LOAN APPROVAL ANALYSIS

Project Overview

This project involves performing an **Exploratory Data Analysis (EDA)** on a home loan approval dataset. The primary objectives are to:

1. Understand the dataset structure and identify missing values.
2. Explore and visualize numerical and categorical variables to uncover patterns and trends.
3. Analyze relationships between variables using advanced visualization techniques.
4. Derive actionable insights to guide decision-making and recommendations.

Dataset Understanding

The dataset consists of **12 columns** and contains information on **367 records**:

1. **Data Types**:
    - The dataset contains a mix of categorical and numerical variables, necessitating appropriate handling for analysis and visualization.
2. **Numerical Variables**:
    - `ApplicantIncome`, `CoapplicantIncome`, `LoanAmount`, and `Loan_Amount_Term` are continuous variables suitable for statistical analysis and visualization.
3. **Categorical Variables**:

- Variables like `Gender`, `Married`, `Education`, `Self_Employed`, and `Property_Area` are categorical and provide opportunities for segmentation and bivariate analysis.

This understanding of the dataset serves as a foundation for further exploration and analysis.

1. Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Loading the Dataset

In [3]:
```
df=pd.read_csv("D:/SKCL/EDA Capstone/Loan Approval EDA/loan_sanction_test.cs
```

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

In [5]:
```
df.head(2)
```

Out[5]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Applica
0	LP001015	Male	Yes	0	Graduate	No	
1	LP001022	Male	Yes	1	Graduate	No	

4 Description of Dataset

In [7]:
```
df.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	
ApplicantIncome	367.0	4805.599455	4910.685399	0.0	2864.00	3786.0	50
CoapplicantIncome	367.0	1569.577657	2334.232099	0.0	0.00	1025.0	24
LoanAmount	362.0	136.132597	61.366652	28.0	100.25	125.0	1
Loan_Amount_Term	361.0	342.537396	65.156643	6.0	360.00	360.0	3
Credit_History	338.0	0.825444	0.380150	0.0	1.00	1.0	

5. Information of Dataset

In [9]:
```
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
```

6. Shape and Size of Dataset

In [11]:
```
df.shape
```

Out[11]:
```
(367, 12)
```

In [12]:
```
df.size
```

Out[12]:
```
4404
```

7. Checking Missing values

In [21]:
```
df.isna().sum()
```

Out[21]:
```
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
```

In [23]:
```
df[df['Gender'].isna()]
```

Out[23]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Appl
22	LP001128	NaN	No	0	Graduate	No	
51	LP001287	NaN	Yes	3+	Not Graduate	No	
106	LP001563	NaN	No	0	Graduate	No	
138	LP001769	NaN	No	NaN	Graduate	No	
209	LP002165	NaN	No	1	Not Graduate	No	
231	LP002298	NaN	No	0	Graduate	Yes	
245	LP002355	NaN	Yes	0	Graduate	No	
279	LP002553	NaN	No	0	Graduate	No	
296	LP002614	NaN	No	0	Graduate	No	
303	LP002657	NaN	Yes	1	Not Graduate	Yes	
318	LP002775	NaN	No	0	Not Graduate	No	

In [25]:
```
df[df['Dependents'].isna()]
```

Out[25]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Appl
46	LP001237	Male	Yes	NaN	Not Graduate	No	
70	LP001366	Female	No	NaN	Graduate	No	
111	LP001587	Male	Yes	NaN	Graduate	No	
138	LP001769	NaN	No	NaN	Graduate	No	
202	LP002111	Male	Yes	NaN	Graduate	No	
247	LP002360	Male	Yes	NaN	Graduate	No	
251	LP002385	Male	Yes	NaN	Graduate	No	
265	LP002441	Male	No	NaN	Graduate	No	
302	LP002654	Female	No	NaN	Graduate	Yes	
312	LP002754	Male	No	NaN	Graduate	No	

In [27]:
```
df[df['Self_Employed'].isna()]
```

Out[27]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Appl
8	LP001059	Male	Yes	2	Graduate	NaN	
11	LP001082	Male	Yes	1	Graduate	NaN	
13	LP001094	Male	Yes	2	Graduate	NaN	
36	LP001208	Male	Yes	2	Graduate	NaN	
72	LP001375	Male	Yes	1	Graduate	NaN	
89	LP001472	Female	No	0	Graduate	NaN	
142	LP001789	Male	Yes	3+	Not Graduate	NaN	
161	LP001906	Male	No	0	Graduate	NaN	
168	LP001950	Female	Yes	3+	Graduate	NaN	
175	LP001999	Male	Yes	2	Graduate	NaN	
192	LP002069	Male	Yes	2	Not Graduate	NaN	
243	LP002346	Male	Yes	0	Graduate	NaN	
255	LP002399	Male	No	0	Graduate	NaN	
259	LP002415	Female	No	1	Graduate	NaN	
276	LP002542	Male	Yes	0	Graduate	NaN	
278	LP002551	Male	Yes	3+	Not Graduate	NaN	
285	LP002572	Male	Yes	1	Graduate	NaN	
287	LP002584	Male	No	0	Graduate	NaN	
294	LP002610	Male	Yes	1	Not Graduate	NaN	
297	LP002630	Male	No	0	Not Graduate	NaN	
301	LP002651	Male	Yes	1	Graduate	NaN	
323	LP002791	Male	No	1	Graduate	NaN	
326	LP002803	Male	Yes	1	Not Graduate	NaN	

In [29]:
```
df[df['LoanAmount'].isna()]
```

Out[29]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Appl
78	LP001415	Male	Yes	1	Graduate	No	
101	LP001542	Female	Yes	0	Graduate	No	
188	LP002057	Male	Yes	0	Not Graduate	No	
247	LP002360	Male	Yes	NaN	Graduate	No	
289	LP002593	Male	Yes	1	Graduate	No	

In [31]:
```
df[df['Loan_Amount_Term'].isna()]
```

Out[31]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Appl
45	LP001232	Male	Yes	0	Graduate	No	
48	LP001268	Male	No	0	Graduate	No	
117	LP001611	Male	Yes	1	Graduate	No	
129	LP001695	Male	Yes	1	Not Graduate	No	
184	LP002045	Male	Yes	3+	Graduate	No	
214	LP002183	Male	Yes	0	Not Graduate	No	

In [33]:
```
df[df['Credit_History'].isna()]
```

Out[33]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Appl
3	LP001035	Male	Yes	2	Graduate	No	
12	LP001083	Male	No	3+	Graduate	No	
26	LP001163	Male	Yes	2	Graduate	No	
28	LP001174	Male	Yes	0	Graduate	No	
45	LP001232	Male	Yes	0	Graduate	No	
90	LP001475	Male	Yes	0	Graduate	Yes	
99	LP001527	Male	Yes	3+	Graduate	No	
104	LP001558	Male	No	0	Graduate	No	
115	LP001601	Male	No	3+	Graduate	No	
139	LP001771	Female	No	3+	Graduate	No	
143	LP001791	Male	Yes	0	Graduate	Yes	
164	LP001921	Male	No	1	Graduate	No	
177	LP002009	Female	No	0	Graduate	No	
179	LP002017	Male	Yes	3+	Graduate	No	
185	LP002046	Male	Yes	0	Not Graduate	No	
202	LP002111	Male	Yes	NaN	Graduate	No	
220	LP002212	Male	Yes	0	Graduate	No	
259	LP002415	Female	No	1	Graduate	NaN	
262	LP002425	Male	No	0	Graduate	No	
265	LP002441	Male	No	NaN	Graduate	No	
282	LP002566	Female	No	0	Graduate	No	
286	LP002581	Male	Yes	0	Not Graduate	No	
305	LP002712	Male	No	2	Not Graduate	No	
329	LP002816	Male	Yes	1	Graduate	No	
336	LP002853	Female	No	0	Not Graduate	No	
351	LP002901	Male	No	0	Graduate	No	
358	LP002954	Male	Yes	2	Not Graduate	No	
360	LP002965	Female	Yes	0	Graduate	No	
364	LP002980	Male	No	0	Graduate	No	

8. Treating Missing Data

8.1 Replacing NaN 'Gender' with "Unknown"

In [37]:
```
df.loc[df['Gender'].isna(),'Gender']='Unknown'
df['Gender']
```

Out[37]:
```
0      Male
1      Male
2      Male
3      Male
4      Male
       ...
362    Male
363    Male
364    Male
365    Male
366    Male
Name: Gender, Length: 367, dtype: object
```

8.2 Dropping 'Dependents' having NaN values

In [40]:
```
df = df.dropna(subset=['Dependents'])
```

8.3 Replacing NaN 'Self_Employed' with "Unknown"

In [43]:
```
df.loc[df['Self_Employed'].isna(),'Self_Employed']='Unknown'
df['Self_Employed']
```

Out[43]:
```
0       No
1       No
2       No
3       No
4       No
      ...
362    Yes
363     No
364     No
365     No
366    Yes
Name: Self_Employed, Length: 357, dtype: object
```

8.4 Replacing NaN 'LoanAmount' with "mean"

In [46]:
```
df['LoanAmount'] = np.where(df['LoanAmount'].isna(),df['LoanAmount'].mean(),
df['LoanAmount']
```

Out[46]:
```
0      110.0
1      126.0
2      208.0
3      100.0
4       78.0
       ...
362    113.0
363    115.0
364    126.0
365    158.0
366     98.0
Name: LoanAmount, Length: 357, dtype: float64
```

8.5 Replacing NaN 'Loan_Amount_Term' with "median"

In [49]:
```
df['Loan_Amount_Term'] = np.where(df['Loan_Amount_Term'].isna(),df['Loan_Amo
df['Loan_Amount_Term']
```

Out[49]:
```
0      360.0
1      360.0
2      360.0
3      360.0
4      360.0
       ...
362    360.0
363    360.0
364    360.0
365    360.0
366    180.0
Name: Loan_Amount_Term, Length: 357, dtype: float64
```

8.6 Replacing NaN 'Credit_History' with "median"

In [52]:
```
df['Credit_History'] = np.where(df['Credit_History'].isna(),df['Credit_Histo
df['Credit_History']
```

Out[52]:
```
0      1.0
1      1.0
2      1.0
3      1.0
4      1.0
      ...
362    1.0
363    1.0
364    1.0
365    1.0
366    1.0
Name: Credit_History, Length: 357, dtype: float64
```

9. Treating Outliers

In [55]:
```
def replace_outliers(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
```

```
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Replace values below lower bound with lower bound and values above upp
    data[column] = data[column].apply(lambda x: lower_bound if x < lower_bou

    return data

data = replace_outliers(df, 'ApplicantIncome')
```

In [57]:
```
data = replace_outliers(df, 'CoapplicantIncome')
```

In [59]:
```
data = replace_outliers(df, 'LoanAmount')
```

10. Univariate Analysis

10.1 Histogram Plots

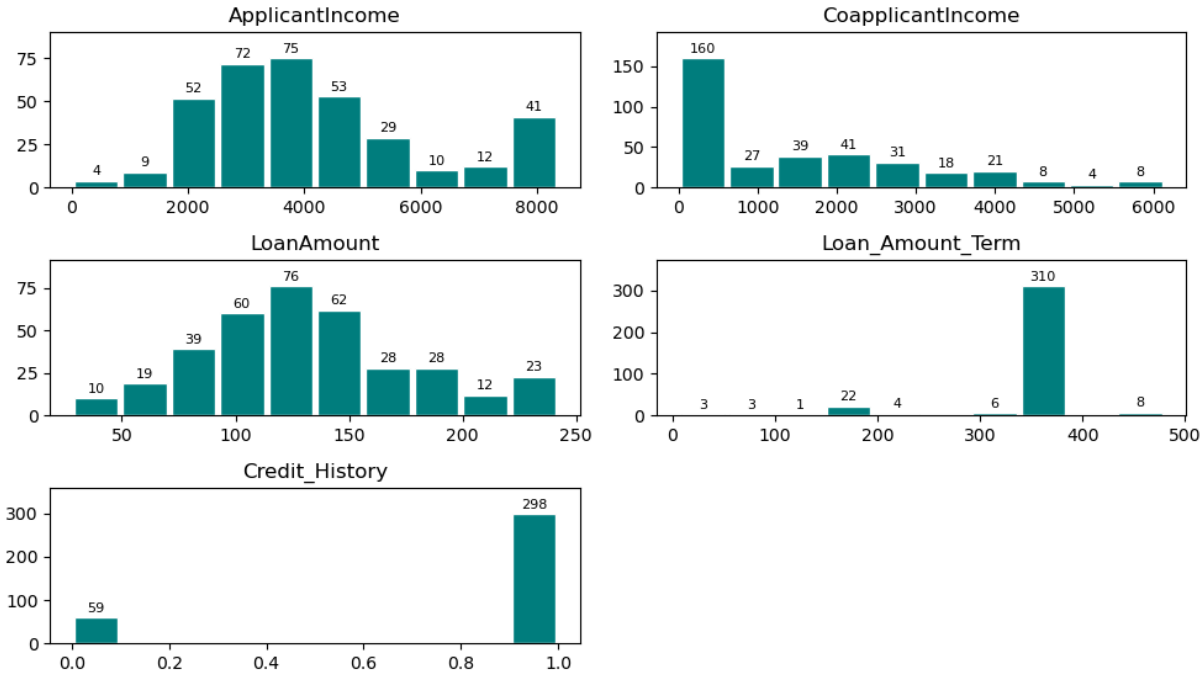
In [63]:
```
axes = df.hist(
    column=['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amou
    color='teal', edgecolor='white', figsize=(10, 6), bins=10, rwidth=0.9, g
)

for ax in axes.flatten():
    if len(ax.patches) > 0:
        y_max = max([patch.get_height() for patch in ax.patches]) * 1.2
        ax.set_ylim(0, y_max)

        for patch in ax.patches:
            height = patch.get_height()
            if height > 0:
                ax.text(
                    patch.get_x() + patch.get_width() / 2,
                    height + y_max * 0.02,
                    f'{int(height)}',
                    ha='center', va='bottom', fontsize=8, color='black'
                )

plt.subplots_adjust(hspace=0.5, wspace=0.4)
plt.tight_layout(rect=[0, 0, 1, 0.95])

plt.show()
```



Insights from the plots:

Applicant Income:

- The distribution is right-skewed, indicating a significant number of applicants with lower incomes.
- A few outliers with exceptionally high incomes are present.

Coapplicant Income:

- Most co-applicants have low or no income.
- The distribution is also right-skewed.

Loan Amount:

- The loan amount distribution is right-skewed, with a majority of loans being for smaller amounts.

Loan Amount Term:

- Most loans have a term of 310 months (30 years).
- There is a smaller number of loans with shorter terms.

Credit History:

- A significant majority of applicants have a good credit history (1).

Recommendations

- **Income Distribution:** Consider offering flexible loan options to accommodate applicants with lower incomes.

- **Co-applicant Income:** Explore strategies to involve co-applicants with higher incomes to improve loan eligibility.
- **Loan Amount:** Offer a wider range of loan amounts to cater to different needs.
- **Loan Term:** Consider offering shorter loan terms to reduce overall interest costs.
- **Credit History:** Prioritize applicants with good credit history for loan approval.

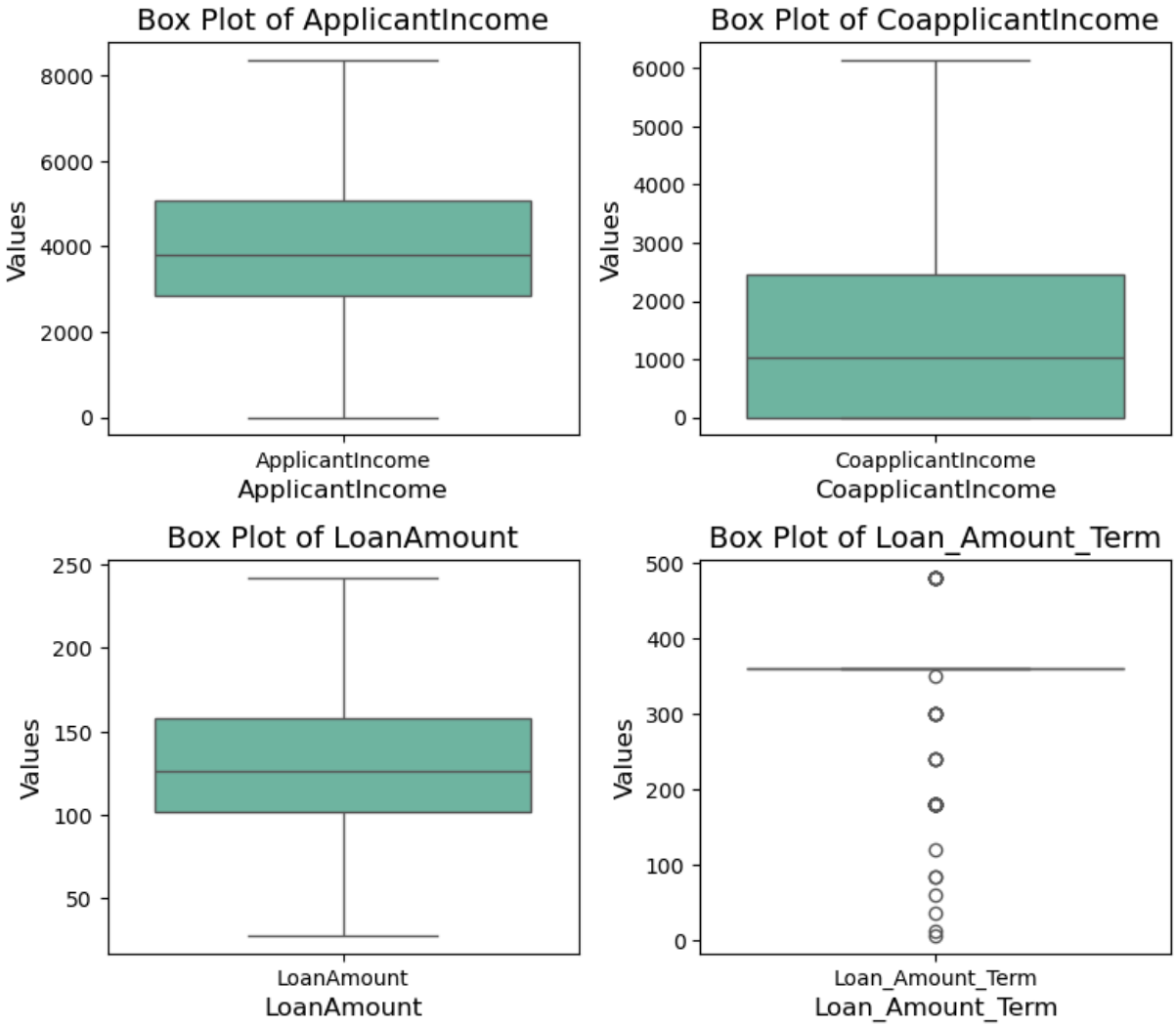
10.2 Box Plots

In [66]:
```
columns = ['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amour

plt.figure(figsize=(8,10))

for i, col in enumerate(columns, 1):
    plt.subplot(3, 2, i)
    sns.boxplot(data=df[[col]], palette='Set2')
    plt.title(f'Box Plot of {col}', fontsize=14)
    plt.xlabel(col, fontsize=12)
    plt.ylabel('Values', fontsize=12)

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



Insights from the box plots:

Applicant Income:

- The distribution is right-skewed, indicating a significant number of applicants with lower incomes.
- There are a few outliers with exceptionally high incomes.
- The median income is around 3000.

Coapplicant Income:

- Most co-applicants have low or no income.
- The median co-applicant income is around 1500.
- There are a few outliers with higher incomes.

Loan Amount:

- The loan amount distribution is right-skewed, with a majority of loans being for smaller amounts.
- The median loan amount is around 120.

Loan Amount Term:

- Most loans have a term of 360 months (30 years).
- There are a few outliers with shorter loan terms, indicating some variations in loan terms.

Recommendations

- **Income Distribution:** Consider offering flexible loan options to accommodate applicants with lower incomes.
- **Co-applicant Income:** Explore strategies to involve co-applicants with higher incomes to improve loan eligibility.
- **Loan Amount:** Offer a wider range of loan amounts to cater to different needs.
- **Loan Term:** Consider offering shorter loan terms to reduce overall interest costs.
- **Credit History:** Prioritize applicants with good credit history for loan approval.

10.3 Bar and Pie Plots

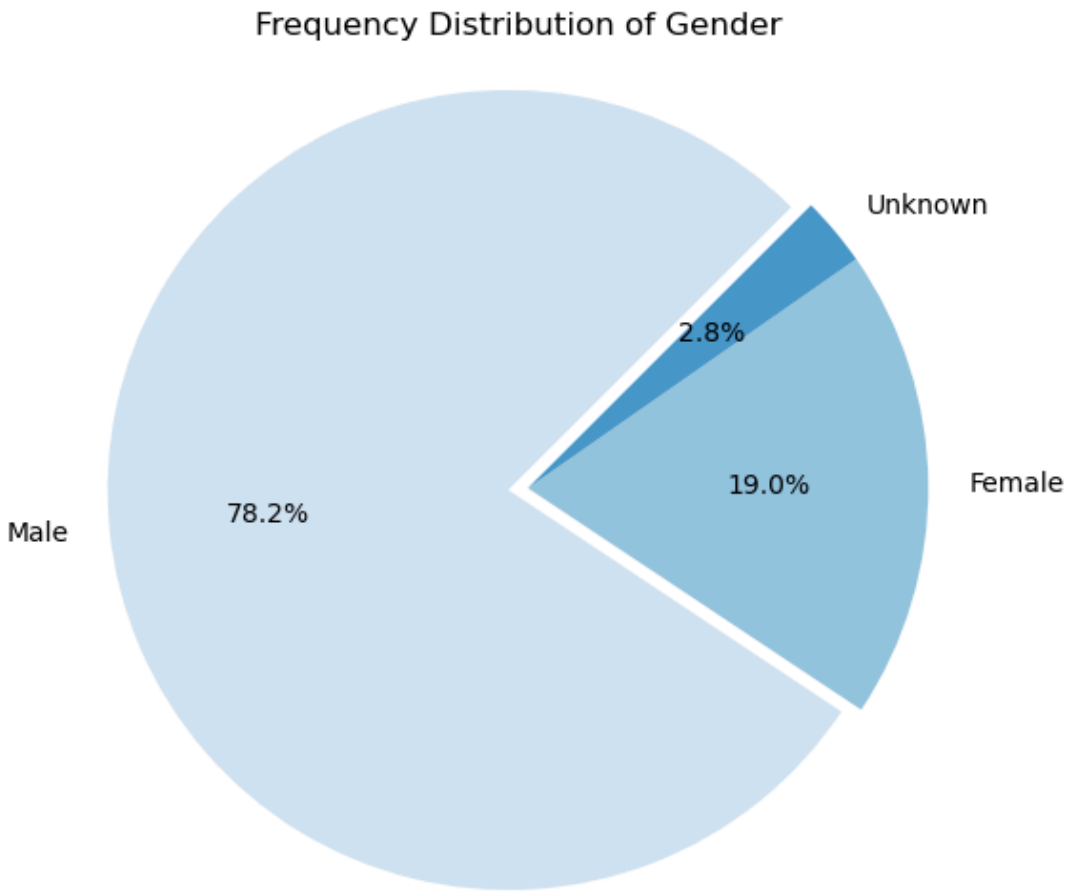
10.3.1 **Frequency Distribution of Gender**

In [61]:
```
gender_counts = df['Gender'].value_counts()

plt.figure(figsize=(8, 6))
colors = plt.cm.Blues([0.2, 0.4, 0.6, 0.8])
explode = [0.05 if i == gender_counts.max() else 0 for i in gender_counts]

plt.pie(
    gender_counts,
    labels=gender_counts.index,
    autopct='%1.1f%%',
    startangle=45,
    colors=colors,
    explode=explode,
    textprops={'color': 'black', 'fontsize': 10}
)

plt.title('Frequency Distribution of Gender', fontsize=12)
plt.axis('equal')
plt.show()
```



Insights from the plot:

Frequency Distribution of Gender

- **Male Dominance:** The majority of individuals in the dataset are male.
- **Female Representation:** There is a significant number of females, though less than males.
- **Unknown Gender:** A small portion of the individuals have an unknown gender.

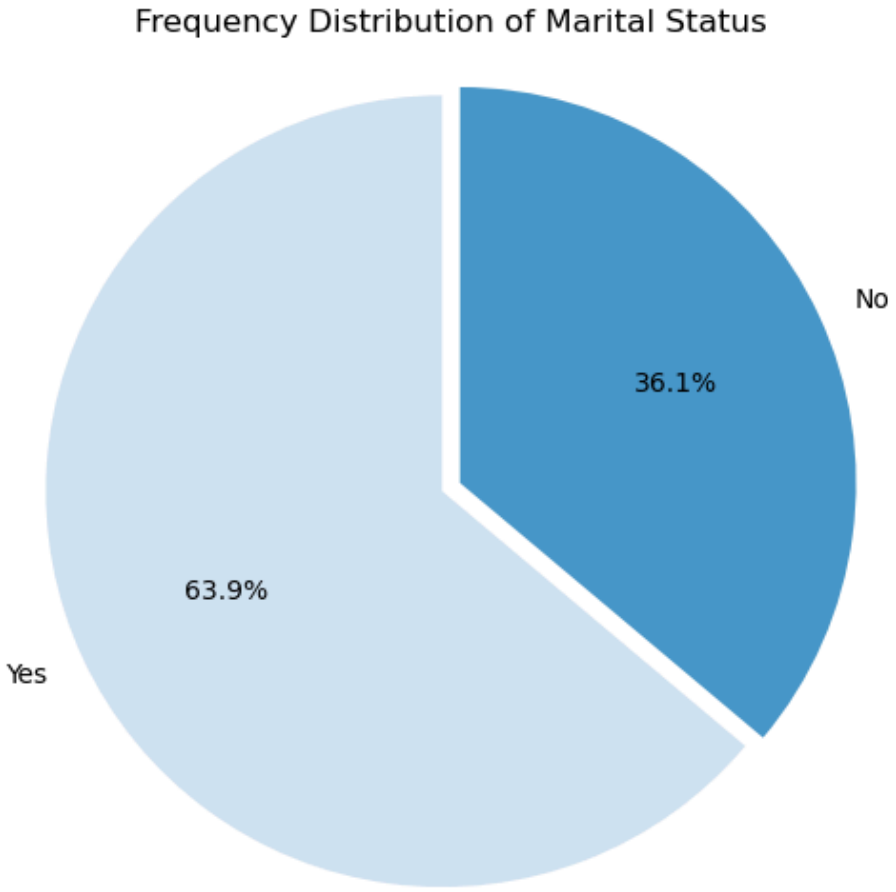
Recommendations

- **Targeted Marketing:** Consider tailoring marketing strategies to the specific needs and preferences of male and female customers.
- **Data Quality:** Investigate the reasons for the "Unknown" gender category and take steps to improve data quality.
- **Diversity and Inclusion:** Promote diversity and inclusion initiatives to ensure that the needs of all genders are considered.

10.3.2 **Frequency Distribution of Married Status**

In [111…

```
married_counts = df['Married'].value_counts()
plt.figure(figsize=(8, 6))
colors = plt.cm.Blues([0.2, 0.6])
explode = [0.05 if i == married_counts.max() else 0 for i in married_counts]
plt.pie(
    married_counts,
    labels=married_counts.index,
    autopct='%1.1f%%',
    startangle=90,
    colors=colors,
    explode=explode,
    textprops={'fontsize': 10, 'color': 'black'}
)
plt.title('Frequency Distribution of Marital Status', fontsize=12)
plt.axis('equal')
plt.show()
```



Frequency Distribution of Marital Status

- **Married Applicants**:
  - Total of **228 married applicants**.
- **Unmarried Applicants**:
  - Total of **129 unmarried applicants**.

Key Insights:

- Higher proportion of married applicants overall, indicating potential family-oriented financial needs.

This distribution could be useful for tailoring loan products to married individuals.

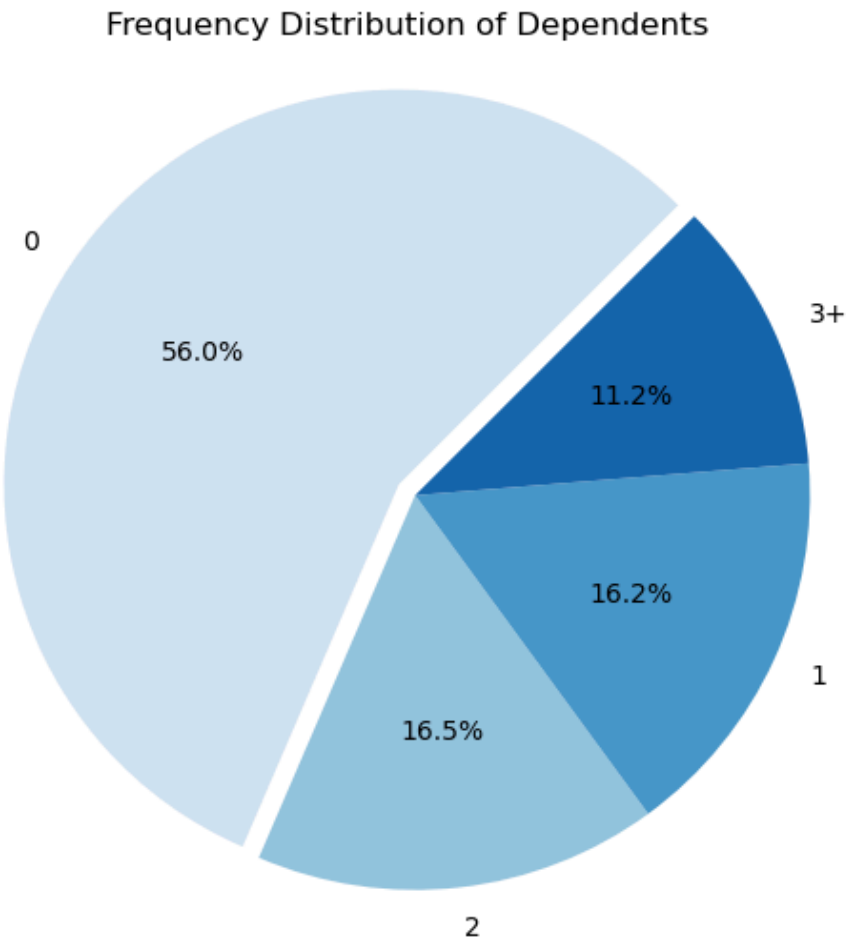
10.3.3 **Frequency Distribution of Dependents**

In [103…

```
dependents_counts = df['Dependents'].value_counts()
plt.figure(figsize=(8, 6))
colors = plt.cm.Blues([0.2, 0.4, 0.6, 0.8])
explode = [0.05 if i == dependents_counts.max() else 0 for i in dependents_c

plt.pie(
    dependents_counts,
    labels=dependents_counts.index,
    autopct='%1.1f%%',
    startangle=45,
    colors=colors,
    explode=explode,
    textprops={'fontsize': 10, 'color': 'black'}
)

plt.title('Frequency Distribution of Dependents', fontsize=12)
plt.axis('equal')
plt.show()
```



Frequency Distribution of Dependents

- **Zero Dependents**:
  - Most applicants (200 in total) report having zero dependents.
- **One Dependent**:
  - **58 applicants** have one dependent.
- **Two Dependents**:
  - **59 applicants** have two dependents.
- **Three or More Dependents**:
  - **40 applicants** report having three or more dependents.

Key Insights:

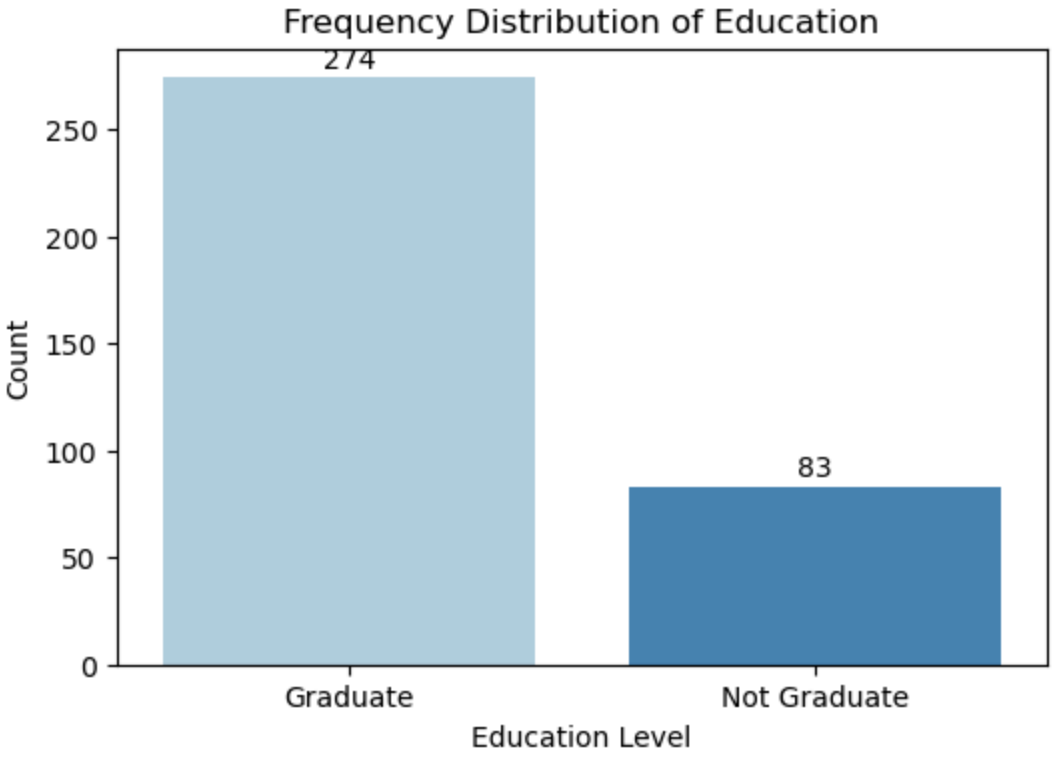
- Majority of applicants have zero dependents, suggesting a potential customer base with fewer family obligations.

10.3.4 **Frequency Distribution of Education**

In [78]:
```
plt.figure(figsize=(6, 4))
sns.countplot(x='Education', hue='Education', data=df, palette="Blues", dodg

education_counts = df['Education'].value_counts()
for index, value in enumerate(education_counts):
    plt.text(index, value + 5, str(value), ha='center', color='black')

plt.title('Frequency Distribution of Education')
plt.xlabel('Education Level')
plt.ylabel('Count')
plt.show()
```



Frequency Distribution of Education

- **Education Levels**:
  - **Graduate**: 274 applicants (approx. 77% of the dataset)
  - **Not Graduate**: 83 applicants (approx. 23% of the dataset)
- **Key Insights**:
  - Majority of loan applicants are graduates, indicating a potential link between education level and likelihood to apply for loans.
  - The predominance of graduates may suggest targeting loan products that cater to their financial stability and awareness.
- **Implications**:
  - **Marketing Focus**: Focus outreach efforts on graduate applicants, who make up the majority.

- **Product Development**: Consider financial products that align with the needs and profiles of a well-educated customer base.d customer base.

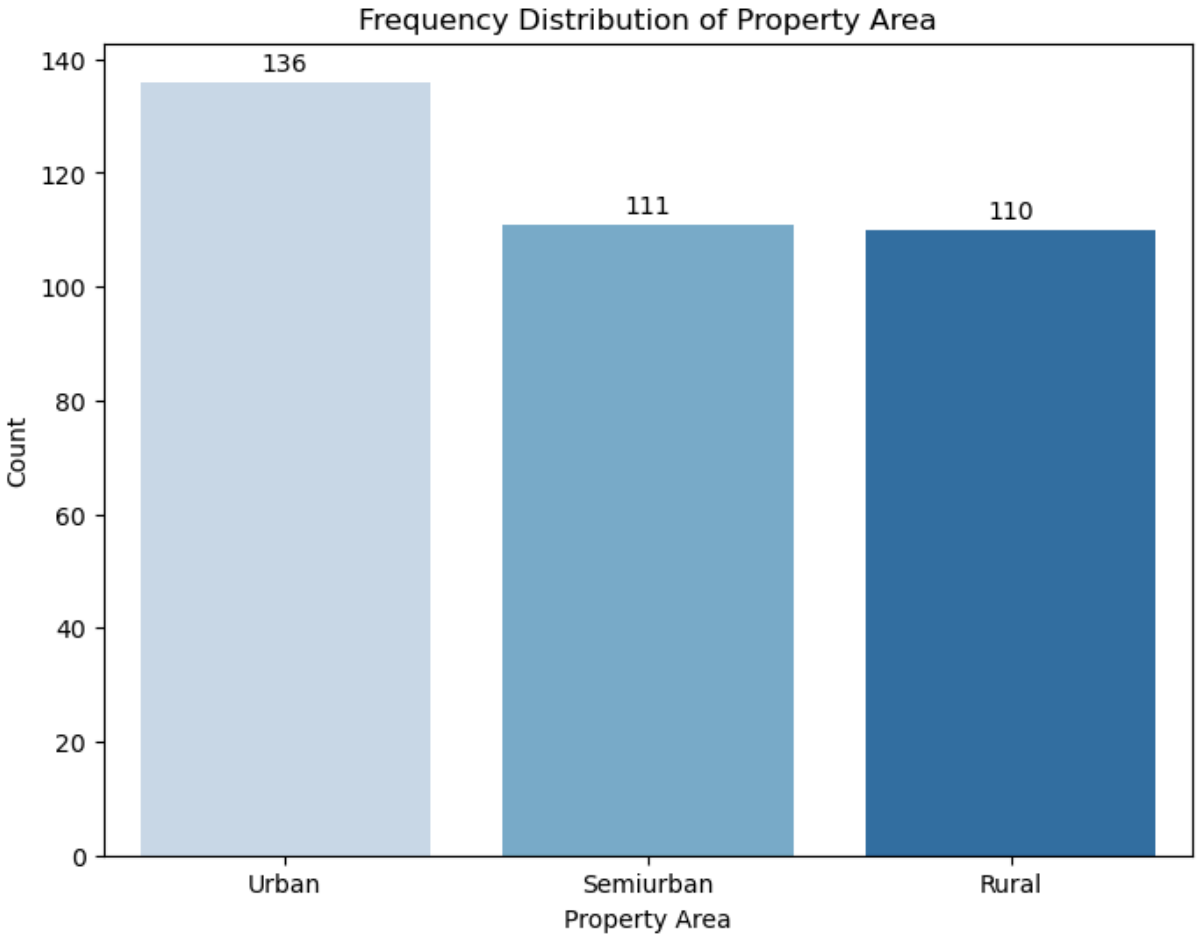
10.3.5 **Frequency Distribution of Property Area**

In [81]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='Property_Area', hue='Property_Area', data=df, palette="Blue

property_area_counts = df['Property_Area'].value_counts().reset_index()
property_area_counts.columns = ['Property_Area', 'count']

for index, row in property_area_counts.iterrows():
    x = row['Property_Area']
    count = row['count']
    x_loc = list(df['Property_Area'].unique()).index(x)
    plt.text(x_loc, count + 2, str(count), ha='center', color='black')

plt.title('Frequency Distribution of Property Area')
plt.xlabel('Property Area')
plt.ylabel('Count')
plt.show()
```



Frequency Distribution of Property Area

- **Urban Dominance**: The majority of properties are located in Urban areas.

- **Semi-Urban and Rural**: A significant portion of properties are in Semi-Urban areas, followed by Rural areas.
- **Recommendations**
  - **Targeted Marketing**: Consider tailoring marketing strategies for different property areas to optimize outreach and engagement.
  - **Inventory Management**: Optimize inventory management to ensure sufficient stock in high-demand Urban areas.
  - **Data-Driven Decisions**: Continuously analyze property area data to identify emerging trends and adjust strategies accordingly.

11. Bivariate Analysis

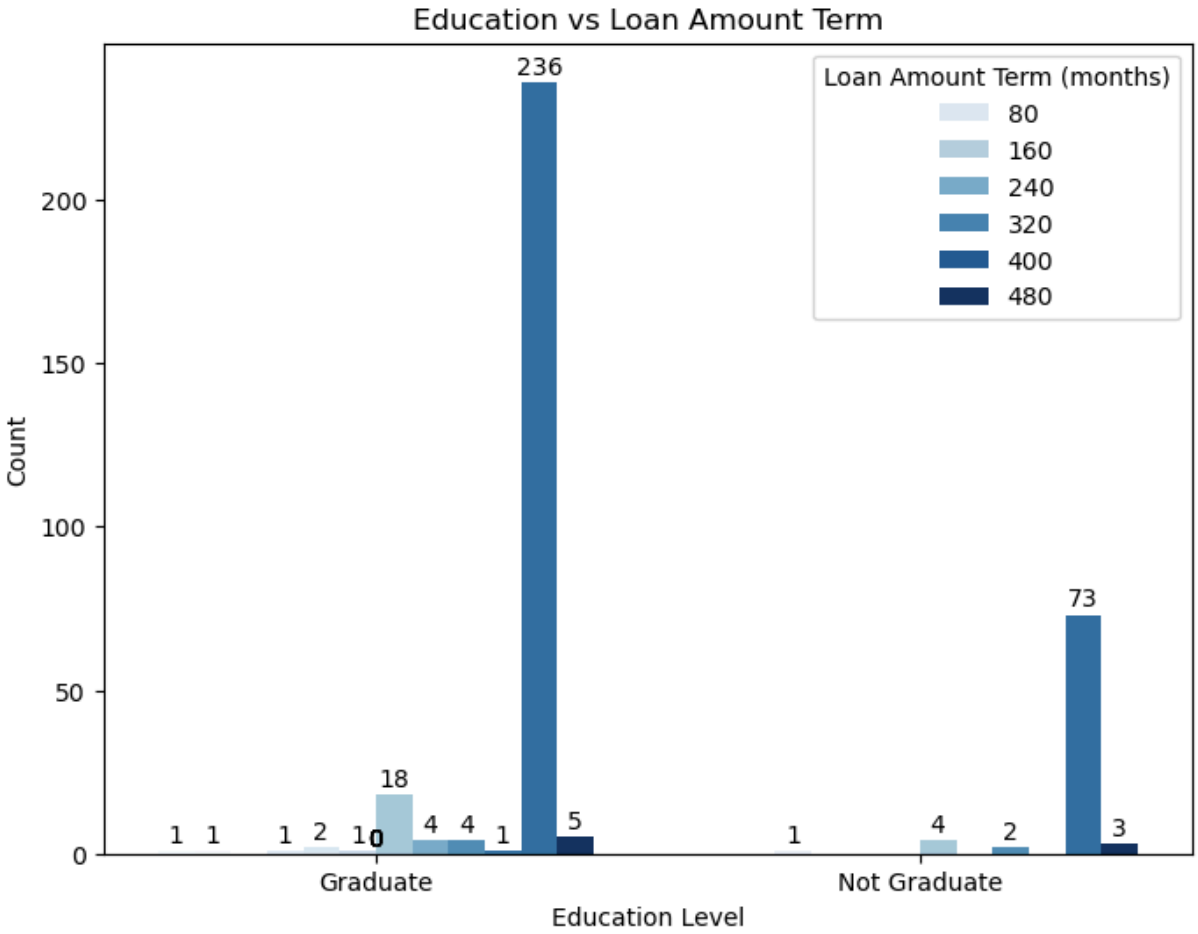
11.1 Education vs Loan Amount Term

In [118…

```
plt.figure(figsize=(8, 6))
sns.countplot(x='Education', hue='Loan_Amount_Term', data=df, palette="Blues

for p in plt.gca().patches:
    plt.gca().annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width()
                       ha='center', va='center', fontsize=10, color='black',
                       textcoords='offset points')

plt.title('Education vs Loan Amount Term')
plt.xlabel('Education Level')
plt.ylabel('Count')
plt.legend(title='Loan Amount Term (months)')
plt.show()
```



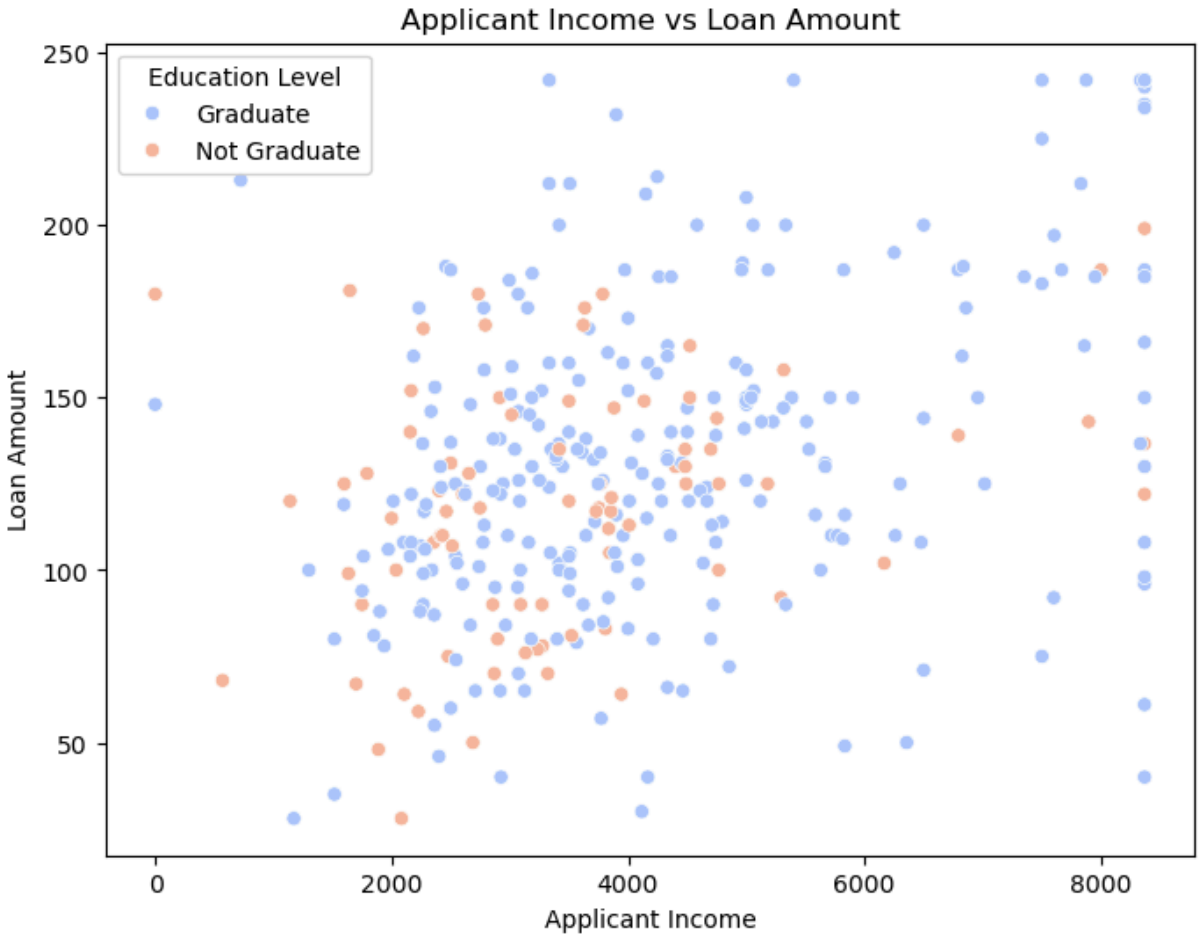
Education vs Loan Amount Term

- **Popular Loan Term**:
  - Majority of applicants, both Graduates (236) and Non-Graduates (73), prefer a **400-month** loan term.
- **Shorter Terms Less Popular**:
  - Few applicants select terms under 240 months, suggesting a preference for extended repayment to reduce monthly payments.
- **Education Influence**:
  - Graduates show a broader range of loan term preferences, including shorter and very long terms.
  - Non-Graduates mostly stick to the 400-month term, with limited variation.
- **Implications**:
  - **Product Design**: Consider tailoring products to the 240-month term, especially for non-graduates.
  - **Risk Profiling**: Graduates' diverse term choices could inform differentiated risk assessment based on education level.

11.2 **Applicant Income vs Loan Amount**

In [116…

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='ApplicantIncome', y='LoanAmount', hue='Education', data=d
plt.title('Applicant Income vs Loan Amount')
plt.xlabel('Applicant Income')
plt.ylabel('Loan Amount')
plt.legend(title='Education Level')
plt.show()
```



Insights from the plot:

Applicant Income vs Loan Amount by Education Level

- **Overall Trend:** There is a positive correlation between applicant income and loan amount, indicating that as income increases, the loan amount tends to increase.
- **Education Level:** Both graduates and non-graduates exhibit similar trends, with higher incomes generally associated with larger loan amounts.
- **Spread:** The data is spread out, suggesting a wide range of loan amounts for different income levels within each education group.

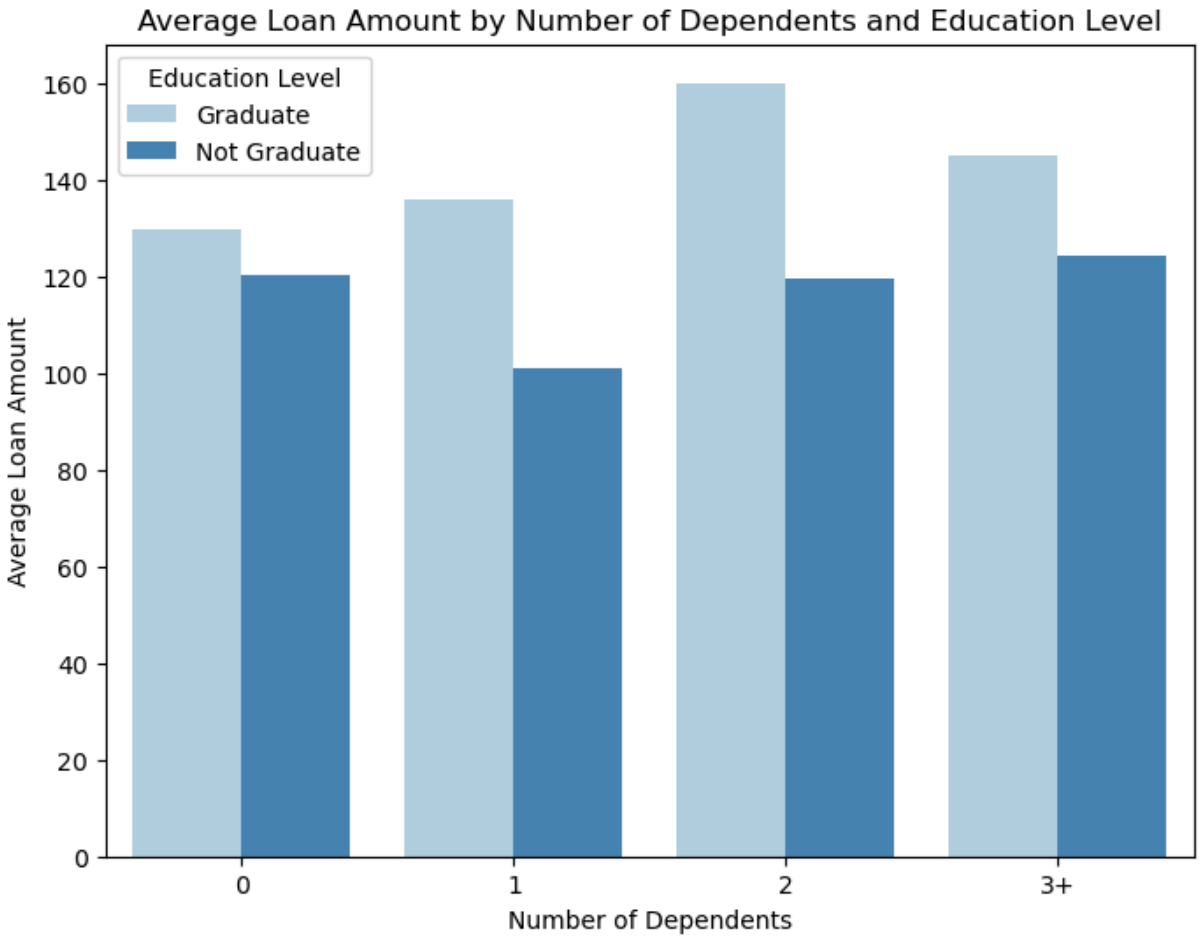
Recommendations

- **Risk Assessment:** Consider income as a key factor in assessing loan risk. Higher income applicants might be considered for larger loans, regardless of education level.
- **Product Differentiation:** Offer a range of loan products to cater to different income groups, considering both graduates and non-graduates.
- **Data-Driven Decisions:** Continuously analyze the relationship between income, loan amount, and education level to refine underwriting and pricing strategies.

11.3 Average Loan Amount by Number of Dependents and Education Level

In [124…

```
plt.figure(figsize=(8, 6))
sns.barplot(x='Dependents', y='LoanAmount', hue='Education', data=df, palett
plt.title('Average Loan Amount by Number of Dependents and Education Level')
plt.xlabel('Number of Dependents')
plt.ylabel('Average Loan Amount')
plt.legend(title='Education Level')
plt.show()
```



Average Loan Amount by Number of Dependents and Education Level

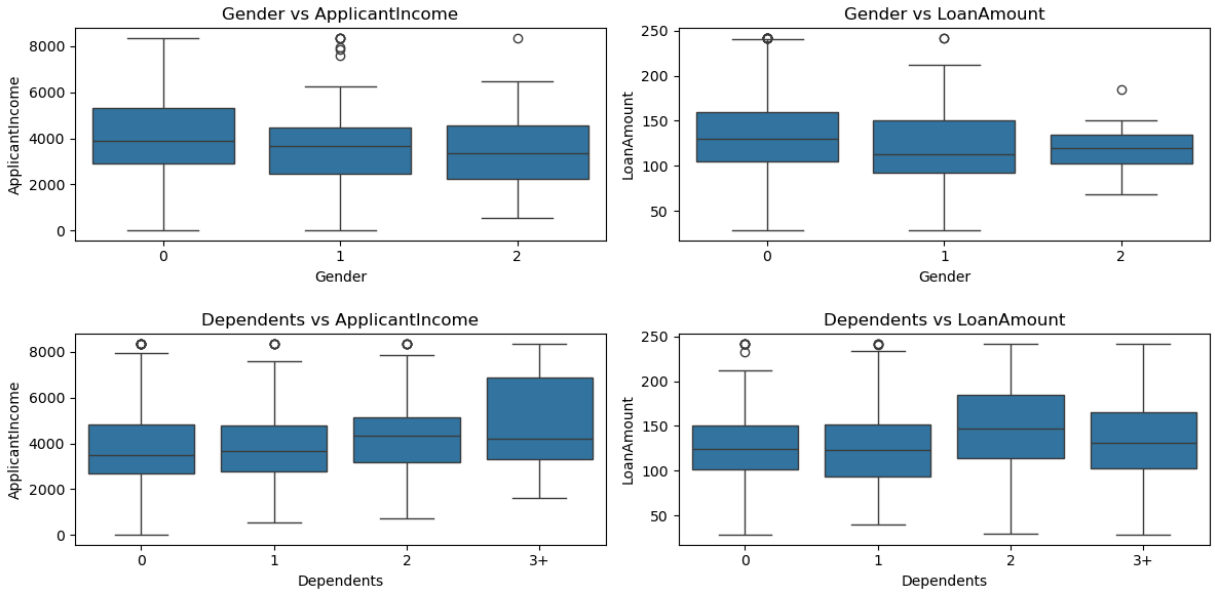
- **Overall Trend:** The average loan amount generally increases with the number of dependents.
- **Education Level:** Graduates tend to have higher average loan amounts compared to non-graduates across all dependent levels.
- **Dependents:** The average loan amount increases significantly with the number of dependents, especially for graduates.
- **Recommendations**
  - **Risk Assessment:** Consider the number of dependents and education level as factors in assessing loan risk.
  - **Product Differentiation:** Offer loan products tailored to different family sizes and education levels.
  - **Data-Driven Decisions:** Continuously analyze the relationship between dependents, education, and loan amount to refine underwriting and pricing strategies.

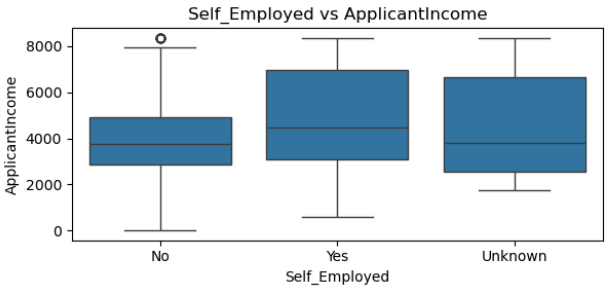
11.4 **Box Plots**

In [157…

```
categorical_var1 = ['Gender', 'Dependents', 'Self_Employed']
numeric_vars = ['ApplicantIncome', 'LoanAmount']

for cat_var in categorical_var1:
    plt.figure(figsize=(12, 3))
    for i, num_var in enumerate(numeric_vars):
        plt.subplot(1, 2, i + 1)
        sns.boxplot(data=data, x=cat_var, y=num_var)
        plt.title(f'{cat_var} vs {num_var}')
    plt.tight_layout()
    plt.show()
```





Insights from the plots:

Box Plot: Gender vs ApplicantIncome

- **Median Income:** The median income is similar across all genders.
- **Spread:** The spread of income is also similar across all genders.
- **Outliers:** There are a few outliers for all genders, indicating some individuals with significantly higher incomes.

Box Plot: Gender vs Loan Amount

- **Median Loan Amount:** The median loan amount is similar across all genders.
- **Spread:** The spread of loan amounts is also similar across all genders.
- **Outliers:** There are a few outliers for all genders, indicating some individuals with significantly higher loan amounts.

Box Plot: Dependents vs ApplicantIncome

- **Median Income:** The median income increases as the number of dependents increases.
- **Spread:** The spread of income is similar across all dependent categories.
- **Outliers:** There are outliers present in all categories, indicating some individuals with significantly higher incomes.

Box Plot: Dependents vs Loan Amount

- **Median Loan Amount:** The median loan amount increases as the number of dependents increases.
- **Spread:** The spread of loan amounts increases with the number of dependents, indicating more variation in loan amounts for families with more dependents.
- **Outliers:** There are outliers present in all categories, indicating some individuals with significantly higher loan amounts, especially for families with fewer dependents.

Box Plot: Self_Employed vs ApplicantIncome

- **Median Income:** The median income is similar across all employment categories.
- **Spread:** The spread of income is also similar across all categories.
- **Outliers:** There are outliers present in all categories, indicating some individuals with significantly higher incomes.

Box Plot: Self_Employed vs Loan Amount

- **Median Loan Amount:** The median loan amount is similar across all employment categories.
- **Spread:** The spread of loan amounts is also similar across all categories.
- **Outliers:** There are outliers present in all categories, indicating some individuals with significantly higher loan amounts, especially for self-employed individuals.

Recommendations

- **Data-Driven Decisions:** Continuously analyze the data to identify trends and patterns. Use data-driven insights to refine underwriting and pricing strategies.
- **Risk Assessment:** Develop a robust risk assessment framework that considers multiple factors, including income, employment status, dependents, and credit history.
- **Product Differentiation:** Offer a range of loan products to cater to different customer segments, based on their needs and preferences.
- **Customer Segmentation:** Segment customers based on various factors, such as income, occupation, and demographics, to tailor marketing and sales strategies.
- **Financial Education:** Provide financial education to customers to help them make informed decisions about borrowing and repayment.

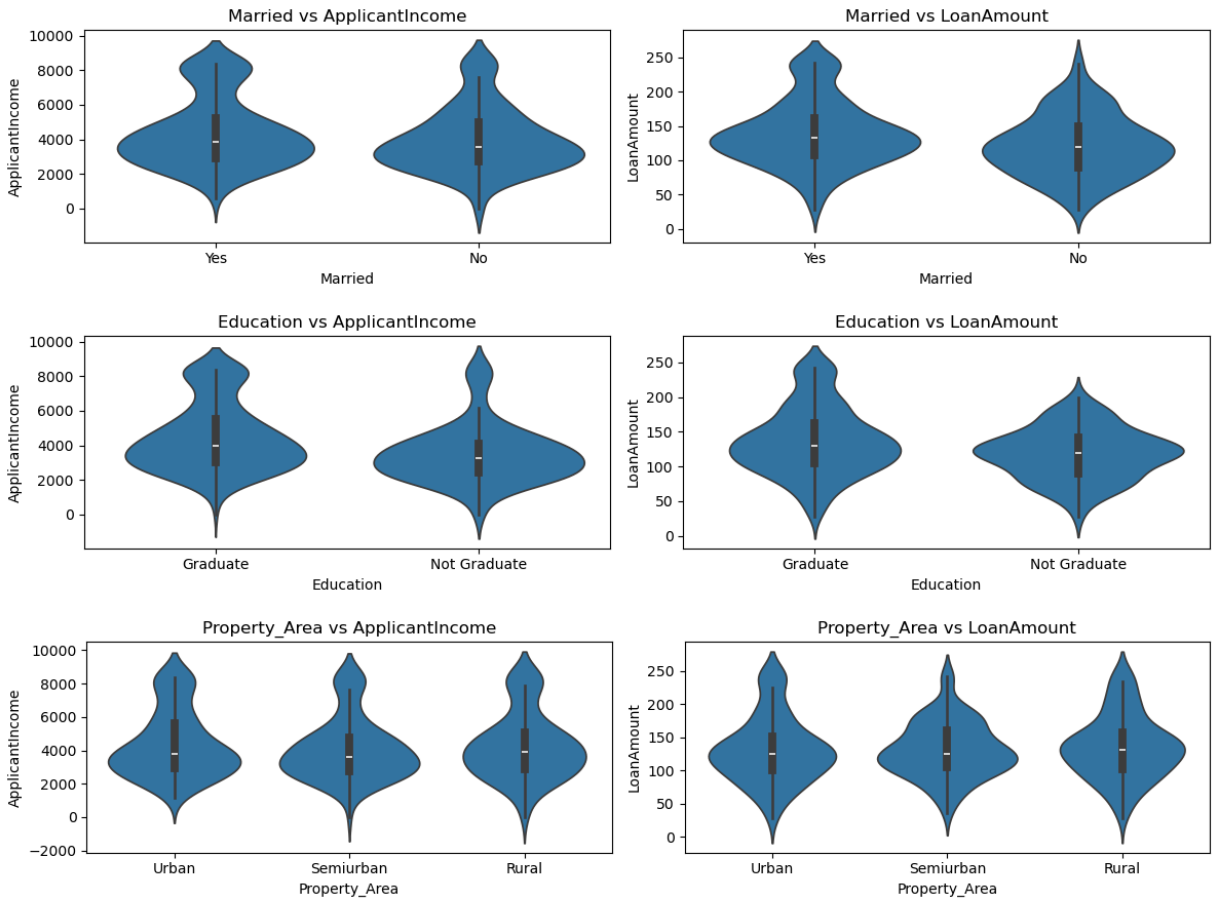
11.5 **Violin Plots**

In [153…

```
categorical_var2 = ['Married', 'Education', 'Property_Area']
numeric_vars = ['ApplicantIncome', 'LoanAmount']

for cat_var in categorical_var2:
    plt.figure(figsize=(12, 3))
    for i, num_var in enumerate(numeric_vars):
        plt.subplot(1, 2, i + 1)
        sns.violinplot(data=data, x=cat_var, y=num_var)
        plt.title(f'{cat_var} vs {num_var}')
```

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



Insights from the plots:

Violin Plot: Married vs Applicant Income

- **Similar Distribution:** The distribution of applicant income appears to be quite similar for both married and unmarried individuals.
- **Median Income:** The median income for married individuals seems slightly higher than that of unmarried individuals.
- **Spread:** The spread of income is similar for both groups, indicating a similar range of incomes.

Violin Plot: Married vs Loan Amount

- **Similar Distribution:** The distribution of loan amounts appears to be quite similar for both married and unmarried individuals.
- **Median Loan Amount:** The median loan amount is slightly higher for married individuals.
- **Spread:** The spread of loan amounts is similar for both groups, indicating a similar range of loan amounts.

Violin Plot: Education vs Applicant Income

- **Similar Distribution:** The distribution of applicant income appears to be quite similar for both graduates and non-graduates.
- **Median Income:** The median income for graduates seems slightly higher than that of non-graduates.
- **Spread:** The spread of income is similar for both groups, indicating a similar range of incomes.

Violin Plot: Education vs Loan Amount

- **Similar Distribution:** The distribution of loan amounts appears to be quite similar for both graduates and non-graduates.
- **Median Loan Amount:** The median loan amount is slightly higher for graduates.
- **Spread:** The spread of loan amounts is similar for both groups, indicating a similar range of loan amounts.

Violin Plot: Property Area vs Applicant Income

- **Similar Distribution:** The distribution of applicant income appears to be quite similar across all property areas (Urban, Semiurban, Rural).
- **Median Income:** The median income seems to be slightly higher for urban areas.
- **Spread:** The spread of income is similar for all groups, indicating a similar range of incomes.

Violin Plot: Property Area vs Loan Amount

- **Similar Distribution:** The distribution of loan amounts appears to be quite similar across all property areas (Urban, Semiurban, Rural).
- **Median Loan Amount:** The median loan amount seems to be slightly higher for urban areas.
- **Spread:** The spread of loan amounts is similar for all groups, indicating a similar range of loan amounts.

Recommendations

- **Income-Based Lending:** Continue to prioritize income as a key factor in assessing loan eligibility.
- **Data-Driven Decisions:** Continuously analyze the data to identify trends and patterns. Use data-driven insights to refine underwriting and pricing strategies.

- **Equal Opportunity:** Ensure that lending practices are fair and equitable for all customer segments, regardless of marital status, education, or property area.
- **Customer Segmentation:** Consider segmenting customers based on factors like income, occupation, and demographics to tailor marketing and sales strategies.
- **Financial Education:** Provide financial education to customers to help them make informed decisions about borrowing and repayment.

12.Multivariate Analysis

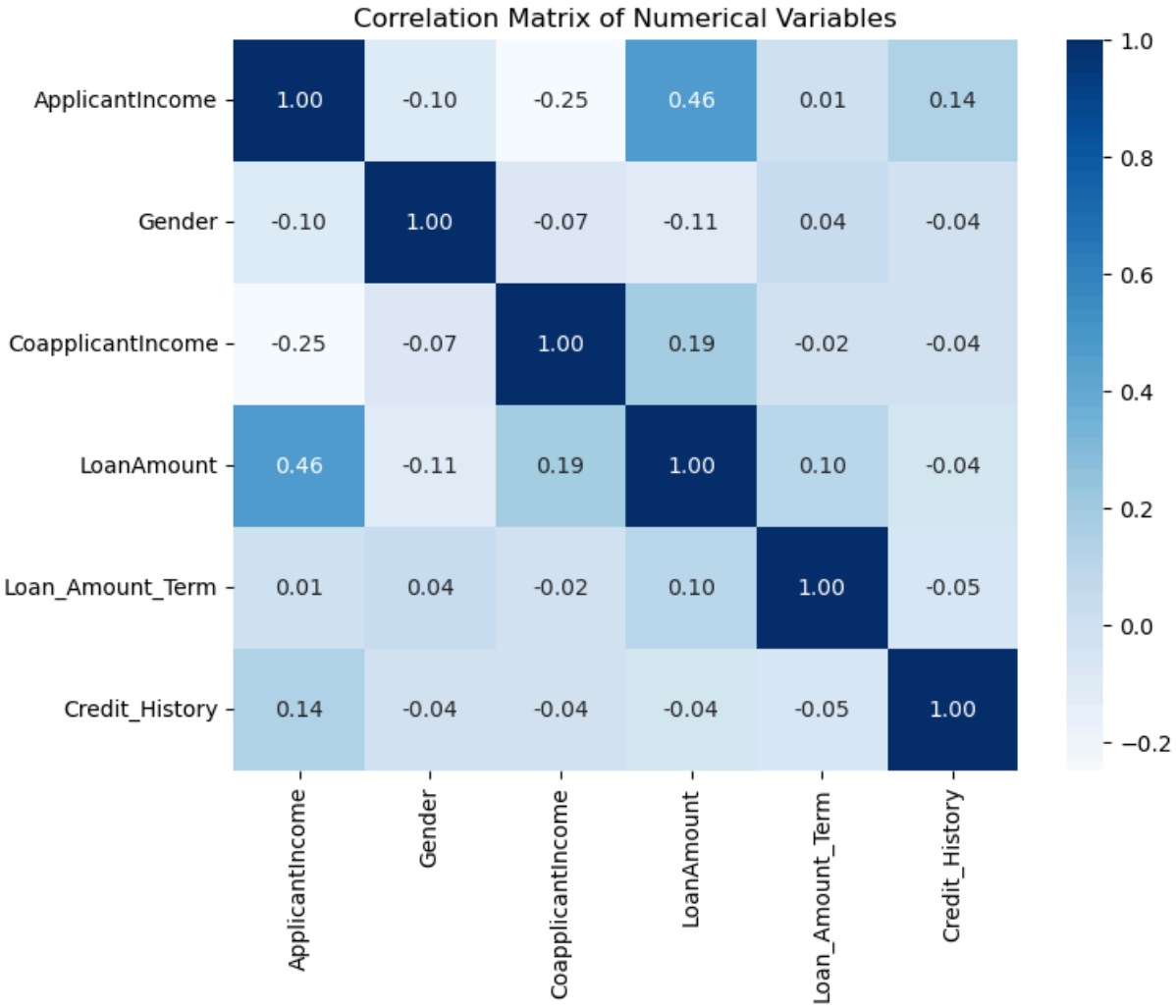
12.1 Correlation Matrix of Numerical Variables

Converting Gender to numerical data from categorical data for correlation

In [62]:
```
df['Gender']=df['Gender'].map({'Male':0, 'Female':1, 'Unknown':2})
df['Gender']
```

Out[62]:
```
0      0
1      0
2      0
3      0
4      0
      ..
362    0
363    0
364    0
365    0
366    0
Name: Gender, Length: 357, dtype: int64
```

In [64]:
```
plt.figure(figsize=(8, 6))
correlation_matrix = df[['ApplicantIncome', 'Gender', 'CoapplicantIncome', '
sns.heatmap(correlation_matrix, annot=True, cmap="Blues", fmt=".2f")
plt.title('Correlation Matrix of Numerical Variables')
plt.show()
```



Insights from the Correlation Matrix

- **ApplicantIncome and LoanAmount**: Moderate positive correlation (0.46), indicating that higher applicant income is associated with larger loan amounts.
- **Credit_History and LoanAmount**: Weak positive correlation (0.14), showing minimal influence of credit history on loan amounts.
- **CoapplicantIncome and LoanAmount**: Weak positive correlation (0.19), indicating a slight relationship between coapplicant income and loan amounts.
- **Loan_Amount_Term**: Nearly uncorrelated with all other variables, reflecting no strong relationships.
- **Gender**: Weak or negligible correlations with all variables, suggesting limited predictive power.
- **Credit_History**: Slightly positive correlation with ApplicantIncome (0.14), but otherwise weak relationships.

Overall, `Applicant Income` shows the strongest correlation with LoanAmount among the variables.

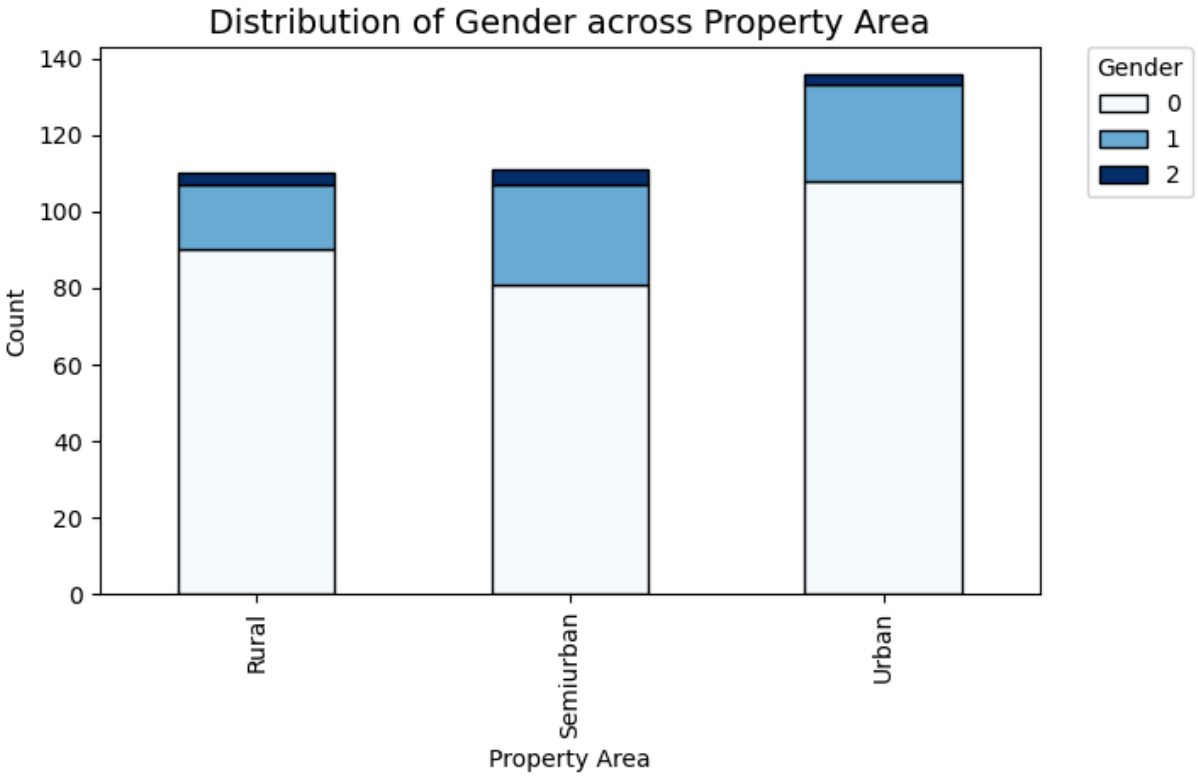
12.2 **Stacked Bar Chart to show the distribution of categorical variables across multiple categories**

12.2.1 Gender Distribution Across Property Area

In [90]:
```
gender_distribution = data.groupby(['Property_Area', 'Gender']).size().unsta

plt.figure(figsize=(8, 6))
ax = gender_distribution.plot(kind='bar', stacked=True, colormap='Blues', ec
plt.title("Distribution of Gender across Property Area", fontsize=14)
plt.ylabel("Count")
plt.xlabel("Property Area")
plt.legend(title="Gender (0=Female, 1=Male, 2=Unknown)")
plt.tight_layout()
ax.legend(title="Gender", bbox_to_anchor=(1.05, 1), loc='upper left', border
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



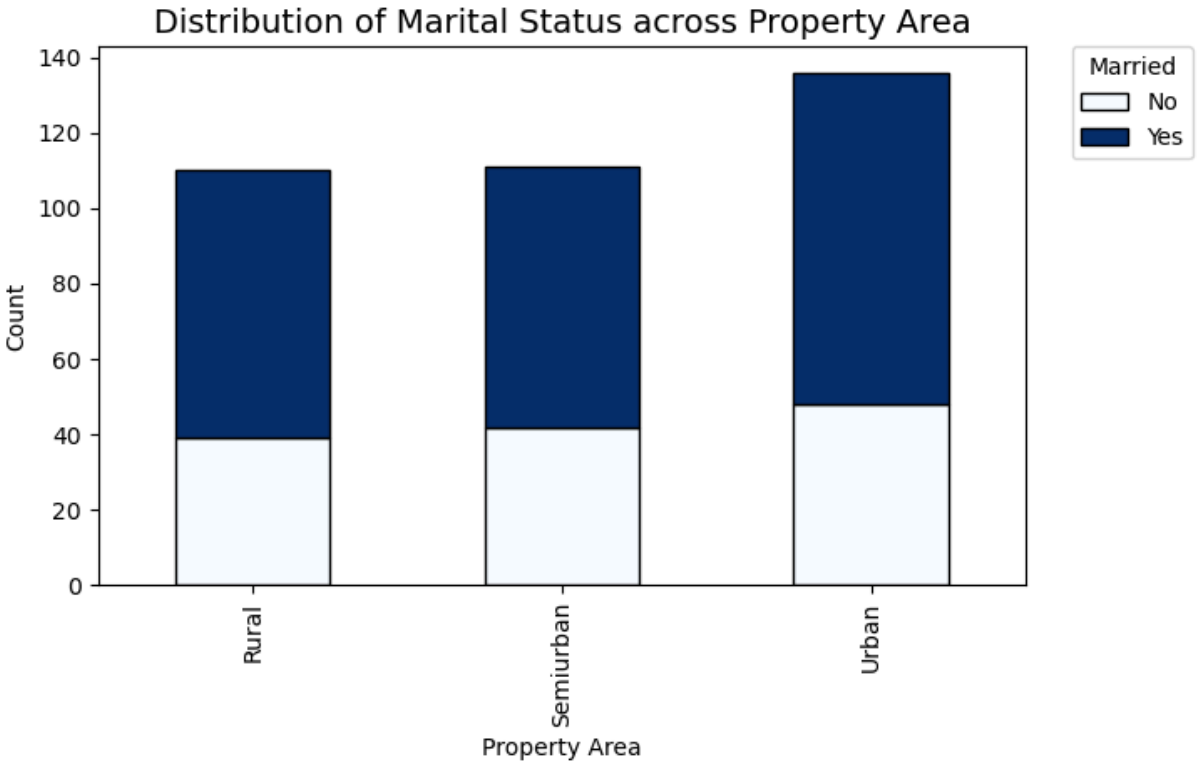
Insights

- **Urban**: Male applicants dominate (**101 males**, **36 females**).
- **Semiurban**: More males (**112 males**) compared to females (**54 females**).
- **Rural**: Majority are males (**85 males**, **23 females**).

12.2.2 Marital Status Distribution Across Property Area

In [94]:
```
marital_distribution = data.groupby(['Property_Area', 'Married']).size().uns
plt.figure(figsize=(8, 6))
ax=marital_distribution.plot(kind='bar', stacked=True, colormap='Blues', edg
plt.title("Distribution of Marital Status across Property Area", fontsize=14
plt.ylabel("Count")
plt.xlabel("Property Area")
plt.legend(title="Married")
plt.tight_layout()
ax.legend(title="Married", bbox_to_anchor=(1.05, 1), loc='upper left', borde
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



Insights

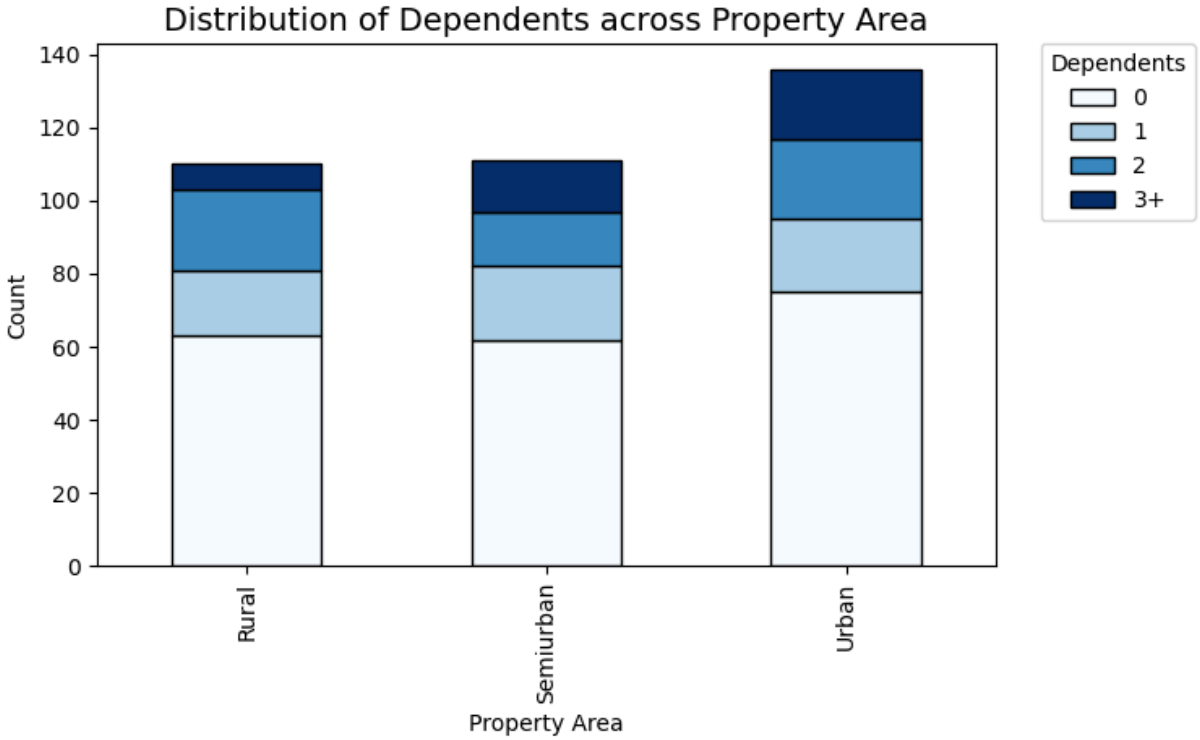
- **Urban**: Most applicants are married (**102 married**, **35 unmarried**).
- **Semiurban**: Higher count of married applicants (**121 married**, **45 unmarried**).
- **Rural**: Married applicants dominate (**82 married**, **26 unmarried**).

12.2.3 Dependents Distribution Across Property Area

In [98]:
```
dependents_distribution = data.groupby(['Property_Area', 'Dependents']).size
plt.figure(figsize=(8, 6))
ax=dependents_distribution.plot(kind='bar', stacked=True, colormap='Blues',
plt.title("Distribution of Dependents across Property Area", fontsize=14)
plt.ylabel("Count")
plt.xlabel("Property Area")
plt.legend(title="Dependents")
plt.tight_layout()
```

```
ax.legend(title="Dependents", bbox_to_anchor=(1.05, 1), loc='upper left', bc
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



Insights

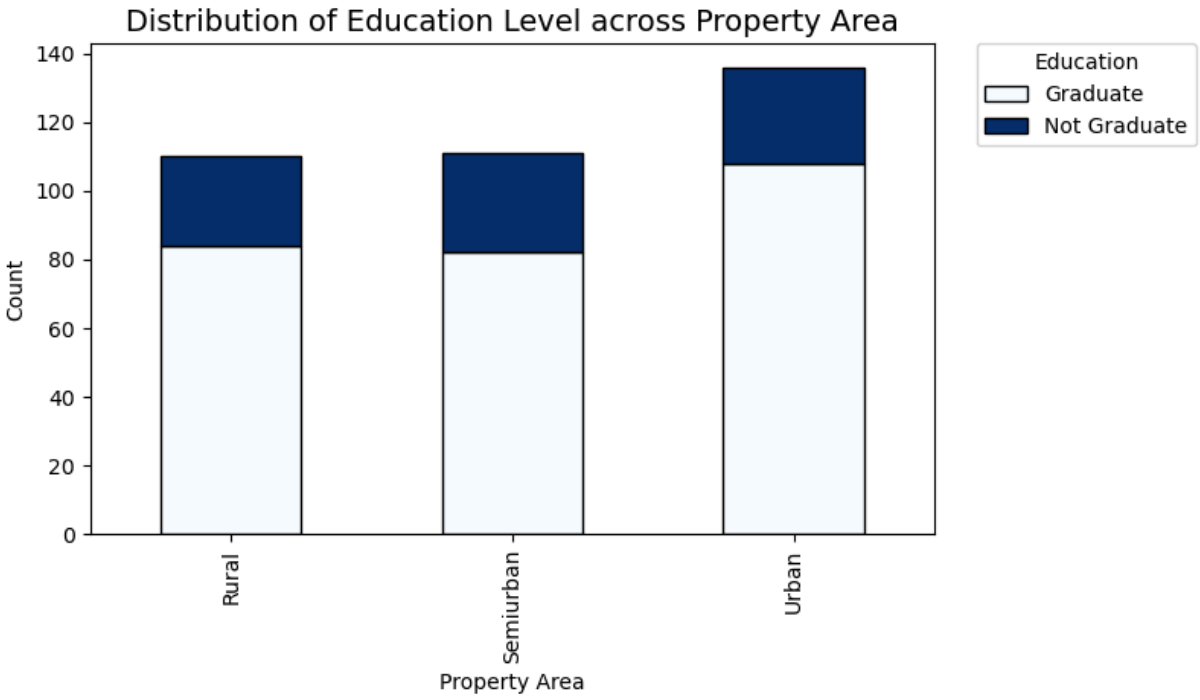
- **Urban**: Most have no dependents (**90 no dependents**, **10 with 3+ dependents**).
- **Semiurban**: Predominantly no dependents (**97 no dependents**, **18 with 1 dependent**).
- **Rural**: Majority have no dependents (**78 no dependents**, **13 with 2 dependents**).

12.2.4 Education Level Distribution Across Property Area.

In [100…

```
education_distribution = data.groupby(['Property_Area', 'Education']).size()
plt.figure(figsize=(8, 6))
ax=education_distribution.plot(kind='bar', stacked=True, colormap='Blues', e
plt.title("Distribution of Education Level across Property Area", fontsize=1
plt.ylabel("Count")
plt.xlabel("Property Area")
plt.legend(title="Education")
plt.tight_layout()
ax.legend(title="Education", bbox_to_anchor=(1.05, 1), loc='upper left', bor
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



Insights

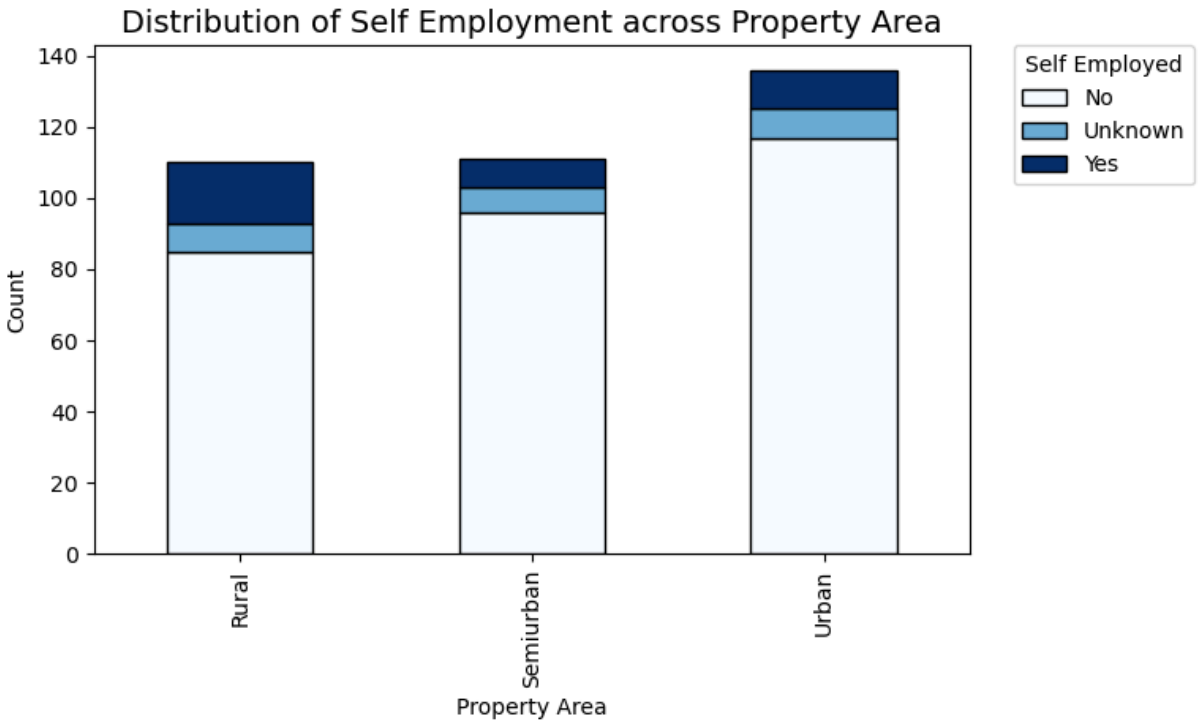
- **Urban**: Majority are graduates (**116 graduates**, **21 non-graduates**).
- **Semiurban**: Graduates dominate (**138 graduates**, **28 non-graduates**).
- **Rural**: Graduates lead (**85 graduates**, **23 non-graduates**).

12.2.5 Self Employment Distribution Across Property Area

In [104…

```
self_employed_distribution = data.groupby(['Property_Area', 'Self_Employed']
plt.figure(figsize=(8, 6))
ax=self_employed_distribution.plot(kind='bar', stacked=True, colormap='Blues
plt.title("Distribution of Self Employment across Property Area", fontsize=1
plt.ylabel("Count")
plt.xlabel("Property Area")
plt.legend(title="Self Employed")
plt.tight_layout()
ax.legend(title="Self Employed", bbox_to_anchor=(1.05, 1), loc='upper left',
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



Insights

- **Urban**: Most applicants are not self-employed (**21 self-employed**, **114 not self-employed**).
- **Semiurban**: Few are self-employed (**28 self-employed**, **126 not self-employed**).
- **Rural**: Similar trend (**20 self-employed**, **88 not self-employed**).

Key Insights

Univariate Analysis

1. **Income Distribution**:
    - Applicant income is highly right-skewed with a significant number of low-income applicants and a few high-income outliers.
    - Most co-applicants have little to no income.
2. **Loan Amount**:
    - Loan amounts are also right-skewed, with most loans falling within a smaller range.
3. **Credit History**:
    - A majority of applicants have a good credit history (coded as "1"), indicating financial responsibility.
4. **Categorical Variables**:

- Gender: Majority of applicants are male.
- Marital Status: Most applicants are married.
- Dependents: Most applicants report zero dependents.
- Education: Approximately 77% of applicants are graduates.
- Property Area: Urban areas dominate, followed by semi-urban areas.

Bivariate Analysis

1. **Income vs Loan Amount**:
   - A positive correlation exists between applicant income and loan amount, with higher income applicants eligible for larger loans.
   - Similar trends are observed for graduates and non-graduates.
2. **Education vs Loan Term**:
   - Graduates prefer a variety of loan terms, while non-graduates mostly opt for 30-year terms.
3. **Dependents vs Loan Amount**:
   - Average loan amount increases with the number of dependents, especially for graduates.

Multivariate Analysis

1. **Correlation Matrix**:
   - Applicant income shows the strongest positive correlation with loan amounts.
   - Loan term is weakly correlated with other variables, suggesting limited influence.
2. **Categorical Variable Distribution**:
   - Urban areas have a higher concentration of graduates and married applicants.
   - Male applicants dominate across all property areas.

Conclusion

1. **Applicant Income**:
   - Plays a critical role in determining loan amounts, indicating the need for income-based risk assessment.
2. **Credit History**:

- A good credit history significantly impacts loan approvals, emphasizing the importance of maintaining a strong financial record.

3. **Loan Products**:
    - Tailored loan products targeting different demographics (graduates, income levels, dependents) can improve customer satisfaction and accessibility.
4. **Data Quality and Inclusivity**:
    - Address missing data and promote financial inclusion for underrepresented groups like female applicants and rural residents.

Recommendations:

- Offer flexible loan terms to cater to diverse customer needs.
- Implement targeted marketing strategies based on education level, property area, and marital status.
- Develop financial education programs to improve credit history awareness and eligibility rates.
