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

In [3]:
```
data = pd.read_csv(r"C:\Users\vasan\OneDrive\Desktop\Python\Loan_Data.csv")
```

In [4]:
```
data.head()
```

Out[4]:

	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

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

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

In [84]:
```
def Risk_Category(row):
    if 'B' not in row['Bounce String'][-6:]:
        return "Low Risk"
    elif row['Bounce String'][-6:].count('B')==2 and 'B' not in row['Bounce String'
        return "Medium Risk"
    else:
        return "High Risk"
```

In [85]:
```
data['Risk_Category'] = data.apply(Risk_Category,axis = 1)
```

In [86]:
```
data.head(5)
```

Out[86]:

	Amount Pending	State	Tenure	Interest Rate	City	Bounce String	Disbursed Amount	Loan Number	risk_category	Te
0	963	Karnataka	11	7.69	Bangalore	SSS	10197	JZ6FS	Low Risk	
1	1194	Karnataka	11	6.16	Bangalore	SSB	12738	RDIOY	High Risk	
2	1807	Karnataka	14	4.24	Hassan	BBS	24640	WNW4L	Medium Risk	
3	2451	Karnataka	10	4.70	Bangalore	SSS	23990	6LBJS	Low Risk	
4	2611	Karnataka	10	4.41	Mysore	SSB	25590	ZFZUA	High Risk	

In [87]:
```
data[data.Risk_Category == 'High Risk']. head(10)
```

Out[87]:

	Amount Pending	State	Tenure	Interest Rate	City	Bounce String	Disbursed Amount	Loan Number	risk_category
1	1194	Karnataka	11	6.16	Bangalore	SSB	12738	RDIOY	High Risk
4	2611	Karnataka	10	4.41	Mysore	SSB	25590	ZFZUA	High Risk
5	2172	Karnataka	14	4.36	DAKSHINA KANNADA	SSB	29596	T07RO	High Risk
6	1041	Karnataka	10	5.77	BANGALORE RURAL	SSB	10140	Y0M29	High Risk
20	1969	Karnataka	10	5.65	RAMANAGAR	SBB	19190	F4Z6I	High Risk
24	3340	Karnataka	11	5.81	Mysore	BB	35695	IUFXZ	High Risk
26	2089	Karnataka	10	5.31	Bangalore	BS	20390	26RBN	High Risk
29	1259	Karnataka	10	4.75	BANGALORE RURAL	BS	12320	4WR05	High Risk
33	1426	Karnataka	11	5.14	DAKSHINA KANNADA	BS	15290	KAIC4	High Risk
34	1333	Karnataka	10	6.52	BANGALORE RURAL	SB	12940	0ZE2T	High Risk

In [90]:
```
data['Tenure'].unique()
```

Out[90]:
```
array([11, 14, 10, 17,  7,  8, 18, 24, 15], dtype=int64)
```

In [95]:
```
data.drop(columns = ['risk_category'], inplace = True)
```

In [96]:
```
data['Repaid_Count'] = data['Bounce String'].apply(lambda x: len(x))
```

In [97]:
```
data.head(5)
```

Out[97]:

	Amount Pending	State	Tenure	Interest Rate	City	Bounce String	Disbursed Amount	Loan Number	Tenure_Status	R
0	963	Karnataka	11	7.69	Bangalore	SSS	10197	JZ6FS	Early Tenure	
1	1194	Karnataka	11	6.16	Bangalore	SSB	12738	RDIOY	Early Tenure	
2	1807	Karnataka	14	4.24	Hassan	BBS	24640	WNW4L	Early Tenure	
3	2451	Karnataka	10	4.70	Bangalore	SSS	23990	6LBJS	Early Tenure	
4	2611	Karnataka	10	4.41	Mysore	SSB	25590	ZFZUA	Early Tenure	

In [69]:
```
data['Repaid_Count'].value_counts()
```

Out[69]:
```
4     6399
2     4696
1     4322
3     3401
5     2273
6     2018
7     1343
8      110
9       13
10       7
Name: Repaid_Count, dtype: int64
```

In [77]:
```
def Tenure_Status (row):
    if row['Repaid_Count'] <= 3:
        return "Early Tenure"
    elif row['Tenure'] - row['Repaid_Count'] <= 3:
        return "Late Tenure"
    else:
        return "Mid Tenure"
```

In [78]:
```
data['Tenure_Status'] = data.apply(Tenure_Status , axis = 1)
```

In [98]:
```
data.head(5)
```

Out[98]:

	Amount Pending	State	Tenure	Interest Rate	City	Bounce String	Disbursed Amount	Loan Number	Tenure_Status	R
0	963	Karnataka	11	7.69	Bangalore	SSS	10197	JZ6FS	Early Tenure	
1	1194	Karnataka	11	6.16	Bangalore	SSB	12738	RDIOY	Early Tenure	
2	1807	Karnataka	14	4.24	Hassan	BBS	24640	WNW4L	Early Tenure	
3	2451	Karnataka	10	4.70	Bangalore	SSS	23990	6LBJS	Early Tenure	
4	2611	Karnataka	10	4.41	Mysore	SSB	25590	ZFZUA	Early Tenure	

In [99]:
```
data[data.Tenure_Status == 'Late Tenure'] .head()
```

Out[99]:

	Amount Pending	State	Tenure	Interest Rate	City	Bounce String	Disbursed Amount	Loan Number	Tenure_Status
351	901	Madhya Pradesh	7	18.10	Vidisha	LSSBBS	5943	ZVXZB	Late Tenure
352	751	Madhya Pradesh	7	21.92	Rewa	LSSSSS	4893	ZCGBR	Late Tenure
353	901	Madhya Pradesh	7	18.10	Damoh	LSBSSS	5943	QEMGC	Late Tenure
354	751	Madhya Pradesh	7	21.92	Sidhi	LBSSSS	4893	HYZZD	Late Tenure
355	751	Madhya Pradesh	7	21.92	Chhatarpur	LSSSSS	4893	T03EE	Late Tenure

In [107…
```
def Ticket_Size (row):
    if row['Amount Pending'] <= 3000:
        return 'Low ticket size'
    elif row['Amount Pending'] > 3000 and row['Amount Pending'] <=5000:
```

```
        return 'Medium ticket size'
    else:
        return 'High ticket size'
```

In [108…
```
data['Ticket_Size'] = data.apply(Ticket_Size, axis = 1)
```

In [109…
```
data['Ticket_Size'] . value_counts()
```

Out[109]:
```
Low ticket size       22859
Medium ticket size     1330
High ticket size        393
Name: Ticket_Size, dtype: int64
```

In [124…
```
def Loan_Range (row):
    if row['Disbursed Amount'] <=20000:
        return 'Low EMI'
    elif row['Disbursed Amount'] >20000 and row['Disbursed Amount'] <=50000:
        return 'Medium EMI'
    else:
        return 'High EMI'

data['Loan_Range'] = data.apply(Loan_Range, axis = 1)

data['Loan_Range']. value_counts()
```

Out[124]:
```
Low EMI       18708
Medium EMI     4901
High EMI        973
Name: Loan_Range, dtype: int64
```

In [137…
```
def Channel (row):
    if row['Bounce String'].count('S') + row['Bounce String'].count('H') >3 \
    or row['Bounce String'] == 'FEMI' or row['Loan_Range'] == 'Low EMI':
        return 'Whatsapp bot'
    elif row['City'] in ('Bangalore','Chennai','Mumbai','Hyderabad','Kolkata','Delh
    and row['Interest Rate'] <=5 and row['State'] in ('Madhya Pradesh','Maharashtra
    and row['Bounce String'].count('B') + row['Bounce String'].count('L') <=2\
    and row['Loan_Range'] in ('Low EMI','Medium EMI'):
        return 'Voice bot'
    elif row['Bounce String'].count('B') + row['Bounce String'].count('L') >2 \
    or 'B' in row['Bounce String'][-2:] or 'L' in row['Bounce String'][-2:] or 'BL'
    or 'LB' in row['Bounce String'][-2:]:
        return 'Human calling (Necessary)'
    else:
        return 'Human calling (Optional)'

data['Channel'] = data.apply(Channel,axis = 1)
data.head()
data['Channel']. value_counts()
```

Out[137]:
```
Whatsapp bot                21119
Human calling (Optional)     2358
Human calling (Necessary)    1098
Voice bot                       7
Name: Channel, dtype: int64
```

In [138…
```
def Channel_Cost(row):
    if row['Channel'] == 'Whatsapp bot':
        return 5
    elif row['Channel'] == 'Voice bot':
        return 10
    else:
        return 50
```

```
data['Channel_Cost'] = data.apply(Channel_Cost, axis = 1)
data.head()
```

Out[138]:

	Amount Pending	State	Tenure	Interest Rate	City	Bounce String	Disbursed Amount	Loan Number	Tenure_Status	R
0	963	Karnataka	11	7.69	Bangalore	SSS	10197	JZ6FS	Early Tenure	
1	1194	Karnataka	11	6.16	Bangalore	SSB	12738	RDIOY	Early Tenure	
2	1807	Karnataka	14	4.24	Hassan	BBS	24640	WNW4L	Early Tenure	
3	2451	Karnataka	10	4.70	Bangalore	SSS	23990	6LBJS	Early Tenure	
4	2611	Karnataka	10	4.41	Mysore	SSB	25590	ZFZUA	Early Tenure	

In [152…

```
data['Tenure Complete Balance'] = data['Tenure'] - data['Repaid_Count']
```

In [160…

```
data[data['Tenure Complete Balance'] == 1].head()
```

Out[160]:

	Amount Pending	State	Tenure	Interest Rate	City	Bounce String	Disbursed Amount	Loan Number	Tenure_Status
351	901	Madhya Pradesh	7	18.10	Vidisha	LSSBBS	5943	ZVXZB	Late Tenure
352	751	Madhya Pradesh	7	21.92	Rewa	LSSSSS	4893	ZCGBR	Late Tenure
353	901	Madhya Pradesh	7	18.10	Damoh	LSBSSS	5943	QEMGC	Late Tenure
354	751	Madhya Pradesh	7	21.92	Sidhi	LBSSSS	4893	HYZZD	Late Tenure
355	751	Madhya Pradesh	7	21.92	Chhatarpur	LSSSSS	4893	T03EE	Late Tenure

In [143…

```
import matplotlib.pyplot as plt
```

In [172…

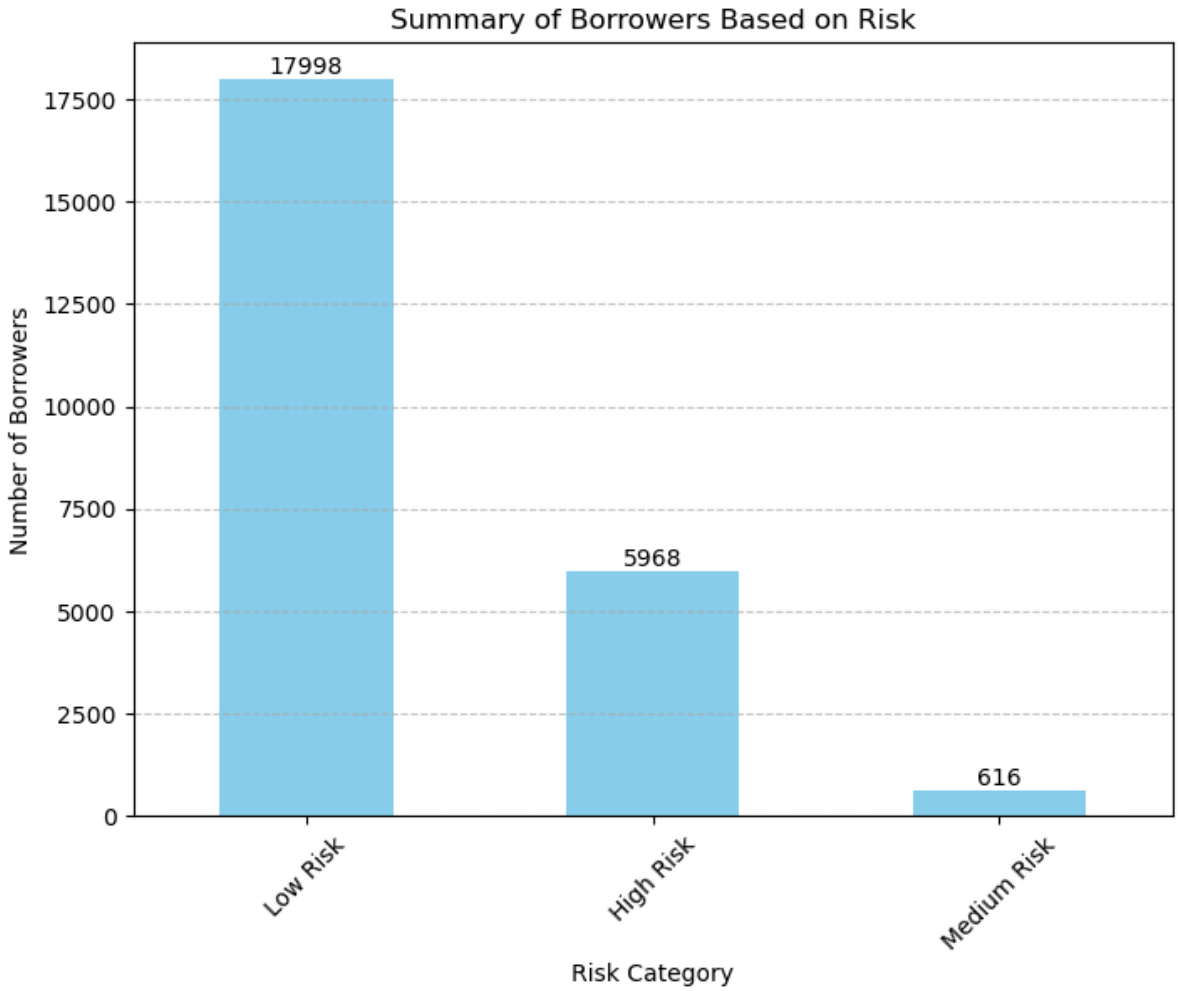
```
#summary of borrowers (with graphs) based on risk

risk_summary = data['Risk_Category'].value_counts()

# Plotting
plt.figure(figsize=(8, 6))
ax = risk_summary.plot(kind='bar', color='skyblue')
plt.title('Summary of Borrowers Based on Risk')
plt.xlabel('Risk Category')
plt.ylabel('Number of Borrowers')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
```

```
# Annotate bars with counts
for i, count in enumerate(risk_summary):
    ax.text(i, count, str(count), ha='center', va='bottom')

plt.show()
```



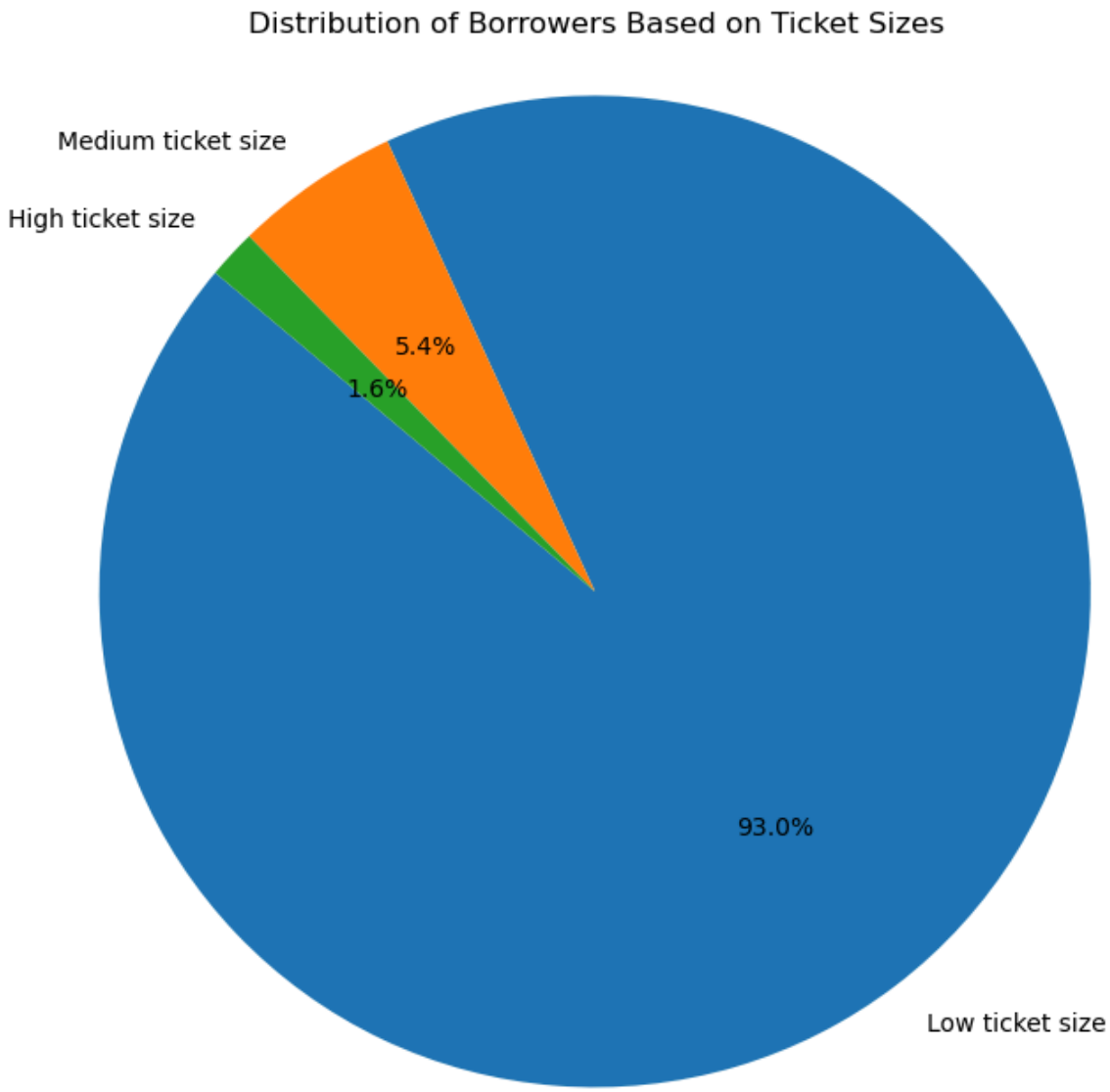
In [170…

```
#summary of borrowers based on ticket sizes

Ticket_size_summary = data['Ticket_Size'].value_counts()

# Plotting
plt.figure(figsize=(8, 8))
plt.pie(Ticket_size_summary, labels=Ticket_size_summary.index, autopct='%1.1f%%', s
plt.title('Distribution of Borrowers Based on Ticket Sizes')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```



In [171…

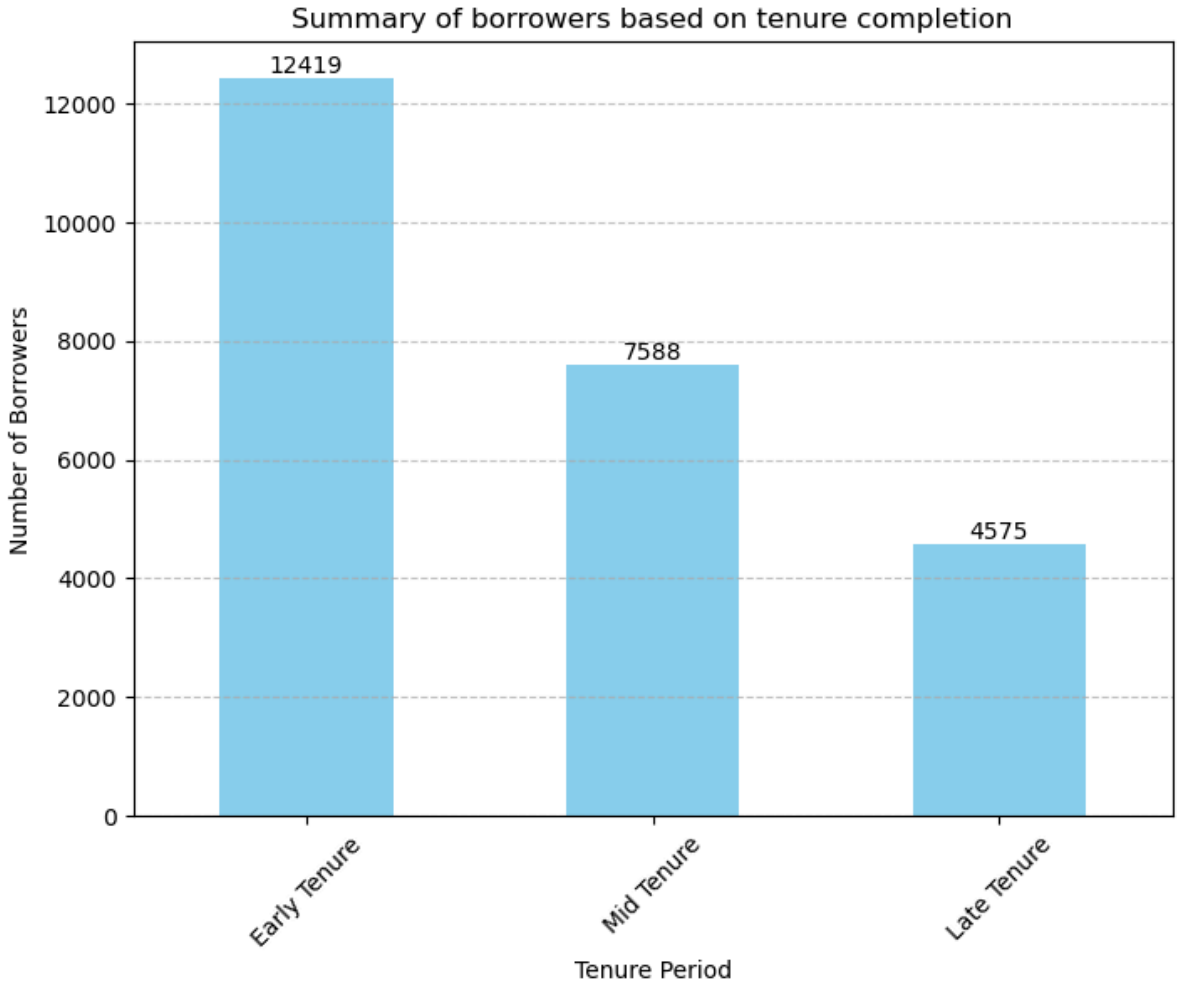
```
#Summary of borrowers based on tenure completion

tenure_completion_summary = data['Tenure_Status'].value_counts()

# Plotting
plt.figure(figsize=(8, 6))
ax = tenure_completion_summary.plot(kind='bar', color='skyblue')
plt.title('Summary of borrowers based on tenure completion')
plt.xlabel('Tenure Period')
plt.ylabel('Number of Borrowers')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Annotate bars with counts
for i, count in enumerate(tenure_completion_summary):
    ax.text(i, count, str(count), ha='center', va='bottom')

plt.show()
```



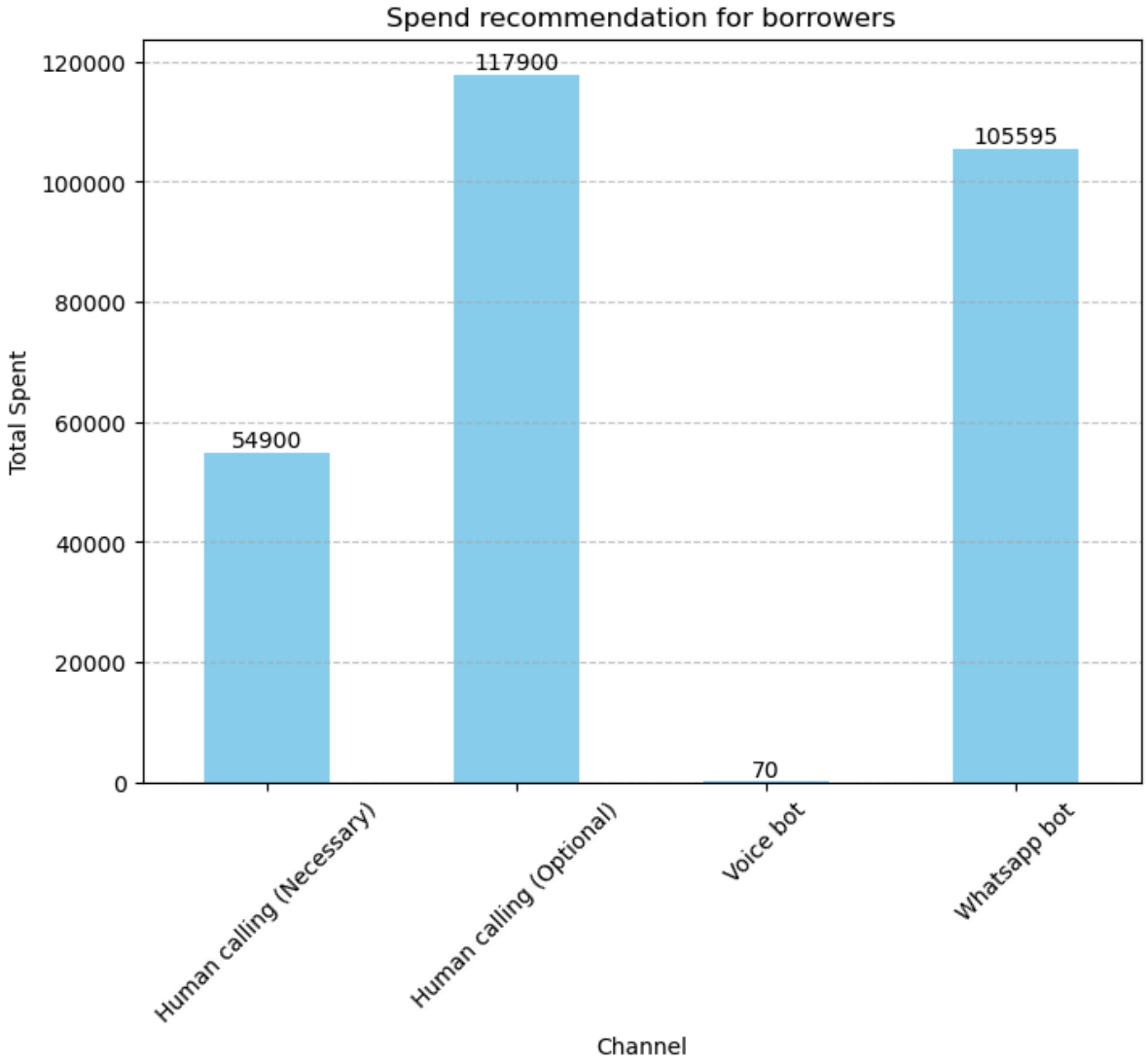
In [168…

```
# Spend recommendation for borrowers

Channel_spending = data.groupby('Channel')['Channel_Cost'].sum()

# Plotting
plt.figure(figsize=(8, 6))
ax = Channel_spending.plot(kind='bar', color='skyblue')
plt.title('Spend recommendation for borrowers')
plt.xlabel('Channel')
plt.ylabel('Total Spent')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Annotate bars with counts
for i, count in enumerate(Channel_spending):
    ax.text(i, count, str(count), ha='center', va='bottom')
```



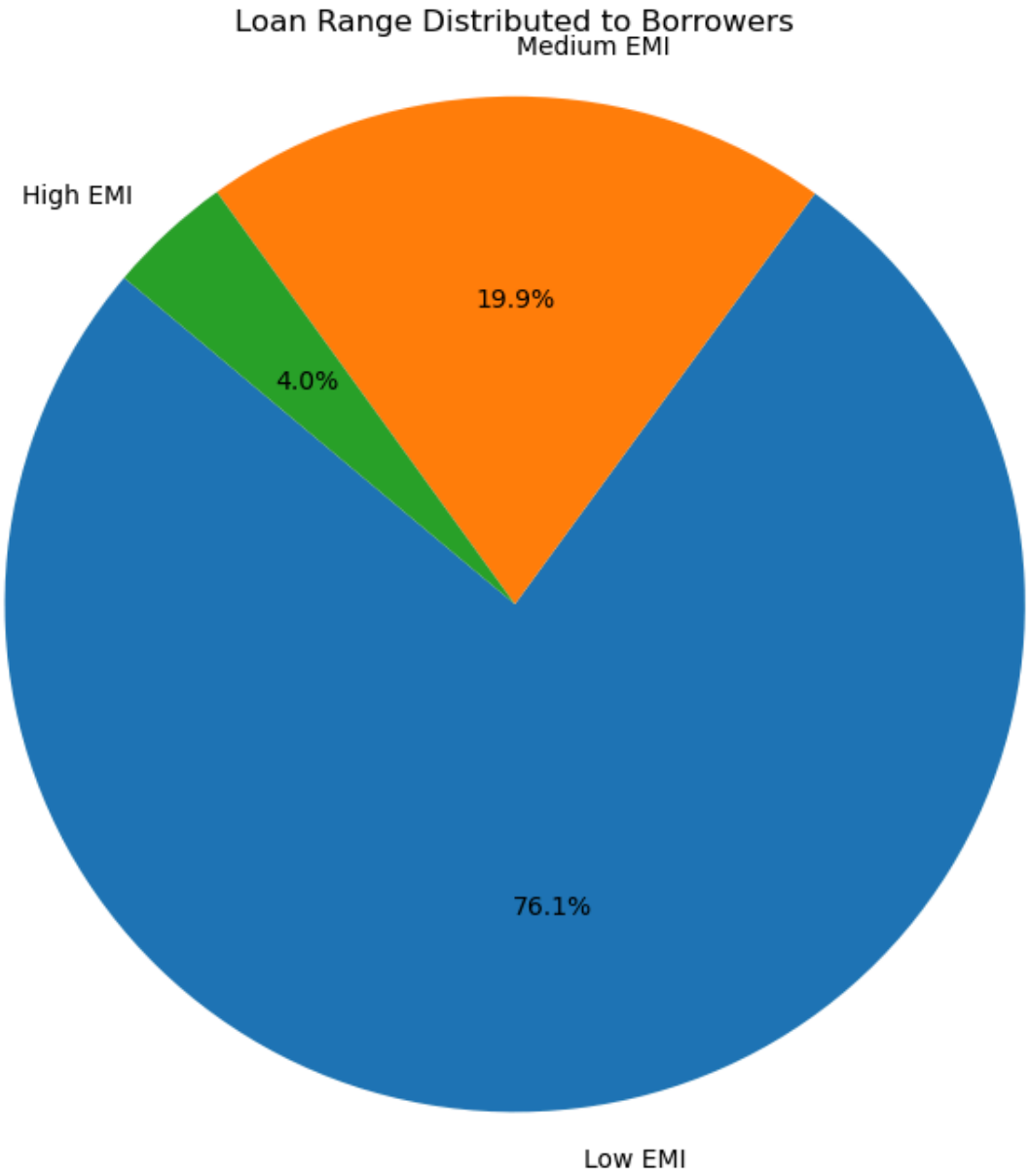
In [169…

```
#Loan Range Distributed to Borrowers

Loan_Range_Summary = data['Loan_Range'].value_counts()

# Plotting
plt.figure(figsize=(8, 8))
plt.pie(Loan_Range_Summary, labels=Loan_Range_Summary.index, autopct='%1.1f%%', sta
plt.title('Loan Range Distributed to Borrowers')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```


Loan Range Distributed to Borrowers
Medium EMI
High EMI
19.9%
4.0%
76.1%
Low EMI