

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

```
In [2]: os.chdir(r"C:\Users\Lenovo\OneDrive\Desktop\cardataset")
```

```
In [4]: df=pd.read_csv("Data_Analyst_Assignment_Dataset.csv")
```

```
In [5]: df
```

Out[5]:

|       | Amount Pending | State          | Tenure | Interest Rate | City      | Bounce String | Disbursed Amount | Loan Number |
|-------|----------------|----------------|--------|---------------|-----------|---------------|------------------|-------------|
| 0     | 963            | Karnataka      | 11     | 7.69          | Bangalore | SSS           | 10197            | JZ6FS       |
| 1     | 1194           | Karnataka      | 11     | 6.16          | Bangalore | SSB           | 12738            | RDIOY       |
| 2     | 1807           | Karnataka      | 14     | 4.24          | Hassan    | BBS           | 24640            | WNW4L       |
| 3     | 2451           | Karnataka      | 10     | 4.70          | Bangalore | SSS           | 23990            | 6LBJS       |
| 4     | 2611           | Karnataka      | 10     | 4.41          | Mysore    | SSB           | 25590            | ZFZUA       |
| ...   | ...            | ...            | ...    | ...           | ...       | ...           | ...              | ...         |
| 24577 | 899            | Andhra Pradesh | 8      | 0.00          | Chittoor  | FEMI          | 7192             | EAX5C       |
| 24578 | 2699           | Andhra Pradesh | 8      | 0.00          | Krishna   | FEMI          | 21592            | 5MCE9       |
| 24579 | 1540           | Andhra Pradesh | 8      | 0.00          | Krishna   | FEMI          | 12320            | 9HO4Q       |
| 24580 | 824            | Andhra Pradesh | 8      | 0.00          | Guntur    | FEMI          | 6592             | 3VV72       |
| 24581 | 2254           | Andhra Pradesh | 11     | 0.00          | Kurnool   | FEMI          | 24794            | 18XBC       |

24582 rows × 8 columns

```
In [6]: df.shape
```

```
Out[6]: (24582, 8)
```

```
In [8]: df.info
```

```
Out[8]: <bound method DataFrame.info of
      st Rate          City \
      0        963    Karnataka    11      7.69  Bangalore
      1       1194    Karnataka    11      6.16  Bangalore
      2       1807    Karnataka    14      4.24   Hassan
      3       2451    Karnataka    10      4.70  Bangalore
      4       2611    Karnataka    10      4.41   Mysore
      ...
      ...
      24577     899 Andhra Pradesh    8      0.00  Chittoor
      24578     2699 Andhra Pradesh    8      0.00  Krishna
      24579     1540 Andhra Pradesh    8      0.00  Krishna
      24580     824  Andhra Pradesh    8      0.00  Guntur
      24581     2254 Andhra Pradesh   11      0.00  Kurnool

      Bounce String Disbursed Amount Loan Number
      0        SSS      10197   JZ6FS
      1        SSB      12738   RDIOY
      2        BBS      24640   WNW4L
      3        SSS      23990   6LBJS
      4        SSB      25590   ZFZUA
      ...
      ...
      24577    FEMI      7192   EAX5C
      24578    FEMI      21592   5MCE9
      24579    FEMI      12320   9H04Q
      24580    FEMI      6592    3VV72
      24581    FEMI      24794   18XBC

[24582 rows x 8 columns]>
```

In [10]: df.dtypes

```
Out[10]: Amount Pending      int64
      State           object
      Tenure          int64
      Interest Rate  float64
      City            object
      Bounce           String
      Disbursed Amount  int64
      Loan Number     object
      dtype: object
```

In [11]: df.describe()

|              | Amount Pending | Tenure       | Interest Rate | Disbursed Amount |
|--------------|----------------|--------------|---------------|------------------|
| <b>count</b> | 24582.000000   | 24582.000000 | 24582.000000  | 24582.000000     |
| <b>mean</b>  | 1791.172687    | 9.415263     | 0.934960      | 17705.195468     |
| <b>std</b>   | 937.565507     | 3.238904     | 3.114732      | 14192.671509     |
| <b>min</b>   | 423.000000     | 7.000000     | 0.000000      | 2793.000000      |
| <b>25%</b>   | 1199.000000    | 8.000000     | 0.000000      | 9857.750000      |
| <b>50%</b>   | 1593.000000    | 8.000000     | 0.000000      | 13592.000000     |
| <b>75%</b>   | 2083.000000    | 11.000000    | 0.000000      | 19968.000000     |
| <b>max</b>   | 13349.000000   | 24.000000    | 37.920000     | 141072.000000    |

```
In [12]: df.isnull()
```

Out[12]:

|       | Amount Pending | State | Tenure | Interest Rate | City  | Bounce String | Disbursed Amount | Loan Number |
|-------|----------------|-------|--------|---------------|-------|---------------|------------------|-------------|
| 0     | False          | False | False  | False         | False | False         | False            | False       |
| 1     | False          | False | False  | False         | False | False         | False            | False       |
| 2     | False          | False | False  | False         | False | False         | False            | False       |
| 3     | False          | False | False  | False         | False | False         | False            | False       |
| 4     | False          | False | False  | False         | False | False         | False            | False       |
| ...   | ...            | ...   | ...    | ...           | ...   | ...           | ...              | ...         |
| 24577 | False          | False | False  | False         | False | False         | False            | False       |
| 24578 | False          | False | False  | False         | False | False         | False            | False       |
| 24579 | False          | False | False  | False         | False | False         | False            | False       |
| 24580 | False          | False | False  | False         | False | False         | False            | False       |
| 24581 | False          | False | False  | False         | False | False         | False            | False       |

24582 rows × 8 columns

```
In [14]: df['Amount Pending']
```

Out[14]:

```
0      963
1     1194
2     1807
3     2451
4     2611
...
24577    899
24578   2699
24579   1540
24580    824
24581   2254
```

Name: Amount Pending, Length: 24582, dtype: int64

```
In [18]: state_counts = df['State'].value_counts()
```

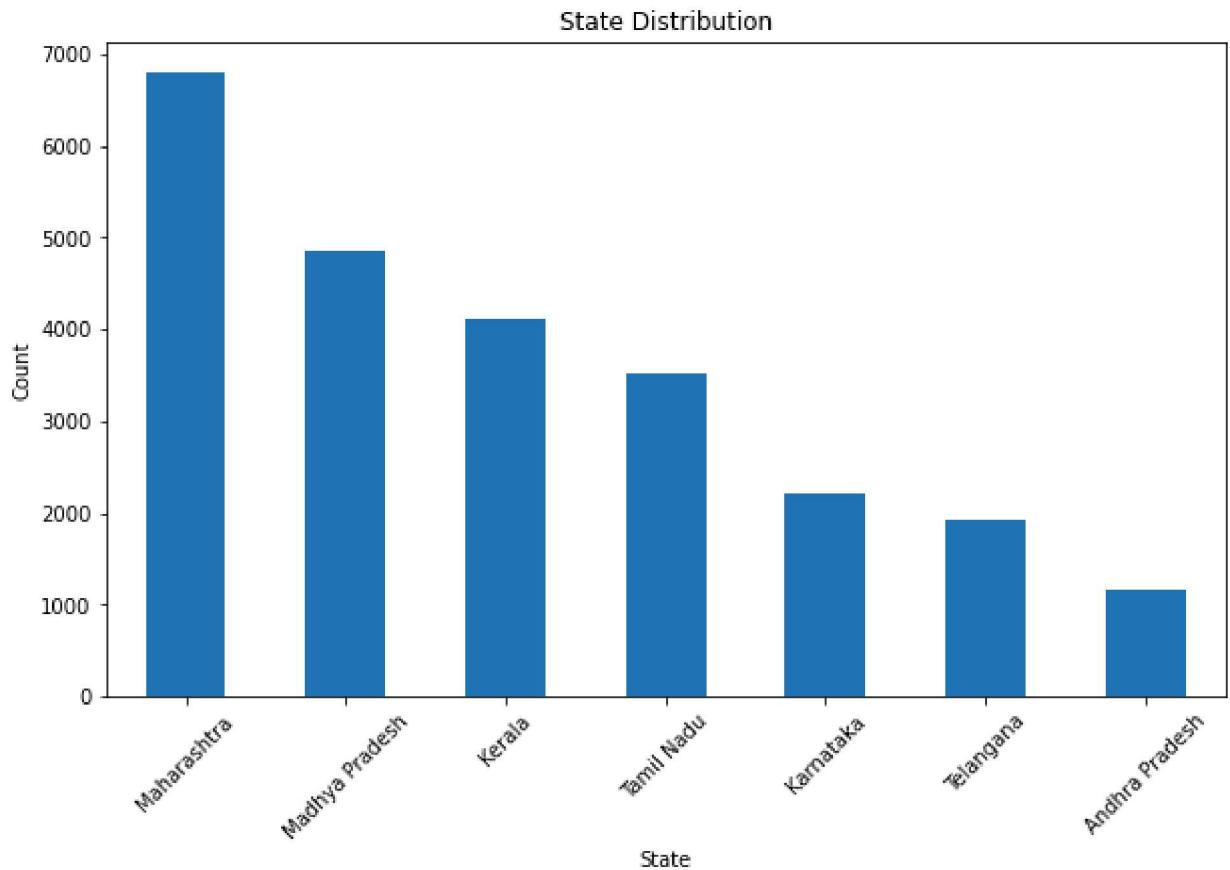
```
print("State Distribution:")
print(state_counts)
```

```
plt.figure(figsize=(10, 6))
state_counts.plot(kind='bar')
plt.title('State Distribution')
plt.xlabel('State')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

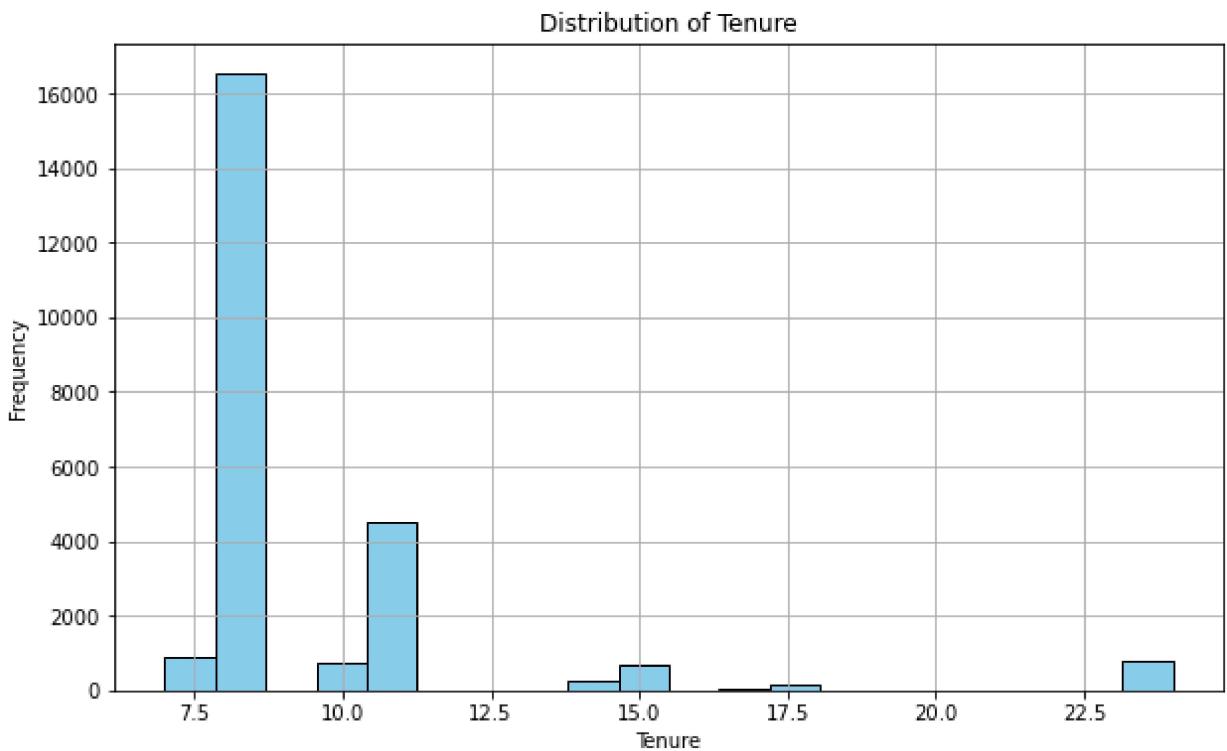
State Distribution:

|                |      |
|----------------|------|
| Maharashtra    | 6793 |
| Madhya Pradesh | 4850 |
| Kerala         | 4116 |
| Tamil Nadu     | 3526 |
| Karnataka      | 2205 |
| Telangana      | 1931 |
| Andhra Pradesh | 1161 |

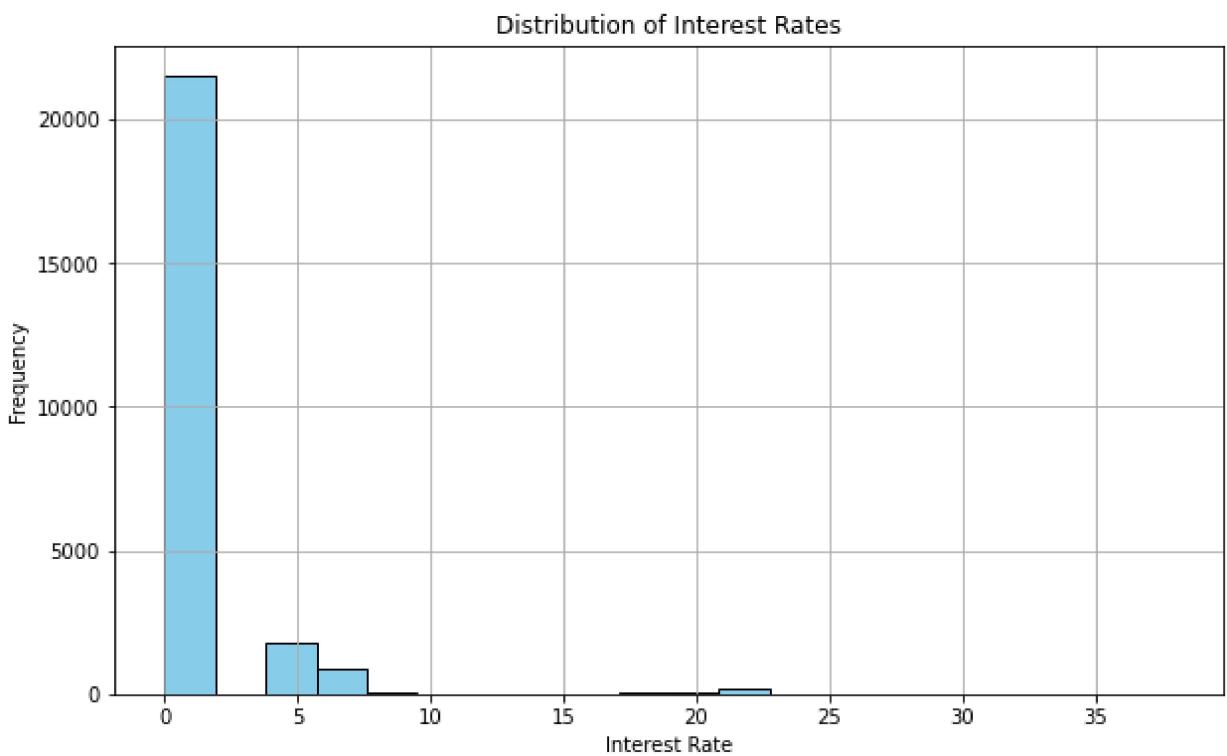
Name: State, dtype: int64



```
In [20]: plt.figure(figsize=(10, 6))
plt.hist(df['Tenure'], bins=20, color='skyblue', edgecolor='black')
plt.title('Distribution of Tenure')
plt.xlabel('Tenure')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



```
In [21]: plt.figure(figsize=(10, 6))
plt.hist(df['Interest Rate'], bins=20, color='skyblue', edgecolor='black')
plt.title('Distribution of Interest Rates')
plt.xlabel('Interest Rate')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



```
In [24]: print("Summary statistics of City:")
print(df['City'].describe())
```

```

city_counts = df['City'].value_counts()

# Display the city distribution
print("\nCity Distribution:")
print(city_counts)

plt.figure(figsize=(12, 6))
city_counts.head(10).plot(kind='bar', color='skyblue')
plt.title('Top 10 City Distribution')
plt.xlabel('City')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()

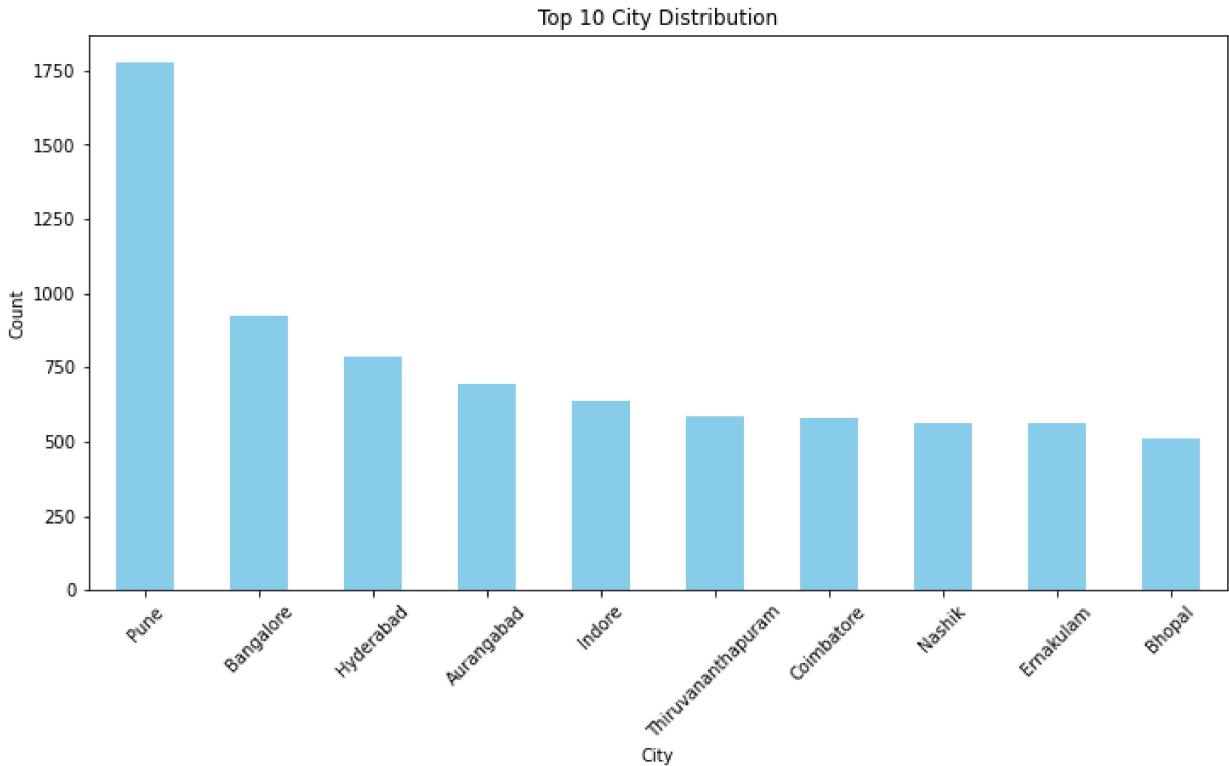
```

Summary statistics of City:

|                    |        |
|--------------------|--------|
| count              | 24582  |
| unique             | 186    |
| top                | Pune   |
| freq               | 1780   |
| Name: City, dtype: | object |

City Distribution:

|                     |                   |
|---------------------|-------------------|
| Pune                | 1780              |
| Bangalore           | 925               |
| Hyderabad           | 784               |
| Aurangabad          | 693               |
| Indore              | 637               |
| ...                 |                   |
| MAHE                | 3                 |
| Chengaianna         | 2                 |
| Mandla              | 2                 |
| BURHANPUR           | 1                 |
| Gadchiroli          | 1                 |
| Name: City, Length: | 186, dtype: int64 |



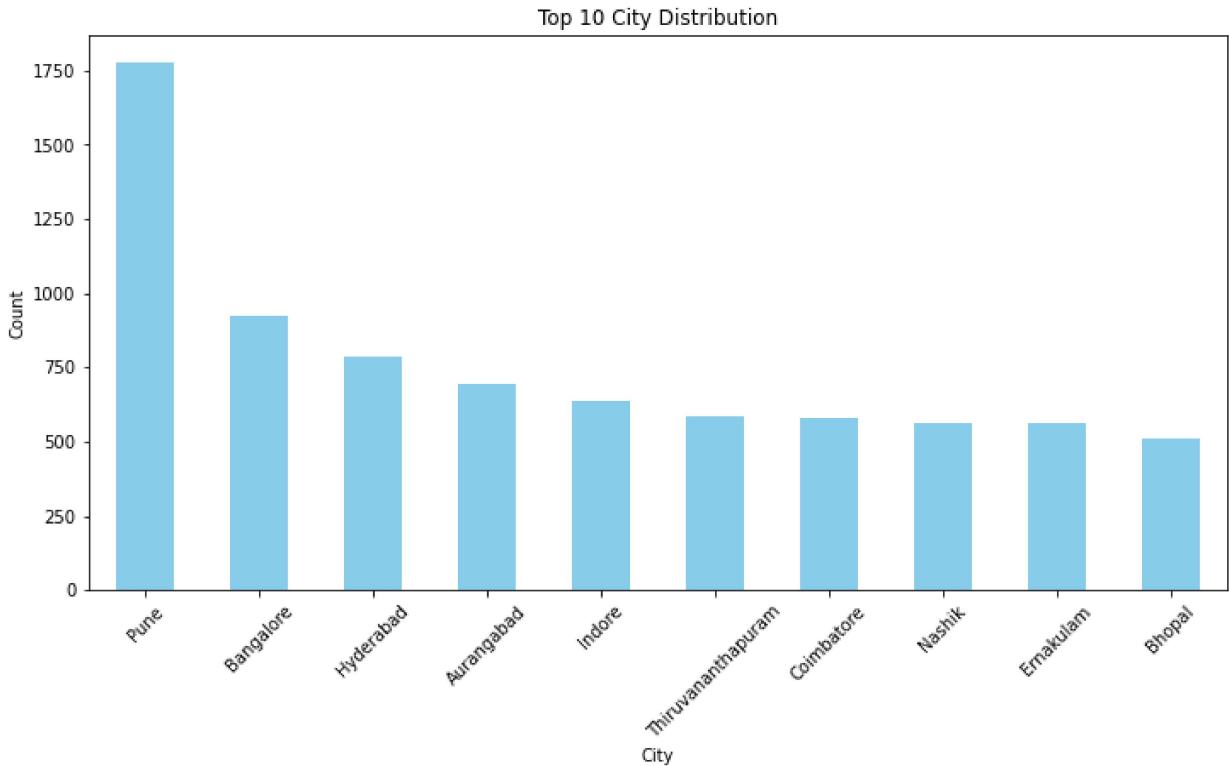
```
In [26]: #df['city'].describe()
df['City'].describe()
```

```
Out[26]: count      24582
unique       186
top          Pune
freq        1780
Name: City, dtype: object
```

```
In [28]: city_counts
```

```
Out[28]:    Pune      1780
    Bangalore      925
    Hyderabad      784
    Aurangabad      693
    Indore      637
    ...
    MAHE         3
    Chengaianna      2
    Mandla         2
    BURHANPUR      1
    Gadchiroli      1
Name: City, Length: 186, dtype: int64
```

```
In [29]: plt.figure(figsize=(12, 6))
city_counts.head(10).plot(kind='bar', color='skyblue')
plt.title('Top 10 City Distribution')
plt.xlabel('City')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



```
In [30]: bounce_counts = df['Bounce String'].value_counts()
```

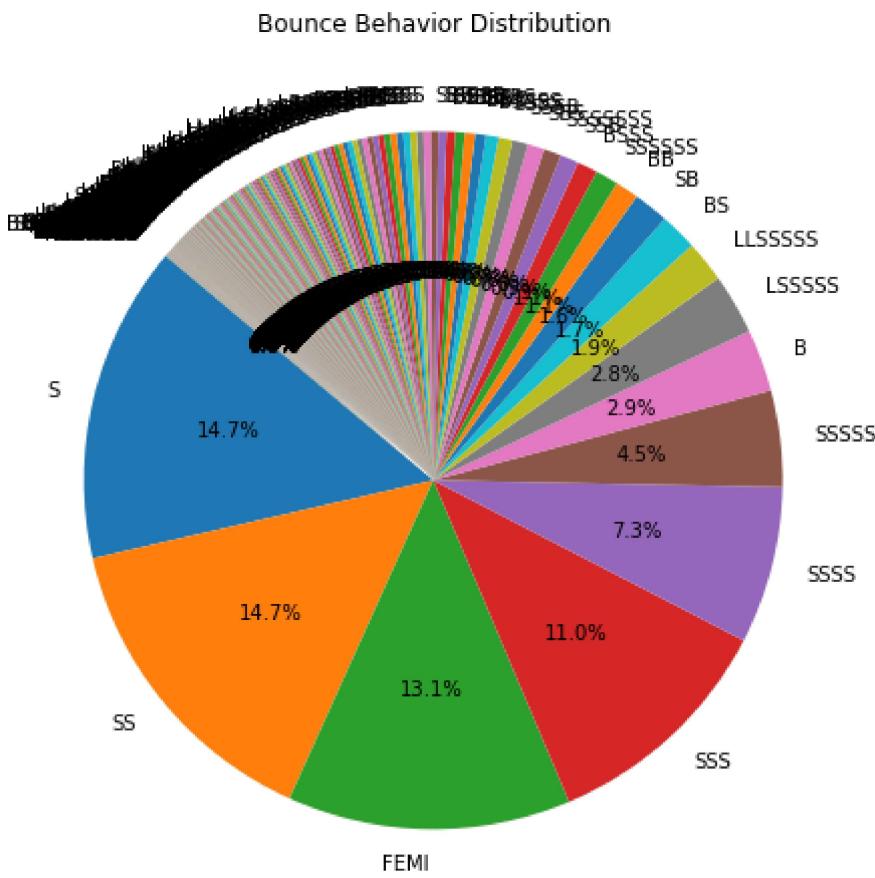
```
print("Bounce Behavior Distribution:")
print(bounce_counts)

plt.figure(figsize=(8, 8))
bounce_counts.plot(kind='pie', autopct='%1.1f%%', startangle=140)
plt.title('Bounce Behavior Distribution')
plt.ylabel('')
plt.show()
```

Bounce Behavior Distribution:

|         |      |
|---------|------|
| S       | 3615 |
| SS      | 3603 |
| FEMI    | 3222 |
| SSS     | 2716 |
| SSSS    | 1790 |
| ...     |      |
| BLBSBBB | 1    |
| HBSHSSS | 1    |
| LLSBBLB | 1    |
| HBBHSSS | 1    |
| BHSBB   | 1    |

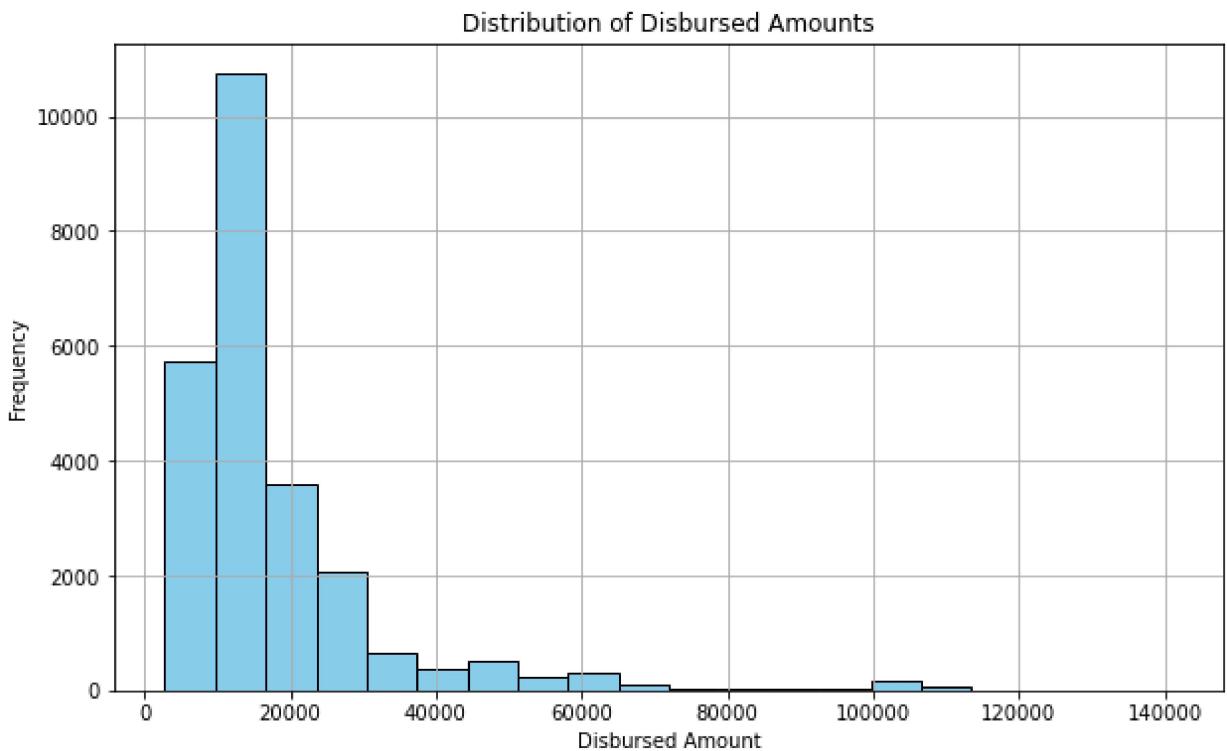
Name: Bounce String, Length: 413, dtype: int64



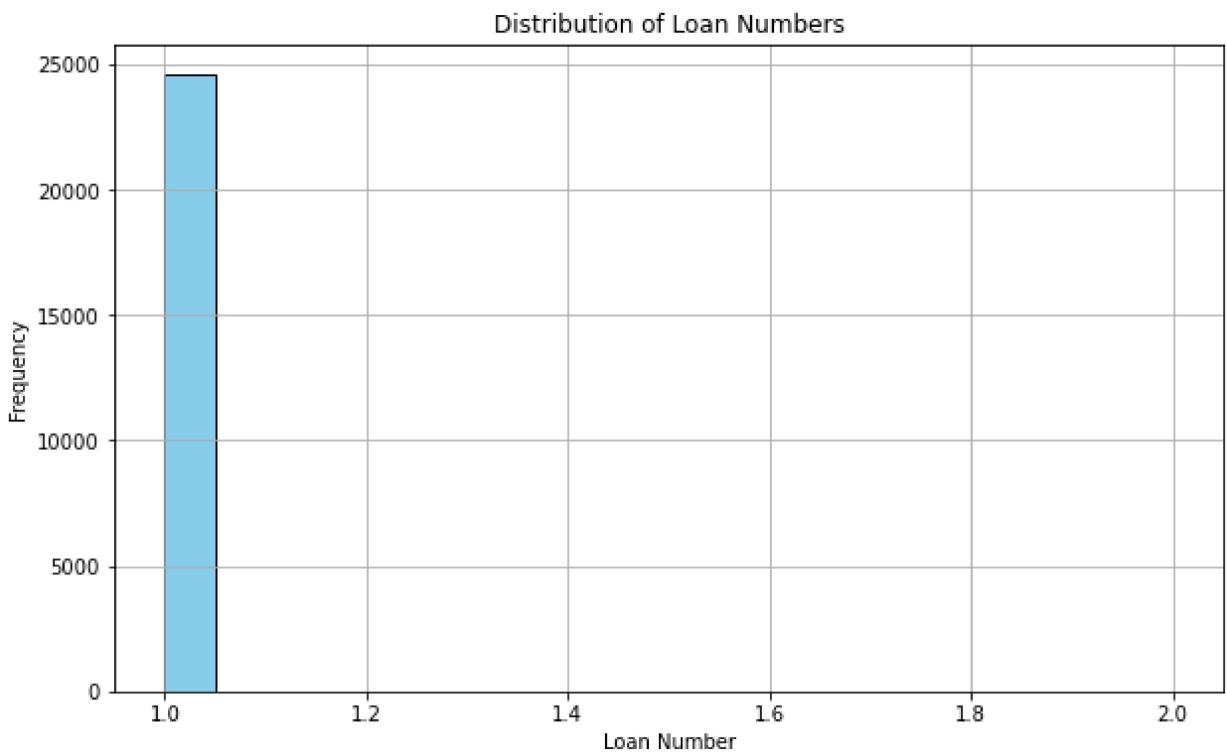
```
In [31]: df['Disbursed Amount'].describe()
```

```
Out[31]: count    24582.000000
mean     17705.195468
std      14192.671509
min      2793.000000
25%     9857.750000
50%    13592.000000
75%    19968.000000
max    141072.000000
Name: Disbursed Amount, dtype: float64
```

```
In [33]: plt.figure(figsize=(10, 6))
plt.hist(df['Disbursed Amount'], bins=20, color='skyblue', edgecolor='black')
plt.title('Distribution of Disbursed Amounts')
plt.xlabel('Disbursed Amount')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



```
In [36]: unique_loan_numbers = df['Loan Number'].nunique()
total_entries = df.shape[0]
plt.figure(figsize=(10, 6))
df['Loan Number'].value_counts().plot(kind='hist', bins=20, color='skyblue', edgecolor='black')
plt.title('Distribution of Loan Numbers')
plt.xlabel('Loan Number')
plt.ylabel('Frequency')
plt.grid(True)
```



```
In [37]: df['Bounces_Last_6_Months'] = df['Bounce String'].apply(lambda x: x[:6].count('B') + x[-1])
```

```

df['Bounced_Last_Month'] = df['Bounce String'].apply(lambda x: x[-1] == 'B' or x[-1] == 'C')

def assign_risk_label(row):
    if row['Bounced_Last_Month']:
        return 'Unknown risk'
    elif row['Bounces_Last_6_Months'] == 0:
        return 'Low risk'
    elif row['Bounces_Last_6_Months'] <= 2:
        return 'Medium risk'
    else:
        return 'High risk'

df['Risk_Label'] = df.apply(assign_risk_label, axis=1)

```

```
print(df[['Loan Number', 'Risk_Label']])
```

|       | Loan Number | Risk_Label   |
|-------|-------------|--------------|
| 0     | JZ6FS       | Low risk     |
| 1     | RDIOY       | Unknown risk |
| 2     | NNW4L       | Medium risk  |
| 3     | 6LBJS       | Low risk     |
| 4     | ZFZUA       | Unknown risk |
| ...   | ...         | ...          |
| 24577 | EAX5C       | Low risk     |
| 24578 | 5MCE9       | Low risk     |
| 24579 | 9H04Q       | Low risk     |
| 24580 | 3VV72       | Low risk     |
| 24581 | 18XBC       | Low risk     |

[24582 rows x 2 columns]

```
In [45]: max_tenure = df['Tenure'].max()
def tenure_status(row):
    if row['Tenure'] <= 3:
        return 'Early Tenure'
    elif row['Tenure'] >= max_tenure - 3:
        return 'Late Tenure'
    else:
        return 'Mid Tenure'

df['Tenure_Status'] = df.apply(tenure_status, axis=1)

print(df[['Loan Number', 'Tenure_Status']])
```

```

      Loan Number Tenure_Status
0           JZ6FS    Mid Tenure
1           RDIOY    Mid Tenure
2           WNW4L    Mid Tenure
3           6LBJS    Mid Tenure
4           ZFZUA    Mid Tenure
...
24577        EAX5C    Mid Tenure
24578        5MCE9    Mid Tenure
24579        9H04Q    Mid Tenure
24580        3VV72    Mid Tenure
24581        18XBC    Mid Tenure

```

[24582 rows x 2 columns]

In [46]: `df_sorted = df.sort_values(by='Amount Pending')`

```

cohort_size = len(df_sorted) // 3

def assign_ticket_size_label(index):
    if index < cohort_size:
        return 'Low ticket size'
    elif index < 2 * cohort_size:
        return 'Medium ticket size'
    else:
        return 'High ticket size'

df_sorted['Ticket Size'] = df_sorted.index.map(assign_ticket_size_label)

print(df_sorted[['Loan Number', 'Amount Pending', 'Ticket Size']])

```

|       | Loan Number | Amount Pending | Ticket Size        |
|-------|-------------|----------------|--------------------|
| 1534  | HEMS0       | 423            | Low ticket size    |
| 1982  | 1BYJD       | 444            | Low ticket size    |
| 889   | 7COLC       | 451            | Low ticket size    |
| 265   | 587TX       | 522            | Low ticket size    |
| 1486  | 5QJN0       | 522            | Low ticket size    |
| ...   | ...         | ...            | ...                |
| 9776  | 8MQRY       | 12500          | Medium ticket size |
| 13946 | 1R840       | 12500          | Medium ticket size |
| 23089 | QUV9D       | 12500          | High ticket size   |
| 14009 | 66HA4       | 12500          | Medium ticket size |
| 13706 | HZ6XJ       | 13349          | Medium ticket size |

[24582 rows x 3 columns]

In [48]: `def assign_spend_category(row):`

```

        if row['Bounce String'].startswith('FEMI'):
            return 'Whatsapp bot'
        elif row['State'] in ['English', 'Hindi'] and row['Bounce String'] == 'Low' and row['Spend Category'] == 'Low':
            return 'Voice bot'
        else:
            return 'Human calling'

```

`df['Spend Category'] = df.apply(assign_spend_category, axis=1)`

```
print(df[['Loan Number', 'Spend Category']])
```

|       | Loan Number | Spend Category |
|-------|-------------|----------------|
| 0     | JZ6FS       | Human calling  |
| 1     | RDIOY       | Human calling  |
| 2     | WNW4L       | Human calling  |
| 3     | 6LBJS       | Human calling  |
| 4     | ZFZUA       | Human calling  |
| ...   | ...         | ...            |
| 24577 | EAX5C       | Whatsapp bot   |
| 24578 | 5MCE9       | Whatsapp bot   |
| 24579 | 9H04Q       | Whatsapp bot   |
| 24580 | 3VV72       | Whatsapp bot   |
| 24581 | 18XBC       | Whatsapp bot   |

[24582 rows x 2 columns]

In [ ]: