

Lokshabha 2024 Election

July 26, 2024

```
[2]: #LOKSHABHA ELECTION 2024 RESULT DATA ANALYSIS
```

```
[4]: #importing libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[6]: #reading our dataset
data=pd.read_csv(r"C:\Users\devis\OneDrive\Desktop\python data_
↳analysis\election_results_2024.csv")
```

```
[8]: data.head()
```

```
[8]:
```

	Constituency	Const. No.	Leading Candidate \
0	AJMER	13	BHAGIRATH CHOUDHARY
1	ALWAR	8	BHUPENDER YADAV
2	AMBALA	1	VARUN CHAUDHRY
3	ANANTNAG-RAJOURI	3	MIAN ALTAF AHMAD
4	ARAKKONAM	7	S JAGATHRATCHAKAN

	Leading Party	Trailing Candidate \
0	Bharatiya Janata Party	RAMCHANDRA CHOUDHARY
1	Bharatiya Janata Party	LALIT YADAV
2	Indian National Congress	BANTO KATARIA
3	Jammu & Kashmir National Conference	MEHBOOBA MUFTI
4	Dravida Munnetra Kazhagam	L VIJAYAN

	Trailing Party	Margin	Status
0	Indian National Congress	329991	Result Declared
1	Indian National Congress	48282	Result Declared
2	Bharatiya Janata Party	49036	Result Declared
3	Jammu & Kashmir Peoples Democratic Party	281794	Result Declared
4	All India Anna Dravida Munnetra Kazhagam	306559	Result Declared

```
[10]: #Party with highest and lowest margin of victory
```

```
[12]: party_votes=data.groupby('Leading Party')['Margin'].sum().
↳sort_values(ascending=False)
```

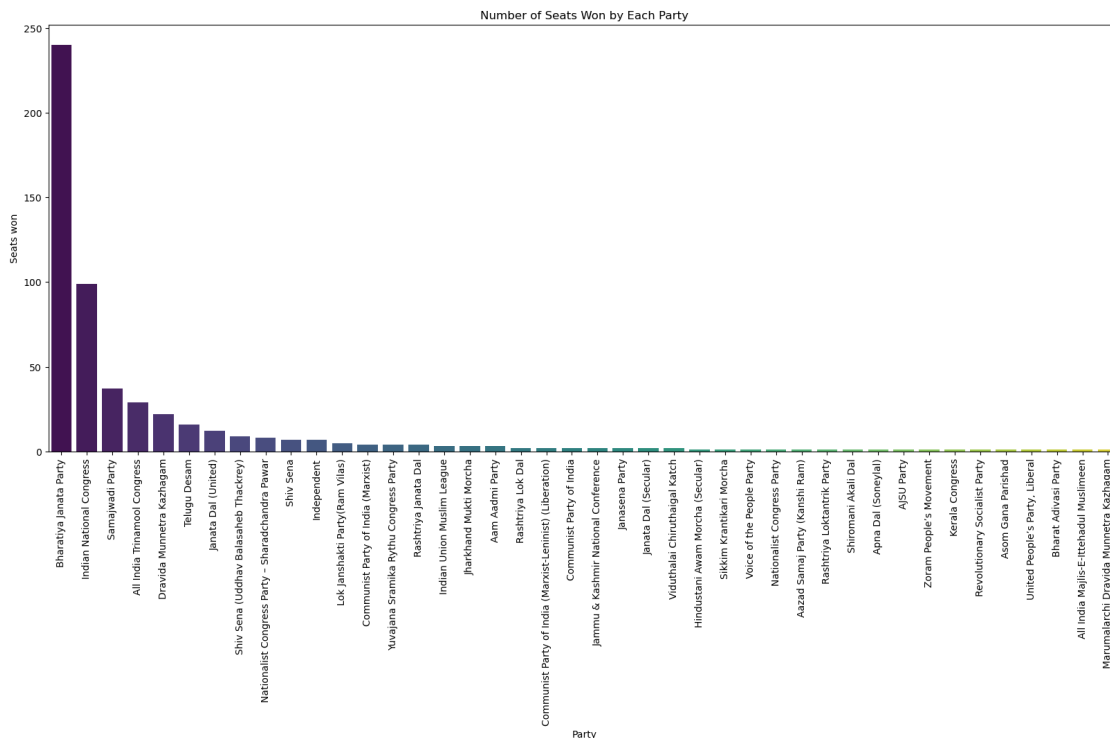
```
data['Margin']=pd.to_numeric(data['Margin'],errors='coerce')
```

```
#Party with highest and lowest margin of victory
highest_margin=data.loc[data['Margin'].idxmax()]
lowest_margin=data.loc[data['Margin'].idxmin()]
```

```
[14]: #Plot number of Seats won by each party
leading_party_highest_votes=party_votes.idxmax()
leading_party_lowest_votes=party_votes.idxmin()

#Number of seats won by each party
seats_won=data['Leading Party'].value_counts()

#Plot number of seats won by each party
plt.figure(figsize=(20,8))
sns.barplot(x=seats_won.index,y=seats_won.values,palette='viridis')
plt.title('Number of Seats Won by Each Party')
plt.xlabel('Party')
plt.ylabel('Seats won')
plt.xticks(rotation=90)
plt.show()
```



```
[16]: #Get the votes for Rahul Gandhi,Narendra Modi,and Amit Shah
```

```

[18]: rahul_entries=data[data['Leading Candidate'] == 'RAHUL GANDHI']
      modi_entries=data[data['Leading Candidate'] == 'NARENDRA MODI']
      amit_entries=data[data['Leading Candidate'] == 'AMIT SHAH']

      #Get the votes for Rahul Gandhi,Narendra Modi,and Amit Shah
      rahul_votes=rahul_entries['Margin'].values
      modi_votes=modi_entries['Margin'].values[0] if not modi_entries.empty else 0
      amit_votes=amit_entries['Margin'].values[0] if not amit_entries.empty else 0

      #Get the original constituency names for Rahul Gandhi
      rahul_constituencies=list(rahul_entries['Constituency'])

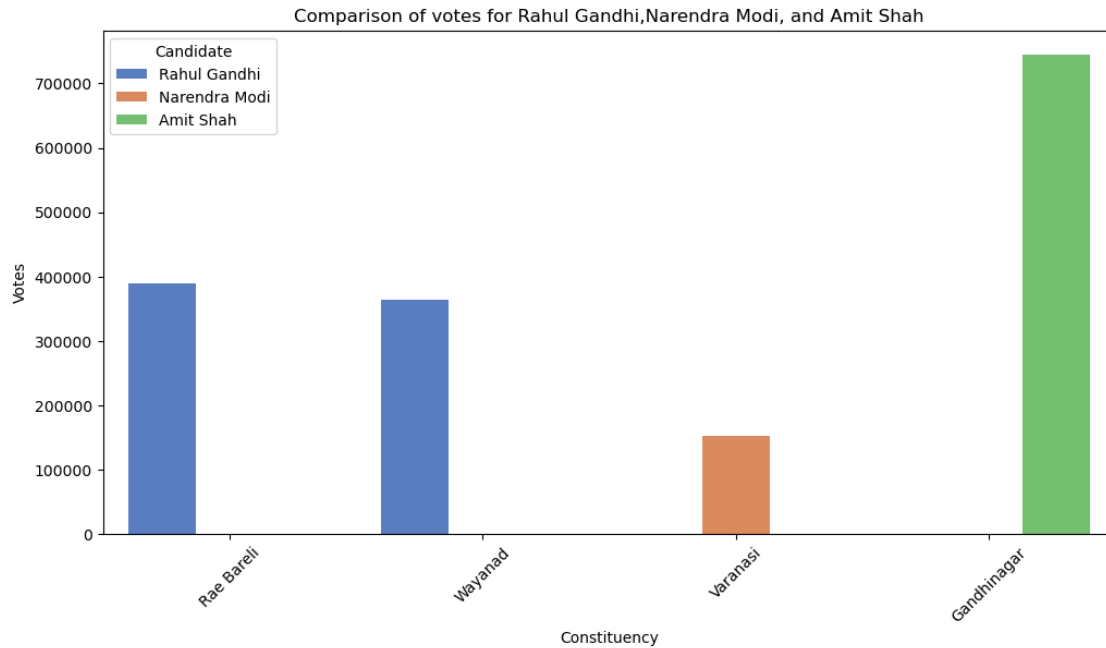
      #Get the original constituency name for Narendra Modi
      modi_constituency=modi_entries['Constituency'].values[0] if not modi_entries.
      ↪empty else "Mo"

      ##Get the original constituency name for Amit Shah
      amit_constituency=amit_entries['Constituency'].values[0] if not amit_entries.
      ↪empty else "Am"

      #Combine the data
      data_to_plot=pd.DataFrame({
          'Candidate':['Rahul Gandhi']*len(rahul_votes) +['Narendra Modi','Amit_
      ↪Shah'],
          'Constituency': rahul_constituencies +_
      ↪[modi_constituency,amit_constituency],
          'Votes': list(rahul_votes) + [modi_votes,amit_votes]
      })

      #Plot the comparison
      plt.figure(figsize=(12,6))
      sns.
      ↪barplot(data=data_to_plot,x='Constituency',y='Votes',hue='Candidate',palette='muted')
      plt.title("Comparison of votes for Rahul Gandhi,Narendra Modi, and Amit Shah")
      plt.xlabel('Constituency')
      plt.ylabel('Votes')
      plt.xticks(rotation=45)
      plt.show()

```

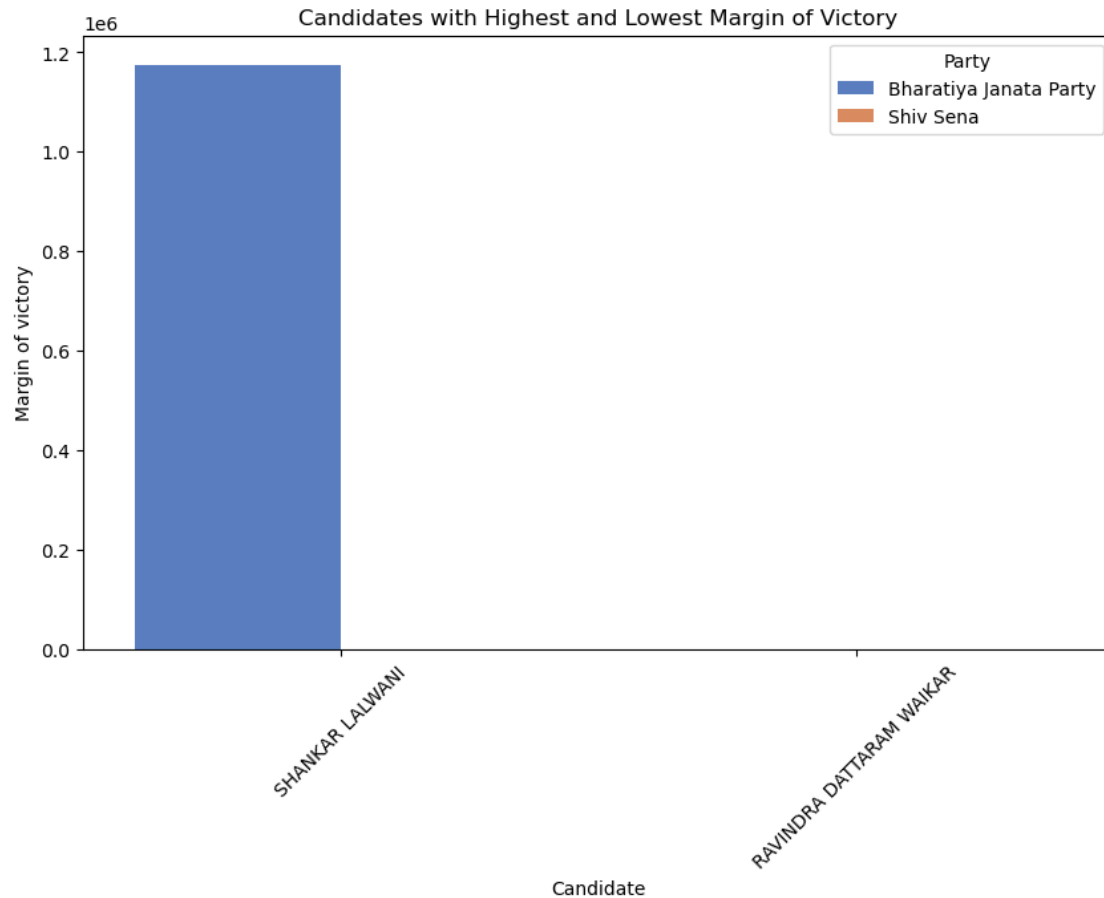


[20]: *#Highest and Lowest Victory Candidate*

```
[22]: highest_margin_entry=data.loc[data['Margin'].idxmax()]
lowest_margin_entry=data.loc[data['Margin'].idxmin()]

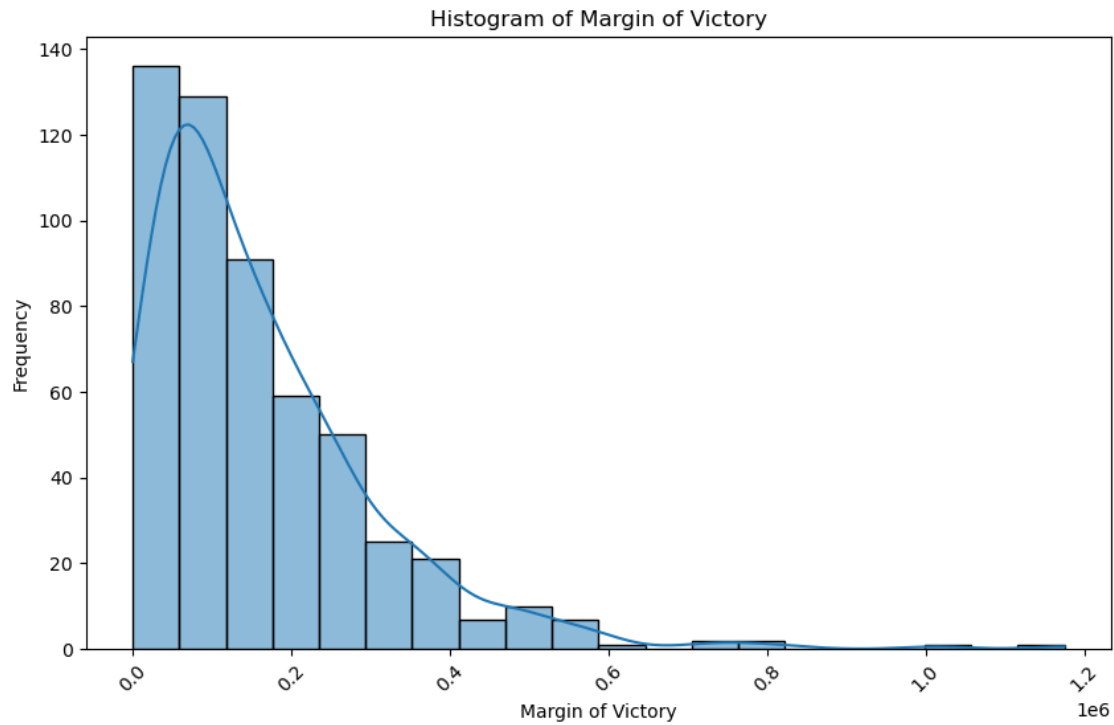
#Combine the data
data_to_plot=pd.DataFrame({
    'Candidate':[highest_margin_entry['Leading_Candidate'],
    ↪Candidate'],lowest_margin_entry['Leading Candidate']],
    'Party':[highest_margin_entry['Leading Party'],lowest_margin_entry['Leading_Party']],
    ↪Party']],
    'Margin':[highest_margin_entry['Margin'],lowest_margin_entry['Margin']]
})

#Plot the comparison
plt.figure(figsize=(10,6))
sns.
    ↪barplot(data=data_to_plot,x='Candidate',y='Margin',hue='Party',palette='muted')
plt.title("Candidates with Highest and Lowest Margin of Victory")
plt.xlabel('Candidate')
plt.ylabel('Margin of victory')
plt.xticks(rotation=45)
plt.show()
```



```
[24]: plt.figure(figsize=(10,6))
sns.histplot(data['Margin'],bins=20,kde=True)
plt.title("Histogram of Margin of Victory")
plt.xlabel('Margin of Victory')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```

C:\Users\devis\anaconda3\Lib\site-packages\seaborn_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
with pd.option_context('mode.use_inf_as_na', True):



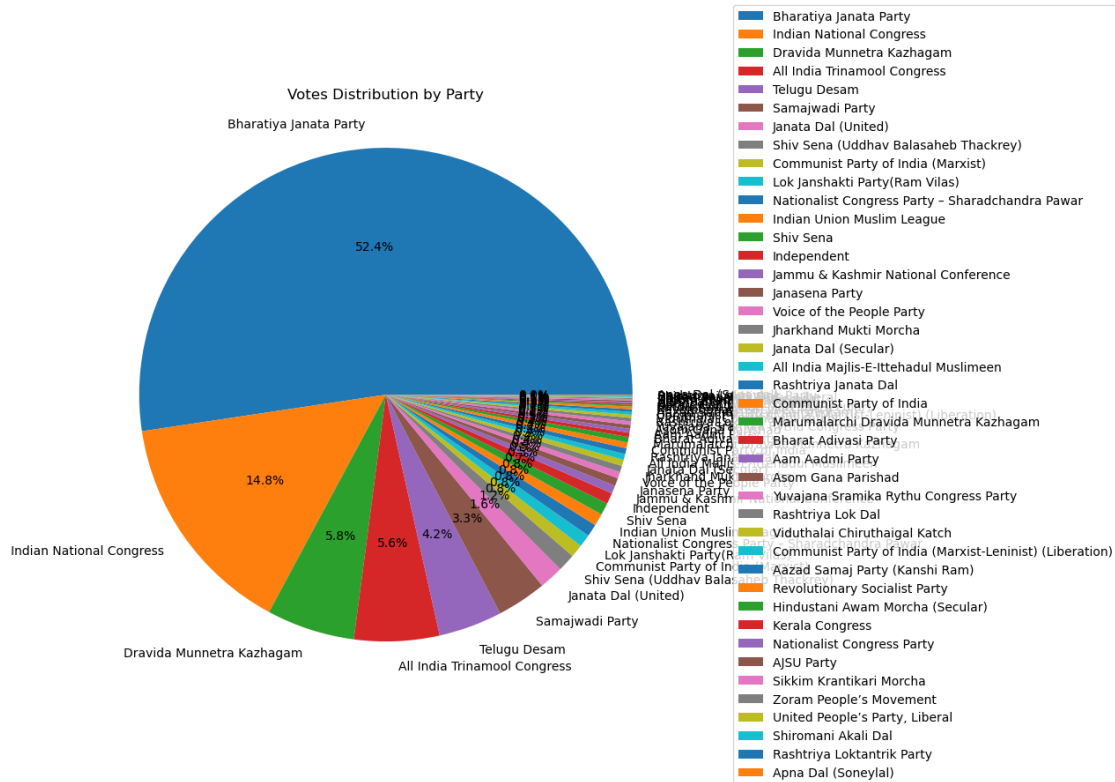
[26]: *#VOTES DISTRIBUTION BY PARTY*

```
[28]: party_votes=data.groupby('Leading Party')['Margin'].sum().
      ↪sort_values(ascending=False)

      #Plot pie chart
      plt.figure(figsize=(10,8))
      plt.pie(party_votes, labels=party_votes.index, autopct='%1.1f%%')
      plt.title('Votes Distribution by Party', pad=20)
      plt.axis('equal')

      plt.legend(labels=party_votes.index, loc='center left', bbox_to_anchor=(1, 0.
      ↪5), fontsize='medium')

      plt.show()
```

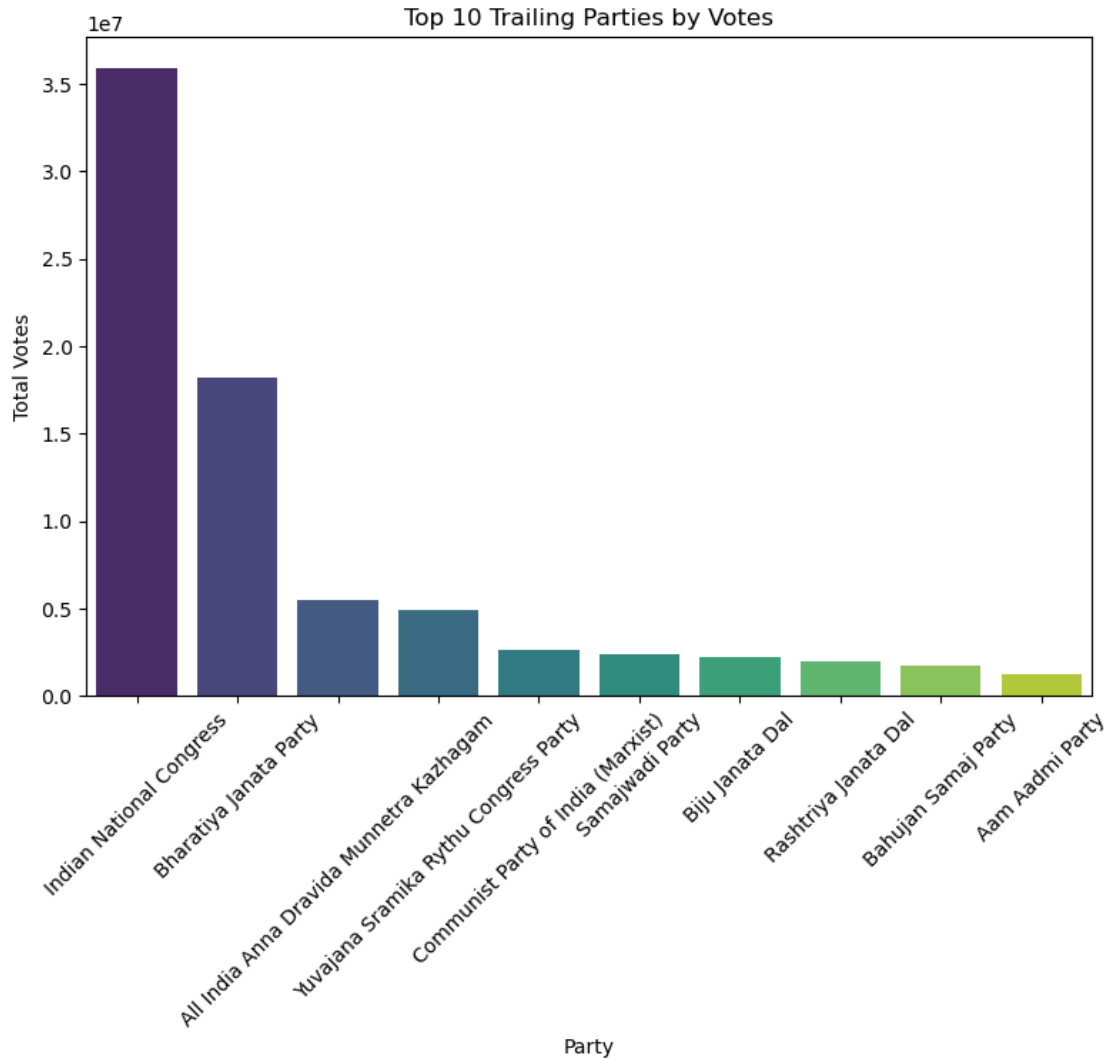


[30]: #Top 10 Trailing Party by Vote

```
[32]: trailing_party_votes=data.groupby('Trailing Party')['Margin'].sum().
      ↪sort_values(ascending=False)
trailing_party_seats=data['Trailing Party'].value_counts()
```

```
[34]: plt.figure(figsize=(20,6))

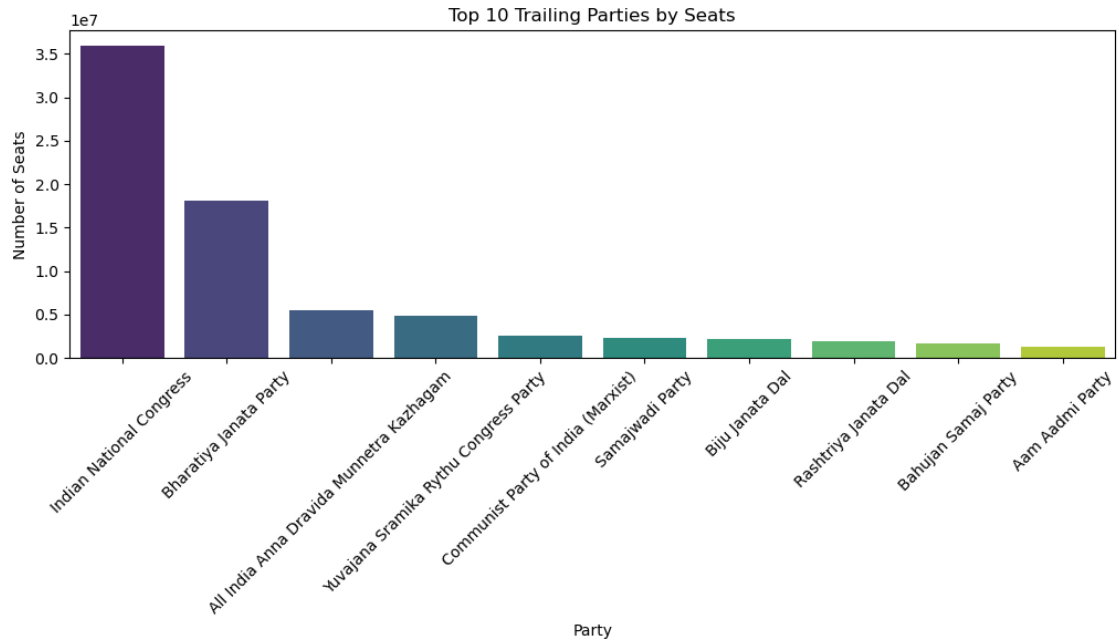
#Plot votes distribution by trailing party
plt.subplot(1,2,1)
sns.barplot(x=trailing_party_votes.index[:10],y=trailing_party_votes.values[:
      ↪10],palette='viridis')
plt.title("Top 10 Trailing Parties by Votes")
plt.xlabel('Party')
plt.ylabel('Total Votes')
plt.xticks(rotation=45)
plt.show()
```



```
[36]: #Top 10 Trailing Party by Seat
```

```
[38]: plt.figure(figsize=(20,6))
plt.subplot(1,2,2)
sns.barplot(x=trailing_party_votes.index[:10],y=trailing_party_votes.values[:
↪10],palette='viridis')
plt.title("Top 10 Trailing Parties by Seats")
plt.xlabel('Party')
plt.ylabel('Number of Seats')
plt.xticks(rotation=45)

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

[]: