



Submitted by: Rohan Negi

Submitted to : Neeraj ma'am

Import Necessary Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import numpy as np
```

Task 1 Data Exploration

Load the dataset into a Python environment

```
df = pd.read_csv()
df
```

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | \ |
|---|----------|--------|---------|------------|-----------|---------------|---|
| 0 | LP001015 | Male | Yes | 0 | Graduate | No | |

| | | | | | | |
|-----|----------|------|-----|-----|--------------|-----|
| 1 | LP001022 | Male | Yes | 1 | Graduate | No |
| 2 | LP001031 | Male | Yes | 2 | Graduate | No |
| 3 | LP001035 | Male | Yes | 2 | Graduate | No |
| 4 | LP001051 | Male | No | 0 | Not Graduate | No |
| .. | ... | ... | ... | ... | ... | ... |
| 362 | LP002971 | Male | Yes | 3+ | Not Graduate | Yes |
| 363 | LP002975 | Male | Yes | 0 | Graduate | No |
| 364 | LP002980 | Male | No | 0 | Graduate | No |
| 365 | LP002986 | Male | Yes | 0 | Graduate | No |
| 366 | LP002989 | Male | No | 0 | Graduate | Yes |

| | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term |
|-----|-----------------|-------------------|------------|------------------|
| \ | | | | |
| 0 | 5720 | 0 | 110.0 | 360.0 |
| 1 | 3076 | 1500 | 126.0 | 360.0 |
| 2 | 5000 | 1800 | 208.0 | 360.0 |
| 3 | 2340 | 2546 | 100.0 | 360.0 |
| 4 | 3276 | 0 | 78.0 | 360.0 |
| .. | ... | ... | ... | ... |
| 362 | 4009 | 1777 | 113.0 | 360.0 |
| 363 | 4158 | 709 | 115.0 | 360.0 |
| 364 | 3250 | 1993 | 126.0 | 360.0 |
| 365 | 5000 | 2393 | 158.0 | 360.0 |
| 366 | 9200 | 0 | 98.0 | 180.0 |

| | Credit_History | Property_Area |
|-----|----------------|---------------|
| 0 | 1.0 | Urban |
| 1 | 1.0 | Urban |
| 2 | 1.0 | Urban |
| 3 | NaN | Urban |
| 4 | 1.0 | Urban |
| .. | ... | ... |
| 362 | 1.0 | Urban |
| 363 | 1.0 | Urban |
| 364 | NaN | Semiurban |
| 365 | 1.0 | Rural |
| 366 | 1.0 | Rural |

[367 rows x 12 columns]

Display the first few rows of the dataset to understand its structure

```
df.head(10)
```

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | \ |
|---|----------|--------|---------|------------|--------------|---------------|---|
| 0 | LP001015 | Male | Yes | 0 | Graduate | No | |
| 1 | LP001022 | Male | Yes | 1 | Graduate | No | |
| 2 | LP001031 | Male | Yes | 2 | Graduate | No | |
| 3 | LP001035 | Male | Yes | 2 | Graduate | No | |
| 4 | LP001051 | Male | No | 0 | Not Graduate | No | |
| 5 | LP001054 | Male | Yes | 0 | Not Graduate | Yes | |
| 6 | LP001055 | Female | No | 1 | Not Graduate | No | |
| 7 | LP001056 | Male | Yes | 2 | Not Graduate | No | |
| 8 | LP001059 | Male | Yes | 2 | Graduate | NaN | |
| 9 | LP001067 | Male | No | 0 | Not Graduate | No | |

| | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | \ |
|---|-----------------|-------------------|------------|------------------|---|
| 0 | 5720 | 0 | 110.0 | 360.0 | |
| 1 | 3076 | 1500 | 126.0 | 360.0 | |
| 2 | 5000 | 1800 | 208.0 | 360.0 | |
| 3 | 2340 | 2546 | 100.0 | 360.0 | |
| 4 | 3276 | 0 | 78.0 | 360.0 | |
| 5 | 2165 | 3422 | 152.0 | 360.0 | |
| 6 | 2226 | 0 | 59.0 | 360.0 | |
| 7 | 3881 | 0 | 147.0 | 360.0 | |
| 8 | 13633 | 0 | 280.0 | 240.0 | |
| 9 | 2400 | 2400 | 123.0 | 360.0 | |

| | Credit_History | Property_Area |
|---|----------------|---------------|
| 0 | 1.0 | Urban |
| 1 | 1.0 | Urban |
| 2 | 1.0 | Urban |
| 3 | NaN | Urban |
| 4 | 1.0 | Urban |
| 5 | 1.0 | Urban |
| 6 | 1.0 | Semiurban |
| 7 | 0.0 | Rural |
| 8 | 1.0 | Urban |
| 9 | 1.0 | Semiurban |

```
df.tail(10)
```

| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | \ |
|-----|----------|--------|---------|------------|--------------|---------------|---|
| 357 | LP002952 | Male | No | 0 | Graduate | No | |
| 358 | LP002954 | Male | Yes | 2 | Not Graduate | No | |
| 359 | LP002962 | Male | No | 0 | Graduate | No | |

| | | | | | | |
|-----|----------|--------|-----|----|--------------|-----|
| 360 | LP002965 | Female | Yes | 0 | Graduate | No |
| 361 | LP002969 | Male | Yes | 1 | Graduate | No |
| 362 | LP002971 | Male | Yes | 3+ | Not Graduate | Yes |
| 363 | LP002975 | Male | Yes | 0 | Graduate | No |
| 364 | LP002980 | Male | No | 0 | Graduate | No |
| 365 | LP002986 | Male | Yes | 0 | Graduate | No |
| 366 | LP002989 | Male | No | 0 | Graduate | Yes |

| \ | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term |
|-----|-----------------|-------------------|------------|------------------|
| | | | | |
| 357 | 2500 | 0 | 60.0 | 360.0 |
| 358 | 3132 | 0 | 76.0 | 360.0 |
| 359 | 4000 | 2667 | 152.0 | 360.0 |
| 360 | 8550 | 4255 | 96.0 | 360.0 |
| 361 | 2269 | 2167 | 99.0 | 360.0 |
| 362 | 4009 | 1777 | 113.0 | 360.0 |
| 363 | 4158 | 709 | 115.0 | 360.0 |
| 364 | 3250 | 1993 | 126.0 | 360.0 |
| 365 | 5000 | 2393 | 158.0 | 360.0 |
| 366 | 9200 | 0 | 98.0 | 180.0 |

| | Credit_History | Property_Area |
|-----|----------------|---------------|
| 357 | 1.0 | Urban |
| 358 | NaN | Rural |
| 359 | 1.0 | Semiurban |
| 360 | NaN | Urban |
| 361 | 1.0 | Semiurban |
| 362 | 1.0 | Urban |
| 363 | 1.0 | Urban |
| 364 | NaN | Semiurban |
| 365 | 1.0 | Rural |
| 366 | 1.0 | Rural |

Check for missing values and handle them if necessary

```
df.isna().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.loc[df['Gender'].isna(),'Gender'] = 'Unknown'

mode_value=df['Dependents'].mode()[0]
df['Dependents'].fillna(mode_value,inplace=True)

df.loc[df['Self_Employed'].isna(),'Self_Employed'] = 'No'

df['LoanAmount']=df['LoanAmount'].fillna(df['LoanAmount'].median())

df['Loan_Amount_Term']=df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].median())

df['Credit_History']=df['Credit_History'].fillna(df['Credit_History'].median())

df.isna().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

blank_values = (df==" ").sum()
print('Blank Values count:\n',blank_values)
```

```
Blank Values count:
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.duplicated().sum()
```

```
0
```

There are no duplicate values in the Data Frame,
Simultaneously there is no relation of Loan_ID with other
columns

Hence we drop the Loan_ID

```
df = df.drop(columns=['Loan_ID'])
```

```
df
```

| | Gender | Married | Dependents | Education | Self_Employed |
|-------------------|--------|---------|------------|--------------|---------------|
| ApplicantIncome \ | | | | | |
| 0 | Male | Yes | 0 | Graduate | No |
| 5720 | | | | | |
| 1 | Male | Yes | 1 | Graduate | No |
| 3076 | | | | | |
| 2 | Male | Yes | 2 | Graduate | No |
| 5000 | | | | | |
| 3 | Male | Yes | 2 | Graduate | No |
| 2340 | | | | | |
| 4 | Male | No | 0 | Not Graduate | No |
| 3276 | | | | | |
| .. | ... | ... | ... | ... | ... |
| ... | | | | | |
| 362 | Male | Yes | 3+ | Not Graduate | Yes |
| 4009 | | | | | |
| 363 | Male | Yes | 0 | Graduate | No |
| 4158 | | | | | |
| 364 | Male | No | 0 | Graduate | No |

```

3250
365  Male      Yes      0      Graduate      No
5000
366  Male      No       0      Graduate      Yes
9200

      CoapplicantIncome  LoanAmount  Loan_Amount_Term
Credit_History \
0              0         110.0         360.0         1.0
1              1500         126.0         360.0         1.0
2              1800         208.0         360.0         1.0
3              2546         100.0         360.0         1.0
4              0           78.0         360.0         1.0
..              ...         ...         ...         ...
362              1777         113.0         360.0         1.0
363              709         115.0         360.0         1.0
364              1993         126.0         360.0         1.0
365              2393         158.0         360.0         1.0
366              0          98.0         180.0         1.0

      Property_Area
0      Urban
1      Urban
2      Urban
3      Urban
4      Urban
..      ...
362     Urban
363     Urban
364  Semiurban
365     Rural
366     Rural

[367 rows x 11 columns]

```

Summarize basic statistics

```
df.describe().T
```

| | count | mean | std | min | 25% |
|-------------------|--------|-------------|-------------|------|--------|
| 50% \ | | | | | |
| ApplicantIncome | 367.0 | 4805.599455 | 4910.685399 | 0.0 | 2864.0 |
| 3786.0 | | | | | |
| CoapplicantIncome | 367.0 | 1569.577657 | 2334.232099 | 0.0 | 0.0 |
| 1025.0 | | | | | |
| LoanAmount | 367.0 | 135.980926 | 60.959739 | 28.0 | 101.0 |
| 125.0 | | | | | |
| Loan_Amount_Term | 367.0 | 342.822888 | 64.658402 | 6.0 | 360.0 |
| 360.0 | | | | | |
| Credit_History | 367.0 | 0.839237 | 0.367814 | 0.0 | 1.0 |
| 1.0 | | | | | |
| | 75% | max | | | |
| ApplicantIncome | 5060.0 | 72529.0 | | | |
| CoapplicantIncome | 2430.5 | 24000.0 | | | |
| LoanAmount | 157.5 | 550.0 | | | |
| Loan_Amount_Term | 360.0 | 480.0 | | | |
| Credit_History | 1.0 | 1.0 | | | |

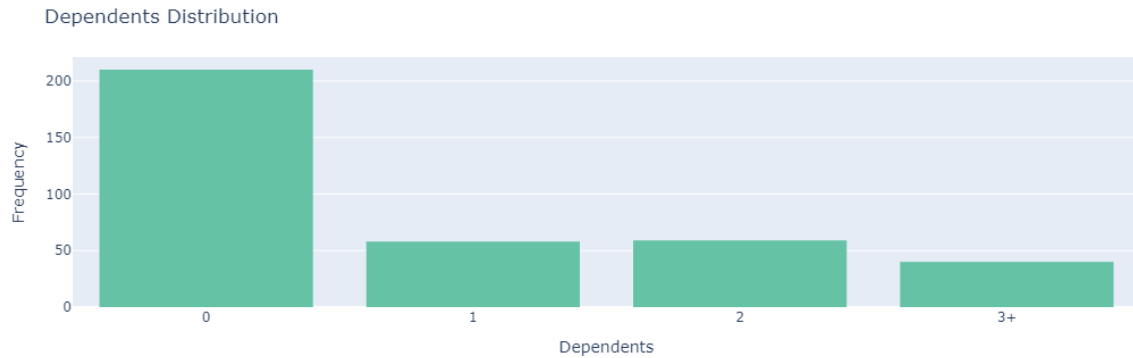
Task 2: Data Visualization

Univariate Analysis

Explore the distribution of numeric columns using the following visualizations

Histograms: Plot the frequency distribution of key numeric variables

```
fig = px.histogram(df, x='Dependents',
color_discrete_sequence=px.colors.qualitative.Set2)
fig.update_layout(
    title="Dependents Distribution",
    xaxis_title="Dependents",
    yaxis_title="Frequency"
)
fig.show()
```

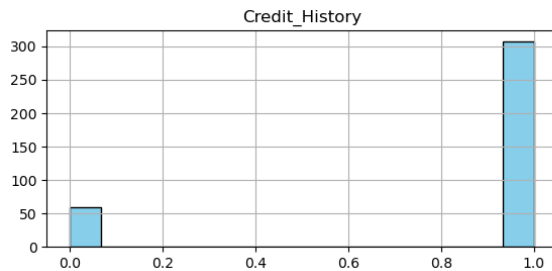
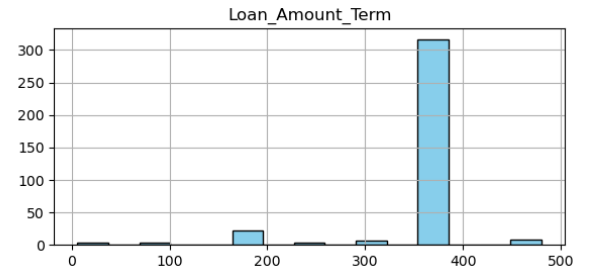
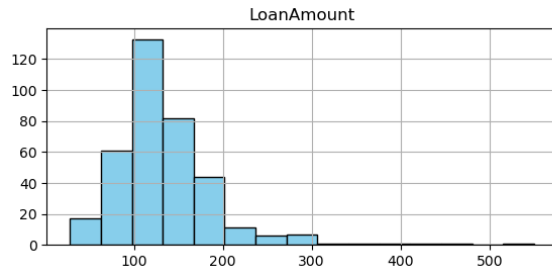
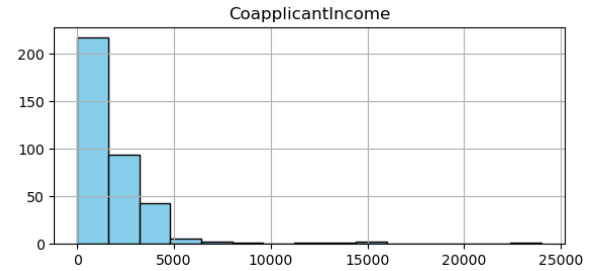
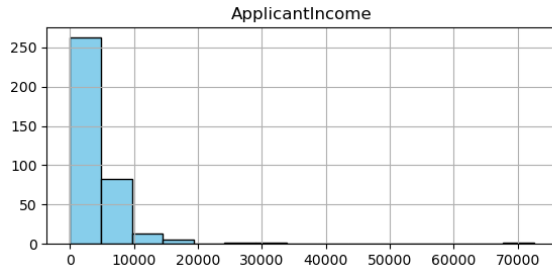
Outcome from above chart

This chart shows the number of people with different dependent counts, where 61.5% individuals have zero dependents, and 15.4% have 1 dependent, 15.4% have 2 dependents and 7.7% have 3+ dependents

Insights to Grow Business

Create tailored loans, targeted campaigns, and flexible rates to support applicants with or without dependents while offering additional products to grow the customer base and reduce risk.

```
numeric_columns = df.select_dtypes(include=['float64',  
'int64']).columns  
# Plot histograms for numeric columns  
df[numeric_columns].hist(bins=15, figsize=(15, 10), color='skyblue',  
edgecolor='black')  
plt.title("Histograms of Numeric Columns")  
plt.show()
```



Outcome from above chart

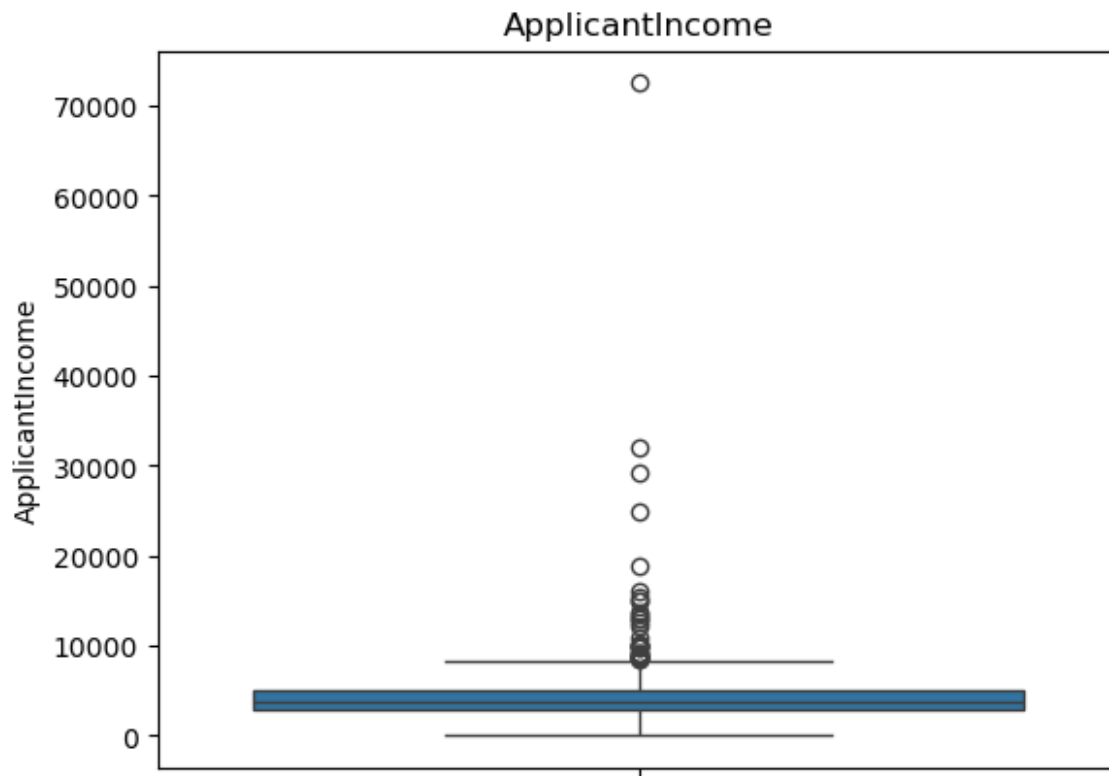
Most applicants make under 20,000, co-applicants earn under 5,000, and loans are usually between 100–200, with many around 400. The bank mainly approves loans for people with good credit history, longer loan terms, and amounts in the lower to middle range, while co-applicants typically earn less.

Insights to Grow Business

Offer affordable loans for low-income applicants and co-applicants, focus on loan amounts between 100-400, provide benefits for those with good credit, help co-applicants improve their finances, and promote longer loan terms for easier repayments.

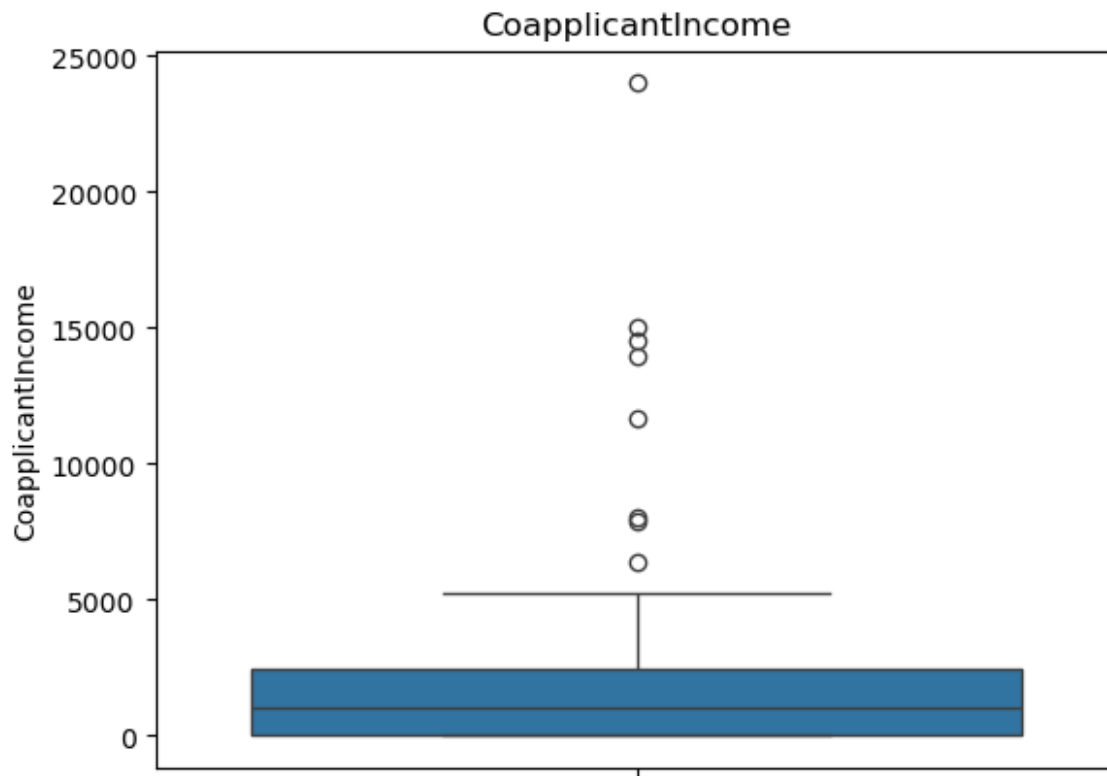
Box Plots: Identify potential outliers and visualize the spread of data

```
sns.boxplot(data=df['ApplicantIncome'])
plt.title("ApplicantIncome")
plt.xticks(rotation=90)
plt.show()
```



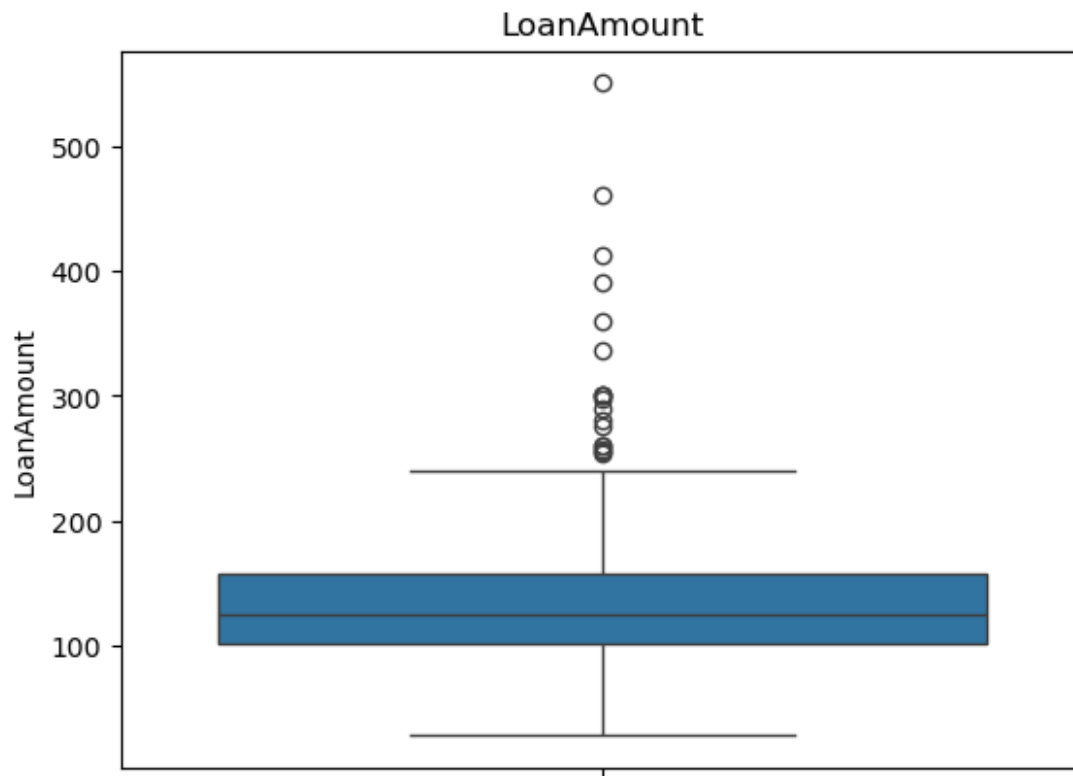
The box plot of applicant income shows a lot of outliers, meaning a few people have very high incomes compared to most. These outliers could impact loan approval decisions and need careful handling to ensure fairness.

```
sns.boxplot(data=df['CoapplicantIncome'])  
plt.title("CoapplicantIncome")  
plt.xticks(rotation=90)  
plt.show()
```



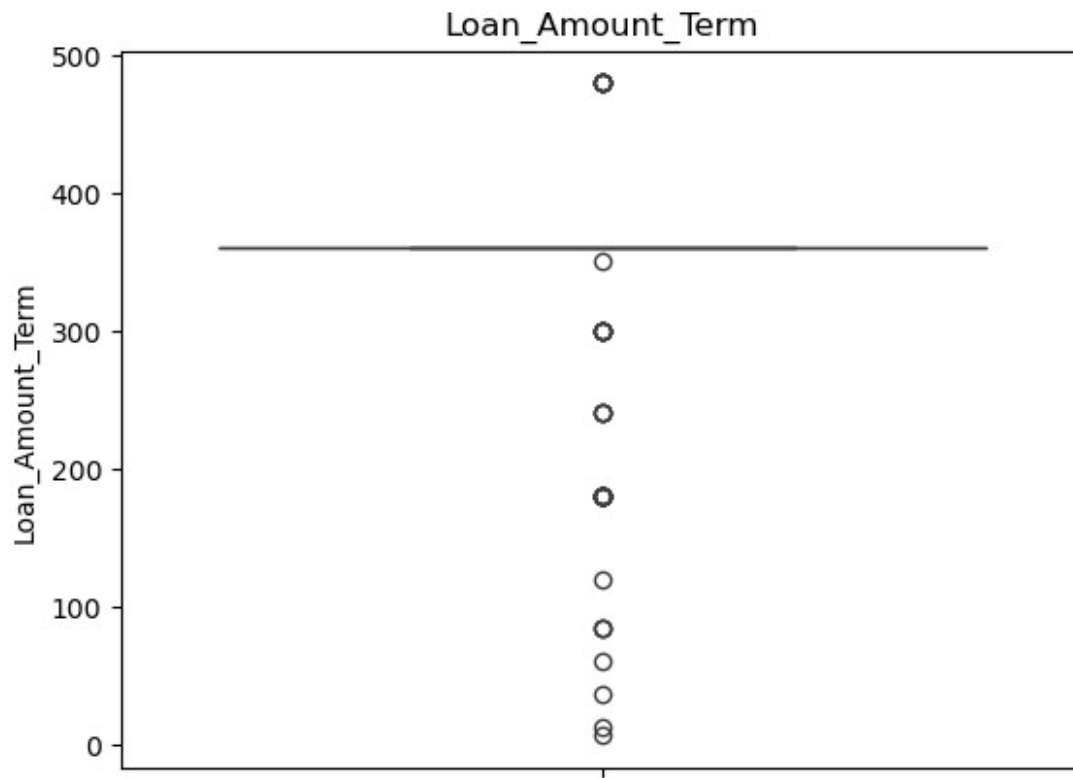
The box plot of co-applicant income shows many outliers, meaning some co-applicants have very high incomes compared to others. These outliers could affect loan approval decisions and should be reviewed to avoid any unfair biases.

```
sns.boxplot(data=df[ 'LoanAmount' ] )  
plt.title("LoanAmount")  
plt.xticks(rotation=90)  
plt.show()
```



The box plot of loan amounts shows many outliers, meaning some people are requesting much higher loans than others. These high values could impact loan decisions and should be reviewed to prevent unfair biases.

```
sns.boxplot(data=df['Loan_Amount_Term'])  
plt.title("Loan_Amount_Term")  
plt.xticks(rotation=90)  
plt.show()
```



The box plot of loan terms shows several outliers, with some loans having unusually long repayment periods. These could affect loan decisions and should be reviewed to ensure fair evaluations.

```
sns.boxplot(data=df['Credit_History'])  
plt.title("Credit_History")  
plt.xticks(rotation=90)  
plt.show()
```



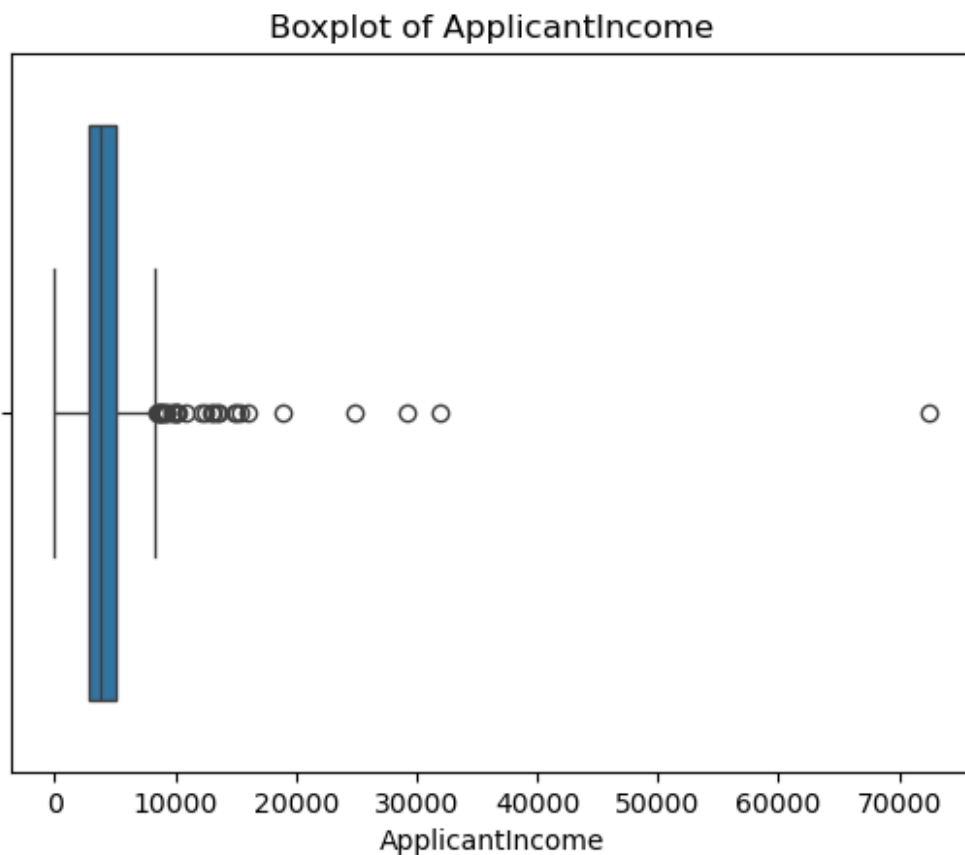
Handling Outliers

```
sns.boxplot(x='ApplicantIncome', data=df)
plt.title('Boxplot of ApplicantIncome')
plt.show()

# Calculate quartiles
Q1 = df['ApplicantIncome'].quantile(0.25)
Q3 = df['ApplicantIncome'].quantile(0.75)
IQR = Q3 - Q1

# Define threshold for outliers
threshold = 1.5 * IQR

# Identify outliers
outliers = df[(df['ApplicantIncome'] < Q1 - threshold) |
              (df['ApplicantIncome'] > Q3 + threshold)]
print(outliers)
```



| ApplicantIncome | Gender | Married | Dependents | Education | Self_Employed |
|-----------------|--------|---------|------------|-----------|---------------|
| 813633 | Male | Yes | 2 | Graduate | No |
| 1312173 | Male | Yes | 2 | Graduate | No |
| 189226 | Male | Yes | 0 | Graduate | No |
| 8172529 | Male | Yes | 3+ | Graduate | No |
| 838449 | Male | Yes | 0 | Graduate | No |
| 9113518 | Male | Yes | 3+ | Graduate | No |
| 989719 | Male | Yes | 3+ | Graduate | No |
| 12412500 | Female | No | 0 | Graduate | No |
| 14332000 | Male | Yes | 0 | Graduate | Yes |
| 14410890 | Male | Yes | 2 | Graduate | Yes |
| 145 | Female | No | 0 | Graduate | No |

| | | | | | | |
|-------|---------|-----|----|--------------|-----|--|
| 12941 | | | | | | |
| 147 | Male | No | 0 | Not Graduate | Yes | |
| 8703 | | | | | | |
| 179 | Male | Yes | 3+ | Graduate | No | |
| 15312 | | | | | | |
| 184 | Male | Yes | 3+ | Graduate | No | |
| 10166 | | | | | | |
| 187 | Male | Yes | 2 | Graduate | No | |
| 9167 | | | | | | |
| 188 | Male | Yes | 0 | Not Graduate | No | |
| 13083 | | | | | | |
| 194 | Male | Yes | 1 | Graduate | No | |
| 10000 | | | | | | |
| 200 | Male | Yes | 0 | Graduate | Yes | |
| 8706 | | | | | | |
| 230 | Male | No | 0 | Graduate | No | |
| 14911 | | | | | | |
| 247 | Male | Yes | 0 | Graduate | No | |
| 10000 | | | | | | |
| 263 | Male | Yes | 1 | Graduate | No | |
| 18840 | | | | | | |
| 272 | Male | No | 1 | Graduate | No | |
| 24797 | | | | | | |
| 279 | Unknown | No | 0 | Graduate | No | |
| 29167 | | | | | | |
| 283 | Male | No | 0 | Not Graduate | No | |
| 9000 | | | | | | |
| 284 | Female | Yes | 2 | Graduate | No | |
| 10000 | | | | | | |
| 285 | Male | Yes | 1 | Graduate | No | |
| 8750 | | | | | | |
| 302 | Female | No | 0 | Graduate | Yes | |
| 14987 | | | | | | |
| 323 | Male | No | 1 | Graduate | No | |
| 16000 | | | | | | |
| 331 | Male | Yes | 3+ | Graduate | No | |
| 9699 | | | | | | |
| 350 | Male | Yes | 2 | Graduate | No | |
| 8667 | | | | | | |
| 360 | Female | Yes | 0 | Graduate | No | |
| 8550 | | | | | | |
| 366 | Male | No | 0 | Graduate | Yes | |
| 9200 | | | | | | |

| CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History \ |
|-------------------|------------|------------------|------------------|
| 0 | 280.0 | 240.0 | 1.0 |
| 0 | 166.0 | 360.0 | 0.0 |

| | | | | |
|-----|-------|-------|-------|-----|
| 18 | 7916 | 300.0 | 360.0 | 1.0 |
| 81 | 0 | 360.0 | 360.0 | 1.0 |
| 83 | 0 | 257.0 | 360.0 | 1.0 |
| 91 | 0 | 390.0 | 360.0 | 1.0 |
| 98 | 0 | 61.0 | 360.0 | 1.0 |
| 124 | 0 | 300.0 | 360.0 | 0.0 |
| 143 | 0 | 550.0 | 360.0 | 1.0 |
| 144 | 0 | 260.0 | 12.0 | 1.0 |
| 145 | 0 | 150.0 | 300.0 | 1.0 |
| 147 | 0 | 199.0 | 360.0 | 0.0 |
| 179 | 0 | 187.0 | 360.0 | 1.0 |
| 184 | 750 | 150.0 | 360.0 | 1.0 |
| 187 | 0 | 235.0 | 360.0 | 1.0 |
| 188 | 0 | 125.0 | 360.0 | 1.0 |
| 194 | 2690 | 412.0 | 360.0 | 1.0 |
| 200 | 0 | 108.0 | 480.0 | 1.0 |
| 230 | 14507 | 130.0 | 360.0 | 1.0 |
| 247 | 0 | 125.0 | 360.0 | 1.0 |
| 263 | 0 | 234.0 | 360.0 | 1.0 |
| 272 | 0 | 240.0 | 360.0 | 1.0 |
| 279 | 0 | 185.0 | 360.0 | 1.0 |
| 283 | 0 | 122.0 | 360.0 | 1.0 |
| 284 | 11666 | 460.0 | 360.0 | 1.0 |
| 285 | 0 | 297.0 | 360.0 | 1.0 |
| 302 | 0 | 177.0 | 360.0 | 1.0 |
| 323 | 5000 | 40.0 | 360.0 | 1.0 |

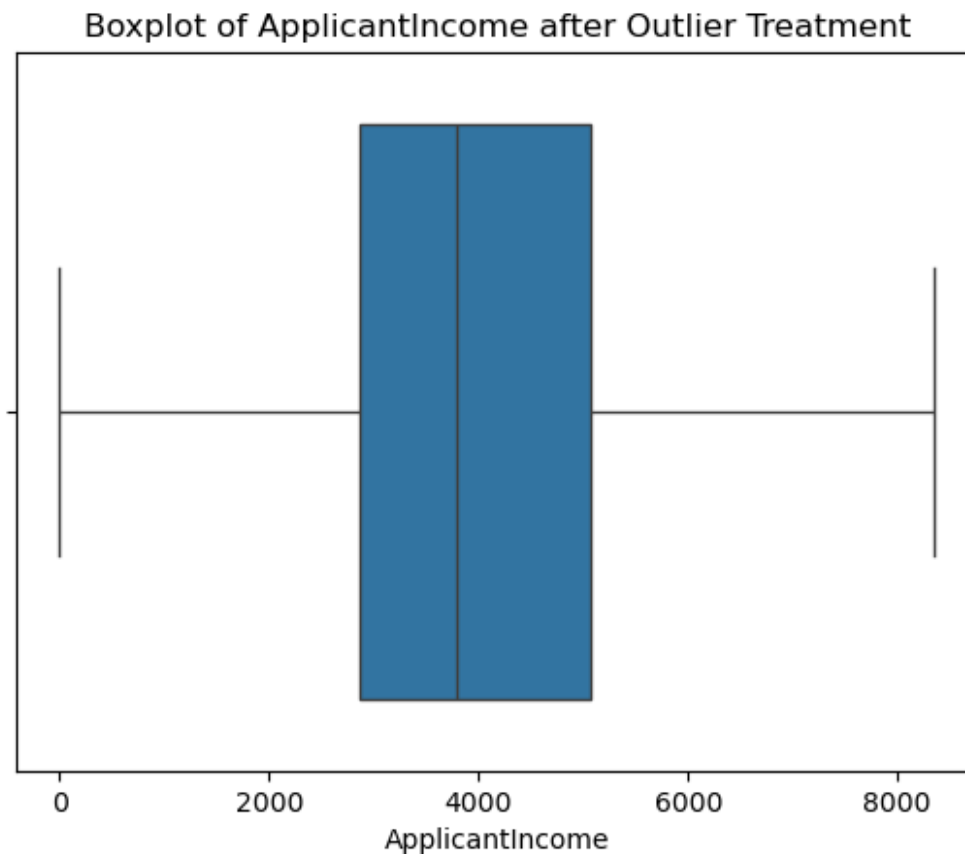
| | | | | |
|-----|------|-------|-------|-----|
| 331 | 0 | 300.0 | 360.0 | 1.0 |
| 350 | 0 | 254.0 | 360.0 | 1.0 |
| 360 | 4255 | 96.0 | 360.0 | 1.0 |
| 366 | 0 | 98.0 | 180.0 | 1.0 |

| | Property_Area |
|-----|---------------|
| 8 | Urban |
| 13 | Semiurban |
| 18 | Urban |
| 81 | Urban |
| 83 | Rural |
| 91 | Rural |
| 98 | Urban |
| 124 | Urban |
| 143 | Semiurban |
| 144 | Rural |
| 145 | Urban |
| 147 | Rural |
| 179 | Urban |
| 184 | Urban |
| 187 | Semiurban |
| 188 | Rural |
| 194 | Semiurban |
| 200 | Rural |
| 230 | Semiurban |
| 247 | Urban |
| 263 | Rural |
| 272 | Semiurban |
| 279 | Semiurban |
| 283 | Rural |
| 284 | Urban |
| 285 | Urban |
| 302 | Rural |
| 323 | Semiurban |
| 331 | Urban |
| 350 | Rural |
| 360 | Urban |
| 366 | Rural |

```
# Handle the outliers at the threshold values
df['ApplicantIncome'] = df['ApplicantIncome'].clip(lower=Q1 -
threshold, upper=Q3 + threshold)

# Recheck the boxplot
sns.boxplot(x='ApplicantIncome', data=df)
```

```
plt.title('Boxplot of ApplicantIncome after Outlier Treatment')
plt.show()
```

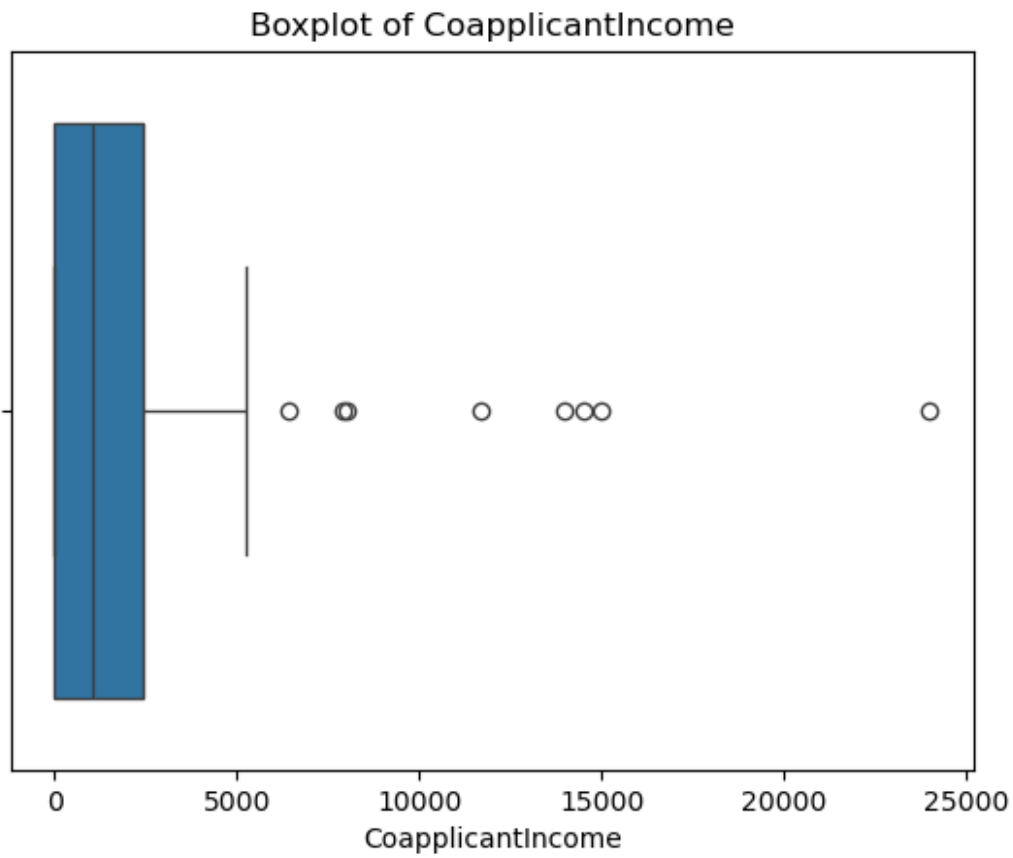


```
sns.boxplot(x='CoapplicantIncome', data=df)
plt.title('Boxplot of CoapplicantIncome')
plt.show()

# Calculate quartiles
Q1 = df['CoapplicantIncome'].quantile(0.25)
Q3 = df['CoapplicantIncome'].quantile(0.75)
IQR = Q3 - Q1

# Define threshold for outliers
threshold = 1.5 * IQR

# Identify outliers
outliers = df[(df['CoapplicantIncome'] < Q1 - threshold) |
               (df['CoapplicantIncome'] > Q3 + threshold)]
print(outliers)
```



| ApplicantIncome \ | Gender | Married | Dependents | Education | Self_Employed |
|-------------------|-------------------|------------|------------------|--------------|---------------|
| 18 | Male | Yes | 0 | Graduate | No |
| 8354 | | | | | |
| 25 | Male | No | 0 | Graduate | No |
| 0 | | | | | |
| 85 | Male | Yes | 2 | Graduate | No |
| 4635 | | | | | |
| 123 | Male | No | 0 | Graduate | No |
| 2500 | | | | | |
| 230 | Male | No | 0 | Graduate | No |
| 8354 | | | | | |
| 237 | Male | Yes | 2 | Not Graduate | No |
| 6166 | | | | | |
| 284 | Female | Yes | 2 | Graduate | No |
| 8354 | | | | | |
| 351 | Male | No | 0 | Graduate | No |
| 2283 | | | | | |
| Credit_History \ | CoapplicantIncome | LoanAmount | Loan_Amount_Term | | |
| 18 | 7916 | 300.0 | 360.0 | | 1.0 |

| | | | | |
|-----|-------|-------|-------|-----|
| 25 | 24000 | 148.0 | 360.0 | 0.0 |
| 85 | 8000 | 102.0 | 180.0 | 1.0 |
| 123 | 6414 | 187.0 | 360.0 | 0.0 |
| 230 | 14507 | 130.0 | 360.0 | 1.0 |
| 237 | 13983 | 102.0 | 360.0 | 1.0 |
| 284 | 11666 | 460.0 | 360.0 | 1.0 |
| 351 | 15000 | 106.0 | 360.0 | 1.0 |

| | Property_Area |
|-----|---------------|
| 18 | Urban |
| 25 | Rural |
| 85 | Rural |
| 123 | Rural |
| 230 | Semiurban |
| 237 | Rural |
| 284 | Urban |
| 351 | Rural |

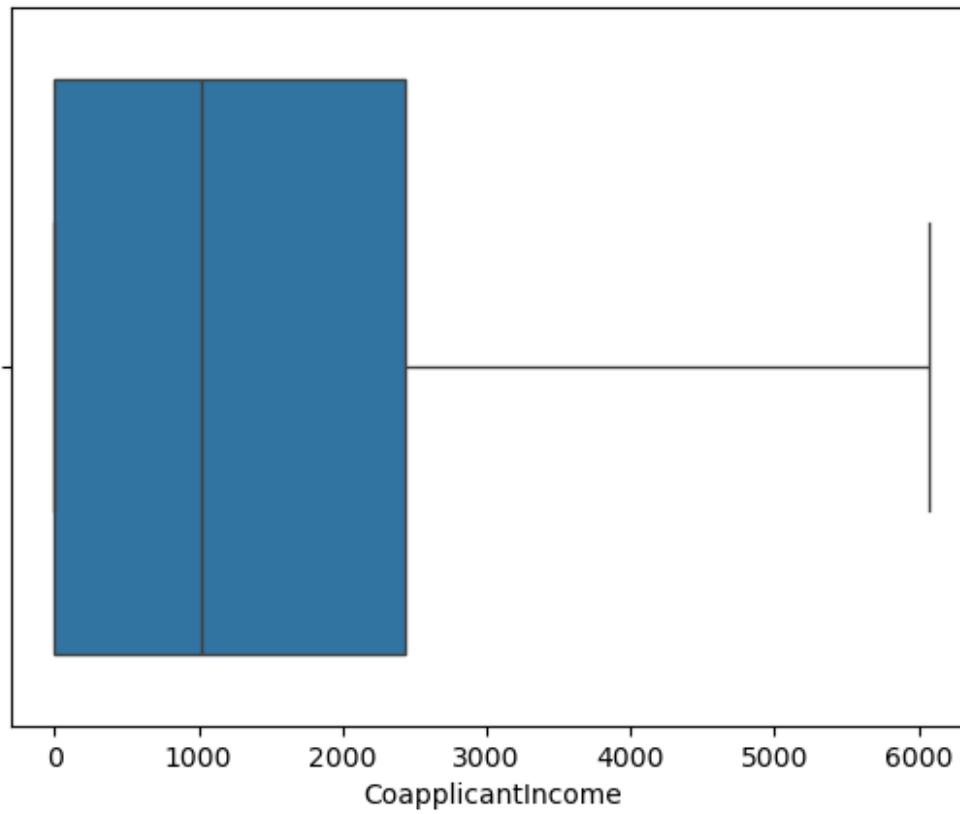
Handle the outliers at the threshold values

```
df['CoapplicantIncome'] = df['CoapplicantIncome'].clip(lower=Q1 -
threshold, upper=Q3 + threshold)
```

Recheck the boxplot

```
sns.boxplot(x='CoapplicantIncome', data=df)
plt.title('Boxplot of CoapplicantIncome after Outlier Treatment')
plt.show()
```

Boxplot of CoapplicantIncome after Outlier Treatment

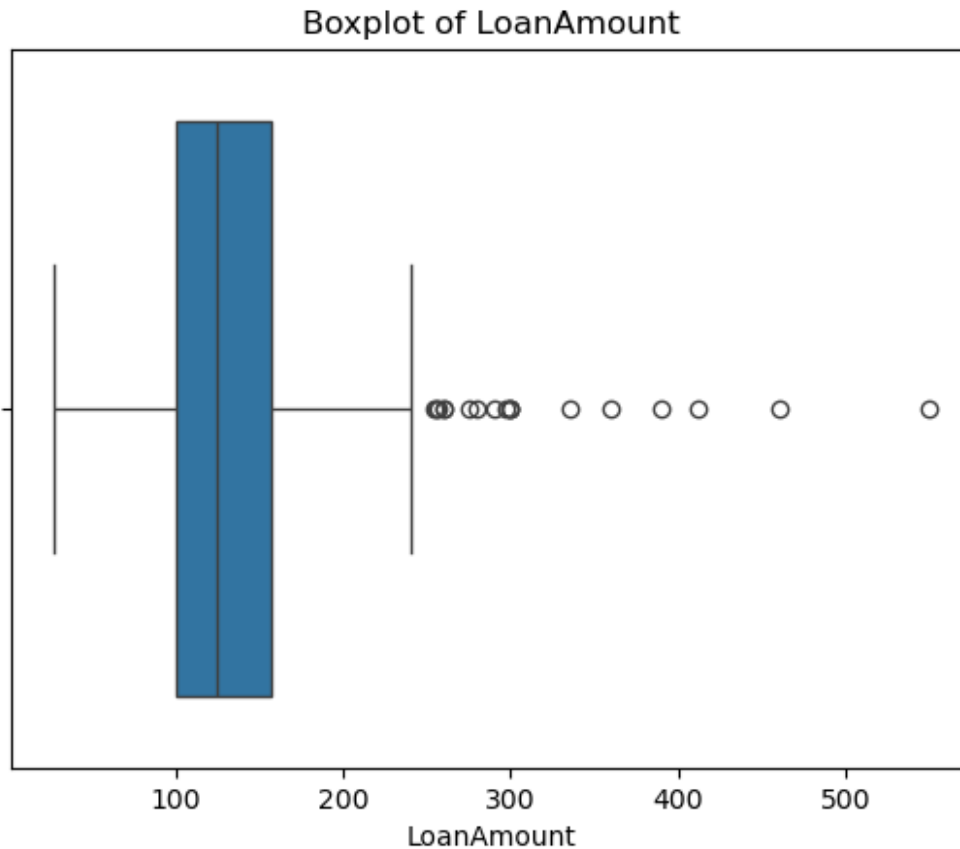


```
sns.boxplot(x='LoanAmount', data=df)
plt.title('Boxplot of LoanAmount')
plt.show()

# Calculate quartiles
Q1 = df['LoanAmount'].quantile(0.25)
Q3 = df['LoanAmount'].quantile(0.75)
IQR = Q3 - Q1

# Define threshold for outliers
threshold = 1.5 * IQR

# Identify outliers
outliers = df[(df['LoanAmount'] < Q1 - threshold) | (df['LoanAmount']
> Q3 + threshold)]
print(outliers)
```



| ApplicantIncome | Gender | Married | Dependents | Education | Self_Employed |
|-----------------|--------|---------|------------|-----------|---------------|
| 8 | Male | Yes | 2 | Graduate | No |
| 8354 | | | | | |
| 18 | Male | Yes | 0 | Graduate | No |
| 8354 | | | | | |
| 24 | Male | Yes | 0 | Graduate | No |
| 5400 | | | | | |
| 27 | Male | Yes | 0 | Graduate | No |
| 7500 | | | | | |
| 81 | Male | Yes | 3+ | Graduate | No |
| 8354 | | | | | |
| 83 | Male | Yes | 0 | Graduate | No |
| 8354 | | | | | |
| 91 | Male | Yes | 3+ | Graduate | No |
| 8354 | | | | | |
| 96 | Male | Yes | 1 | Graduate | No |
| 3333 | | | | | |
| 124 | Female | No | 0 | Graduate | No |
| 8354 | | | | | |
| 143 | Male | Yes | 0 | Graduate | Yes |
| 8354 | | | | | |
| 144 | Male | Yes | 2 | Graduate | Yes |

| | | | | | |
|------|--------|-----|----|----------|----|
| 8354 | | | | | |
| 189 | Male | Yes | 2 | Graduate | No |
| 7874 | | | | | |
| 194 | Male | Yes | 1 | Graduate | No |
| 8354 | | | | | |
| 284 | Female | Yes | 2 | Graduate | No |
| 8354 | | | | | |
| 285 | Male | Yes | 1 | Graduate | No |
| 8354 | | | | | |
| 331 | Male | Yes | 3+ | Graduate | No |
| 8354 | | | | | |
| 345 | Male | Yes | 3+ | Graduate | No |
| 8334 | | | | | |
| 350 | Male | Yes | 2 | Graduate | No |
| 8354 | | | | | |

| | CoapplicantIncome | LoanAmount | Loan_Amount_Term | |
|------------------|-------------------|------------|------------------|-----|
| Credit_History \ | | | | |
| 8 | 0.00 | 280.0 | 240.0 | 1.0 |
| 18 | 6076.25 | 300.0 | 360.0 | 1.0 |
| 24 | 4380.00 | 290.0 | 360.0 | 1.0 |
| 27 | 3750.00 | 275.0 | 360.0 | 1.0 |
| 81 | 0.00 | 360.0 | 360.0 | 1.0 |
| 83 | 0.00 | 257.0 | 360.0 | 1.0 |
| 91 | 0.00 | 390.0 | 360.0 | 1.0 |
| 96 | 4200.00 | 256.0 | 360.0 | 1.0 |
| 124 | 0.00 | 300.0 | 360.0 | 0.0 |
| 143 | 0.00 | 550.0 | 360.0 | 1.0 |
| 144 | 0.00 | 260.0 | 12.0 | 1.0 |
| 189 | 3967.00 | 336.0 | 360.0 | 1.0 |
| 194 | 2690.00 | 412.0 | 360.0 | 1.0 |
| 284 | 6076.25 | 460.0 | 360.0 | 1.0 |
| 285 | 0.00 | 297.0 | 360.0 | 1.0 |
| 331 | 0.00 | 300.0 | 360.0 | 1.0 |
| 345 | 0.00 | 260.0 | 360.0 | 1.0 |

| | | | | |
|-----|------|-------|-------|-----|
| 350 | 0.00 | 254.0 | 360.0 | 1.0 |
|-----|------|-------|-------|-----|

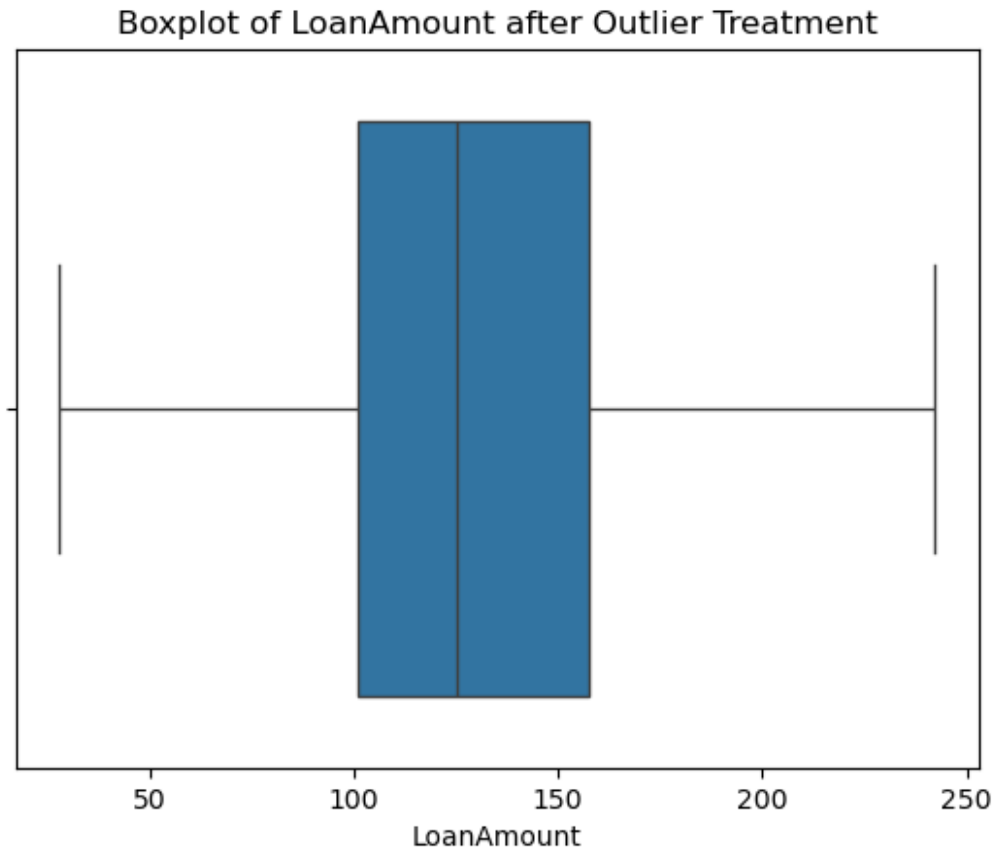
| | Property_Area |
|-----|---------------|
| 8 | Urban |
| 18 | Urban |
| 24 | Urban |
| 27 | Urban |
| 81 | Urban |
| 83 | Rural |
| 91 | Rural |
| 96 | Urban |
| 124 | Urban |
| 143 | Semiurban |
| 144 | Rural |
| 189 | Rural |
| 194 | Semiurban |
| 284 | Urban |
| 285 | Urban |
| 331 | Urban |
| 345 | Urban |
| 350 | Rural |

Handle the outliers at the threshold values

```
df['LoanAmount'] = df['LoanAmount'].clip(lower=Q1 - threshold,  
upper=Q3 + threshold)
```

Recheck the boxplot

```
sns.boxplot(x='LoanAmount', data=df)  
plt.title('Boxplot of LoanAmount after Outlier Treatment')  
plt.show()
```



The box plot of credit history shows some outliers, with a few individuals having very low scores. These could impact loan decisions and should be reviewed carefully to ensure unbiased evaluations.

The box plots reveal that 'ApplicantIncome' varies the most and has a few outliers. 'LoanAmount' and 'Loan_Term' have less variation but still show some outliers, while 'Credit_History' has very little variation and almost no outliers.

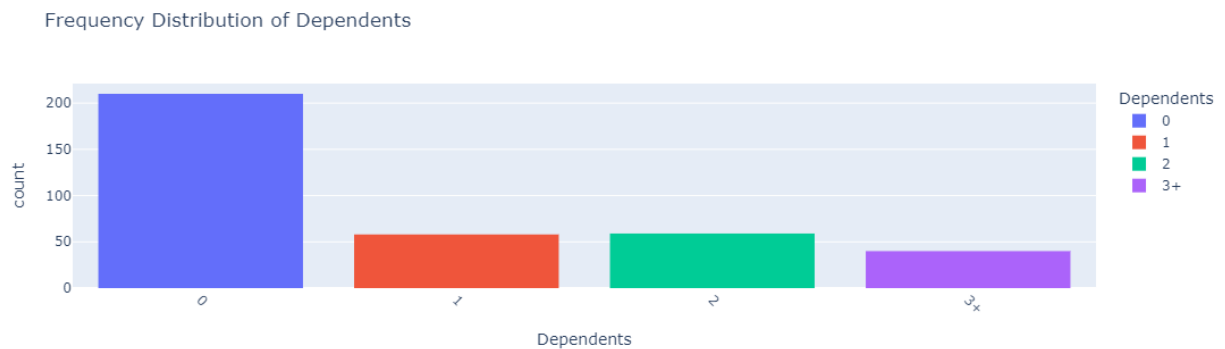
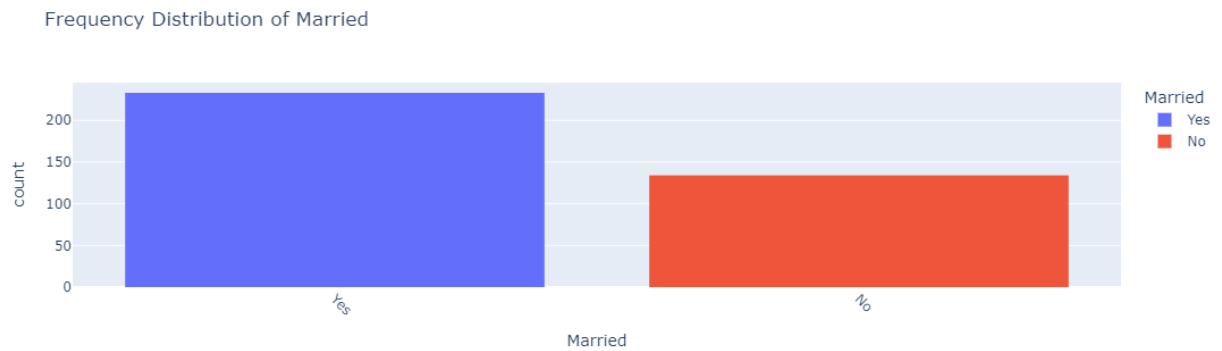
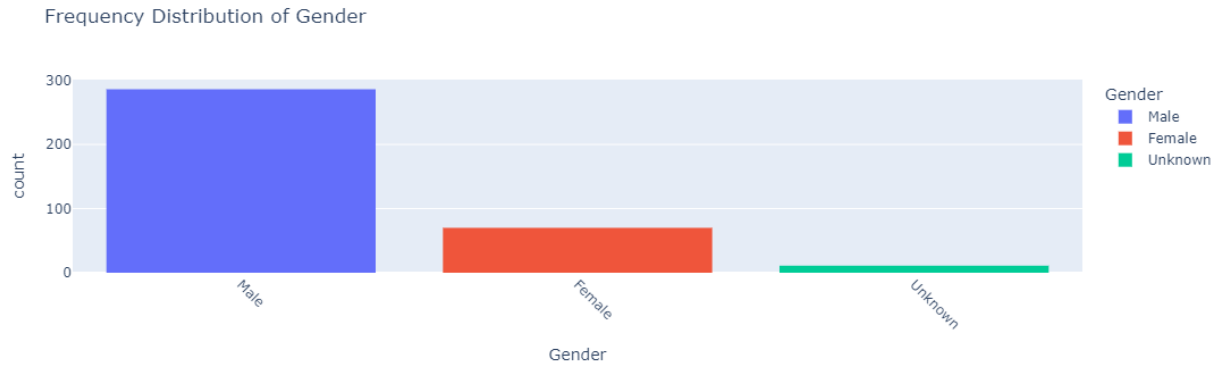
Analyze categorical variables by creating the following plots

Bar Charts: Visualize the frequency distribution of categorical variables

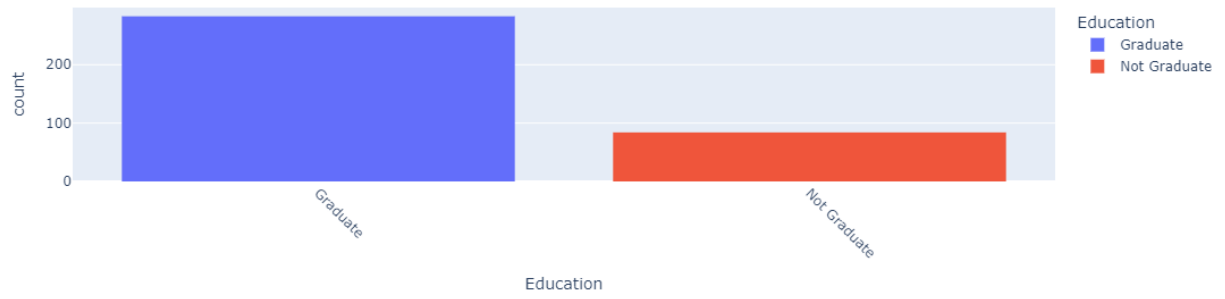
```
# Select categorical columns
categorical_column = df.select_dtypes(include=['object']).columns

# Plot bar charts for each categorical column
```

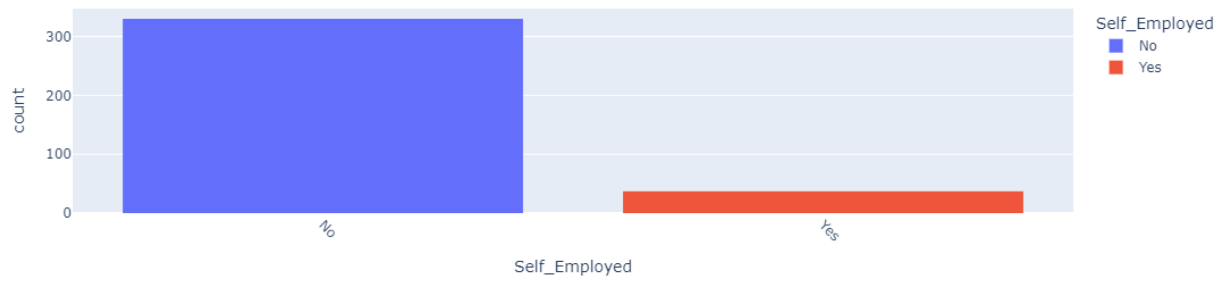
```
for column in categorical_column:
    fig = px.histogram(df, x=column, title=f"Frequency Distribution of {column}", color=column)
    fig.update_xaxes(tickangle=45) # Rotate x-axis labels
    fig.show()
```



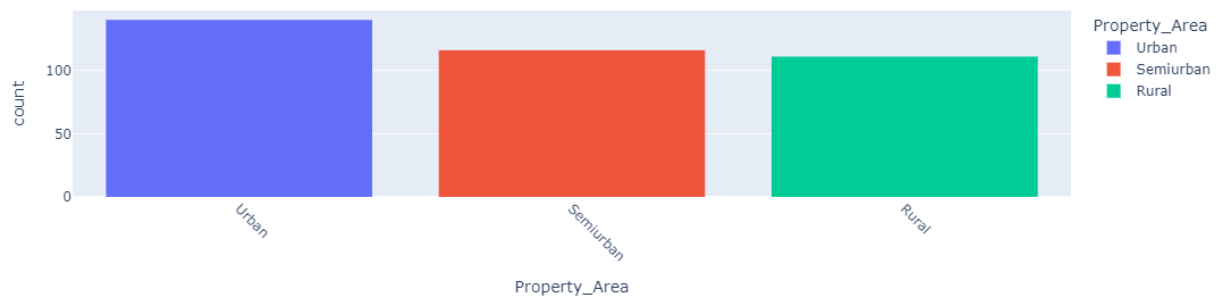
Frequency Distribution of Education



Frequency Distribution of Self_Employed



Frequency Distribution of Property_Area



Outcome from above charts

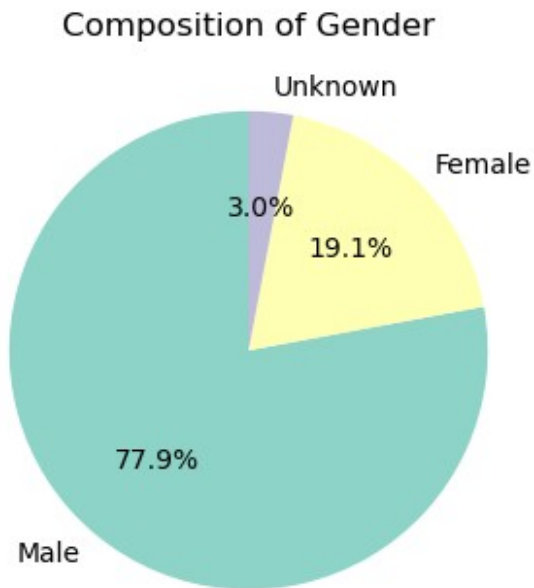
Most applicants are male, married, and graduates with no dependents, living in urban areas and not self-employed, while there is a notable portion of female, unmarried, and self-employed applicants with dependents.

Insights to Grow Business

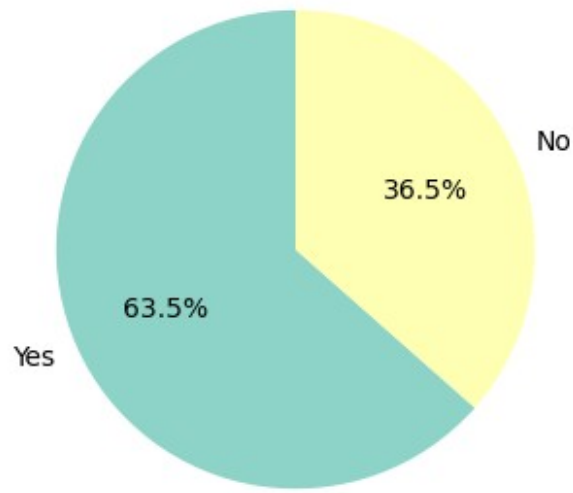
To grow the business, offer affordable and flexible loans for low-income applicants and co-applicants, focus on mid-range loan amounts, reward good credit history, help co-applicants improve their finances, and promote long-term loans for easier repayments.

Pie Charts: Represent the composition of categorical variables

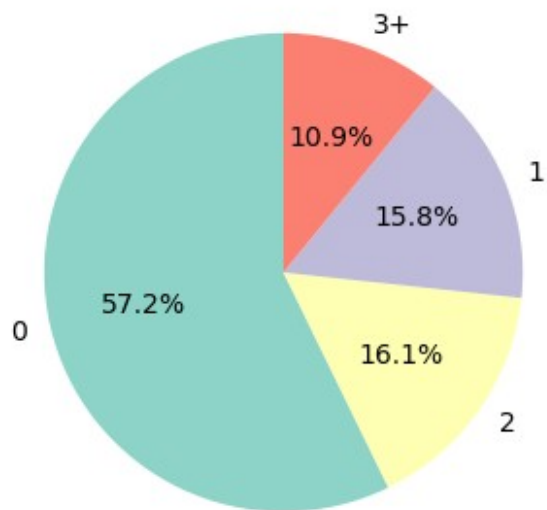
```
categorical_columns = df.select_dtypes(include=['object']).columns
# Plot pie charts for categorical columns
for column in categorical_columns:
    plt.figure(figsize=(10,4))
    df[column].value_counts().plot.pie(autopct='%1.1f%%',
startangle=90, colors=sns.color_palette("Set3"))
    plt.title(f"Composition of {column}")
    plt.ylabel('')
    plt.show()
```



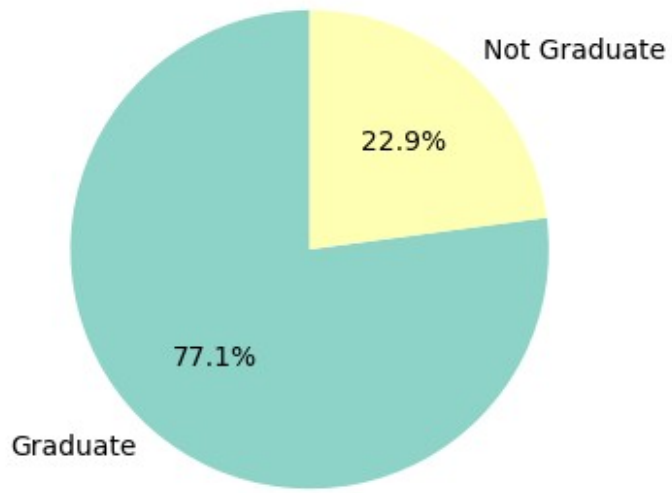
Composition of Married



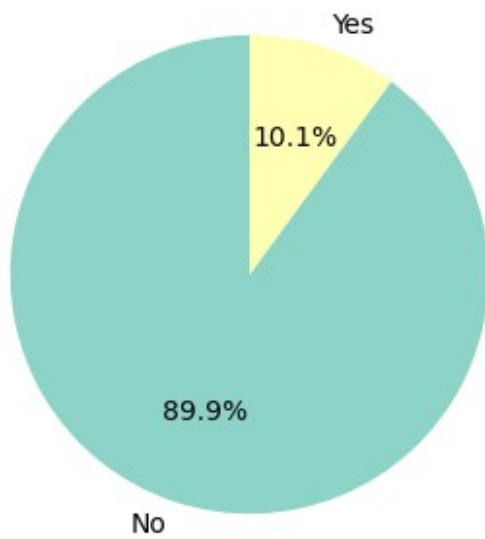
Composition of Dependents



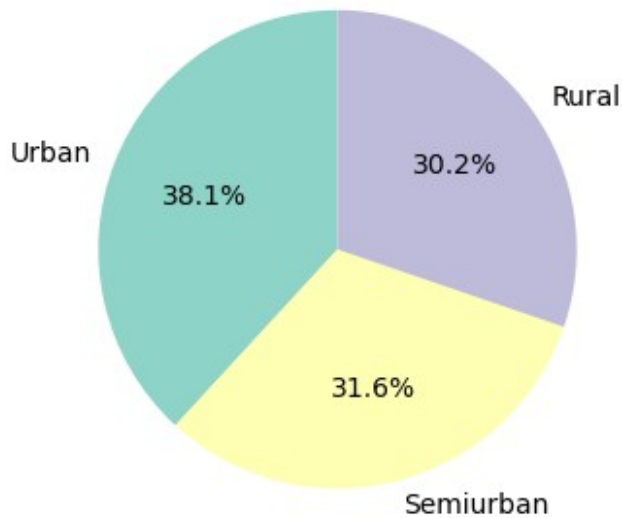
Composition of Education



Composition of Self_Employed



Composition of Property_Area



Outcome from above charts

Most applicants are male, married, and graduates with no dependents, living in urban areas and not self-employed, while there is a notable portion of female, unmarried, and self-employed applicants with dependents.

Insights to Grow Business

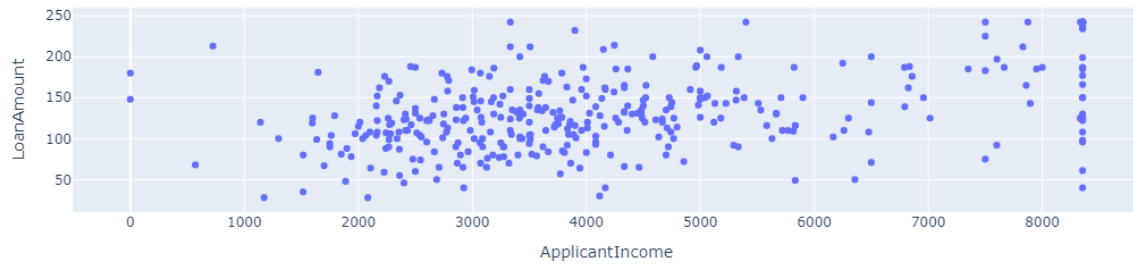
To grow the business, offer affordable and flexible loans for low-income applicants and co-applicants, focus on mid-range loan amounts, reward good credit history, help co-applicants improve their finances, and promote long-term loans for easier repayments.

Bivariate Analysis

Create scatter plots to explore relationships between pairs of numeric variables

```
# Create the scatter plot
fig = px.scatter(df, x='ApplicantIncome', y='LoanAmount',
title="Scatter Plot between Applicant Income and Loan Amount")
# Show the plot
fig.show()
```

Scatter Plot between Applicant Income and Loan Amount



Outcome from above charts

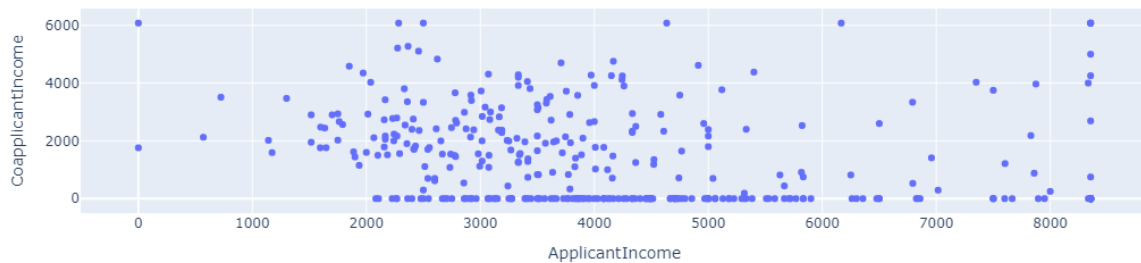
The above scatter plot shows a slight positive link between Applicant Income and Loan Amount, with some outliers where lower incomes have higher loan amounts.

Insights to Grow Business

The scatter plot shows that while higher-income applicants usually get bigger loans, some low-income applicants still receive large loans. This could be a chance to improve risk assessment or create loan options for low-income individuals with good repayment potential.

```
# Create the scatter plot
fig = px.scatter(df, x='ApplicantIncome', y='CoapplicantIncome',
title='Scatter Plot Comparing ApplicantIncome and CoapplicantIncome',
                labels={'ApplicantIncome': 'ApplicantIncome',
                        'CoapplicantIncome': 'CoapplicantIncome'})
# Show the plot
fig.show()
```

Scatter Plot Comparing ApplicantIncome and CoapplicantIncome



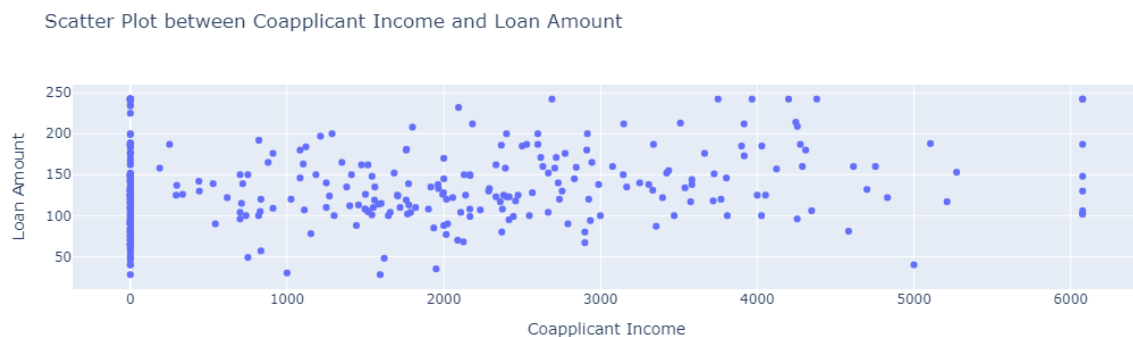
Outcome from above charts

The above scatter plot shows a slight positive relationship between Applicant Income and Coapplicant Income, with most points clustered in the lower income ranges.

Insights to Grow Business

The scatter plot shows that applicants and co-applicants tend to have similar, lower incomes. This presents an opportunity to create affordable joint loan products for low-income applicants and their co-applicants.

```
fig = px.scatter(df, x='CoapplicantIncome', y='LoanAmount',  
title='Scatter Plot between Coapplicant Income and Loan Amount',  
labels={'CoapplicantIncome': 'Coapplicant Income',  
'LoanAmount': 'Loan Amount'})  
fig.show()
```



Outcome from above charts

The scatter plot shows a weak positive correlation between Coapplicant Income and Loan Amount, with most data points concentrated in the lower income ranges.

Insights to Grow Business

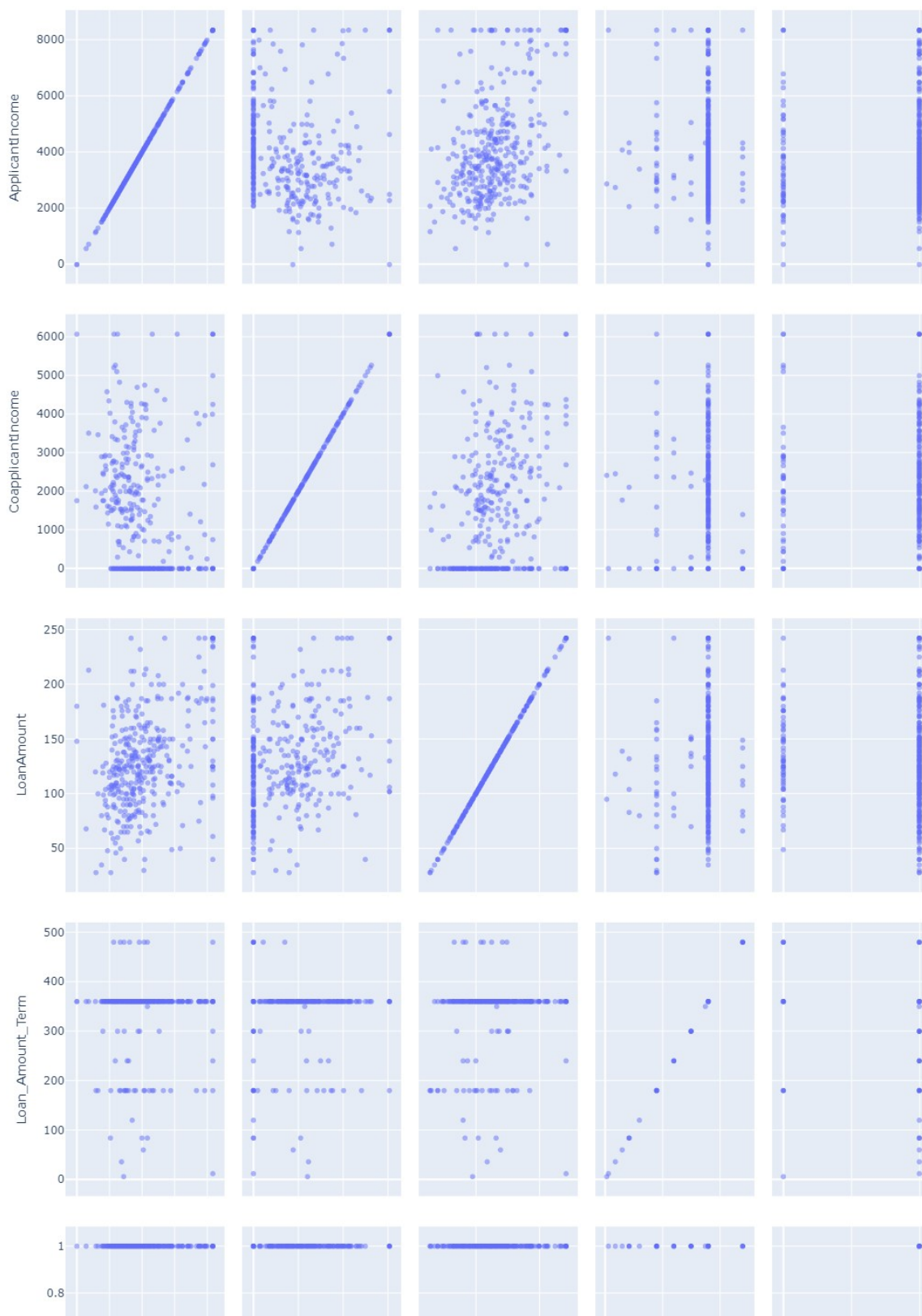
The scatter plot shows a small link between co-applicant income and loan amounts, with most applicants earning lower incomes. This indicates a chance to offer affordable loans with flexible terms for those with lower co-applicant incomes.

Use pair plots (scatter matrix) to visualize interactions between multiple numeric variables

```
# Create the scatter matrix plot  
fig = px.scatter_matrix(df[numeric_columns],  
title="Pair Plot of Numeric Columns",
```

```
        dimensions=numeric_columns,  
        opacity=0.5)  
  
# Update layout for better spacing  
fig.update_layout(height=1800, width=1000, title_x=0.5)  
# Show the plot  
fig.show()
```

Pair Plot of Numeric Columns



Outcome from above charts

The pair plot shows that higher incomes for both applicants and co-applicants are linked to larger loans and longer terms, while credit history seems to be an independent factor, suggesting it should be prioritized in loan approvals.

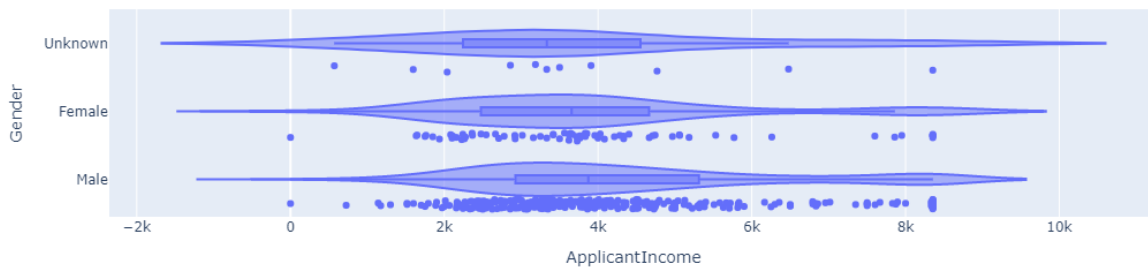
Insights to Grow Business

The pair plot shows that higher incomes lead to larger loans and longer terms, suggesting loan models could be improved. It also points out that credit history should be a key factor in loan approval, independent of other variables.

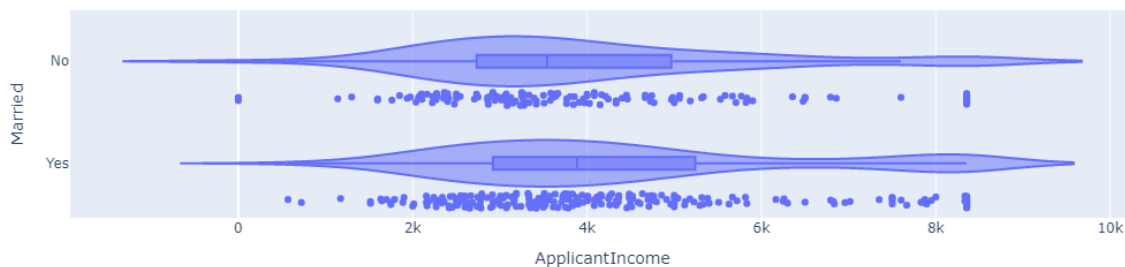
Investigate the relationship between categorical and numeric variables using box plots or violin plots

```
numeric = df[numeric_columns]
categorical = df[categorical_columns]
# Loop through the numeric and categorical columns to create violin plots
for x in numeric_columns:
    for y in categorical_columns:
        fig = px.violin(df, x=x, y=y, title=f'Violin Plot of {x} by {y}', box=True, points="all")
        fig.show()
```

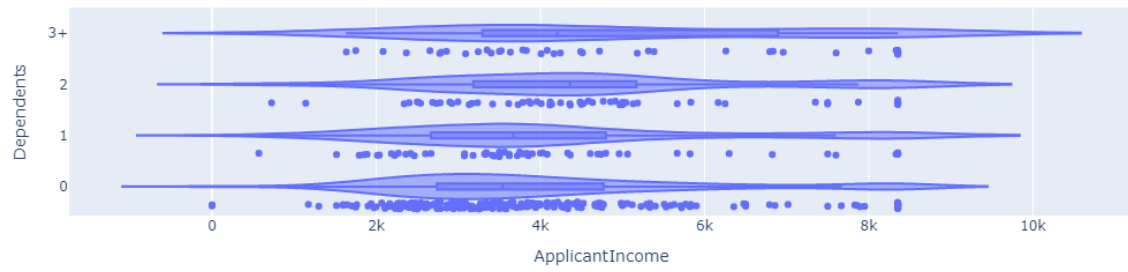
Violin Plot of ApplicantIncome by Gender



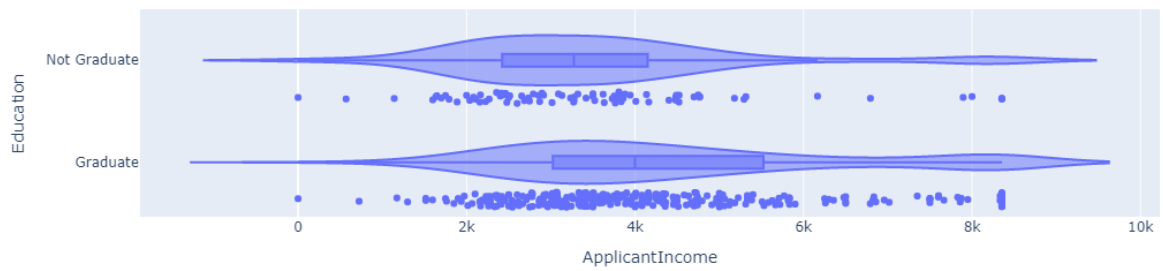
Violin Plot of ApplicantIncome by Married



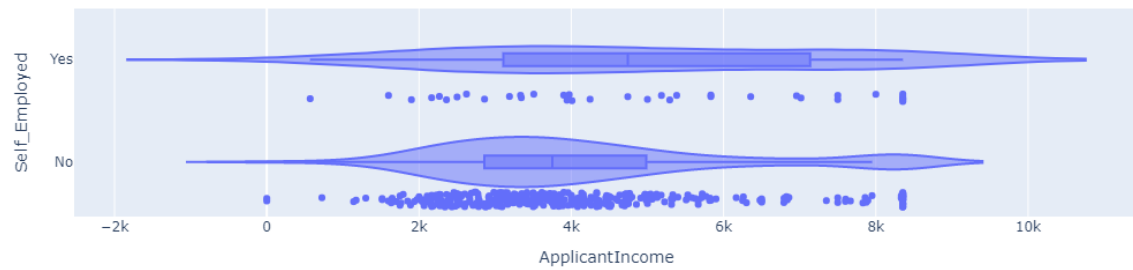
Violin Plot of ApplicantIncome by Dependents



Violin Plot of ApplicantIncome by Education



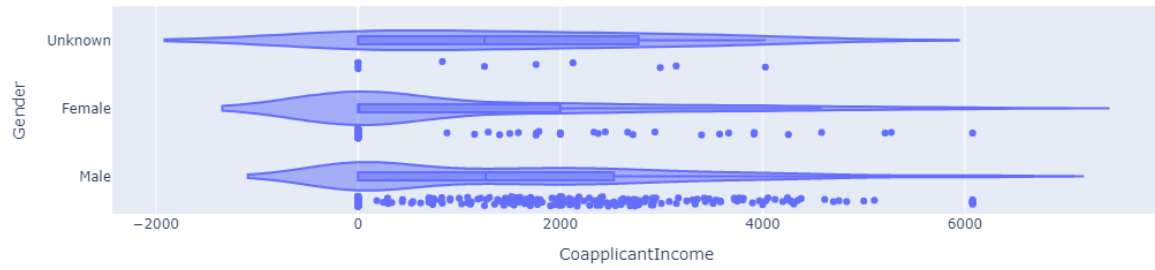
Violin Plot of ApplicantIncome by Self_Employed



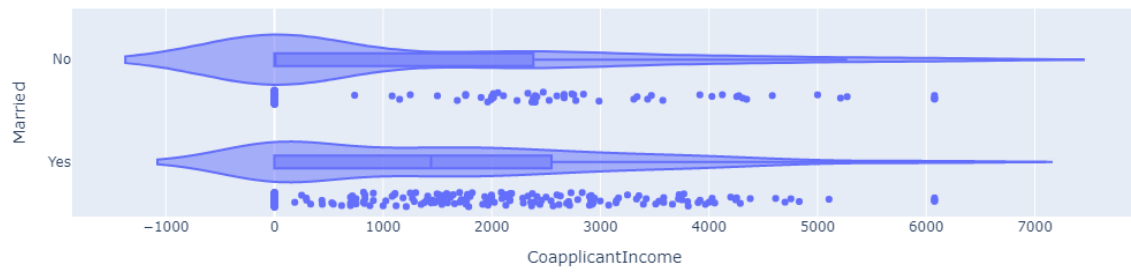
Violin Plot of ApplicantIncome by Property_Area



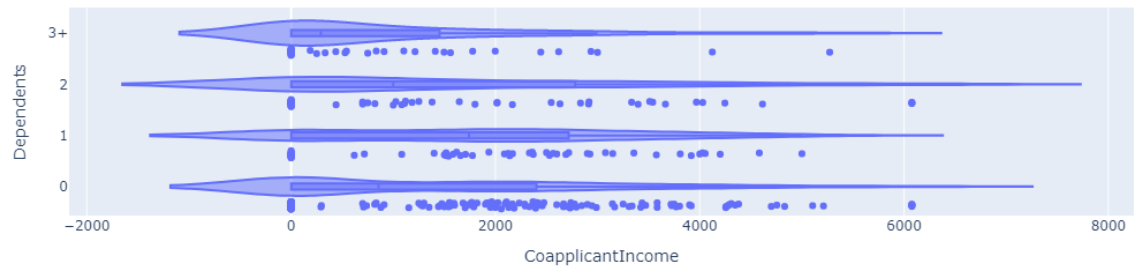
Violin Plot of CoapplicantIncome by Gender



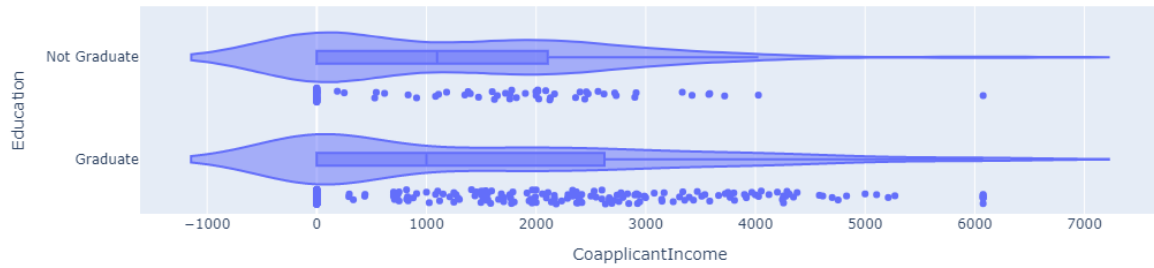
Violin Plot of CoapplicantIncome by Married



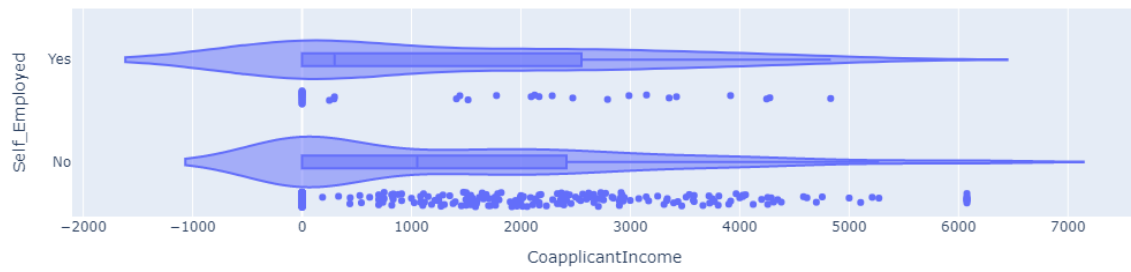
Violin Plot of CoapplicantIncome by Dependents



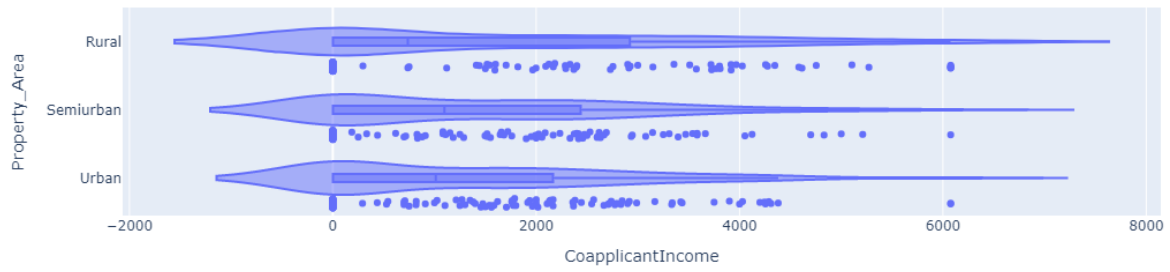
Violin Plot of CoapplicantIncome by Education



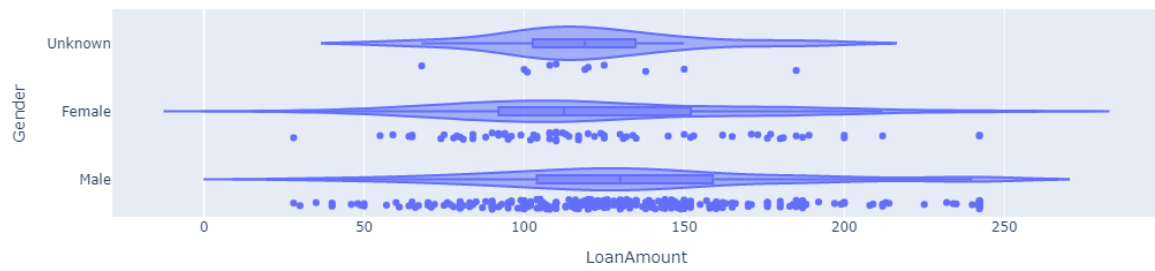
Violin Plot of CoapplicantIncome by Self_Employed



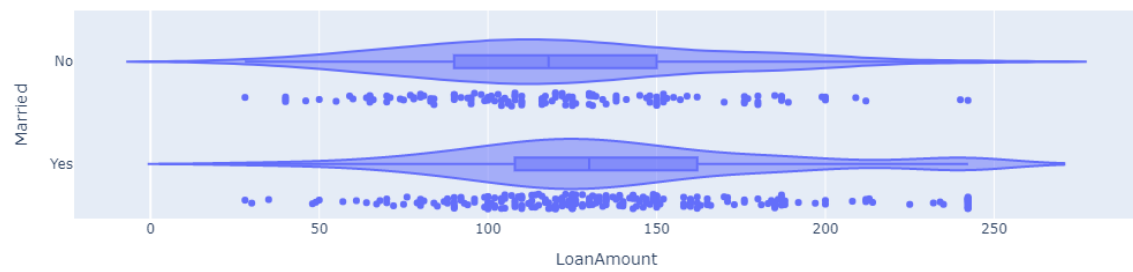
Violin Plot of CoapplicantIncome by Property_Area



Violin Plot of LoanAmount by Gender



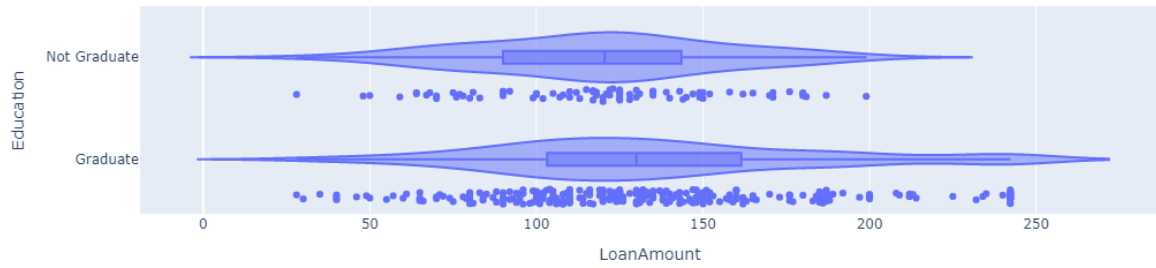
Violin Plot of LoanAmount by Married



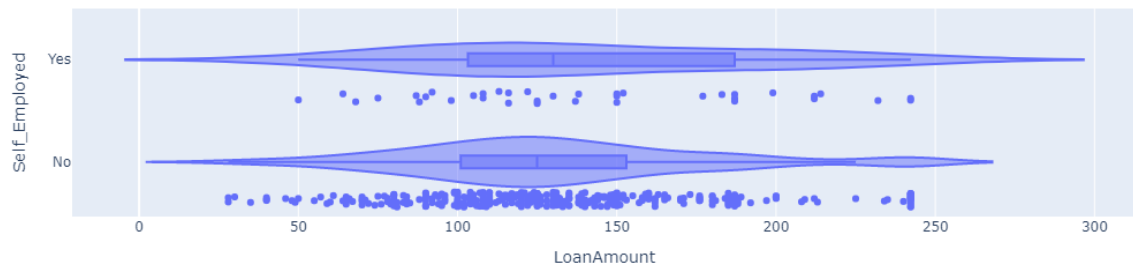
Violin Plot of LoanAmount by Dependents



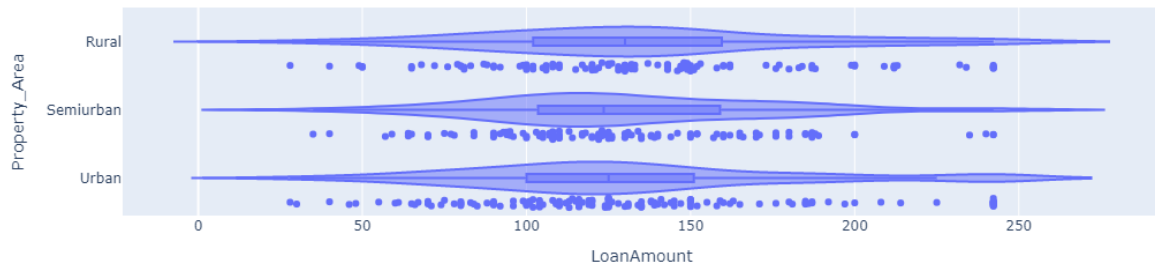
Violin Plot of LoanAmount by Education



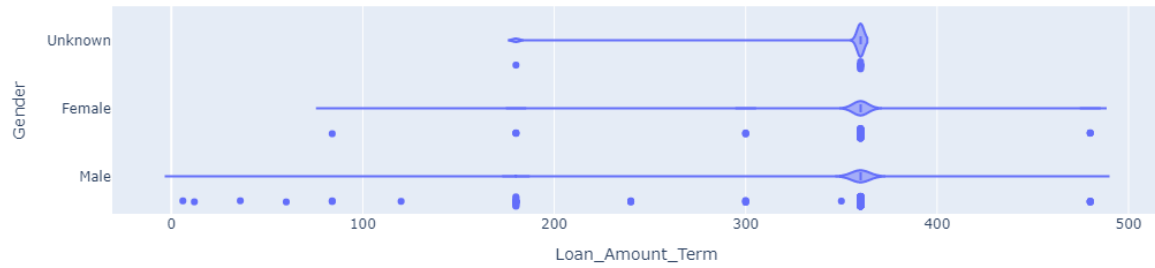
Violin Plot of LoanAmount by Self_Employed



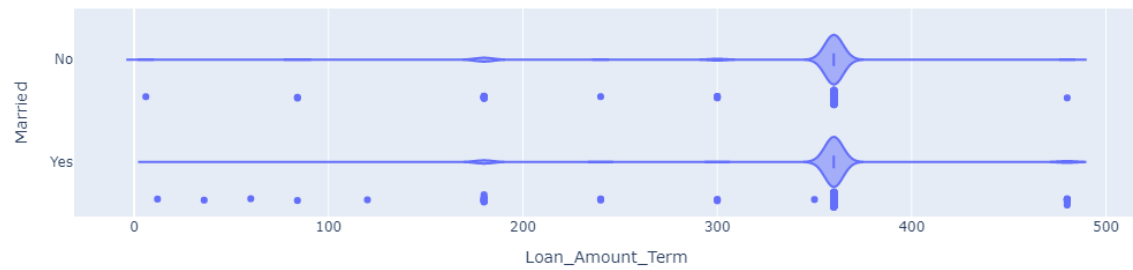
Violin Plot of LoanAmount by Property_Area



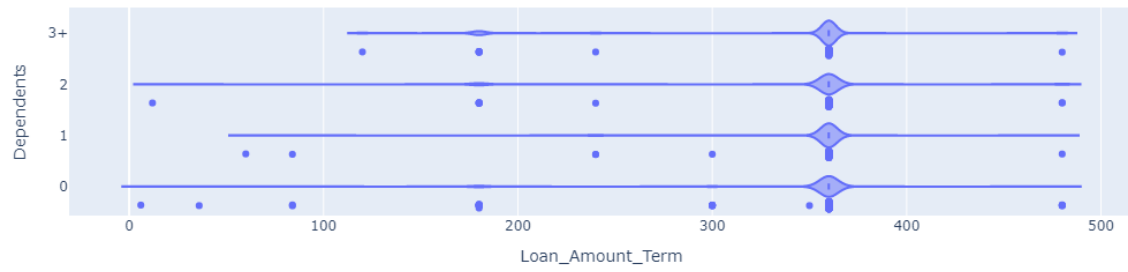
Violin Plot of Loan_Amount_Term by Gender



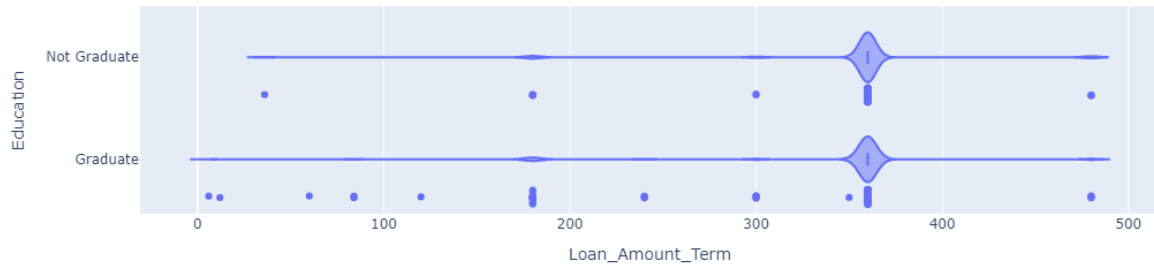
Violin Plot of Loan_Amount_Term by Married



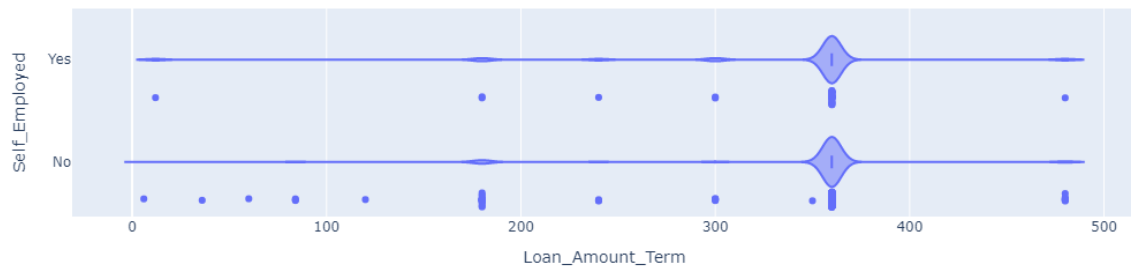
Violin Plot of Loan_Amount_Term by Dependents



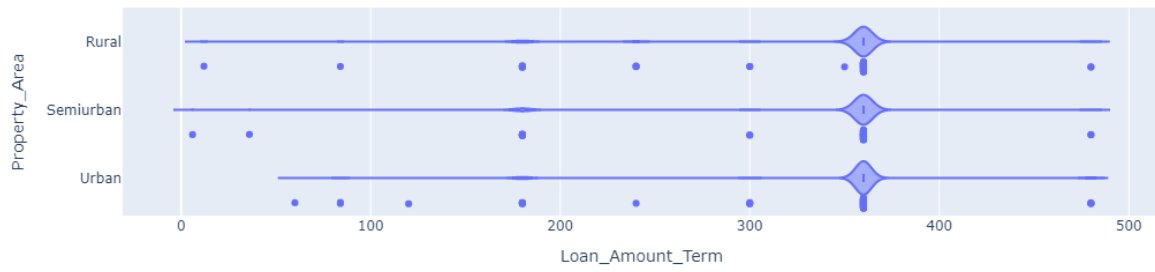
Violin Plot of Loan_Amount_Term by Education



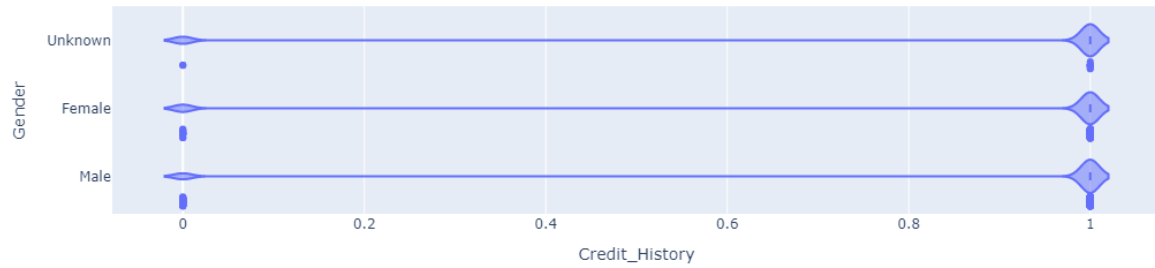
Violin Plot of Loan_Amount_Term by Self_Employed



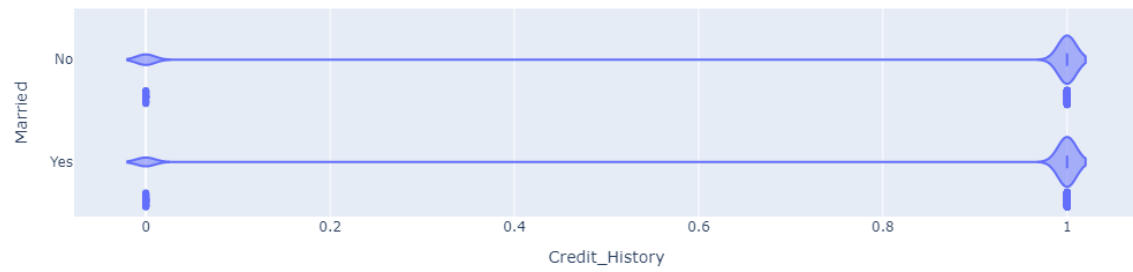
Violin Plot of Loan_Amount_Term by Property_Area



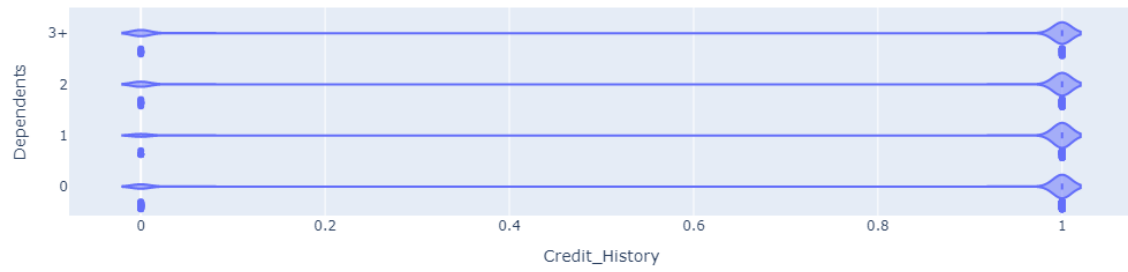
Violin Plot of Credit_History by Gender



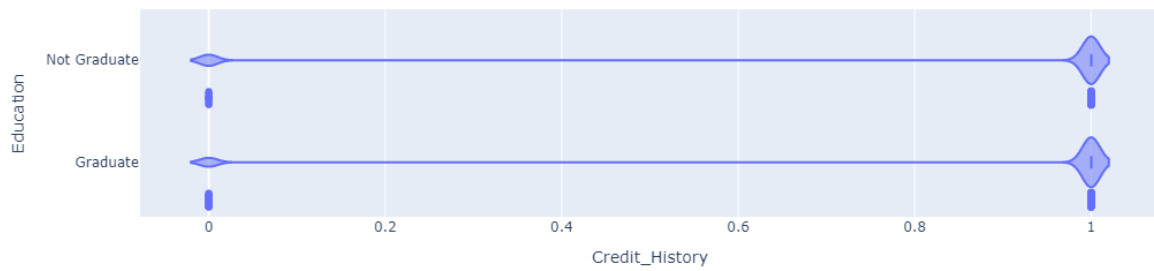
Violin Plot of Credit_History by Married



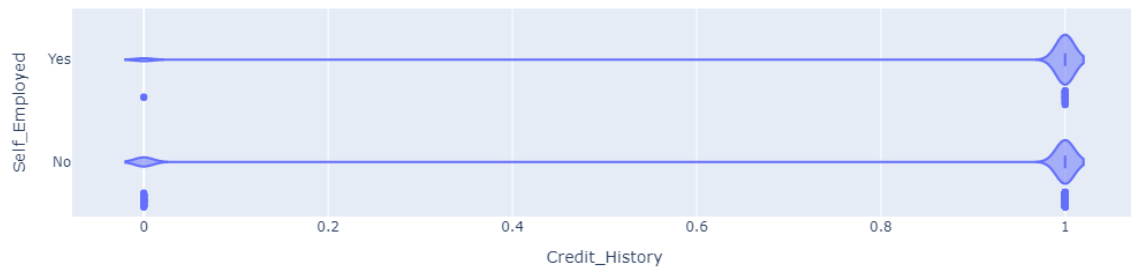
Violin Plot of Credit_History by Dependents

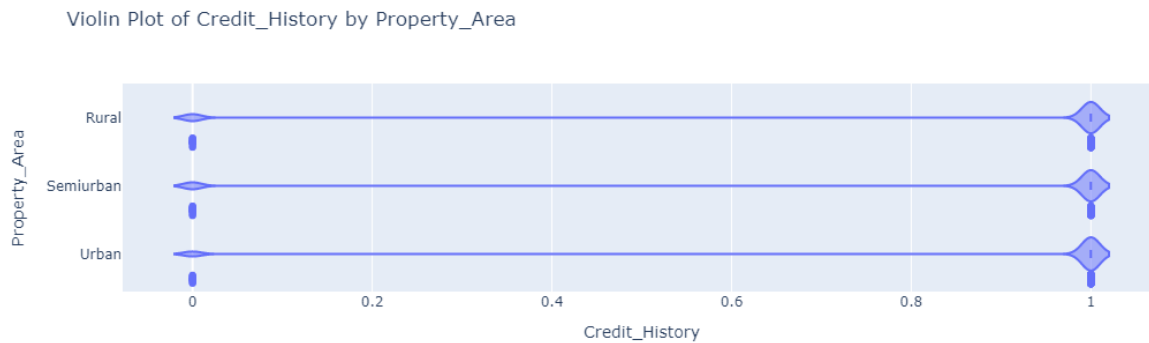


Violin Plot of Credit_History by Education



Violin Plot of Credit_History by Self_Employed





Outcome from above charts

The violin plots show that men, married people, graduates, self-employed individuals, and those living in urban areas tend to earn more, take larger loans, and opt for longer repayment terms. On the other hand, co-applicants, especially women and those from rural areas, generally have lower incomes. Credit history remains similar across genders, marital status, and education, making it a crucial factor in loan decisions.

Insights to Grow Business

To grow the business, focus on offering bigger loans and longer terms to higher-income groups like men, married people, self-employed, and urban residents. At the same time, create special loan options for co-applicants with lower incomes, especially women and those from rural areas. Also, make credit history a key factor in approval decisions.

Multivariate Analysis

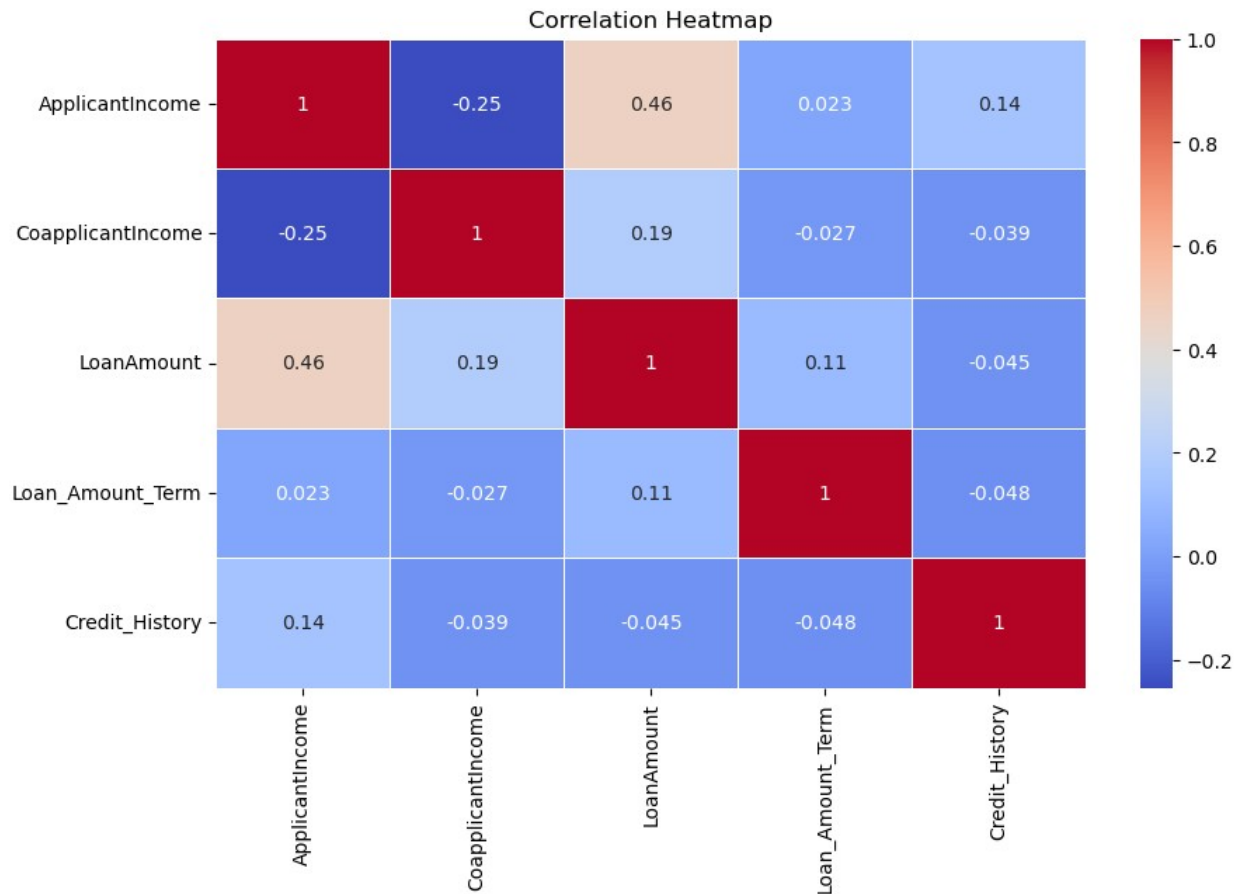
Perform a correlation analysis to identify relationships between numeric variables. Visualize correlations using a heatmap

```
# Assuming df is your DataFrame
df1 = pd.DataFrame(df)

# Select only numeric columns
numeric_df = df1.select_dtypes(include='number')

# Calculate correlation
correlation_matrix = numeric_df.corr()

# Plot heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```

Outcome from above charts

The heatmap shows that as applicant income goes up, the loan amount also tends to increase. There's also a moderate link between loan amount and loan term. However, applicant income and co-applicant income are only weakly related, and there's a slight positive connection between applicant income and credit history.

Insights to Grow Business

To grow the business, focus on offering larger loan amounts and longer terms to higher-income applicants. Also, consider enhancing credit history as a key factor in loan approval while exploring ways to support applicants with lower co-applicant incomes.

Create a stacked bar chart to show the distribution of categorical variables across multiple categories

```
import plotly.graph_objs as go
# Assuming df4 is your DataFrame and you want to create a crosstab
crosstab_data = pd.crosstab(df['Gender'], df['Education'])

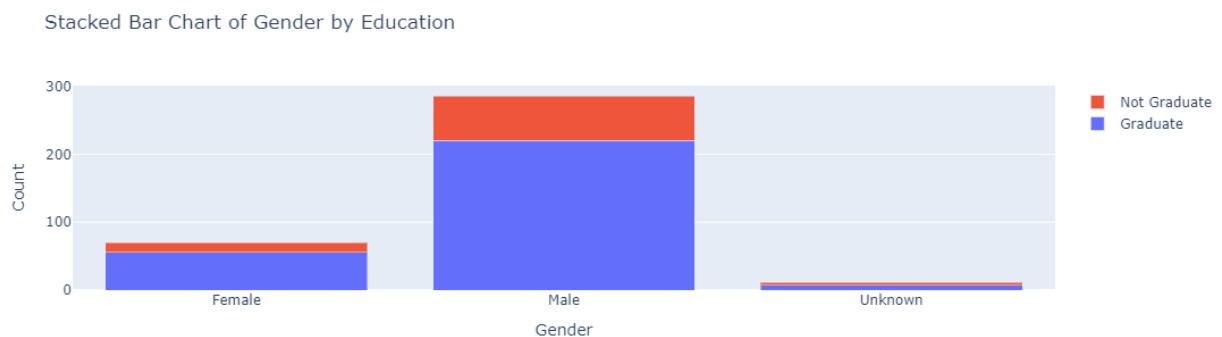
# Create traces for the stacked bar chart
data = []
```

```

for col in crosstab_data.columns:
    data.append(go.Bar(name=col, x=crosstab_data.index,
y=crosstab_data[col]))

# Create the layout
layout = go.Layout(
    barmode='stack',
    title='Stacked Bar Chart of Gender by Education',
    xaxis=dict(title='Gender'),
    yaxis=dict(title='Count'),
)
# Create the figure and plot it
fig = go.Figure(data=data, layout=layout)
fig.show()

```



Outcome from above charts

The stacked bar chart reveals that most loan applicants are male graduates, while a smaller portion are female graduates and non-graduates

Insights to Grow Business

To grow the business, focus on male graduates while creating special offers to attract more female graduates and non-graduates.

Summary

The analysis highlights that factors like income, credit history, and demographics play a key role in loan approval decisions. To grow the business, focus on offering products that cater to high-income applicants with strong credit histories while exploring ways to support applicants with lower incomes through tailored loan options.