In [1]:
```
import pandas as pd
```

In [78]:
```
df = pd.read_csv("Data_Analyst_Assignment_Dataset.csv")
```

In [79]:
```
df
```

Out[79]:

	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 [80]:
```
def calculate_risk_label(row):
    if 'FEMI' in row['Bounce String']:
        return 'Unknown risk'
    if 'B' not in row['Bounce String'] and 'L' not in row['Bounce String']:
        return 'Low risk'
    bounce_count = row['Bounce String'].count('B') + row['Bounce String'].count('L')
    if bounce_count < 2 and 'B' not in row['Bounce String'][-1]:
        return 'Medium risk'
        return 'High risk'
df['Risk Label'] = df.apply(calculate_risk_label, axis=1)
df = pd.read_csv("Risk_Labels.csv")
df
```

Out[80]:

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

24582 rows × 10 columns

In [47]:
```
def calculate_tenure_label(row):

    if row['Tenure'] == 3:
        return 'Early tenure'

    if row['Tenure'] == row['Tenure'] - 3:
        return 'Late tenure'

    return 'Mid tenure'
```

```
df['Tenure Label'] = df.apply(calculate_tenure_label, axis=1)

df.to_csv("Tenure_Label.csv")
df
```

Out[47]:

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

24582 rows × 11 columns

In [81]:

```
df = pd.read_csv("Tenure_Label.csv")
df_sorted = df.sort_values(by='Amount Pending')
df_sorted['Cumulative Amount Pending'] = df_sorted['Amount Pending'].cumsum()
total_amount_pending = df_sorted['Amount Pending'].sum()
threshold_low = total_amount_pending / 3
threshold_high = total_amount_pending * 2 / 3


def assign_ticket_size_label(row):
    if row['Cumulative Amount Pending'] <= threshold_low:
        return 'Low ticket size'
    elif row['Cumulative Amount Pending'] <= threshold_high:
        return 'Medium ticket size'
    else:
        return 'High ticket size'
df_sorted['Ticket Size Label'] = df_sorted.apply(assign_ticket_size_label, axis=1)


df_sorted.to_csv("Amount_Pending.csv")
df_sorted
```

Out[81]:

	Unnamed: 0.1	Unnamed: 0	Amount Pending	State	Tenure	Interest Rate	City	Bounce String	Disbursed Amount	Loan Number	Risk Label	T
1534	1534	1534	423	Maharashtra	11	11.84	Sangli	FEMI	4389	HEMS0	Unknown risk	
1982	1982	1982	444	Tamil Nadu	11	12.23	VIRUDHUNAGAR	FEMI	4598	1BYJD	Unknown risk	
889	889	889	451	Maharashtra	7	37.92	Pune	LSSSSB	2793	7COLC	NaN	
265	265	265	522	Karnataka	11	12.83	Bagalkot	FEMI	5390	587TX	Unknown risk	
1486	1486	1486	522	Maharashtra	11	12.83	Pune	S	5390	5QJN0	Low risk	
...	...	...	...	...	...	...	...	...	...	...	...	
9776	9776	9776	12500	Maharashtra	8	0.00	Kolhapur	LLSSSSS	100000	8MQRY	NaN	
13946	13946	13946	12500	Maharashtra	8	0.00	Pune	S	100000	1R840	Low risk	
23089	23089	23089	12500	Kerala	8	0.00	MALAPPURAM	S	100000	QUV9D	Low risk	
14009	14009	14009	12500	Maharashtra	8	0.00	Sangli	S	100000	66HA4	Low risk	
13706	13706	13706	13349	Maharashtra	8	0.00	Nagpur	S	106792	HZ6XJ	Low risk	

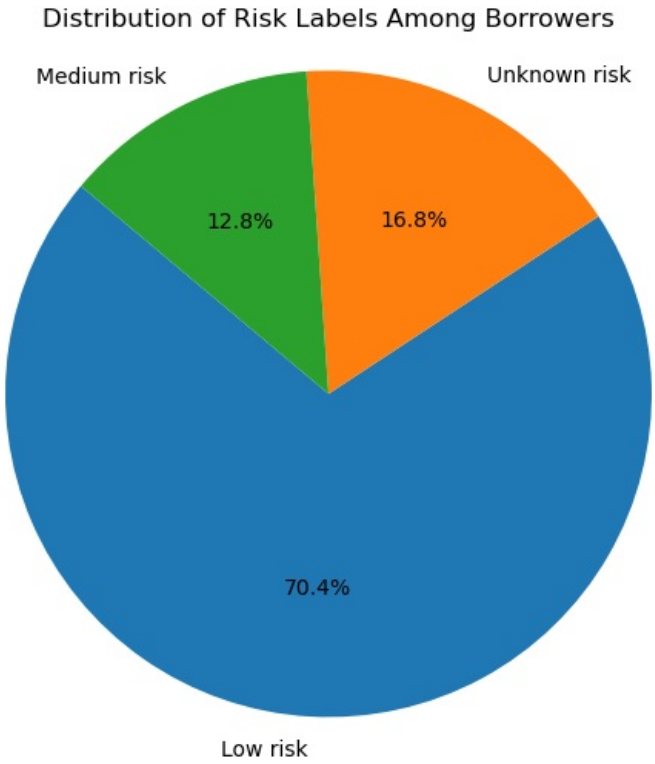
24582 rows × 14 columns

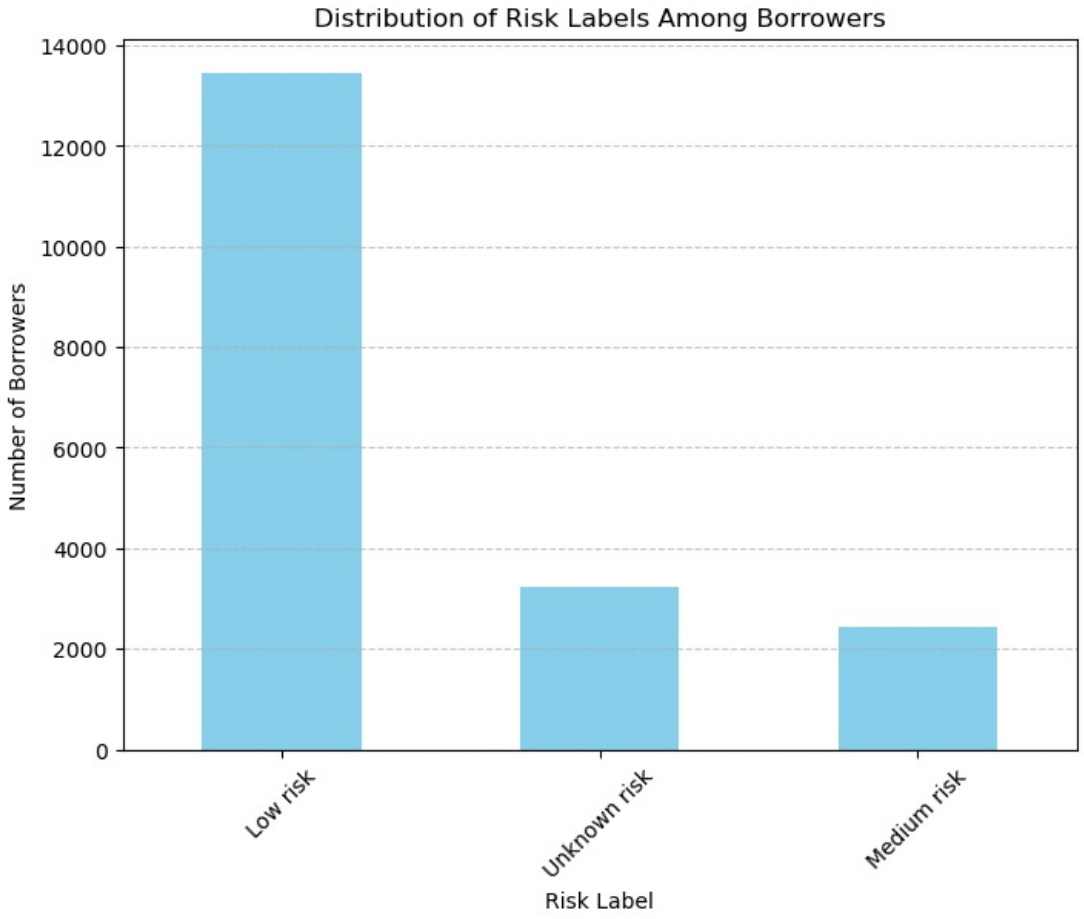
In [83]:
```
import pandas as pd

df = pd.read_csv("Amount_Pending.csv")
def assign_spend_category(row):

    if 'FEMI' in row['Bounce String'] or row['Amount Pending'] == 'Low':
        return 'Whatsapp bot'

    elif 'B' not in row['Bounce String'] or row['Amount Pending'] in ['Low', 'Medium']:
        return 'Voice bot'

    else:
        return 'Human calling'

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

whatsapp_cost = df[df['Spend Category'] == 'Whatsapp bot'].shape[0] * 5
voice_cost = df[df['Spend Category'] == 'Voice bot'].shape[0] * 10
human_cost = df[df['Spend Category'] == 'Human calling'].shape[0] * 50

print("Total cost for Whatsapp bot:", whatsapp_cost, "rupees")
print("Total cost for Voice bot:", voice_cost, "rupees")
print("Total cost for Human calling:", human_cost, "rupees")
```

```
Total cost for Whatsapp bot: 16110 rupees
Total cost for Voice bot: 147610 rupees
Total cost for Human calling: 329950 rupees
```

In [84]:
```
import pandas as pd
import matplotlib.pyplot as plt

risk_counts = df['Risk Label'].value_counts()

plt.figure(figsize=(8, 6))
plt.pie(risk_counts, labels=risk_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Risk Labels Among Borrowers')
plt.axis('equal')
plt.show()
```

```
plt.figure(figsize=(8, 6))
risk_counts.plot(kind='bar', color='skyblue')
plt.title('Distribution of Risk Labels Among Borrowers')
plt.xlabel('Risk Label')
plt.ylabel('Number of Borrowers')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

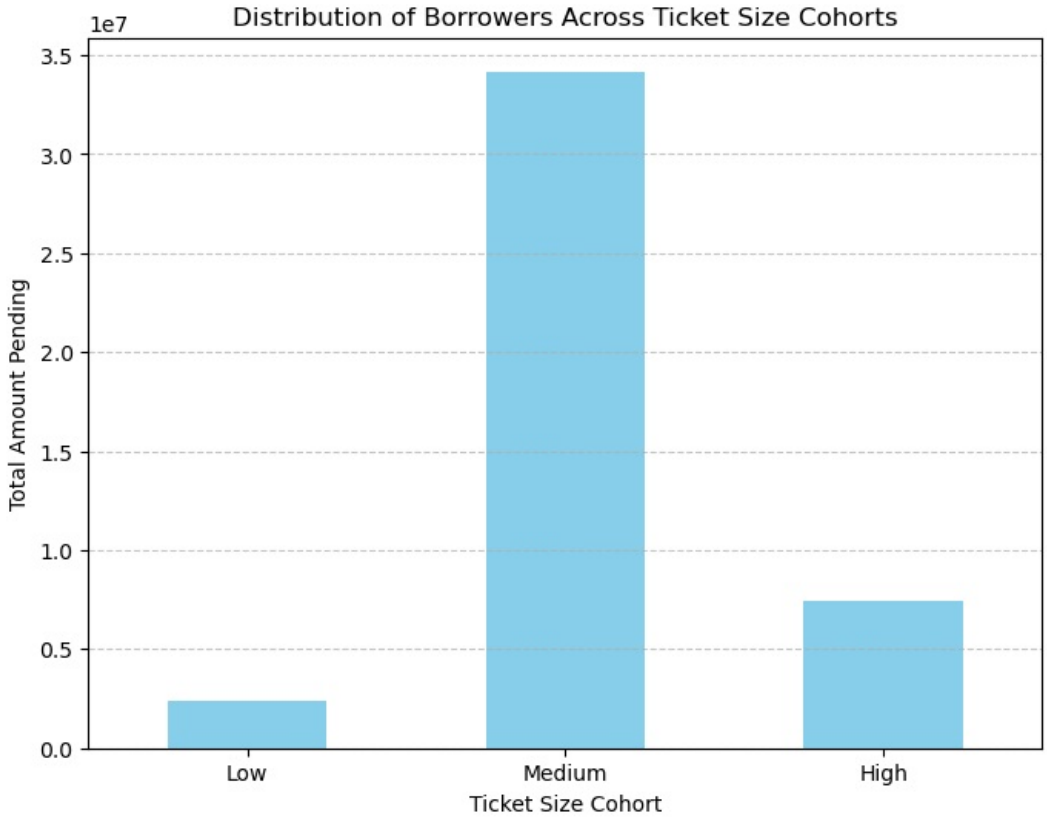


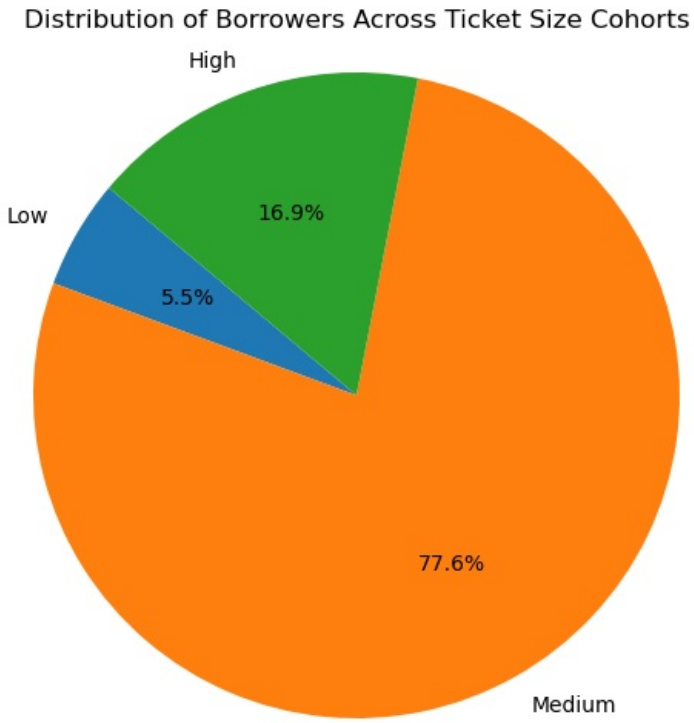


In [85]:
```
import pandas as pd
import matplotlib.pyplot as plt

max_possible_tenure = df['Tenure'].max()
df['Tenure Completion'] = df['Tenure'] / max_possible_tenure

low_threshold = 1000
high_threshold = 3000
```

```
df['Total Amount Pending'] = df['Amount Pending'].groupby(df['Loan Number']).transform('sum')

df['Ticket Size'] = pd.cut(df['Total Amount Pending'], bins=[0, low_threshold, high_threshold, float('inf')],
                           labels=['Low', 'Medium', 'High'])

cohort_amounts = df.groupby('Ticket Size')['Total Amount Pending'].sum()

plt.figure(figsize=(8, 6))
cohort_amounts.plot(kind='bar', color='skyblue')
plt.title('Distribution of Borrowers Across Ticket Size Cohorts')
plt.xlabel('Ticket Size Cohort')
plt.ylabel('Total Amount Pending')
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

plt.figure(figsize=(8, 6))
plt.pie(cohort_amounts, labels=cohort_amounts.index, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Borrowers Across Ticket Size Cohorts')
plt.axis('equal')
plt.show()
```

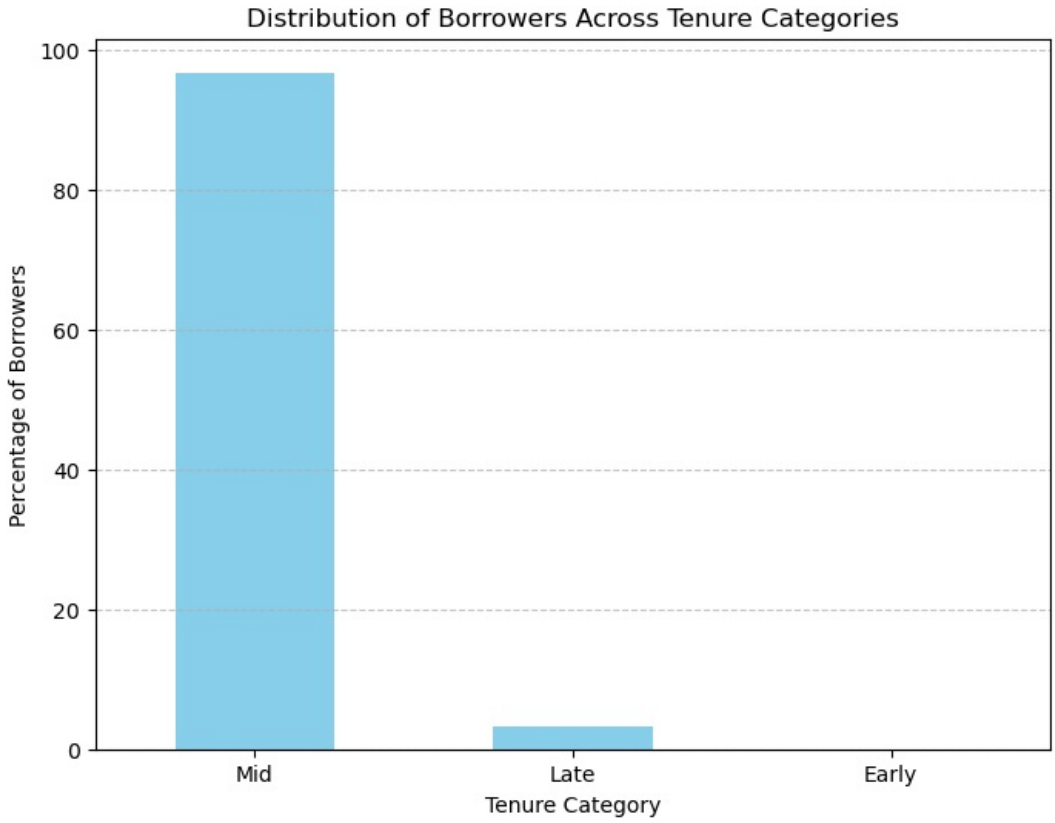
```
C:\Users\madhu\AppData\Local\Temp\ipykernel_24860\3903343702.py:15: FutureWarning: The default of observed=False
is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current b
ehavior or observed=True to adopt the future default and silence this warning.
  cohort_amounts = df.groupby('Ticket Size')['Total Amount Pending'].sum()
```

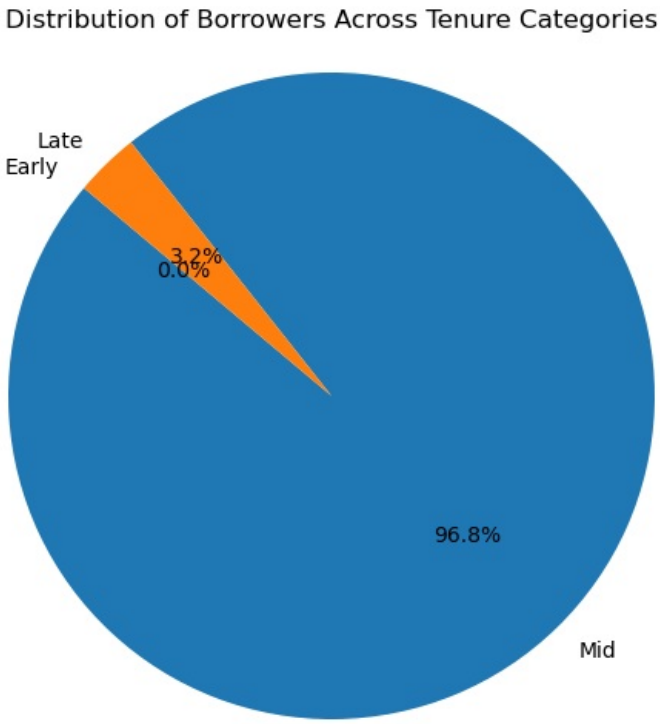




In [86]:

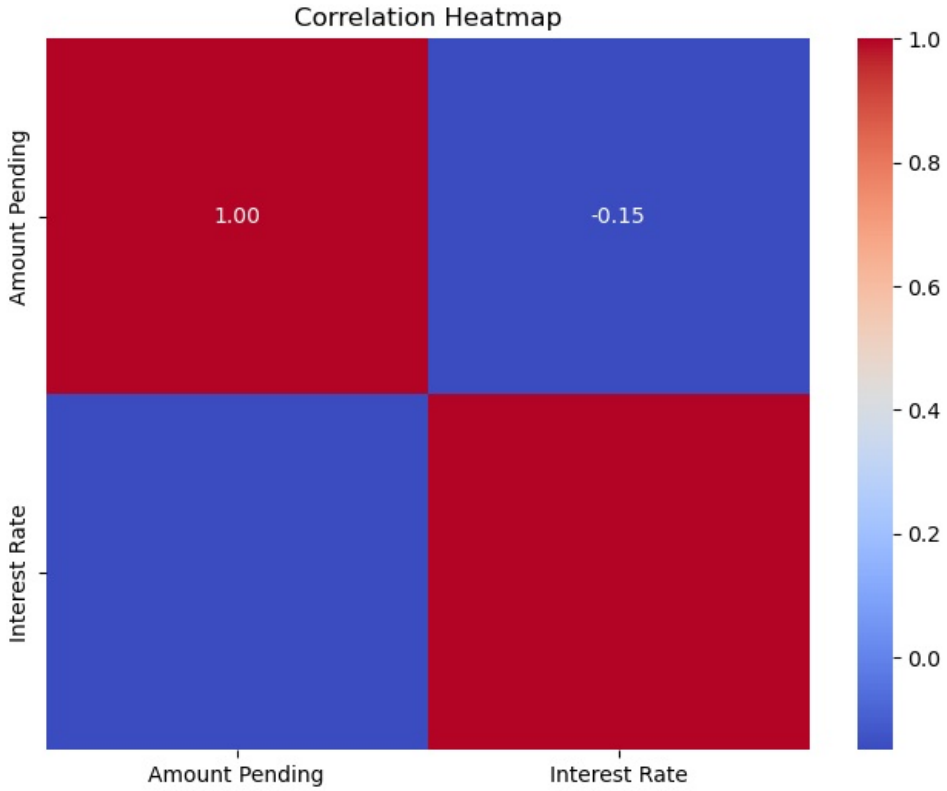
```
import pandas as pd
import matplotlib.pyplot as plt
max_possible_tenure = df['Tenure'].max()
df['Tenure Completion'] = df['Tenure'] / max_possible_tenure

early_threshold = 0.25
late_threshold = 0.75

df['Tenure Category'] = pd.cut(df['Tenure Completion'], bins=[0, early_threshold, late_threshold, float('inf')]
                               labels=['Early', 'Mid', 'Late'])

tenure_counts = df['Tenure Category'].value_counts(normalize=True) * 100

plt.figure(figsize=(8, 6))
tenure_counts.plot(kind='bar', color='skyblue')
plt.title('Distribution of Borrowers Across Tenure Categories')
plt.xlabel('Tenure Category')
plt.ylabel('Percentage of Borrowers')
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

plt.figure(figsize=(8, 6))
plt.pie(tenure_counts, labels=tenure_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Borrowers Across Tenure Categories')
plt.axis('equal')
plt.show()
```

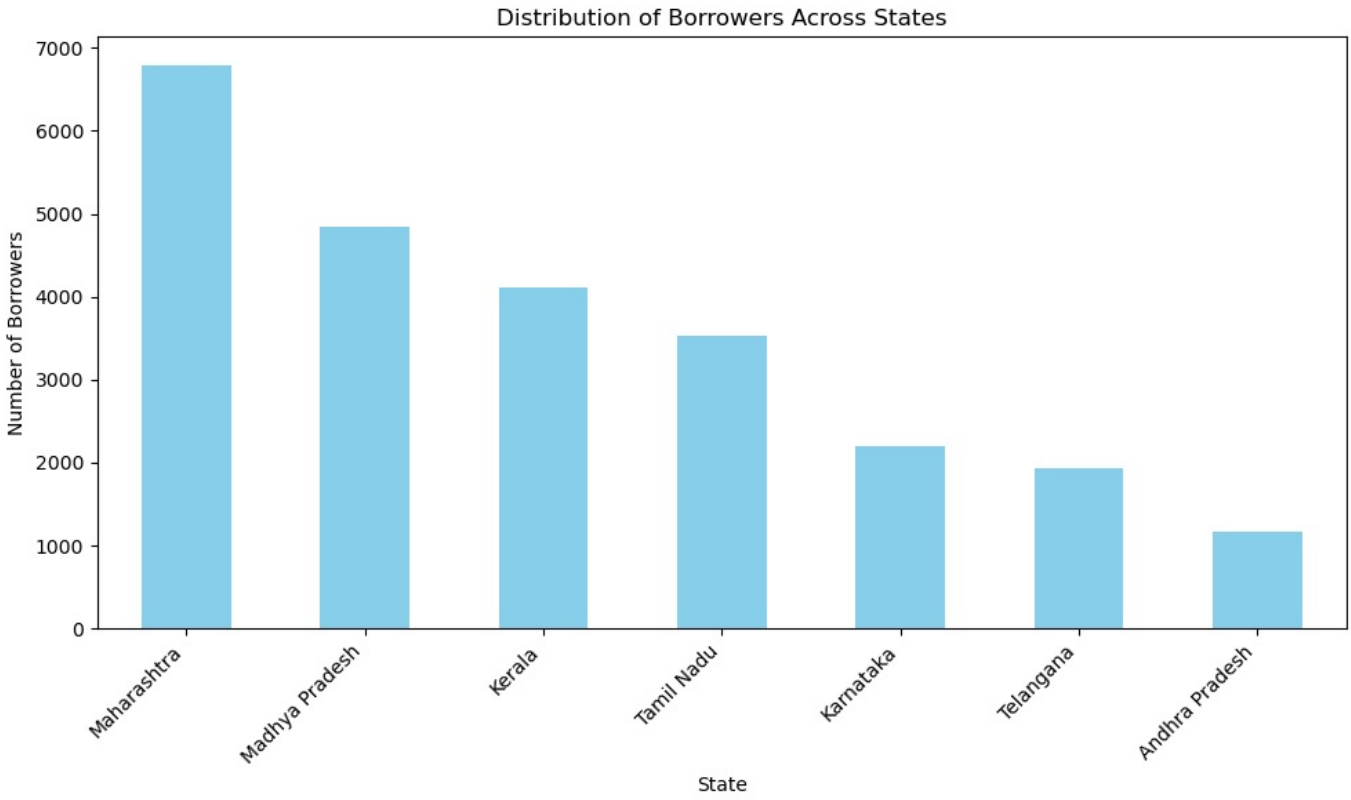


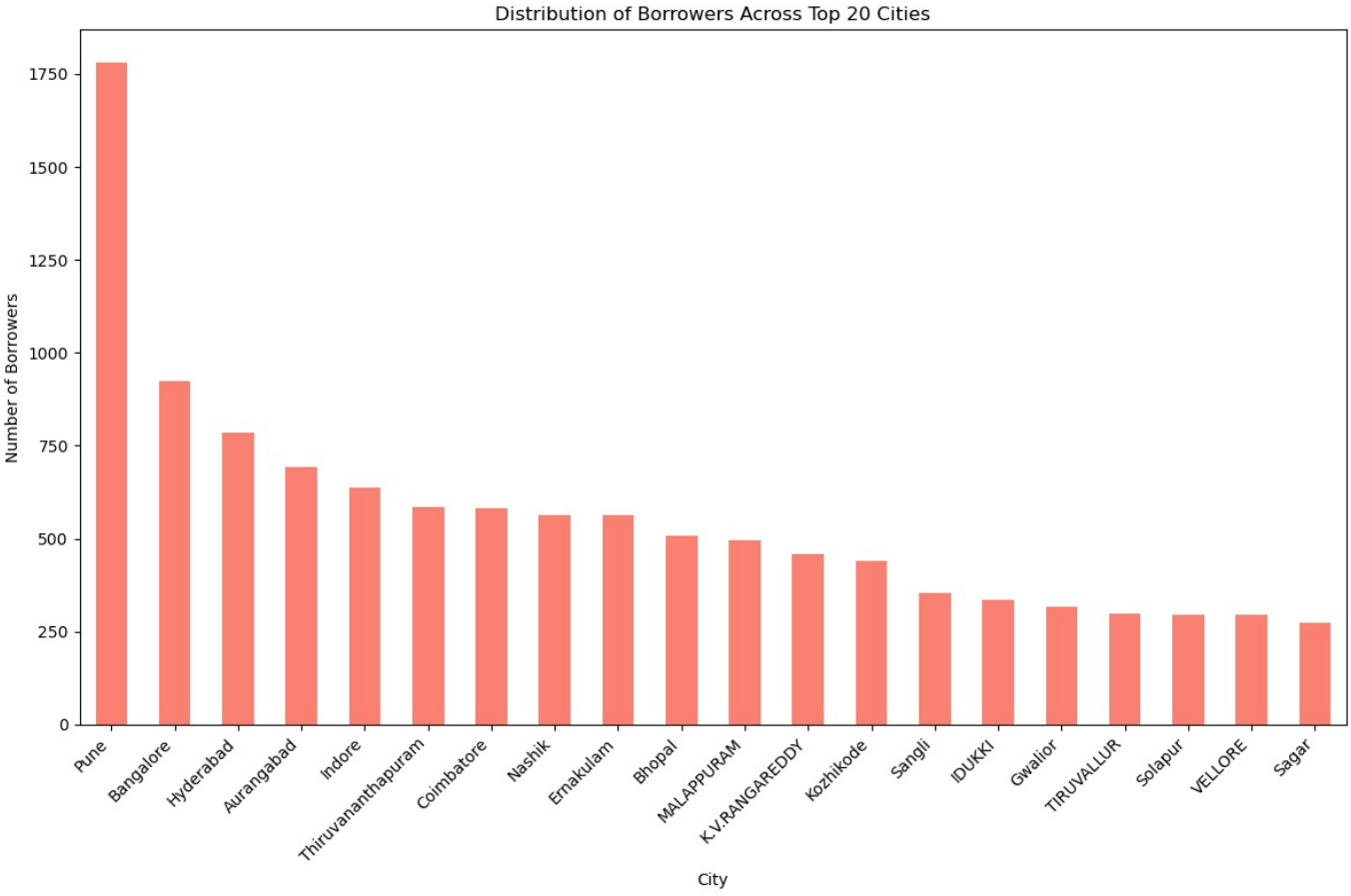


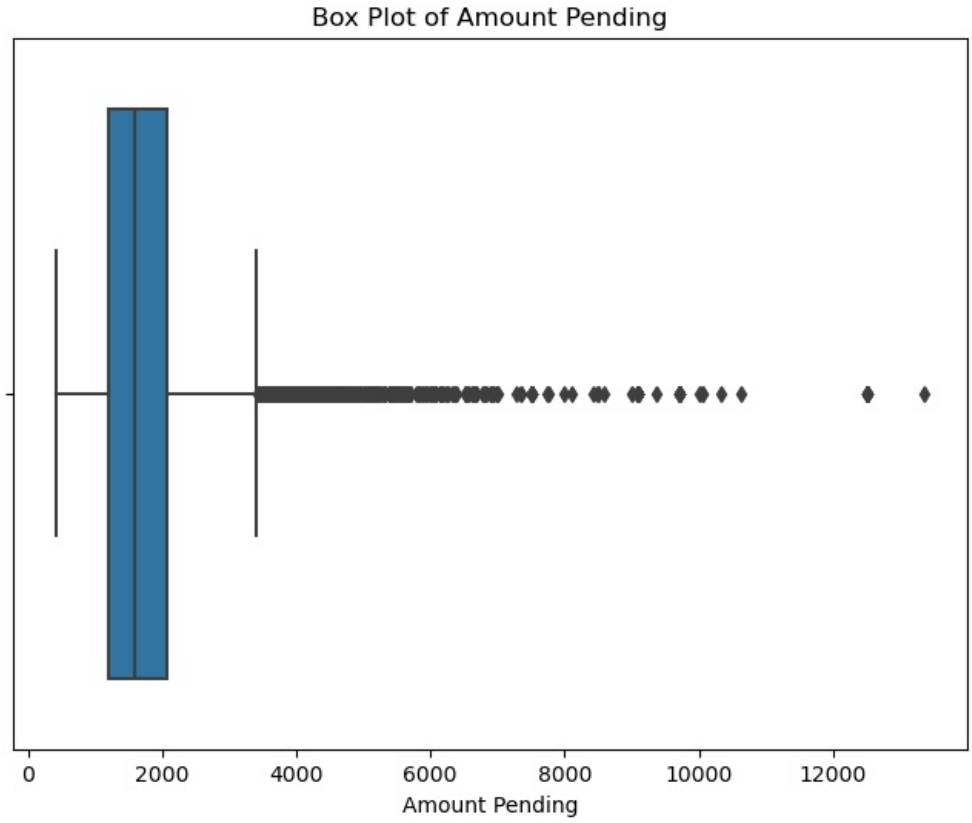
In [87]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


correlation_matrix = df[['Amount Pending', 'Interest Rate']].corr()


plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()


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


plt.figure(figsize=(10, 6))
state_counts.plot(kind='bar', color='skyblue')
```

```
plt.title('Distribution of Borrowers Across States')
plt.xlabel('State')
plt.ylabel('Number of Borrowers')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()


plt.figure(figsize=(12, 8))
city_counts[:20].plot(kind='bar', color='salmon')
plt.title('Distribution of Borrowers Across Top 20 Cities')
plt.xlabel('City')
plt.ylabel('Number of Borrowers')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()


plt.figure(figsize=(8, 6))
sns.boxplot(x=df['Amount Pending'])
plt.title('Box Plot of Amount Pending')
plt.xlabel('Amount Pending')
plt.show()
```




Distribution of Borrowers Across States
Number of Borrowers
0
1000
2000
3000
4000
5000
6000
7000
Maharashtra
Madhya Pradesh
Kerala
Tamil Nadu
Karnataka
Telangana
Andhra Pradesh
State


Distribution of Borrowers Across Top 20 Cities
Number of Borrowers
0
250
500
750
1000
1250
1500
1750
Pune
Bangalore
Hyderabad
Aurangabad
Indore
Thiruvananthapuram
Coimbatore
Nashik
Ernakulam
Bhopal
MALAPPURAM
K.V.RANGAREDDY
Kozhikode
Sangli
IDUKKI
Gwalior
TIRUVALLUR
Solapur
VELLORE
Sagar
City



In [ ]: