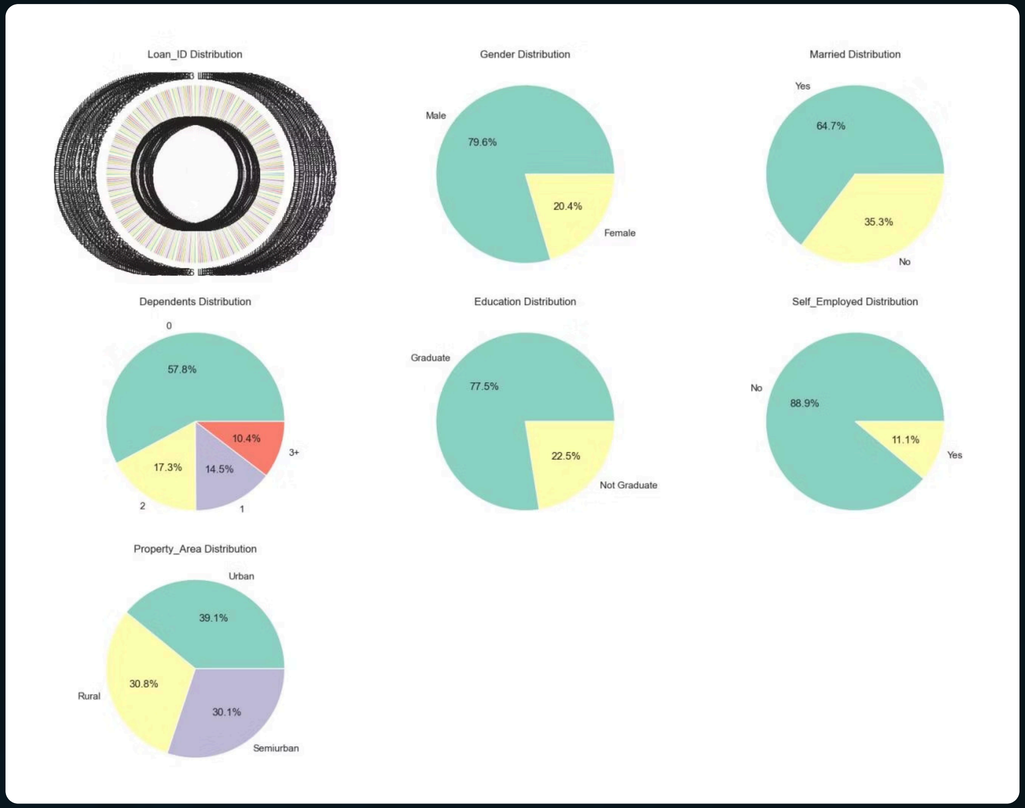
Made with GAMMA

Univariate Analysis: Categorical Variables

We looked at text-based information like gender, education, and property area. Bar charts show the number of people in each group. Pie charts show the percentage share of each group. We saw that most applicants are married male graduates from semi-urban areas.



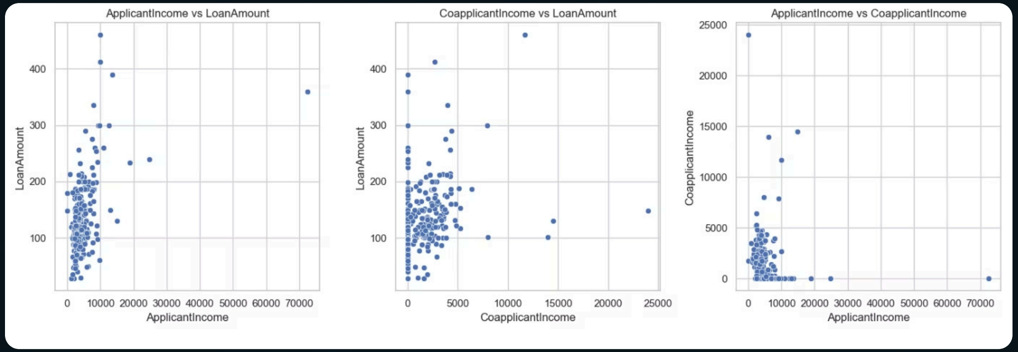
Bar charts: Show counts (e.g., how many men vs. women applied)



Pie charts: Show portions like slices (e.g., % of graduates vs. non-graduates)

Bivariate Analysis: Numeric Relationships

- **Income vs. Loan Amount**: Higher income usually means bigger loan.
- **Boxplots**: Show how loan amounts change based on education or area.



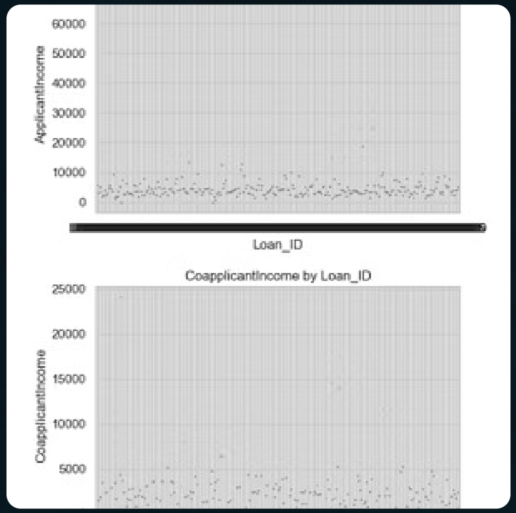
Here we compared two things at a time. For example, we checked if people who earn more get bigger loans — the answer is mostly yes. We also looked at how loan amounts change based on education or where the applicant lives (rural, semiurban, urban).

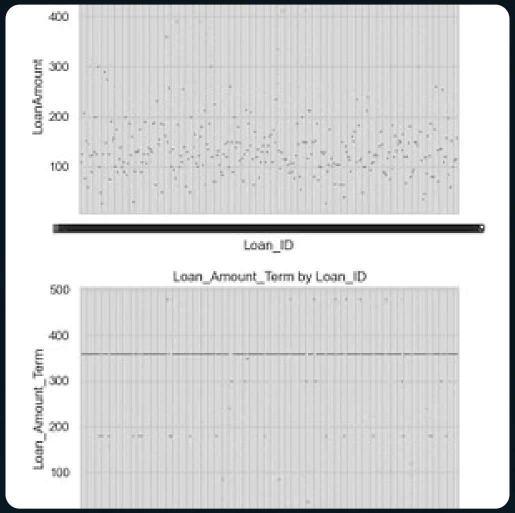
```
sns.set(style="whitegrid")
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
sns.scatterplot(x=df['ApplicantIncome'], y=df['LoanAmount'])
plt.title('ApplicantIncome vs LoanAmount')

plt.subplot(1, 3, 2)
sns.scatterplot(x=df['CoapplicantIncome'], y=df['LoanAmount'])
plt.title('CoapplicantIncome vs LoanAmount')

plt.subplot(1, 3, 3)
sns.scatterplot(x=df['ApplicantIncome'], y=df['CoapplicantIncome'])
plt.title('ApplicantIncome vs CoapplicantIncome')

plt.tight_layout()
plt.show()
```

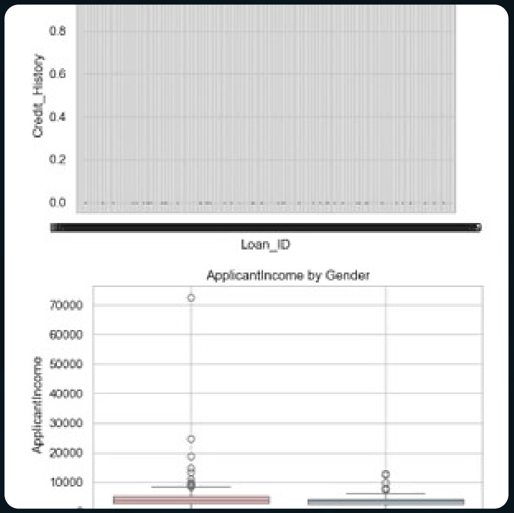
Bivariate Analysis: Categorical & Numeric

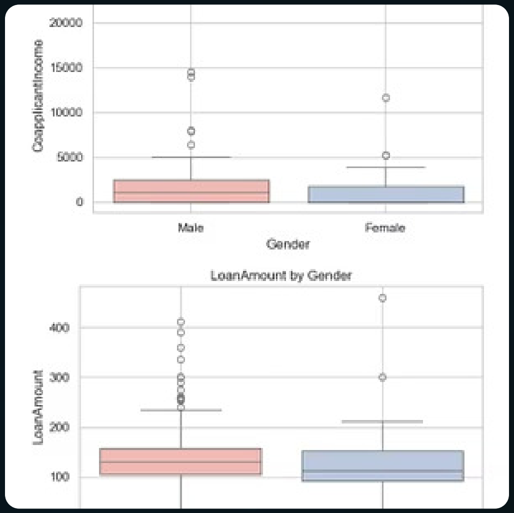
The analysis utilized box plots and violin plots to explore the relationship between categorical and numeric variables. These visualizations provided valuable insights into how the distributions of numeric variables varied across different categories.

Box plots allowed us to examine the central tendency, spread, and outliers of the numeric data for each category. This helped identify any significant differences in the numeric distributions between the groups defined by the categorical variables.





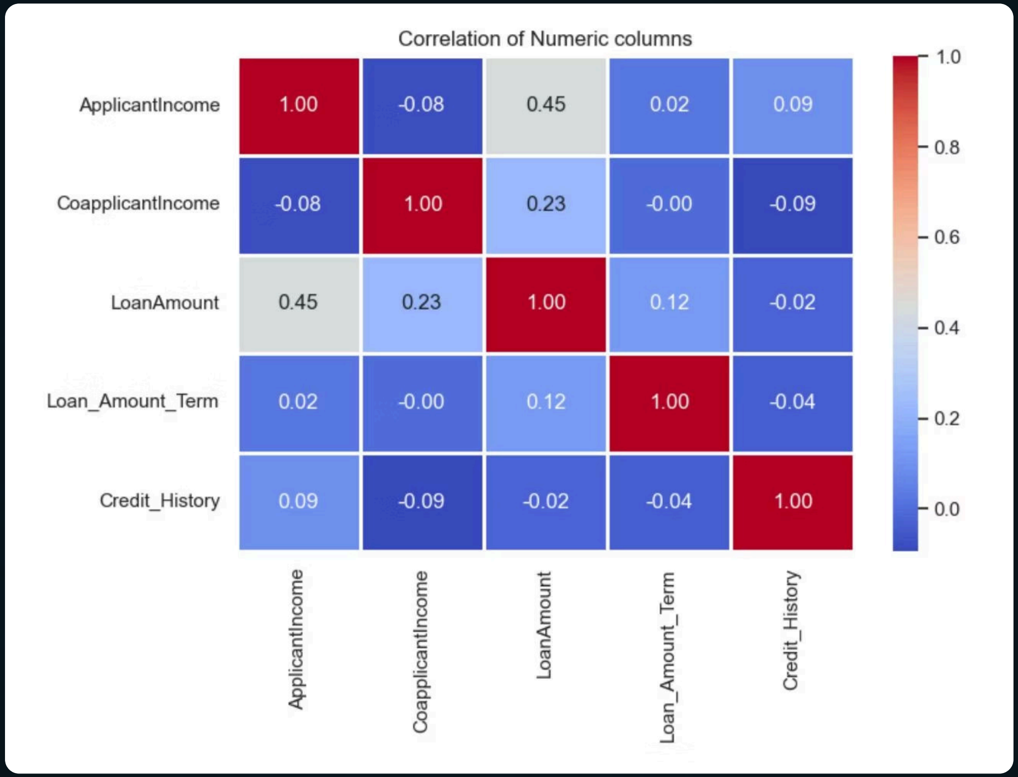




Multivariate Analysis: Correlations

- **Heatmap**: A colorful chart that shows relationships between numbers.
- Income and loan amount have a medium connection.
- Credit history doesn't strongly connect with income or loan size.

We used a heatmap to show how all the numbers relate to each other. A brighter color means a stronger connection. For example, people with higher incomes often ask for higher loans, but the connection is not very strong. Credit history doesn't show much direct link in this chart but is still important.



Key Takeaways & Insights

1. People with higher income ask for bigger loans.
2. **Credit history is very important** — may affect approval.
3. **Semiurban areas** have more loan applicants than other areas.
4. Some data has unusual or very large values (outliers).

From our study, we found that income affects the loan amount. Credit history, even though it doesn't show strong connection in the heatmap, is likely used by banks to make final decisions. Most loan applications come from semiurban areas. We also found a few very high incomes and loan amounts that are far from the average — these are outliers.

Conclusion

In conclusion, we cleaned the data, explored it with charts, and found several interesting insights. We now understand what types of people ask for loans and what may affect approvals. This project is a great starting point if we want to build a system that automatically predicts loan approvals.

- Data was cleaned and explored carefully.
- We found useful patterns in the loan data.
- Charts helped us see what matters most in loan approvals.
- This analysis can be used in the future for building **loan prediction models**.

References

Tools: Python (pandas, seaborn, matplotlib), Jupyter Notebook

LinkedIn- www.linkedin.com/in/abhishek-bhagat-15a005370

GitHub- [hhttps://github.com/Abhishek-0502-Bhagat/Loan-Approval]

Dataset- [hhttps://drive.google.com/file/d/1lCRryHkGizmdtDMK3DbVjAeWkZUCdMkz/view?usp=sharing]