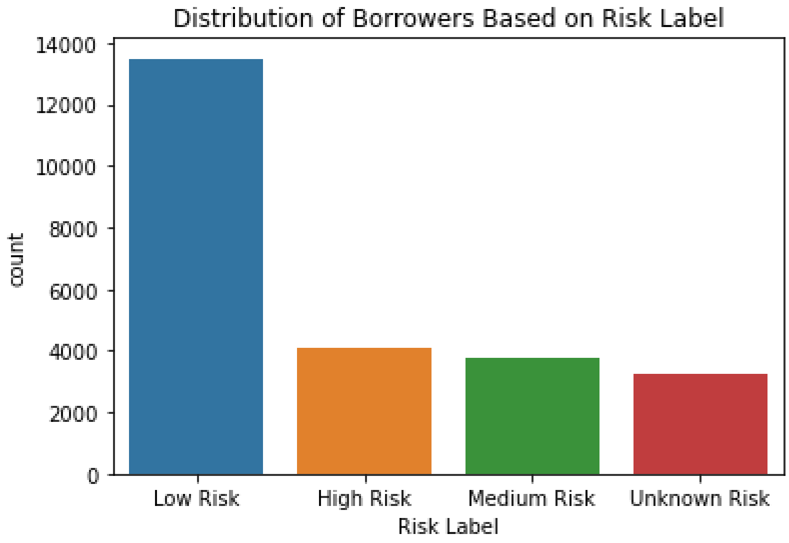
In [11]:

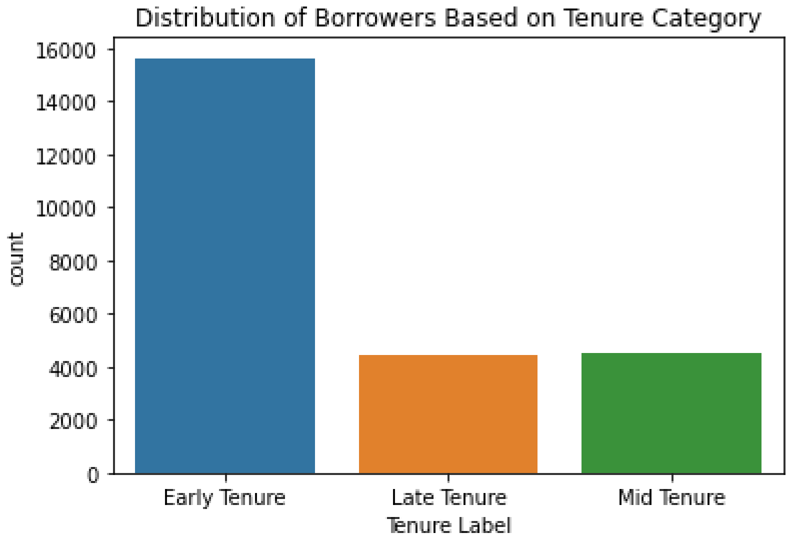
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv("Data_Analyst_Assignment_Dataset.csv")

print(data.head())
print(data.info())
print(data.describe())

#Calculate the risk labels for all the borrowers.
def get_risk_label(bounce_string):
    last_6_months = bounce_string[-6:]
    last_month = last_6_months[-1]
    bounce_count = sum([char in ["B", "L"] for char in last_6_months])

    if bounce_string == "FEMI":
        return "Unknown Risk"

    if bounce_count == 0:
        return "Low Risk"

    if bounce_count <= 2 and last_month not in ["B", "L"]:
        return "Medium Risk"

    return "High Risk"

data["Risk Label"] = data["Bounce String"].apply(get_risk_label)

sns.countplot(x="Risk Label", data=data)
plt.title("Distribution of Borrowers Based on Risk Label")
plt.show()
```

```
   Amount Pending      State  Tenure  Interest Rate       City Bounce Stri
ng  \
0             963  Karnataka      11           7.69  Bangalore           S
SS
1            1194  Karnataka      11           6.16  Bangalore           S
SB
2            1807  Karnataka      14           4.24     Hassan           B
BS
3            2451  Karnataka      10           4.70  Bangalore           S
SS
4            2611  Karnataka      10           4.41     Mysore           S
SB

   Disbursed Amount Loan Number
0             10197       JZ6FS
1             12738       RDIOY
2             24640       WNW4L
3             23990       6LBJS
4             25590       ZFZUA
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24582 entries, 0 to 24581
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Amount Pending    24582 non-null  int64  
 1   State             24582 non-null  object 
 2   Tenure            24582 non-null  int64  
 3   Interest Rate     24582 non-null  float64
 4   City              24582 non-null  object 
 5   Bounce String     24582 non-null  object 
 6   Disbursed Amount  24582 non-null  int64  
 7   Loan Number       24582 non-null  object 
dtypes: float64(1), int64(3), object(4)
memory usage: 1.5+ MB
None
       Amount Pending        Tenure  Interest Rate  Disbursed Amount
count    24582.000000  24582.000000   24582.000000      24582.000000
mean      1791.172687      9.415263       0.934960      17705.195468
std        937.565507      3.238904       3.114732      14192.671509
min        423.000000      7.000000       0.000000       2793.000000
25%       1199.000000      8.000000       0.000000       9857.750000
50%       1593.000000      8.000000       0.000000      13592.000000
75%       2083.000000     11.000000       0.000000      19968.000000
max      13349.000000     24.000000      37.920000     141072.000000
```



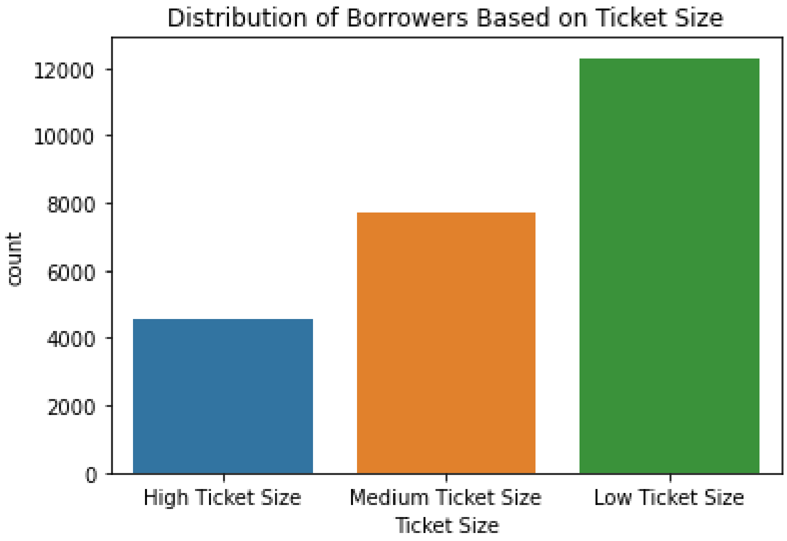
In [12]:

```
#label all customers based on where they are in their tenure
def get_tenure_label(tenure, bounce_string):
    on_book_length = len(bounce_string.replace("FEMI", "").replace(" ", ""))

    if on_book_length <= 3:
        return "Early Tenure"

    if on_book_length >= tenure - 3:
        return "Late Tenure"

    return "Mid Tenure"

data["Tenure Label"] = data.apply(lambda row: get_tenure_label(row["Tenure"]

sns.countplot(x="Tenure Label", data=data)
plt.title("Distribution of Borrowers Based on Tenure Category")
plt.show()
```

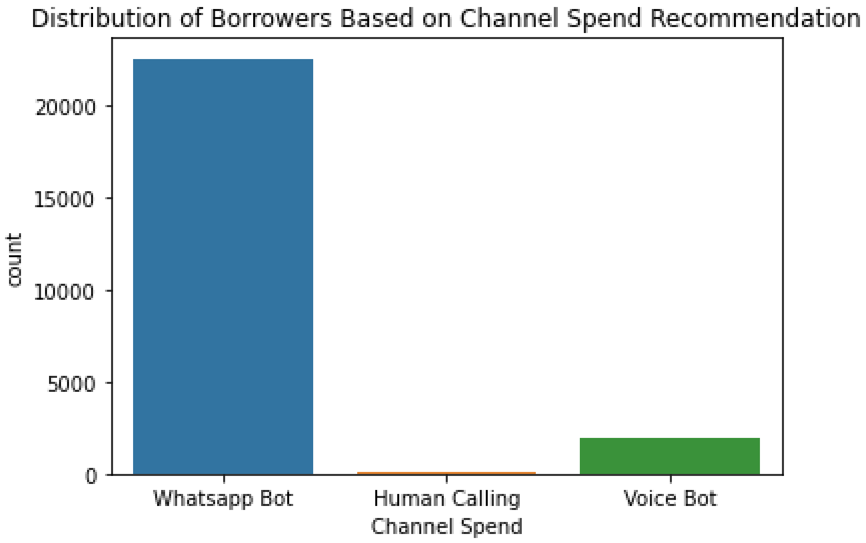


In [8]:

```
#Segment borrowers based on ticket size
data = data.sort_values(by="Amount Pending", ascending=False)

total_pending = data["Amount Pending"].sum()

threshold = total_pending / 3

cumulative_sum = data["Amount Pending"].cumsum()

data["Ticket Size"] = np.where(cumulative_sum <= threshold, "High Ticket Siz
                             np.where(cumulative_sum <= 2 * threshold, "Medi

sns.countplot(x="Ticket Size", data=data)
plt.title("Distribution of Borrowers Based on Ticket Size")
plt.show()
```



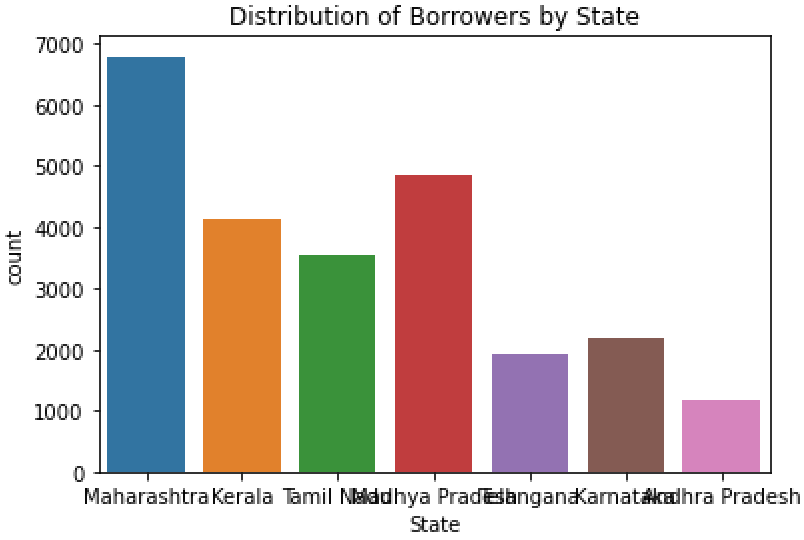
```
In [9]: #Give channel spend recommendations
        def get_channel_spend(row):
            risk_label = row["Risk Label"]
            emi = row["Amount Pending"]
            city = row["City"]
            interest_rate = row["Interest Rate"]

            if risk_label == "Low Risk" or row["Bounce String"] == "FEMI" or emi <
                return "Whatsapp Bot"

            if ((city in ["Delhi", "Mumbai", "Bangalore", "Chennai", "Kolkata"]) or
                return "Voice Bot"

            return "Human Calling"

        data["Channel Spend"] = data.apply(get_channel_spend, axis=1)

        sns.countplot(x="Channel Spend", data=data)
        plt.title("Distribution of Borrowers Based on Channel Spend Recommendation"
        plt.show()
```



In [10]:

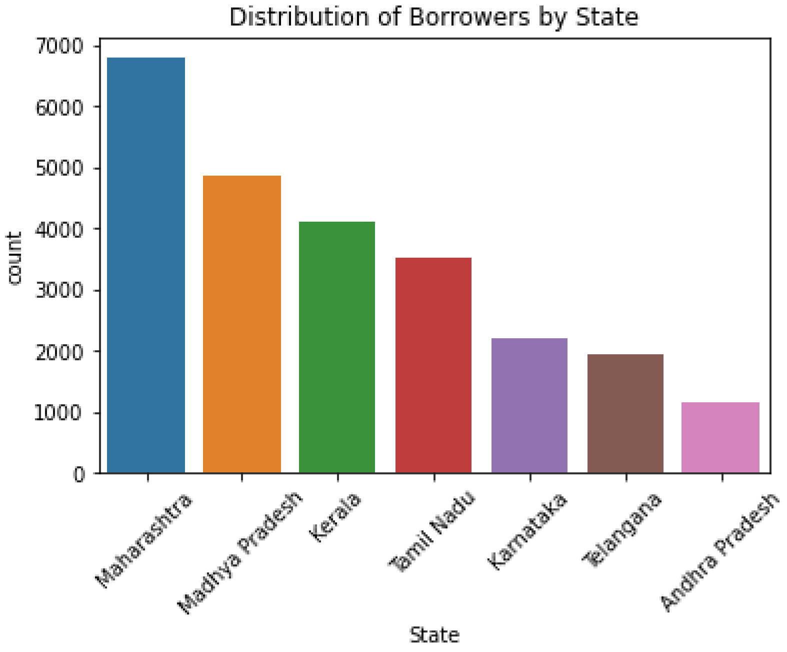
```
#Top 5 states with the most borrowers
print("Top 5 States with the most borrowers:")
print(data["State"].value_counts().head(5))

sns.countplot(x="State", data=data)
plt.title("Distribution of Borrowers by State")
plt.show()
```

```
Top 5 States with the most borrowers:
Maharashtra       6793
Madhya Pradesh    4850
Kerala            4116
Tamil Nadu        3526
Karnataka         2205
Name: State, dtype: int64
```



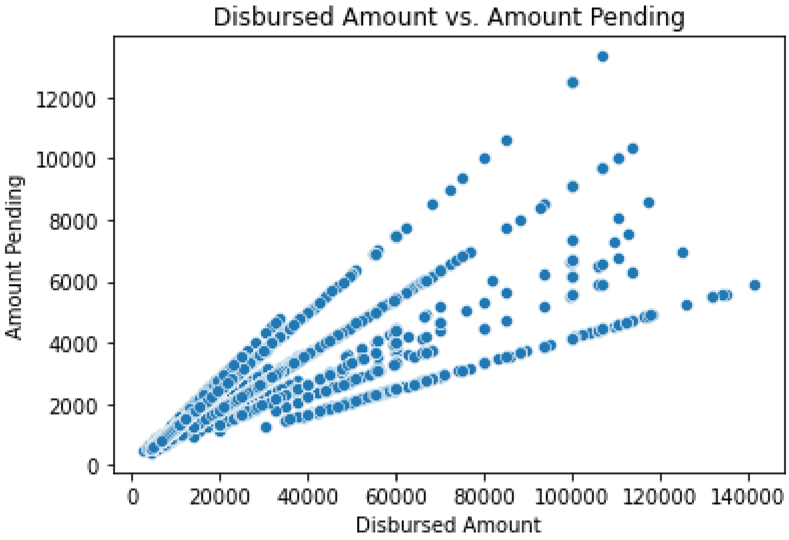
In [13]:

```
# Distribution of borrowers by state
sns.countplot(x="State", data=data, order=data["State"].value_counts().inde
plt.title("Distribution of Borrowers by State")
plt.xticks(rotation=45)
plt.show()
```



In [14]:

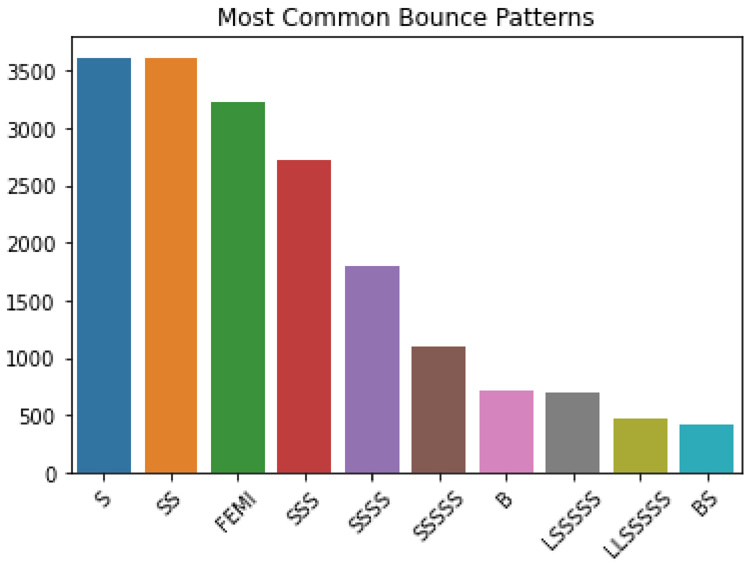
```
# Scatter plot of disbursed amount vs. amount pending
sns.scatterplot(x="Disbursed Amount", y="Amount Pending", data=data)
plt.title("Disbursed Amount vs. Amount Pending")
plt.xlabel("Disbursed Amount")
plt.ylabel("Amount Pending")
plt.show()
```



In [15]:

```
# Analyze common bounce patterns
common_bounce_patterns = data["Bounce String"].value_counts().head(10)
print("Most Common Bounce Patterns:")
print(common_bounce_patterns)

sns.barplot(x=common_bounce_patterns.index, y=common_bounce_patterns.values
plt.title("Most Common Bounce Patterns")
plt.xticks(rotation=45)
plt.show()
```

```
Most Common Bounce Patterns:
S          3615
SS         3603
FEMI       3222
SSS        2716
SSSS       1790
SSSSS      1096
B           707
LSSSSS      687
LLSSSSS     474
BS          425
Name: Bounce String, dtype: int64
```



In [ ]: