Loan Approval Analysis

Name: Aman

Role: Data Analyst

Contact: 8400795449

Introduction

This capstone project presents a comprehensive exploratory data analysis of loan approval patterns using a real-world financial dataset. The analysis examines the relationship between applicant characteristics and loan approval decisions, providing critical insights into the factors that influence lending outcomes.

Key Research Questions

- **Do married applicants request higher loan amounts compared to unmarried applicants?**
- **Does gender influence the amount of loan requested?**
- **Do applicants with more dependents take larger loans?**
- **Is there a strong relationship between applicant income and the loan amount they request?**
- **Do applicants in Urban areas request bigger loans compared to Rural or Semi-Urban areas?**
- **Do self employed applicants request larger loans compared to salaried applicants?**

Data Cleaning

Step 1: Importing The Dataset

```
import pandas as pd
import numpy as np
```

```
df = pd.read_csv("/content/loan_sanction_test.csv")
df
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area
0	LP001015	Male	Yes	0	Graduate	No	5720	0	110.0	360.0	1.0	Urban
1	LP001022	Male	Yes	1	Graduate	No	3076	1500	126.0	360.0	1.0	Urban
2	LP001031	Male	Yes	2	Graduate	No	5000	1800	208.0	360.0	1.0	Urban
3	LP001035	Male	Yes	2	Graduate	No	2340	2546	100.0	360.0	NaN	Urban
4	LP001051	Male	No	0	Not Graduate	No	3276	0	78.0	360.0	1.0	Urban
...	...	...	...	...	...	...	...	...	...	...	...	...
362	LP002971	Male	Yes	3+	Not Graduate	Yes	4009	1777	113.0	360.0	1.0	Urban
363	LP002975	Male	Yes	0	Graduate	No	4158	709	115.0	360.0	1.0	Urban
364	LP002980	Male	No	0	Graduate	No	3250	1993	126.0	360.0	NaN	Semiurban
365	LP002986	Male	Yes	0	Graduate	No	5000	2393	158.0	360.0	1.0	Rural
366	LP002989	Male	No	0	Graduate	Yes	9200	0	98.0	180.0	1.0	Rural

367 rows × 12 columns

Step 2: Information About Dataset

```
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)
memory usage: 34.5+ KB
```

Shape

```
df.shape
```

```
(367, 12)
```

Step 3: Finding Missing Values

```
print("Null Values By %")
print((df.isnull().sum() / len(df)*100).map("{:.2f} %".format))
```

```
Null Values By %
Loan_ID             0.00 %
Gender              3.00 %
Married             0.00 %
Dependents          2.72 %
Education           0.00 %
Self_Employed       6.27 %
ApplicantIncome     0.00 %
CoapplicantIncome   0.00 %
LoanAmount          1.36 %
Loan_Amount_Term    1.63 %
Credit_History      7.90 %
Property_Area       0.00 %
dtype: object
```

Step 4: Basic Statistics By Numerical Columns

```
df.describe()
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	367.000000	367.000000	362.000000	361.000000	338.000000
mean	4805.599455	1569.577657	136.132597	342.537396	0.825444
std	4910.685399	2334.232099	61.366652	65.156643	0.380150
min	0.000000	0.000000	28.000000	6.000000	0.000000
25%	2864.000000	0.000000	100.250000	360.000000	1.000000
50%	3786.000000	1025.000000	125.000000	360.000000	1.000000
75%	5060.000000	2430.500000	158.000000	360.000000	1.000000
max	72529.000000	24000.000000	550.000000	480.000000	1.000000

Step 5: Handing Missing Values

1.

I Filled Missing Values With Most Suitable Alternatives

```
df['Credit_History'].unique()
array([ 1., nan,  0.])

df["Credit_History"] = df['Credit_History'].fillna("No Records")

df['Self_Employed'].unique()
array(['No', 'Yes', nan], dtype=object)

df["Self_Employed"] = df['Self_Employed'].fillna("No Records")

df['Dependents'].unique()
array(['0', '1', '2', '3+', nan], dtype=object)

df["Dependents"] = df['Dependents'].fillna("Unknown")

df['Gender'].unique()
array(['Male', 'Female', nan], dtype=object)

df['Gender'] = df['Gender'].fillna("Unknown")

df = df.dropna(subset="LoanAmount")
```

2.

```
df = df.dropna(subset="LoanAmount")

df['Loan_Amount_Term'].unique()
array([360., 240., 180.,  nan,  60., 480.,  84.,  12., 300., 350.,  36.,
       120.,   6.])

df['Loan_Amount_Term'].median()
360.0

df['Loan_Amount_Term'] = df['Loan_Amount_Term'].fillna(float(360.0))
```

3.

All Missing Values Gone

```
print("Null Values By %")
print((df.isnull().sum() / len(df)*100).map("{:.2f} %".format))

Null Values By %
Loan_ID              0.00 %
Gender               0.00 %
Married              0.00 %
Dependents           0.00 %
Education            0.00 %
Self_Employed        0.00 %
ApplicantIncome      0.00 %
CoapplicantIncome    0.00 %
LoanAmount           0.00 %
Loan_Amount_Term     0.00 %
Credit_History       0.00 %
Property_Area        0.00 %
dtype: object
```

Step 6: Checking Statistics By Numerical Columns Again

```
df.describe()
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
count	362.000000	362.000000	362.000000	362.000000
mean	4769.513812	1569.011050	136.132597	342.254144
std	4911.744038	2338.842716	61.366652	64.675294
min	0.000000	0.000000	28.000000	6.000000
25%	2862.000000	0.000000	100.250000	360.000000
50%	3785.500000	1054.000000	125.000000	360.000000
75%	5030.750000	2416.750000	158.000000	360.000000
max	72529.000000	24000.000000	550.000000	480.000000

Step 7: Checking There Are Duplicate Rows

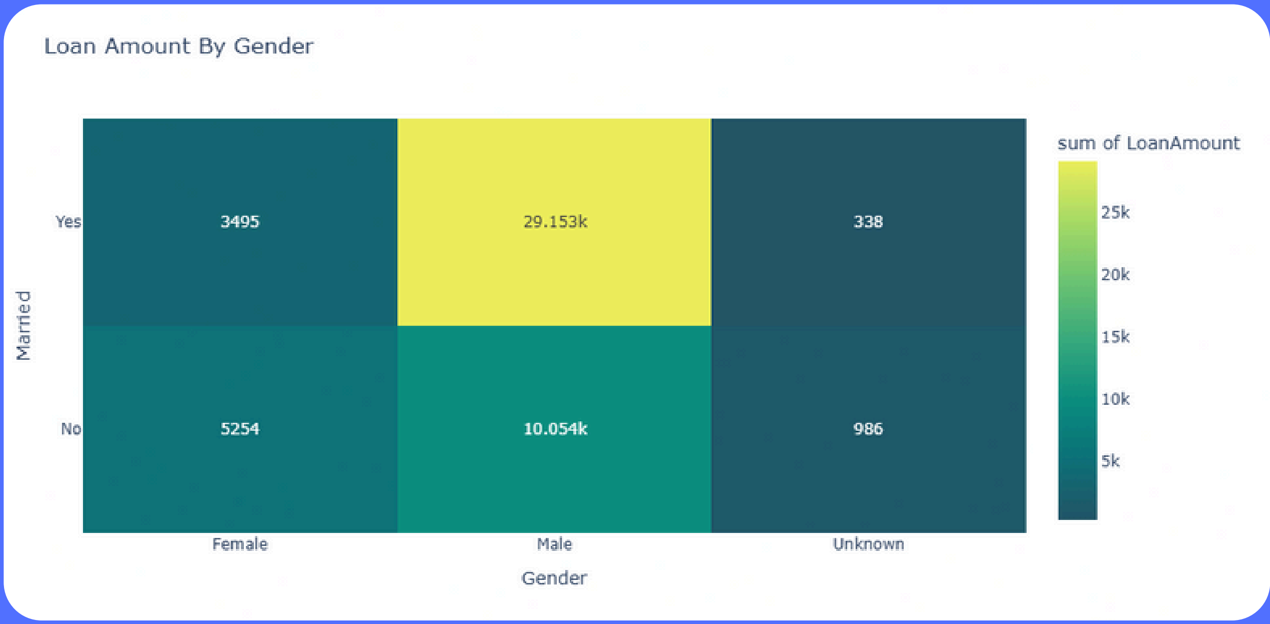
```
df['Loan_ID'].duplicated().sum()
```

```
np.int64(0)
```

```
df.info()
```

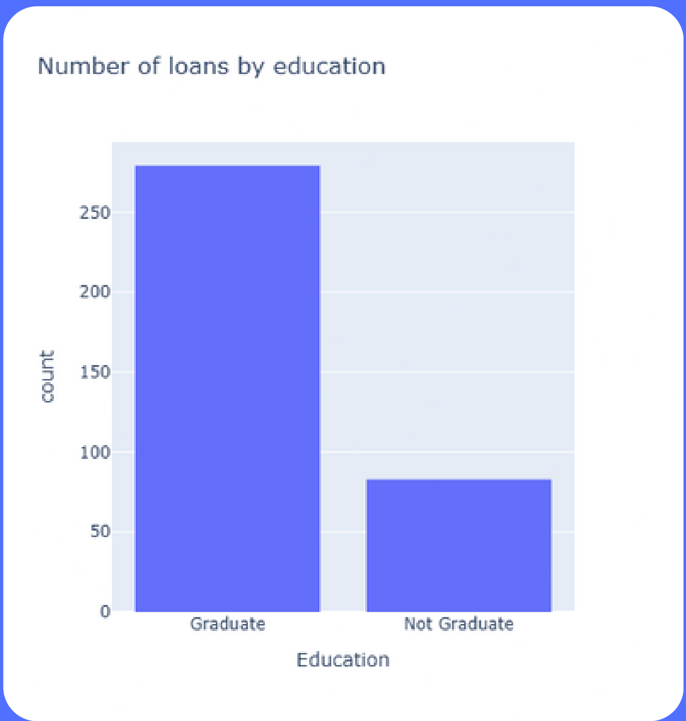
```
<class 'pandas.core.frame.DataFrame'>
Index: 362 entries, 0 to 366
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            362 non-null    object
 1   Gender             362 non-null    object
 2   Married            362 non-null    object
 3   Dependents         362 non-null    object
 4   Education          362 non-null    object
 5   Self_Employed      362 non-null    object
 6   ApplicantIncome    362 non-null    int64
 7   CoapplicantIncome  362 non-null    int64
 8   LoanAmount         362 non-null    float64
 9   Loan_Amount_Term   362 non-null    float64
 10  Credit_History     362 non-null    object
 11  Property_Area      362 non-null    object
dtypes: float64(2), int64(2), object(8)
memory usage: 36.8+ KB
```

Step 8: Analysis on gender



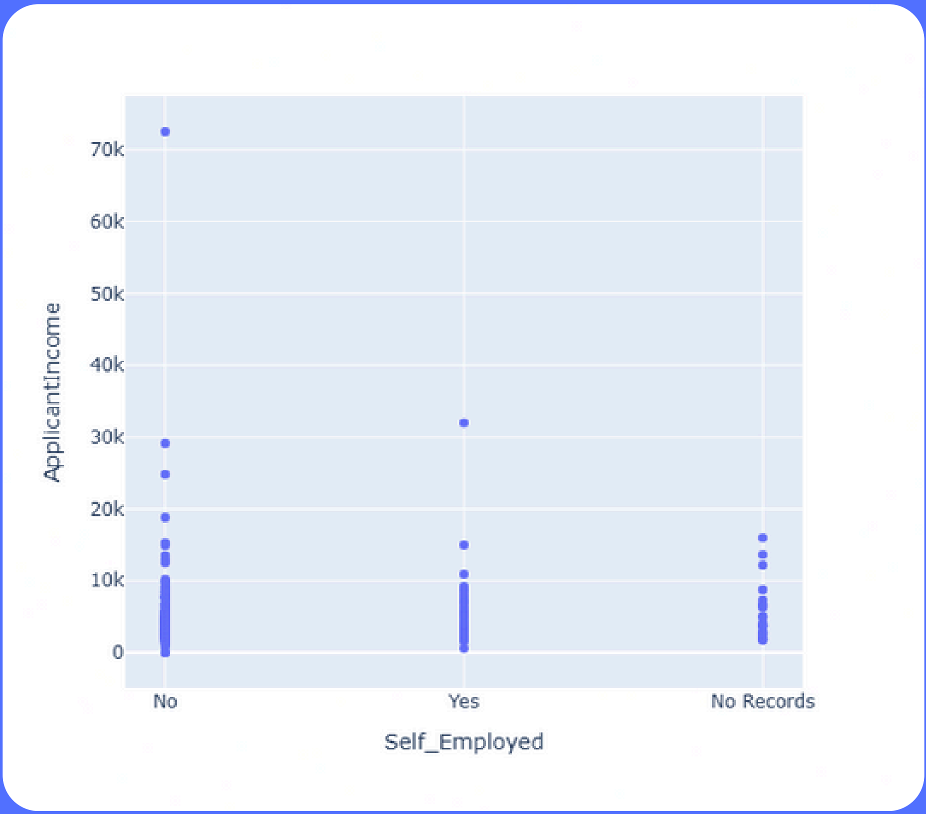
This map clearly indicates that married men tend to require loans

Step 9: Analysis on education level

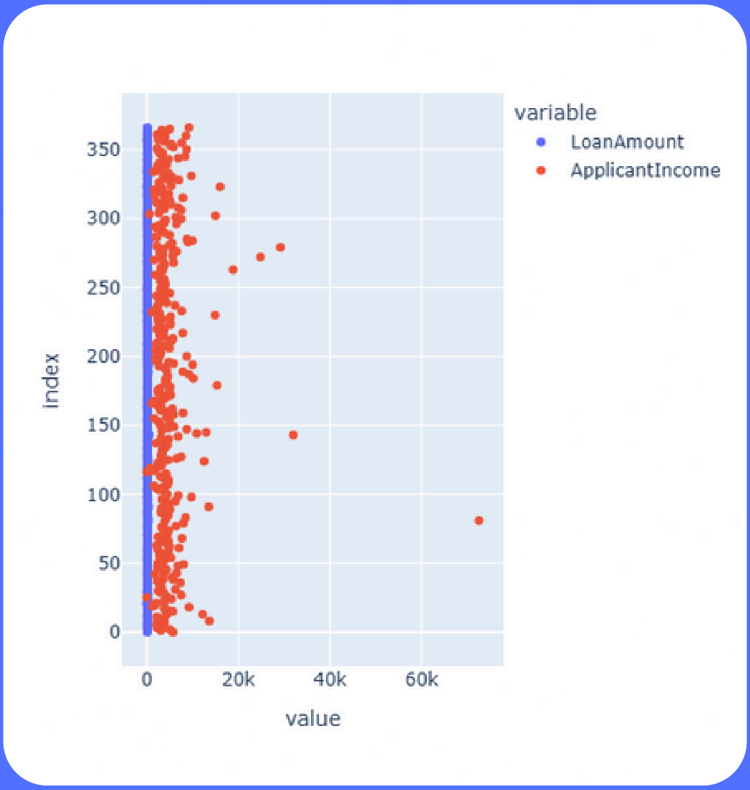


These bars showing educated peoples are taking more loan

Step 10: Income vs Self Employed

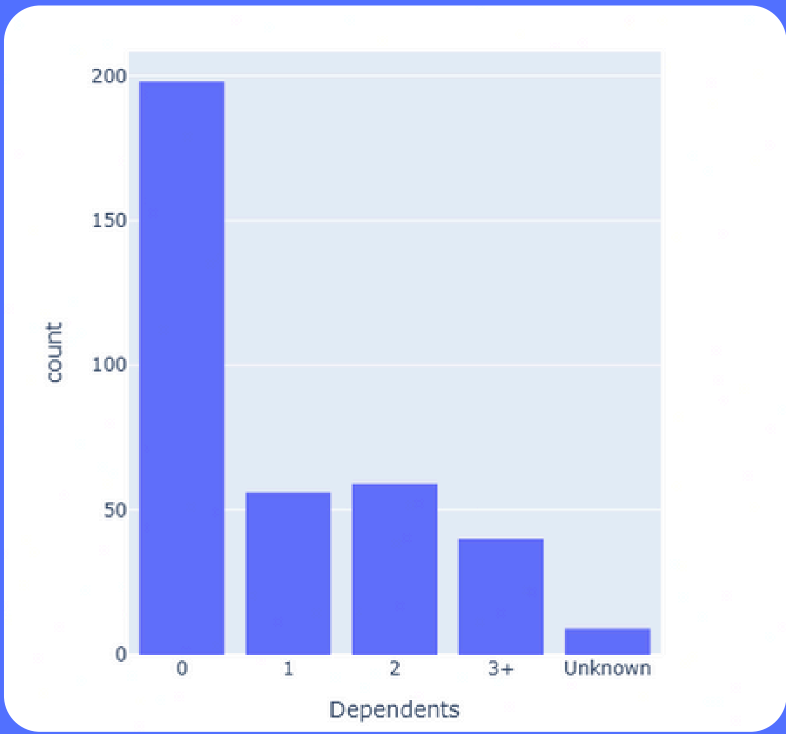


Step 11: Loan Amount vs Income



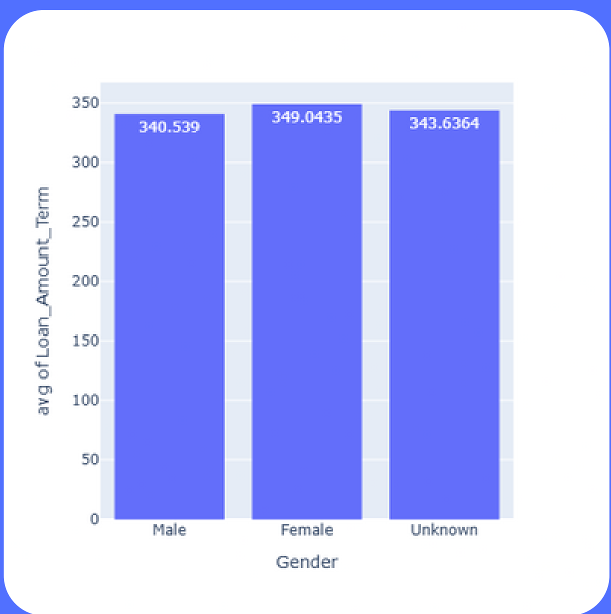
There is no relation between loan amount and income

Step 12: Analysis on Dependents



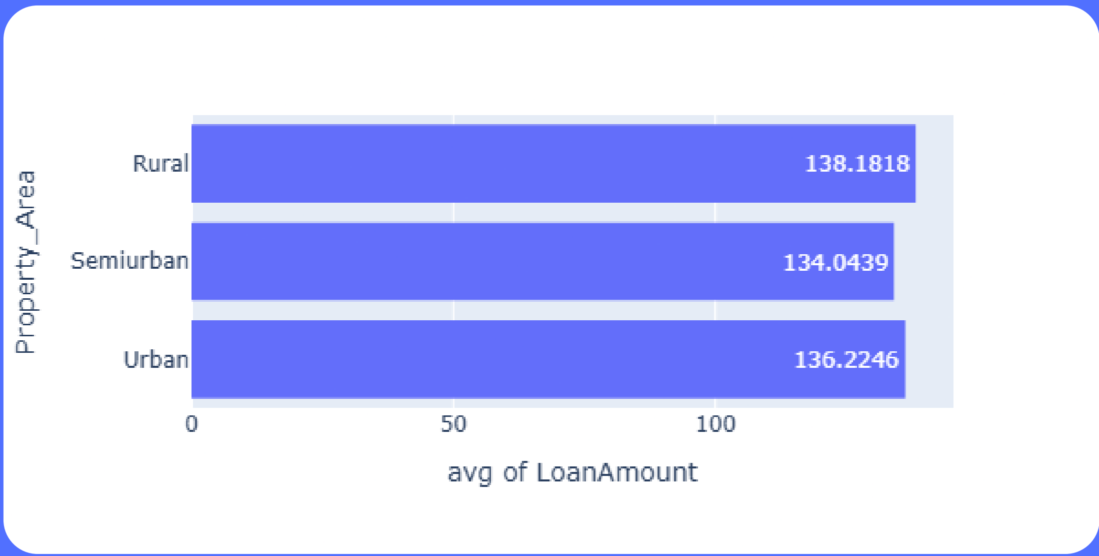
People with low dependents are taking more loan

Step 13: Who is taking long term loans

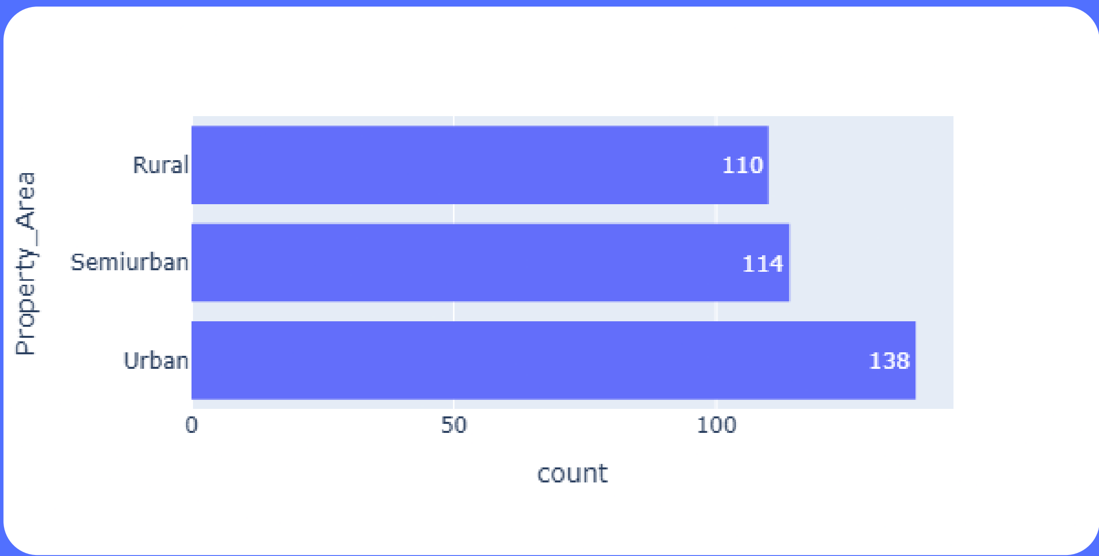


In this data we can see females are taking more long term loans compare to others

Step 13: Avg Loan Amount by area



People from rural areas are taking more loan amount compare to others at avg rate.



Applicants living in urban areas tend to take high number of loans, likely due to the higher cost of living and lifestyle expenses in those regions.

Outcomes

- Married applicants usually need larger loans due to family size, housing needs, etc.
- Male Applicants have more financial burden.
- Educated people are taking more loan compare to uneducated.
- There is no relation between loan amount and income.
- People those are not self employed having more income compared to self employed
- Females are usually taking long term loans compare to others.
- Applicants living in urban areas tend to take high number of loans, likely due to the higher cost of living and lifestyle expenses in those regions.
- People from rural areas are taking more loan amount compare to others at avg rate.