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

In [ ]:
```
df = pd.read_csv('cleaned_data.csv')
```

In [ ]:
```
df
```

Out[ ]:
```
                           state               constituency                                            image   result
   0   Andaman & Nicobar Islands  Andaman & Nicobar Islands  https://results.eci.gov.in/uploads4/candprofil...      won
   1   Andaman & Nicobar Islands  Andaman & Nicobar Islands  https://results.eci.gov.in/uploads4/candprofil...     lost
   2   Andaman & Nicobar Islands  Andaman & Nicobar Islands  https://results.eci.gov.in/uploads4/candprofil...     lost
   3   Andaman & Nicobar Islands  Andaman & Nicobar Islands  https://results.eci.gov.in/uploads4/candprofil...     lost
   4   Andaman & Nicobar Islands  Andaman & Nicobar Islands  https://results.eci.gov.in/uploads4/candprofil...     lost
 ...                         ...                        ...                                              ...      ...
8897                 West Bengal                 Coochbehar  https://results.eci.gov.in/uploads4/candprofil...     lost
8898                 West Bengal                 Coochbehar  https://results.eci.gov.in/uploads4/candprofil...     lost
8899                 West Bengal                 Coochbehar  https://results.eci.gov.in/uploads4/candprofil...     lost
8900                 West Bengal                    Bolpur   https://results.eci.gov.in/uploads4/candprofil...     lost
8901                 West Bengal                   Uluberia                                     img/nota.jpg      NaN

8902 rows × 8 columns
```

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

Out[ ]:

```
state             0
constituency      0
image             0
result          542
votes             0
margin            0
name              0
party             0
dtype: int64
```

In [ ]:

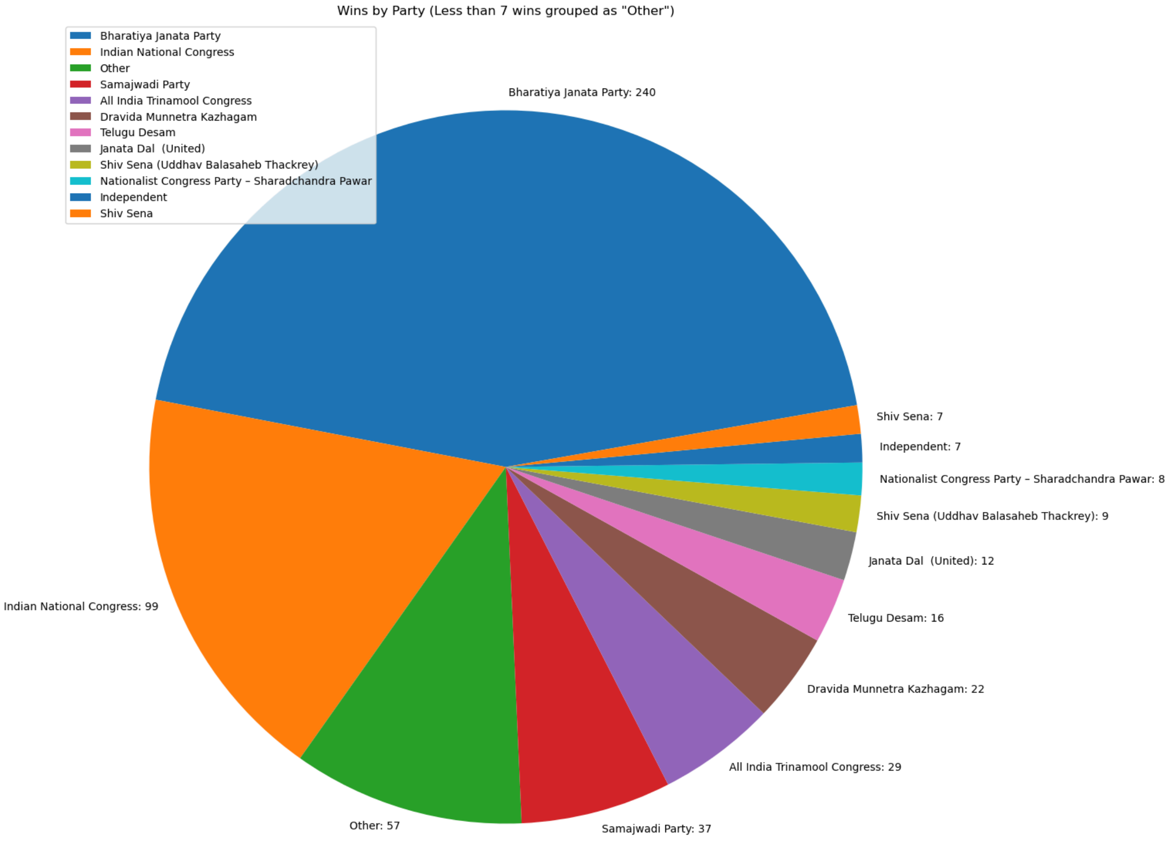
```
df.describe()
```

Out[ ]:

	votes
count	8.902000e+03
mean	7.249646e+04
std	1.798988e+05
min	0.000000e+00
25%	1.094250e+03
50%	2.781000e+03
75%	9.759500e+03
max	1.471885e+06

In [ ]:

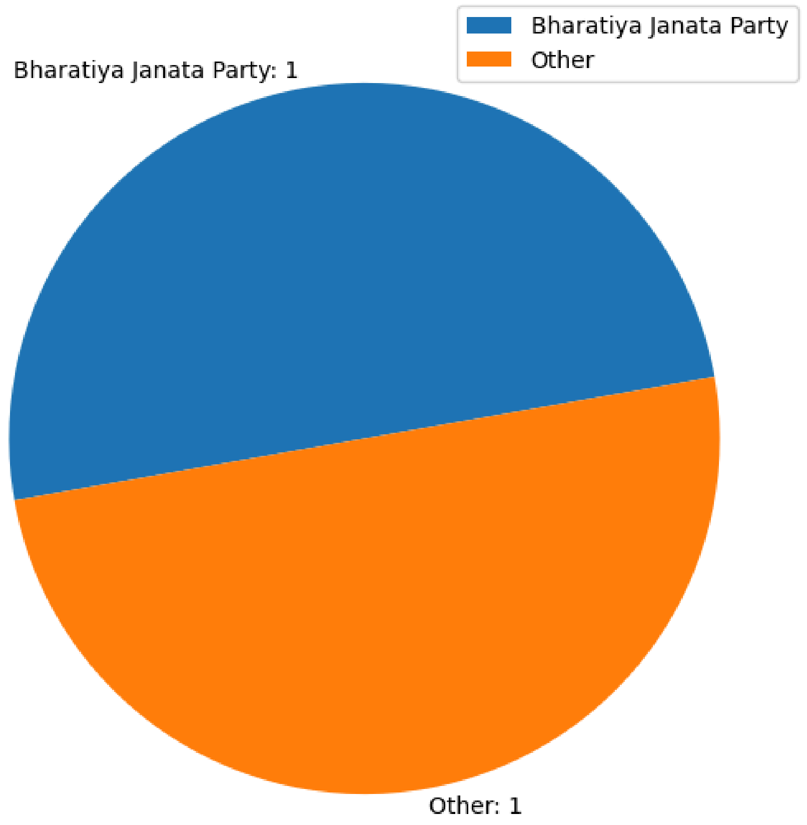
```
df_won = df[df['result'] == 'won']
final_counts = df_won['party'].value_counts()

# Group parties with less than 7 wins as 'Other'
final_counts['Other'] = final_counts[final_counts < 7].sum()
final_counts = final_counts[final_counts >= 7]

# Sort the final counts
final_counts = final_counts.sort_values(ascending=False)



# Construct new labels that include both party names and their counts
labels_with_counts = [f'{label}: {count}' for label, count in zip(final_c

plt.figure(figsize=(20, 15))
plt.pie(final_counts, labels=labels_with_counts, startangle=10, labeldist
plt.title('Wins by Party (Less than 7 wins grouped as "Other")')
plt.legend(final_counts.index, loc='upper left')
plt.show()
```



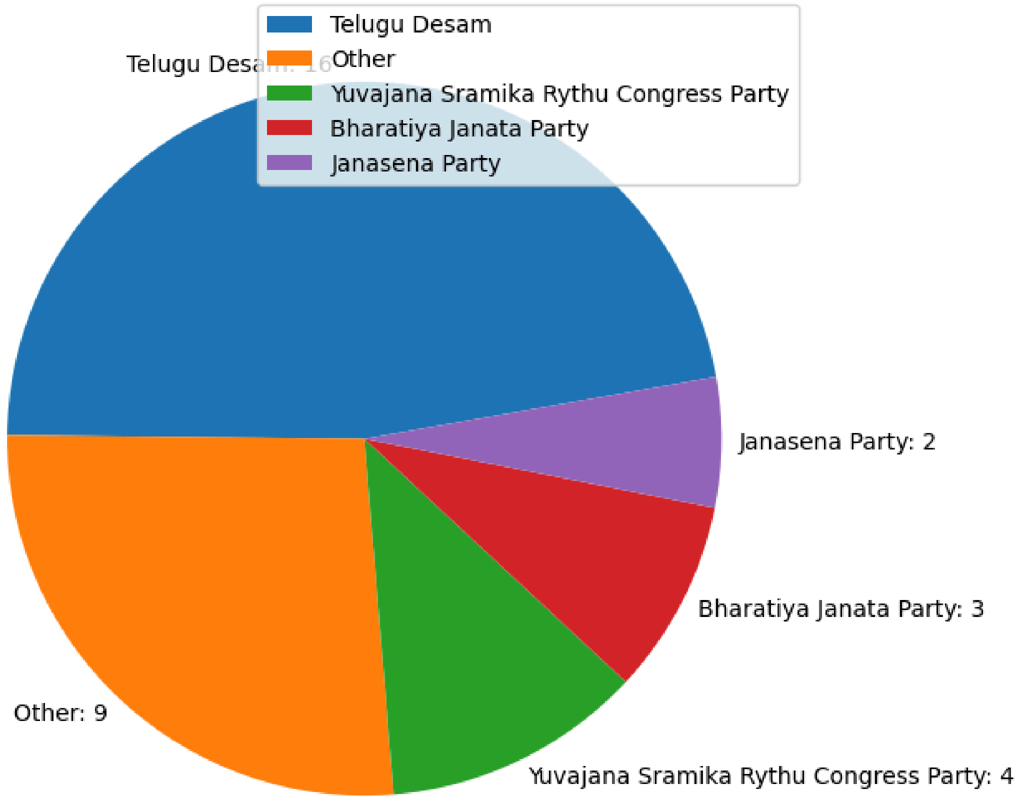
In [ ]:

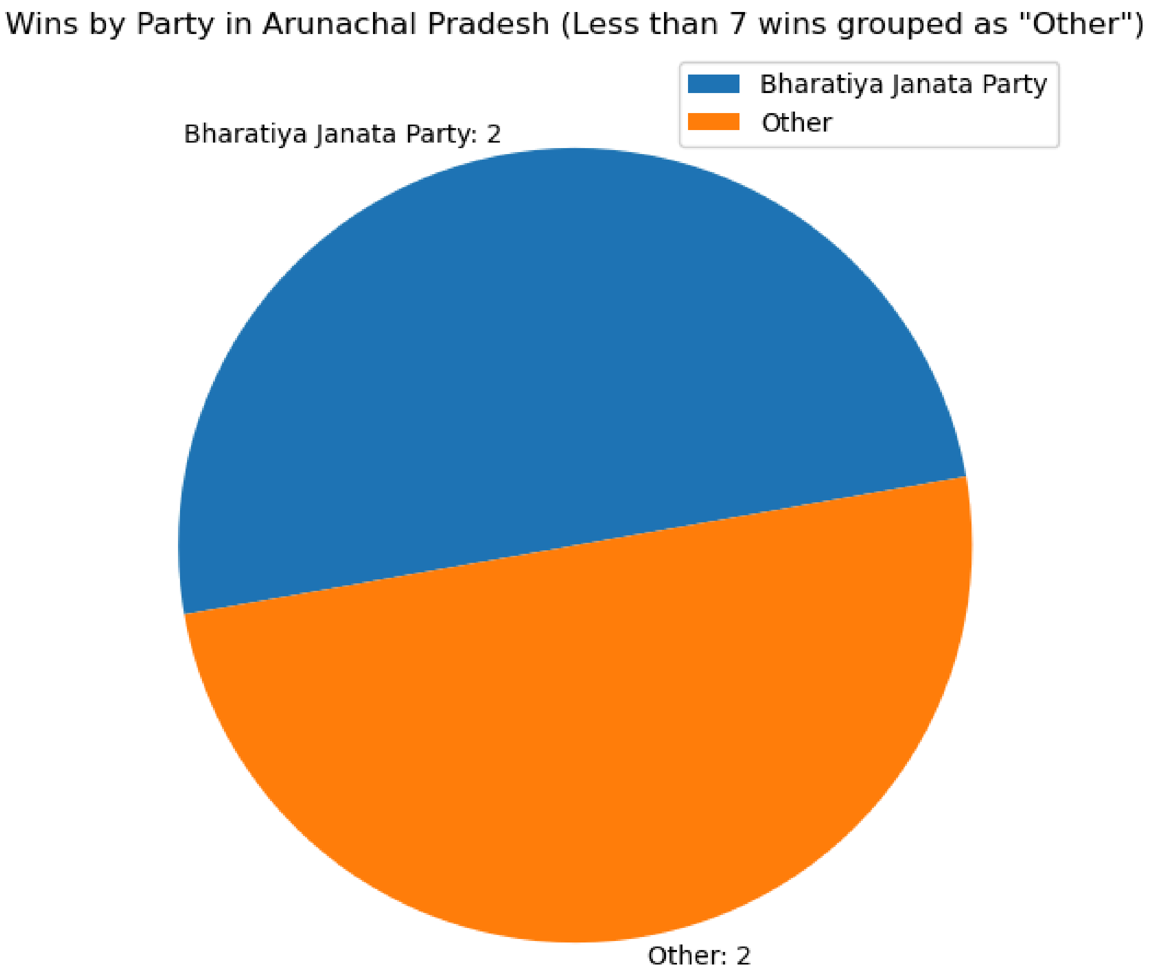
```
for state in df['state'].unique():
    df_state = df[df['state'] == state]
    df_state_won = df_state[df_state['result'] == 'won']
    final_counts = df_state_won['party'].value_counts()

    # Group parties with less than 7 wins as 'Other'
    final_counts['Other'] = final_counts[final_counts < 7].sum()
    # final_counts = final_counts[final_counts >= 7]

    # Sort the final counts
    final_counts = final_counts.sort_values(ascending=False)

    # Construct new labels that include both party names and their counts
    labels_with_counts = [f'{label}: {count}' for label, count in zip(fin

    plt.figure(figsize=(10, 7))
    plt.pie(final_counts, labels=labels_with_counts, startangle=10, label
    plt.title(f'Wins by Party in {state} (Less than 7 wins grouped as "Ot
    plt.legend(final_counts.index, loc='best')
    plt.show()
```

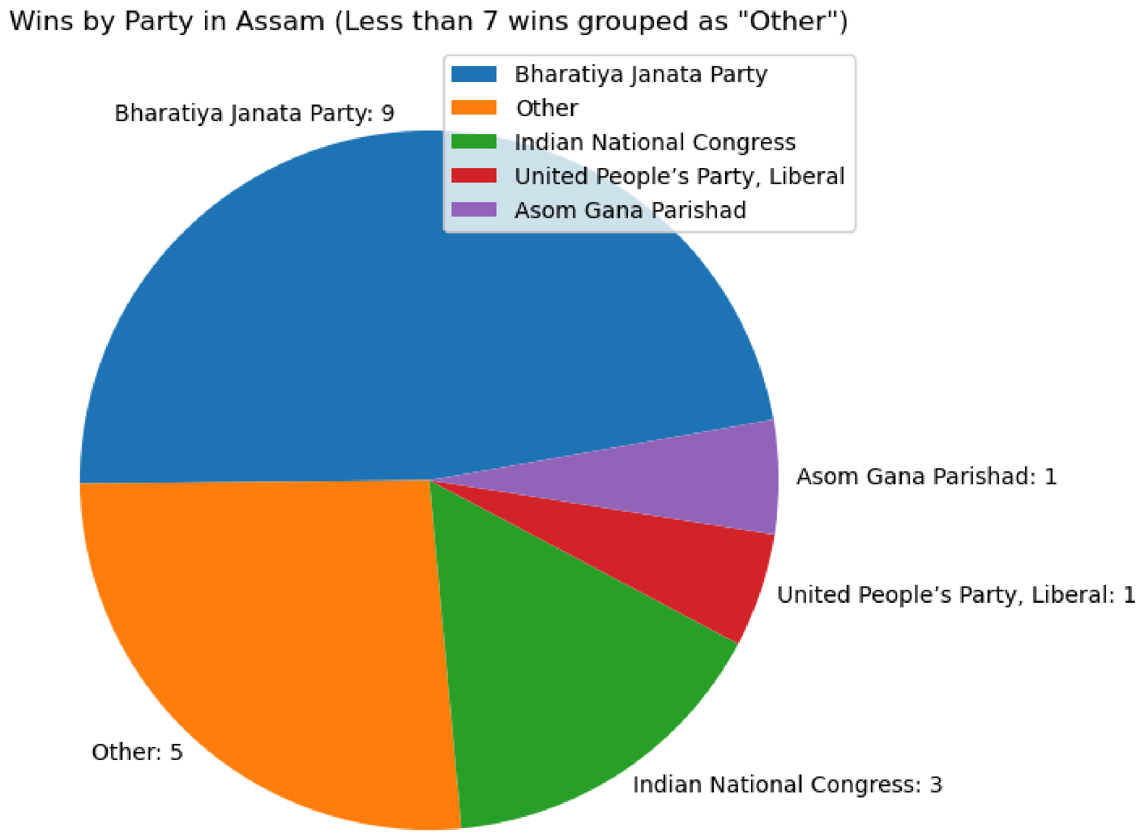
Wins by Party in Andaman & Nicobar Islands (Less than 7 wins grouped as "Other")



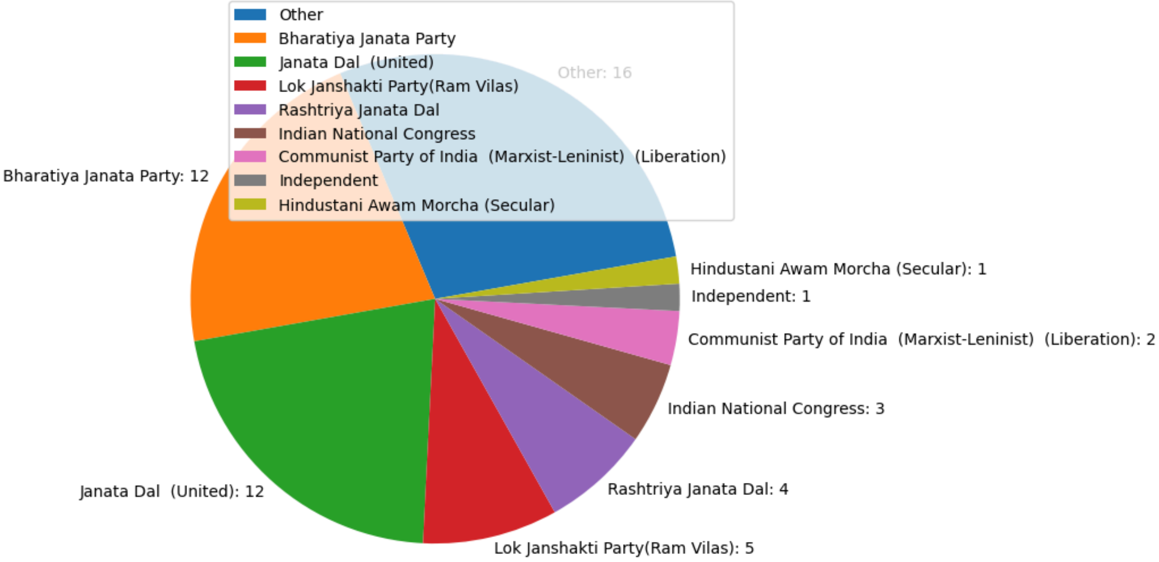
Wins by Party in Andhra Pradesh (Less than 7 wins grouped as "Other")



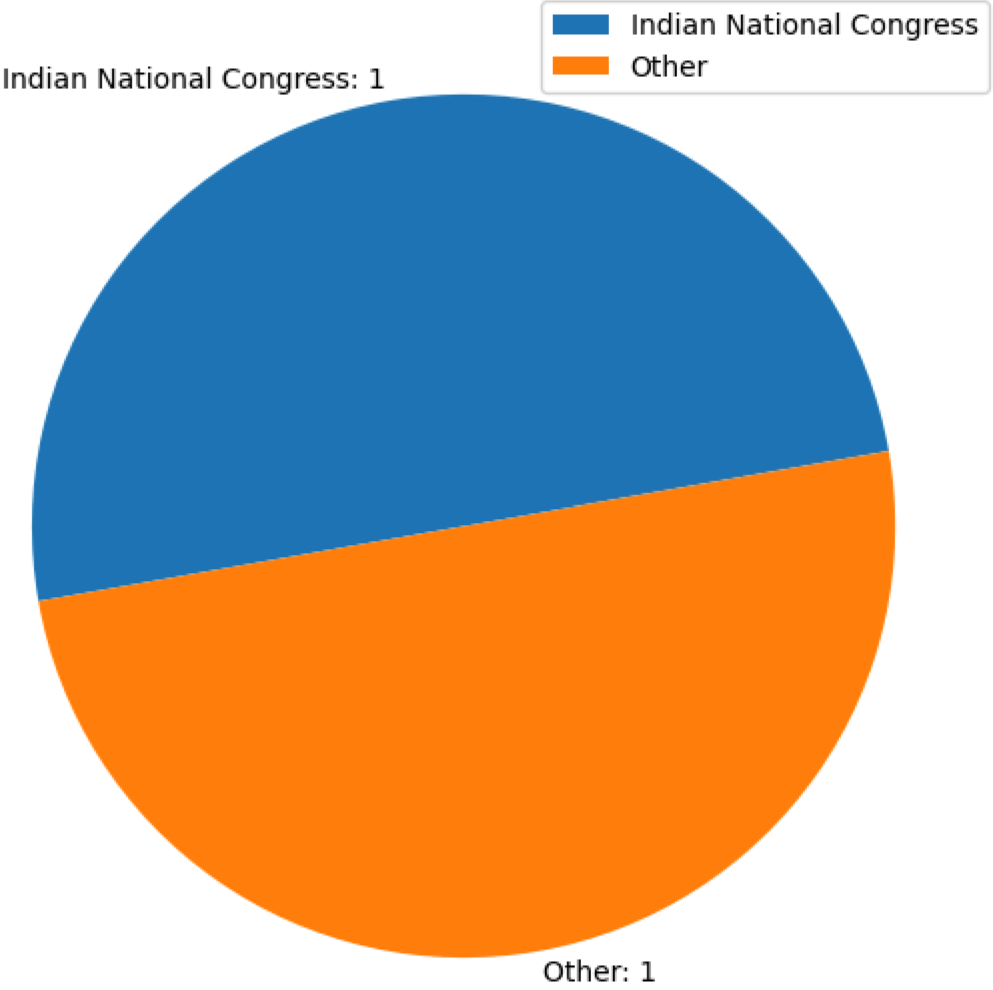

Wins by Party in Arunachal Pradesh (Less than 7 wins grouped as "Other")
Bharatiya Janata Party
Other
Bharatiya Janata Party: 2
Other: 2

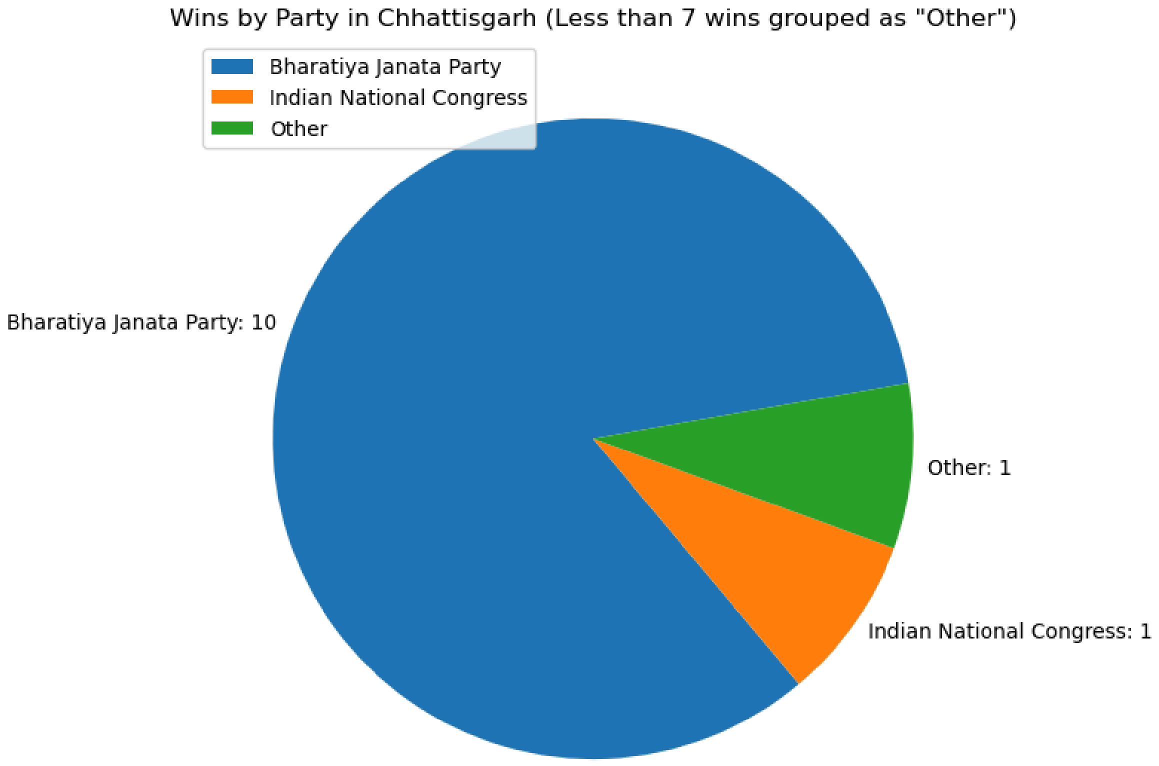

Wins by Party in Assam (Less than 7 wins grouped as "Other")
Bharatiya Janata Party
Other
Indian National Congress
United People’s Party, Liberal
Asom Gana Parishad
Bharatiya Janata Party: 9
Asom Gana Parishad: 1
United People’s Party, Liberal: 1
Indian National Congress: 3
Other: 5

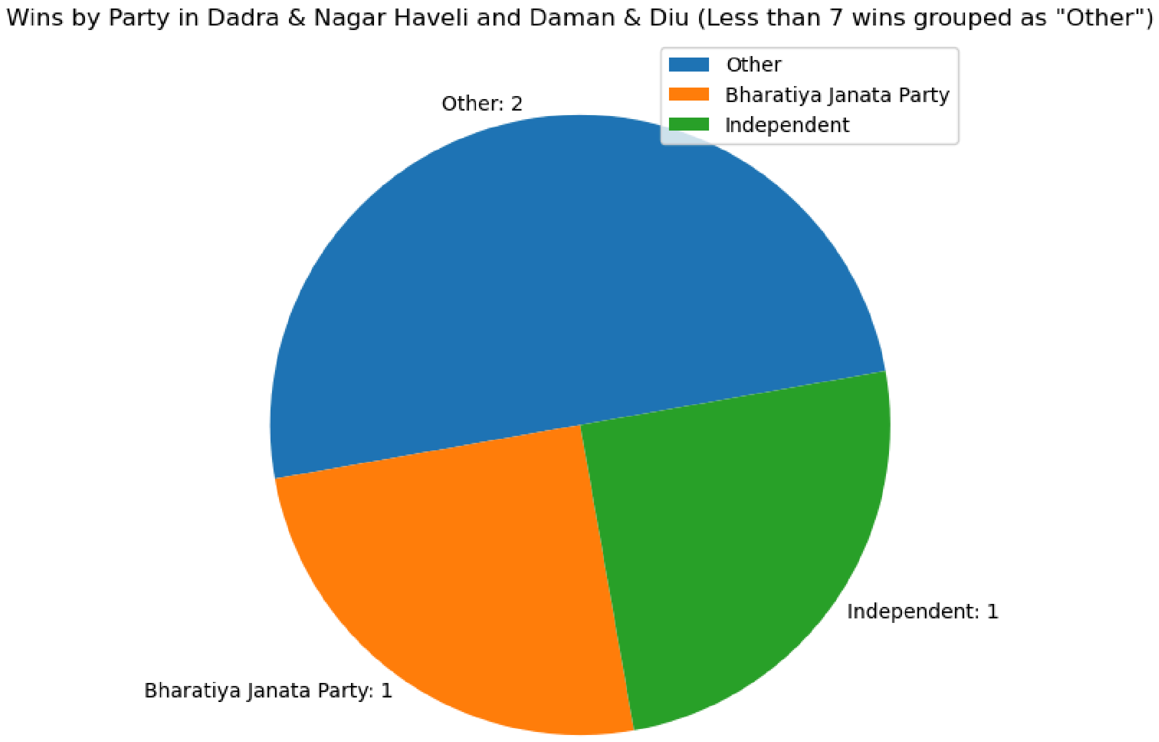
Wins by Party in Bihar (Less than 7 wins grouped as "Other")

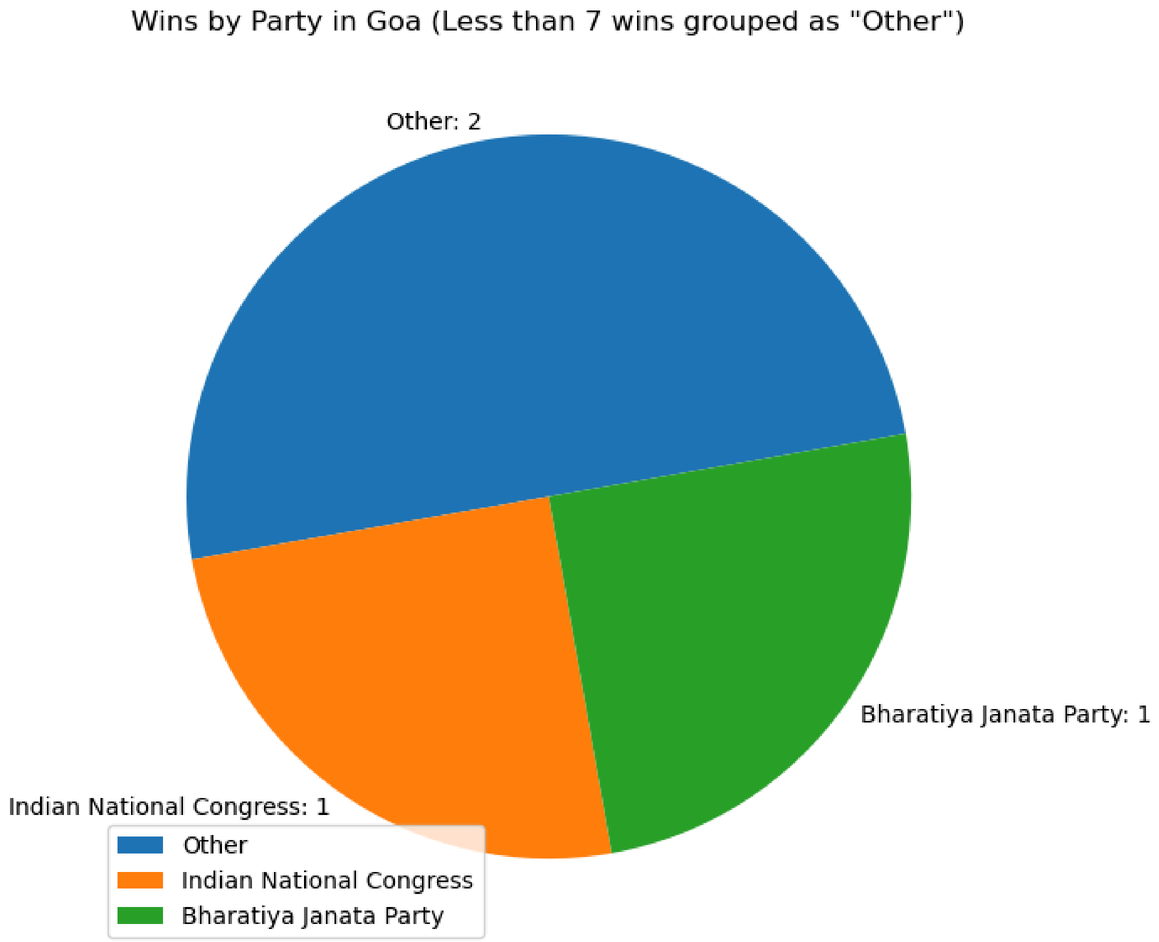


Wins by Party in Chandigarh (Less than 7 wins grouped as "Other")

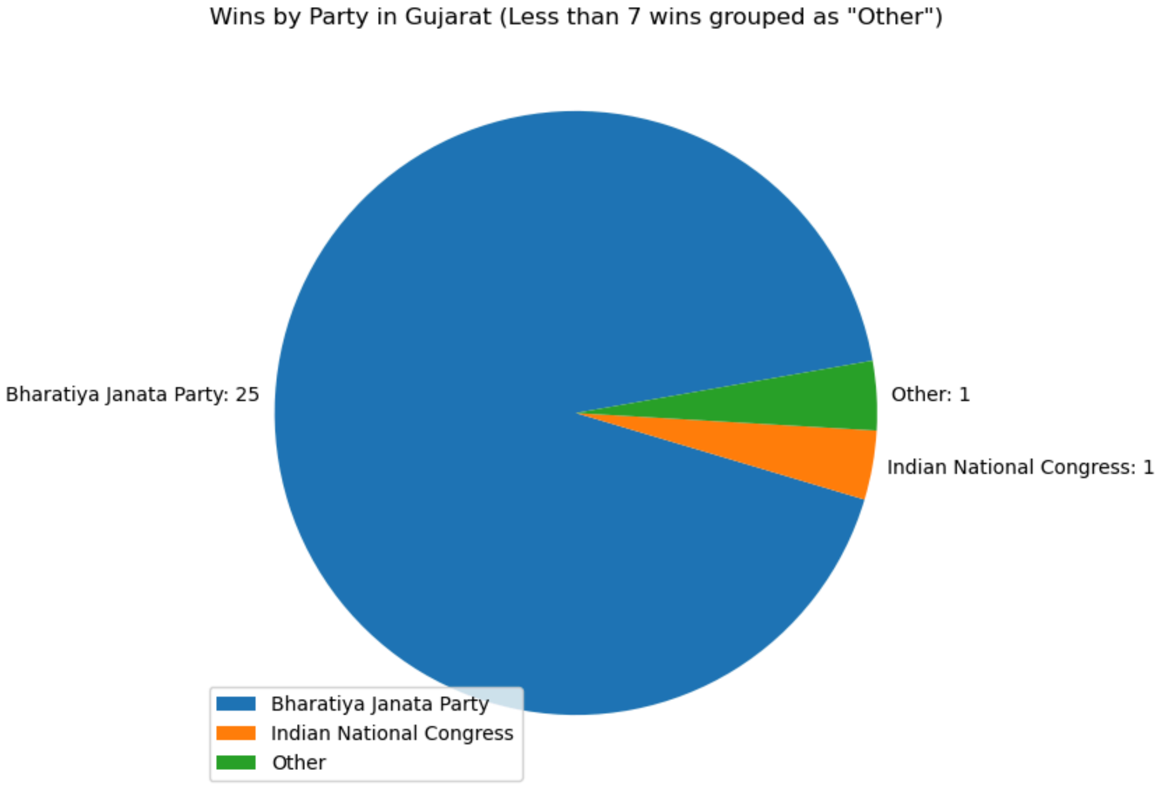



Wins by Party in Chhattisgarh (Less than 7 wins grouped as "Other")
Bharatiya Janata Party
Indian National Congress
Other
Bharatiya Janata Party: 10
Other: 1
Indian National Congress: 1

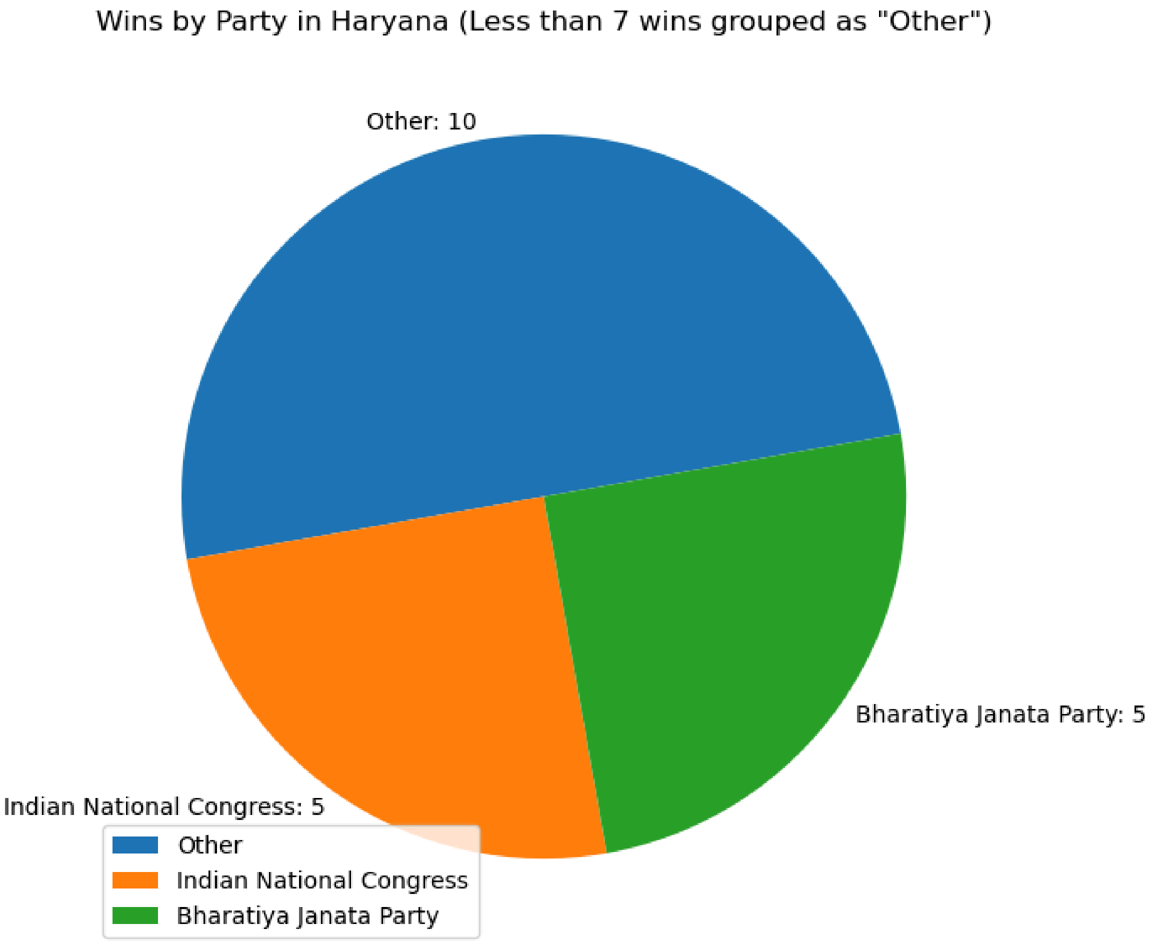

Wins by Party in Dadra & Nagar Haveli and Daman & Diu (Less than 7 wins grouped as "Other")
Other
Bharatiya Janata Party
Independent
Other: 2
Independent: 1
Bharatiya Janata Party: 1



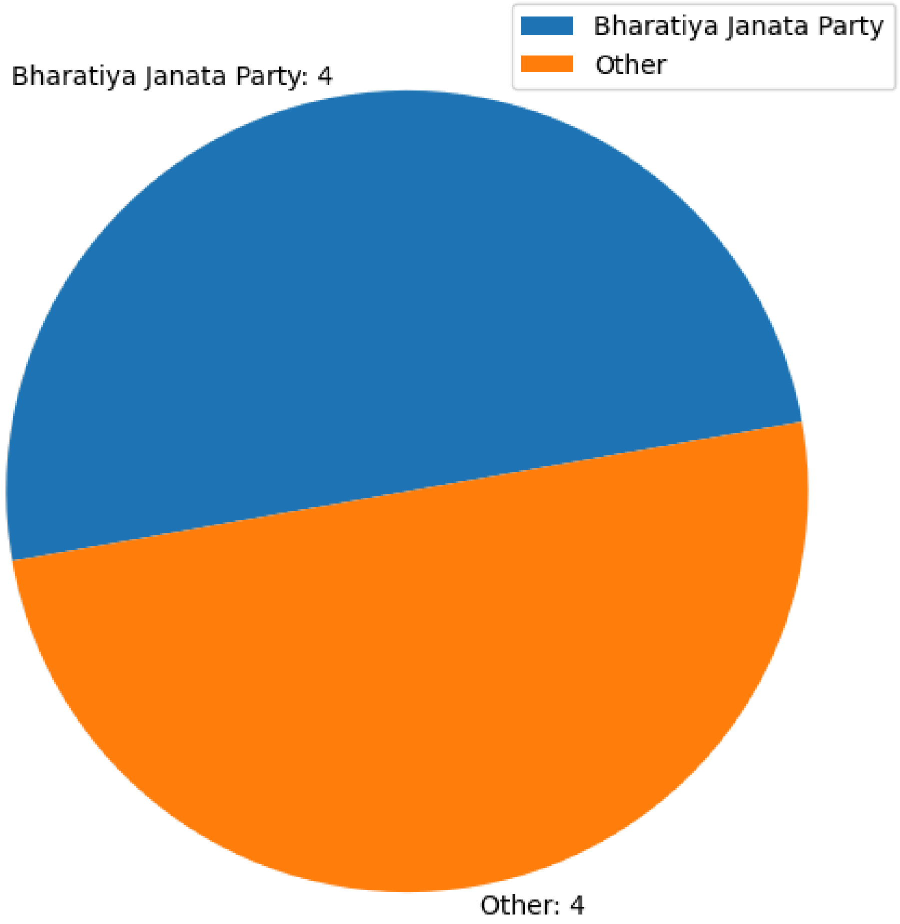
Wins by Party in Goa (Less than 7 wins grouped as "Other")



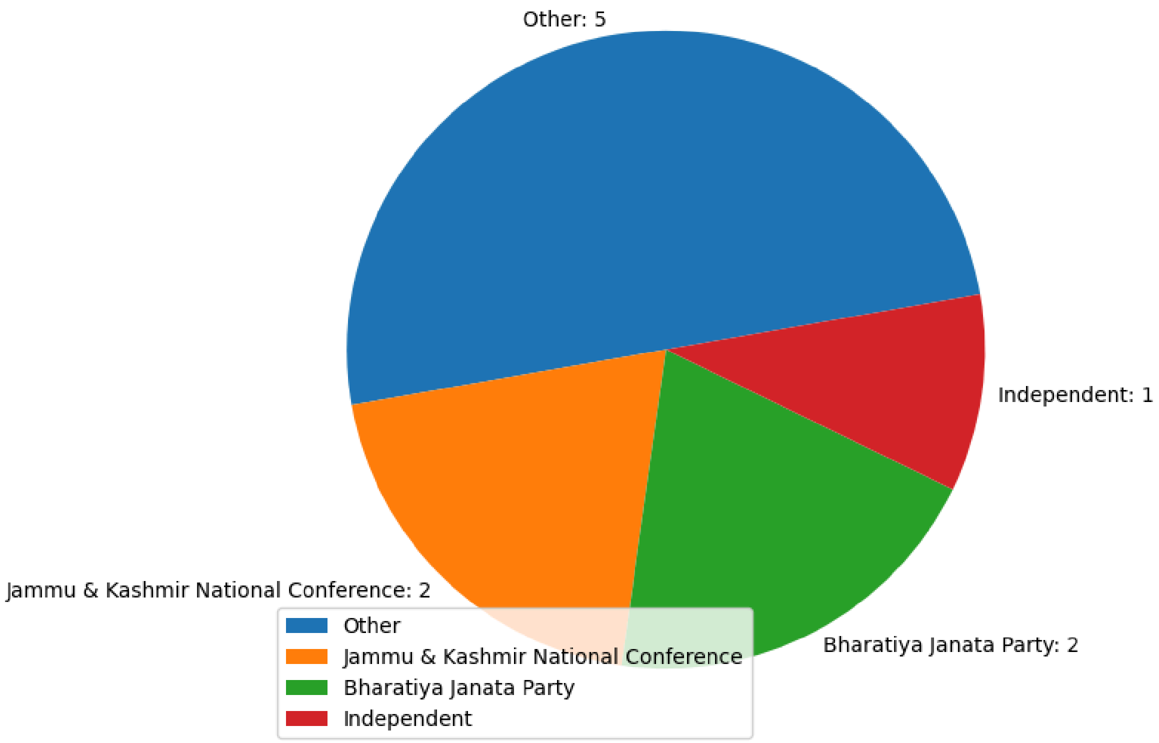
Wins by Party in Gujarat (Less than 7 wins grouped as "Other")

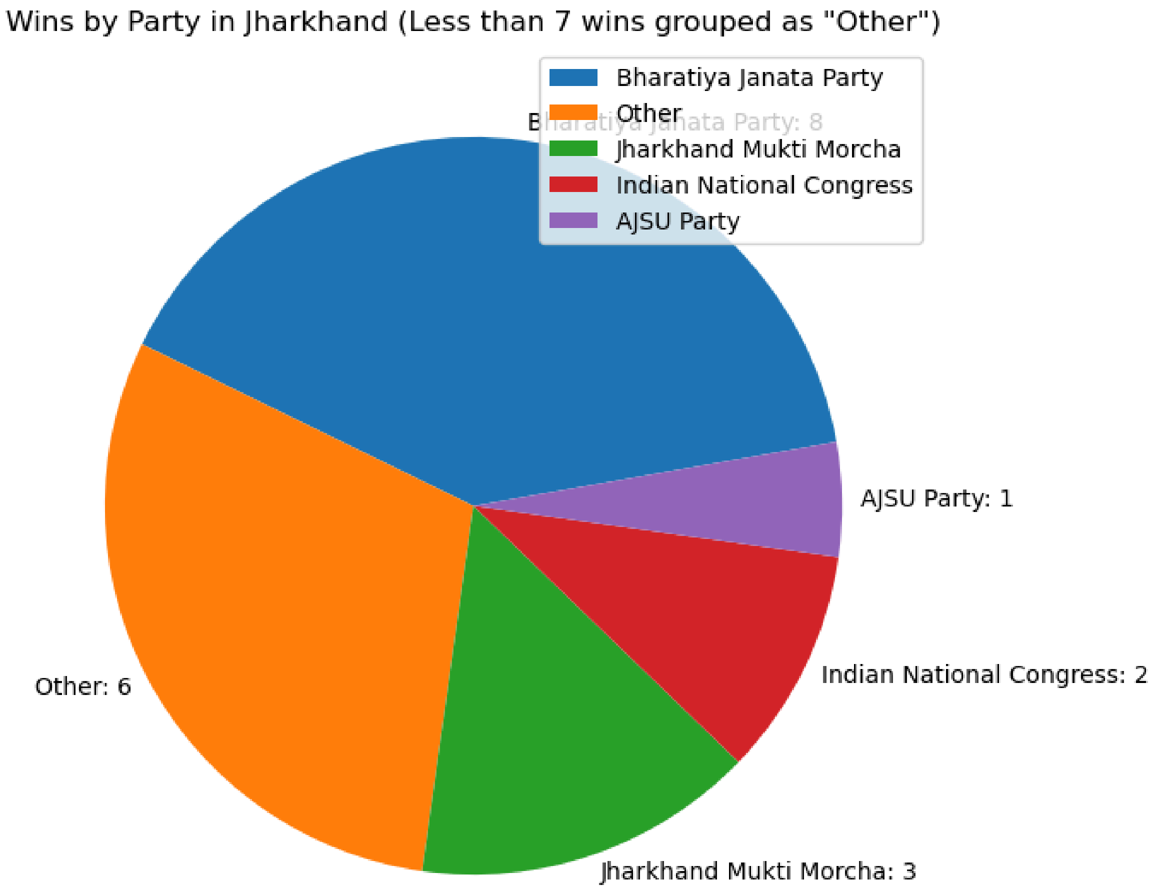

Wins by Party in Haryana (Less than 7 wins grouped as "Other")
Other: 10
Bharatiya Janata Party: 5
Indian National Congress: 5
Other
Indian National Congress
Bharatiya Janata Party

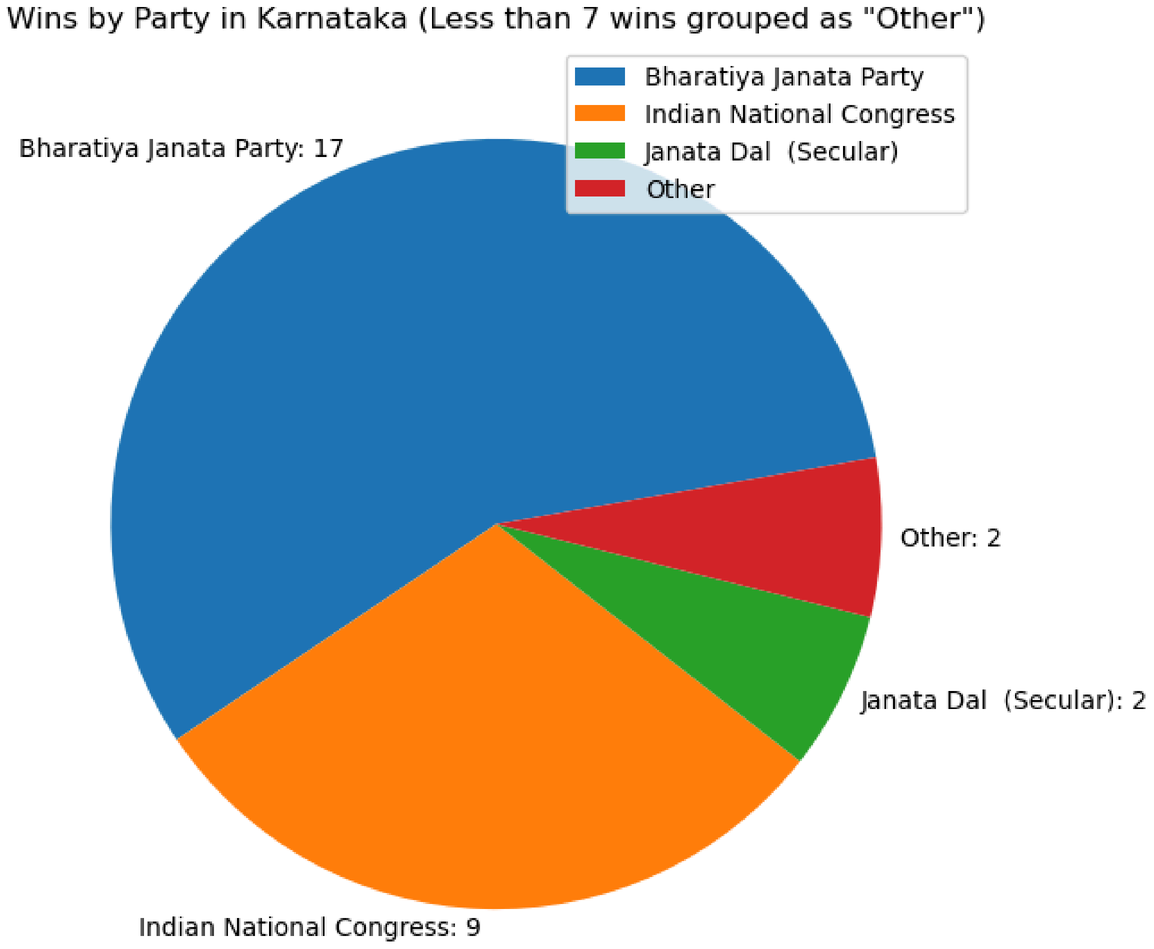
Wins by Party in Himachal Pradesh (Less than 7 wins grouped as "Other")



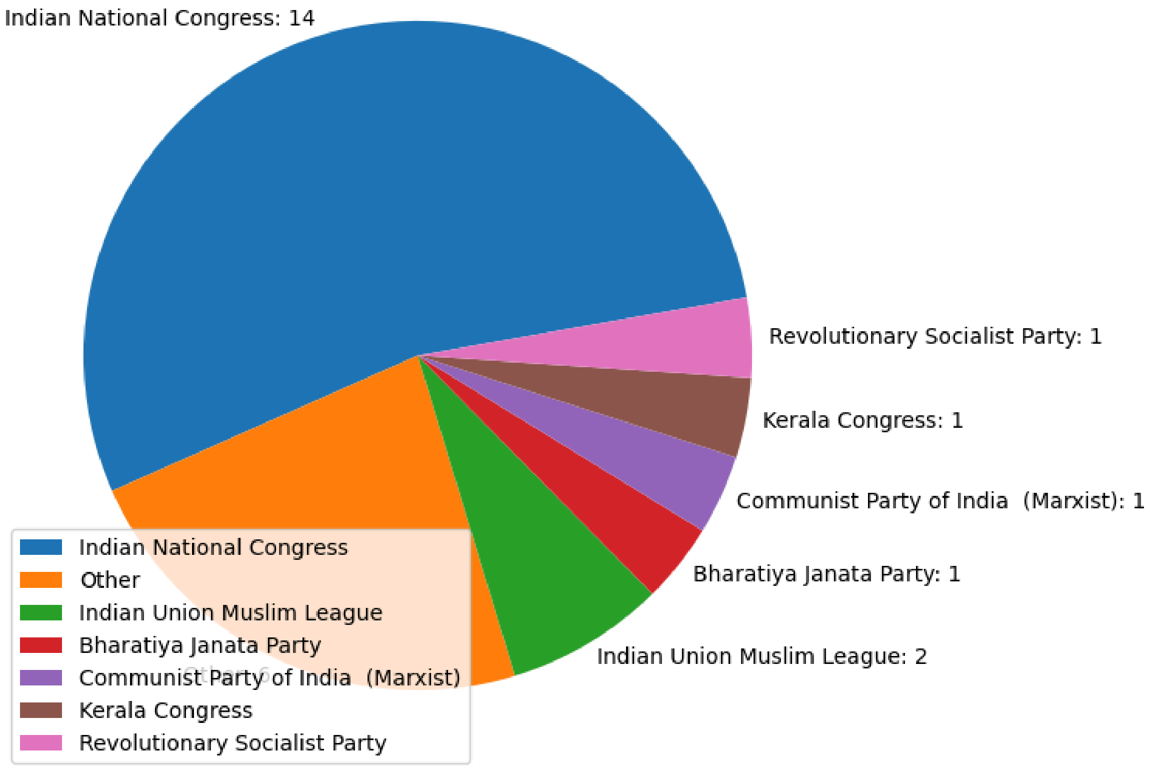
Wins by Party in Jammu and Kashmir (Less than 7 wins grouped as "Other")

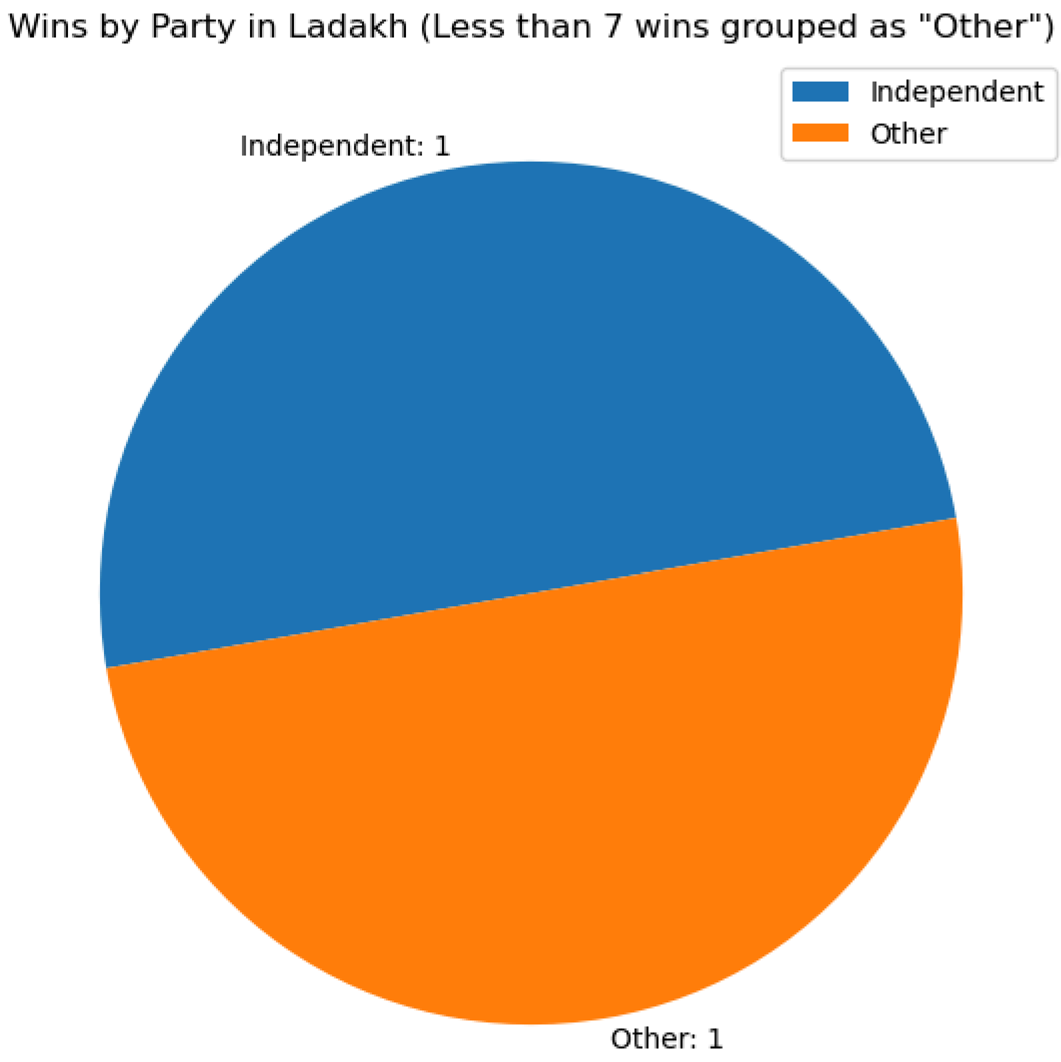


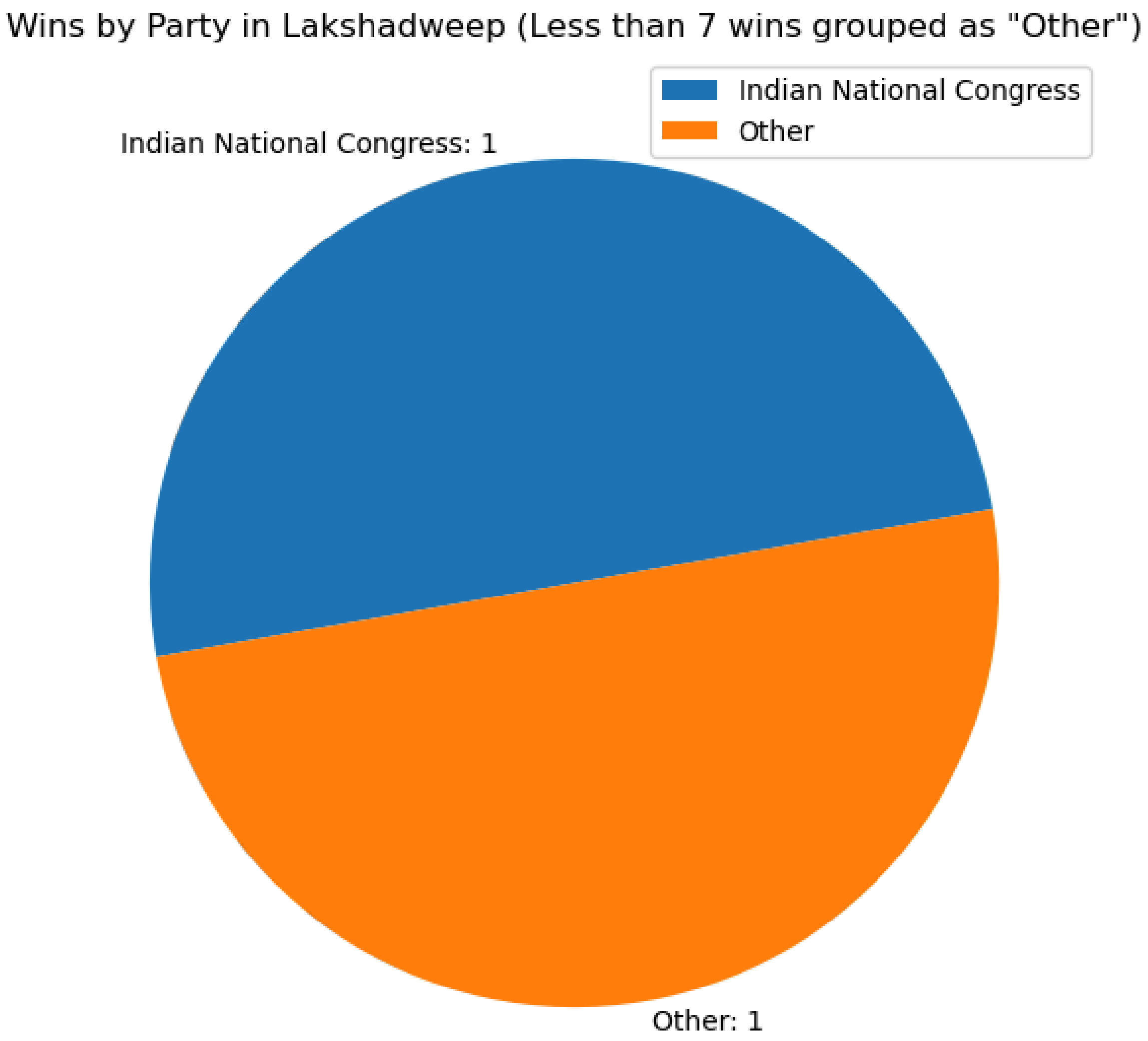

Wins by Party in Jharkhand (Less than 7 wins grouped as "Other")
Bharatiya Janata Party
Other
Jharkhand Mukti Morcha
Indian National Congress
AJSU Party
Bharatiya Janata Party: 8
AJSU Party: 1
Indian National Congress: 2
Jharkhand Mukti Morcha: 3
Other: 6

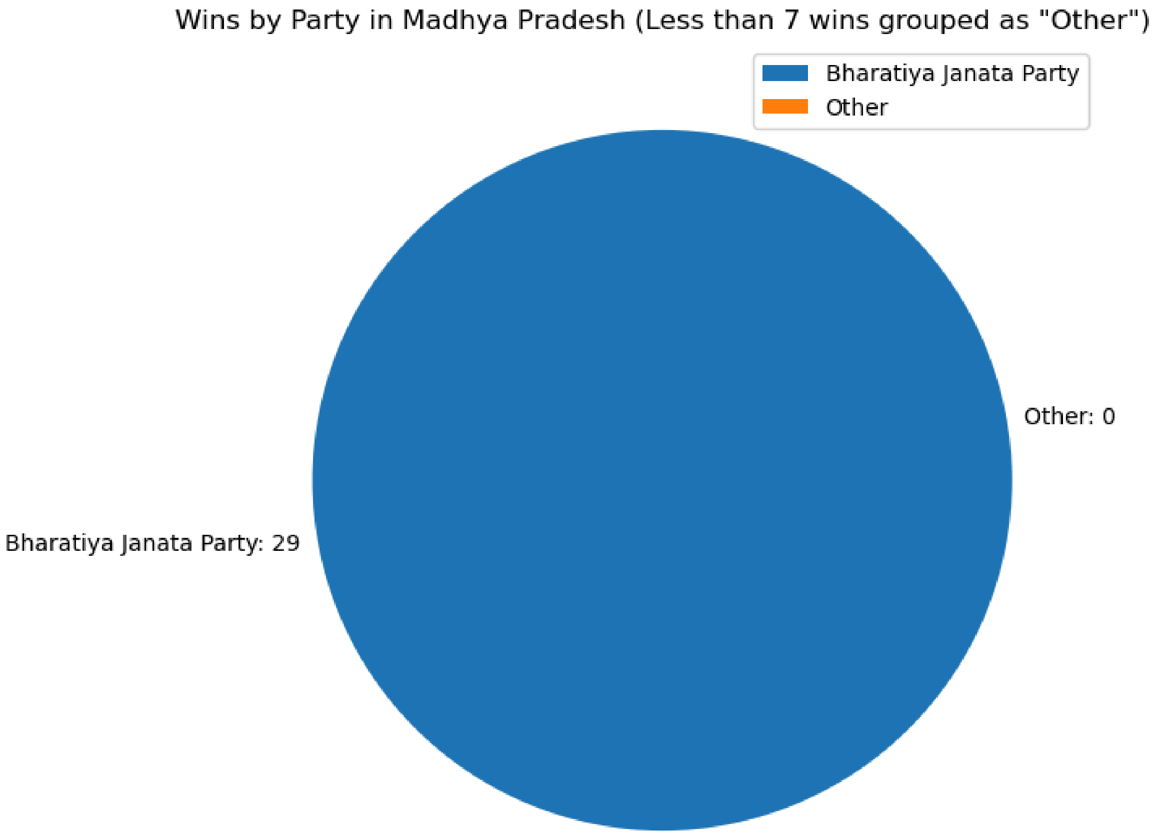

Wins by Party in Karnataka (Less than 7 wins grouped as "Other")
Bharatiya Janata Party
Indian National Congress
Janata Dal (Secular)
Other
Bharatiya Janata Party: 17
Other: 2
Janata Dal (Secular): 2
Indian National Congress: 9

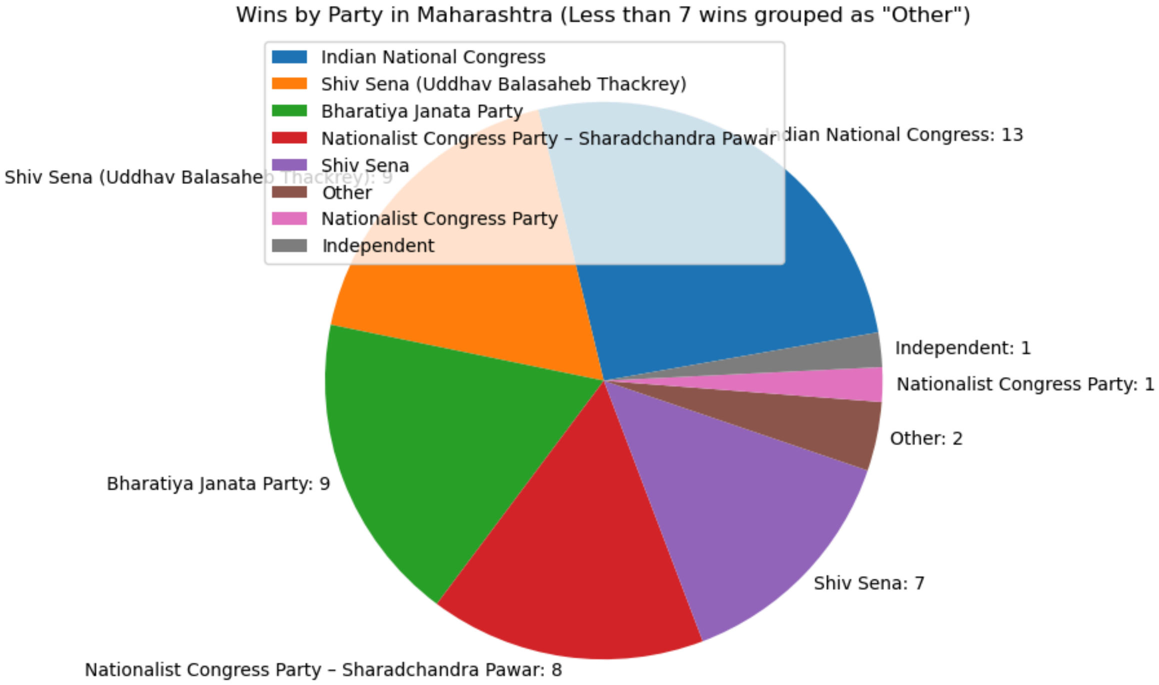
Wins by Party in Kerala (Less than 7 wins grouped as "Other")

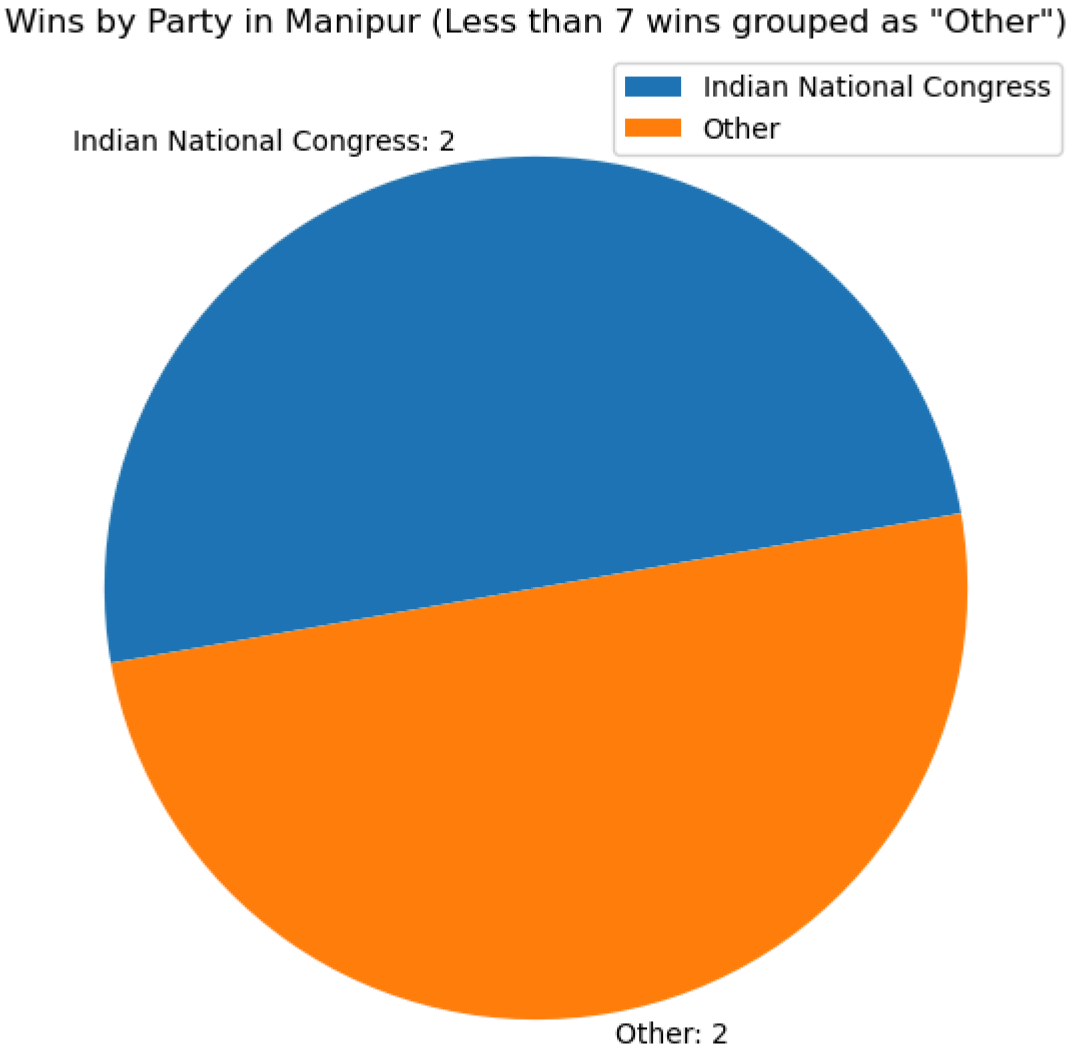


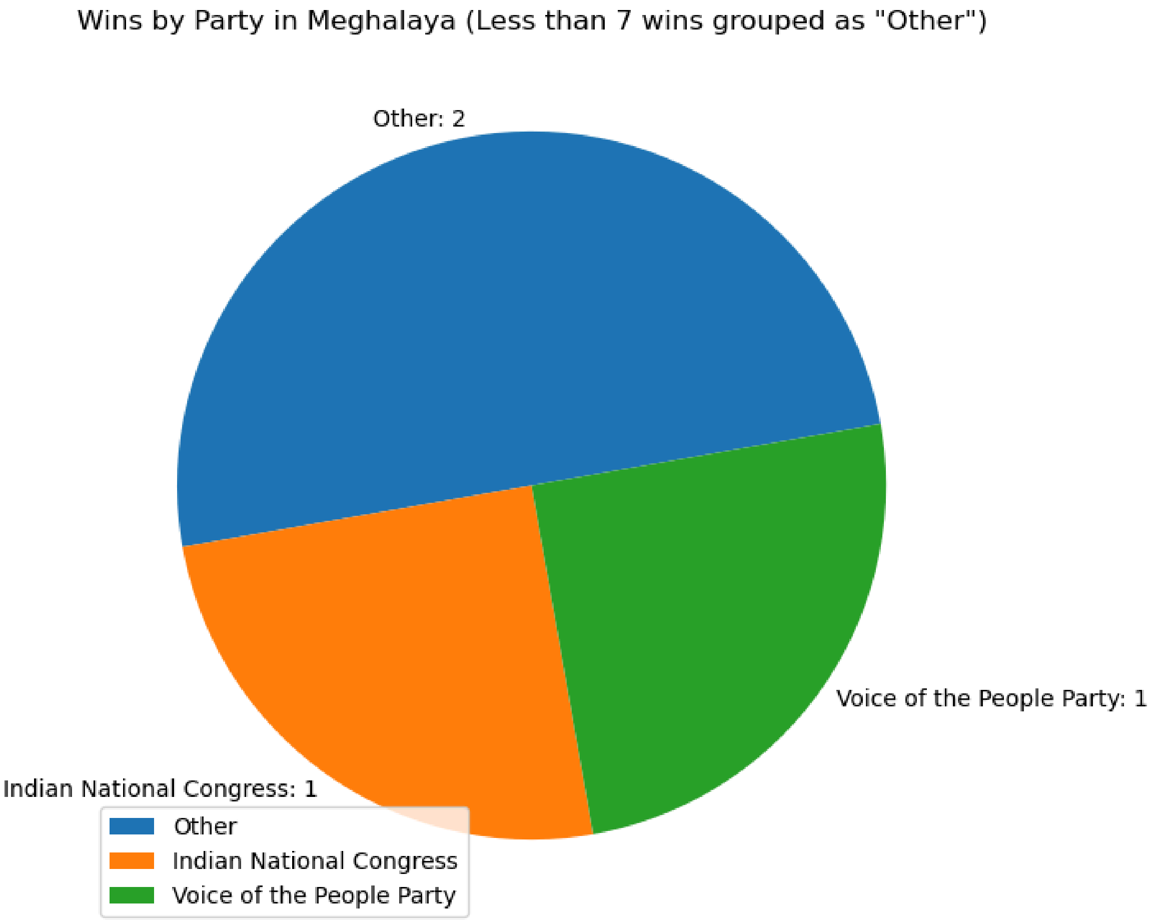

Wins by Party in Ladakh (Less than 7 wins grouped as "Other")
Independent
Other
Independent: 1
Other: 1

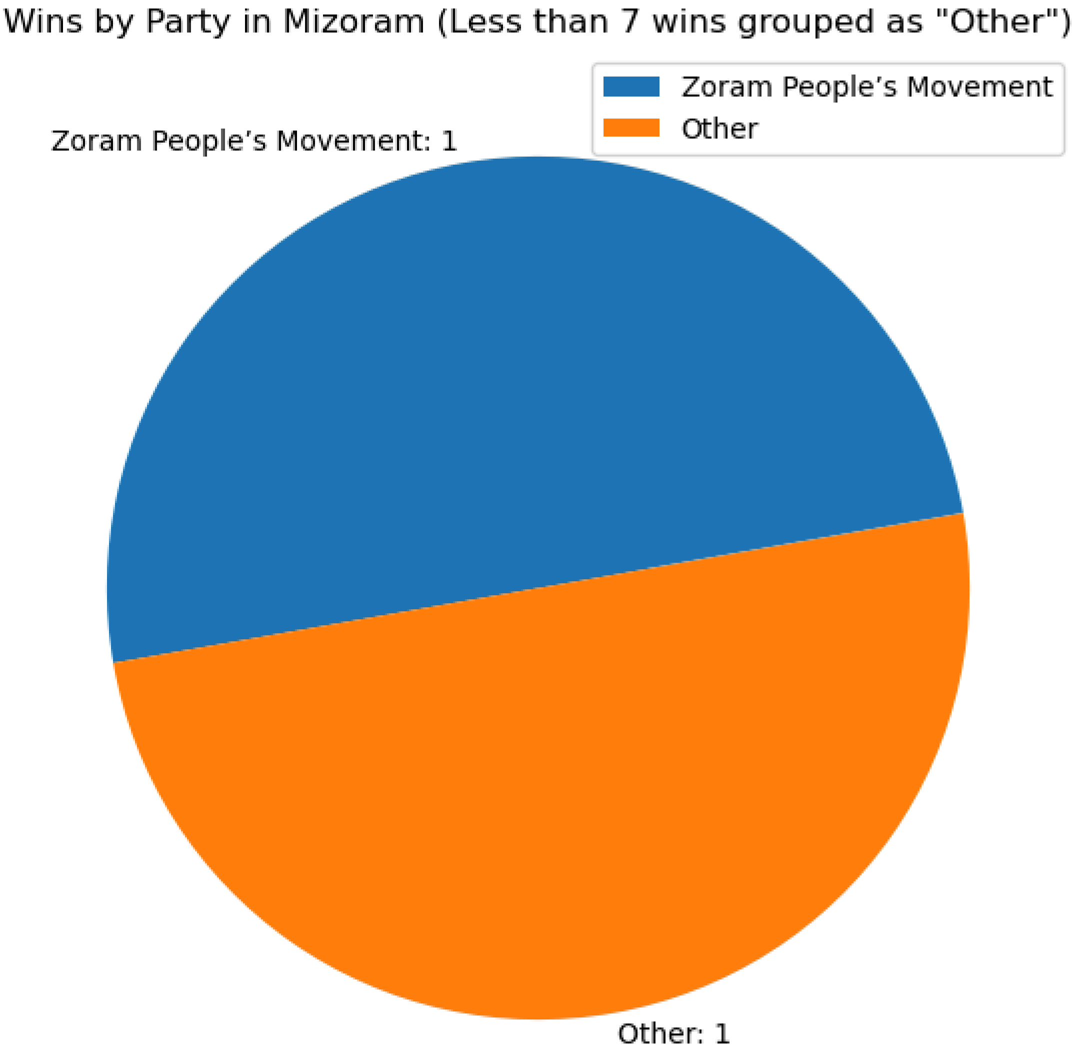

Wins by Party in Lakshadweep (Less than 7 wins grouped as "Other")
Indian National Congress
Other
Indian National Congress: 1
Other: 1

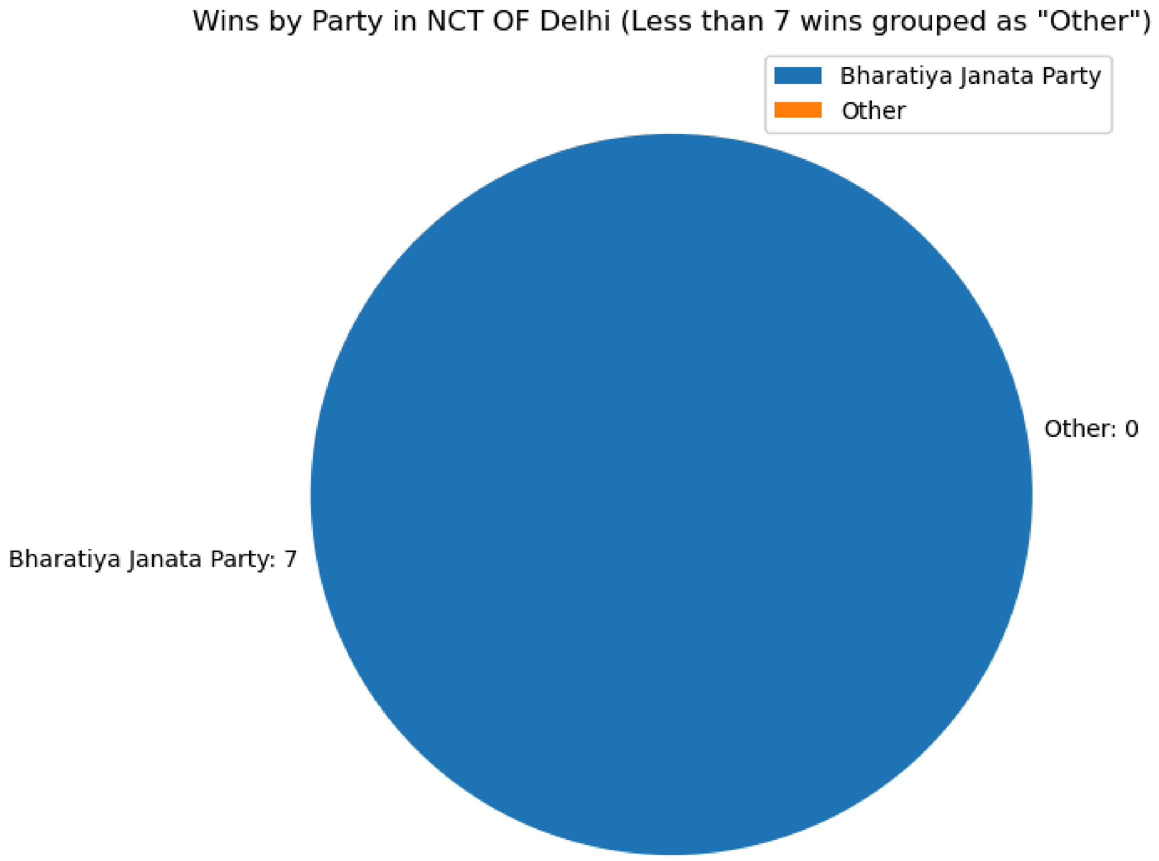

Wins by Party in Madhya Pradesh (Less than 7 wins grouped as "Other")
Bharatiya Janata Party
Other
Other: 0
Bharatiya Janata Party: 29

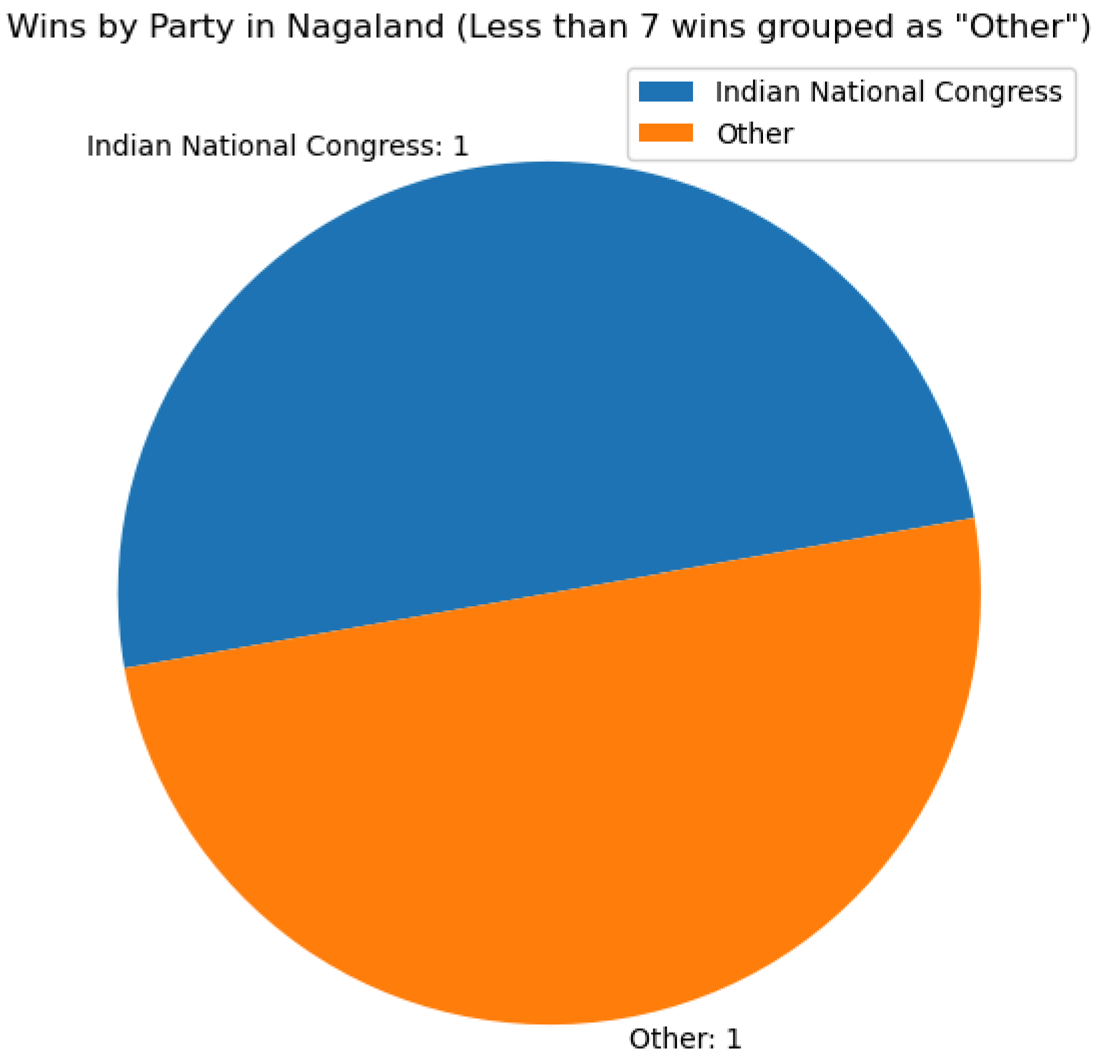

Wins by Party in Maharashtra (Less than 7 wins grouped as "Other")
Indian National Congress
Shiv Sena (Uddhav Balasaheb Thackrey)
Bharatiya Janata Party
Nationalist Congress Party – Sharadchandra Pawar
Shiv Sena
Other
Nationalist Congress Party
Independent
Indian National Congress: 13
Shiv Sena (Uddhav Balasaheb Thackrey): 9
Independent: 1
Nationalist Congress Party: 1
Other: 2
Bharatiya Janata Party: 9
Shiv Sena: 7
Nationalist Congress Party – Sharadchandra Pawar: 8

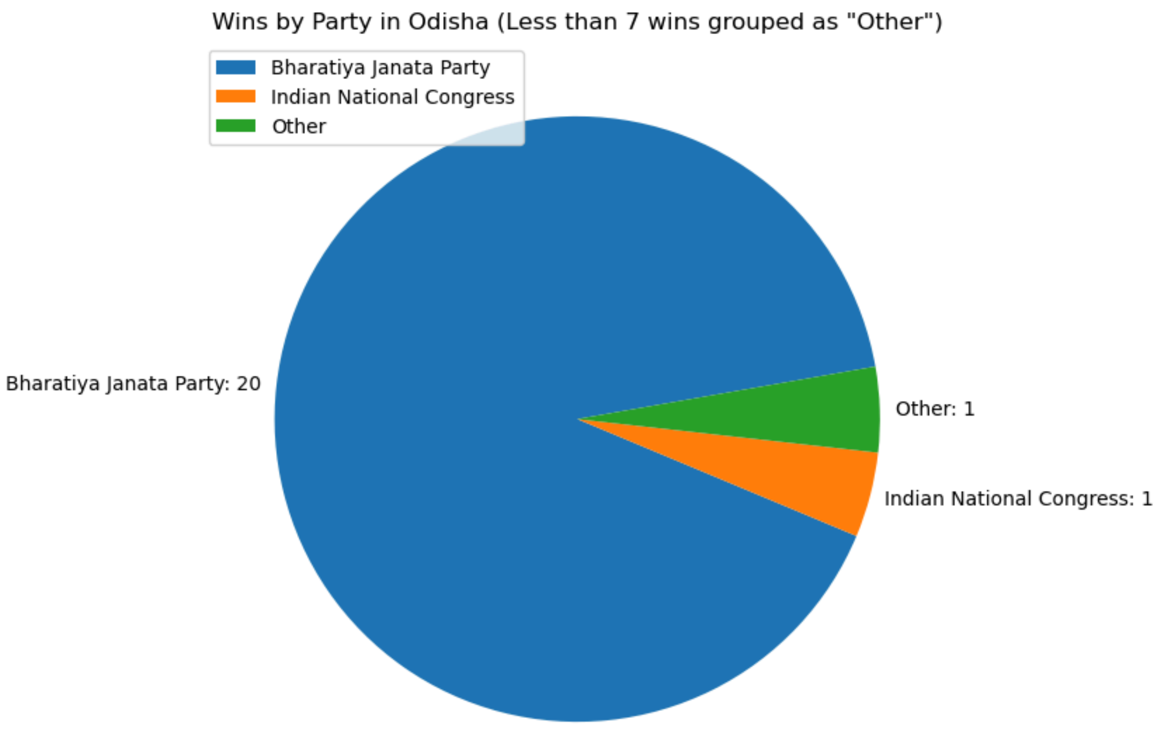

Wins by Party in Manipur (Less than 7 wins grouped as "Other")
Indian National Congress
Other
Indian National Congress: 2
Other: 2

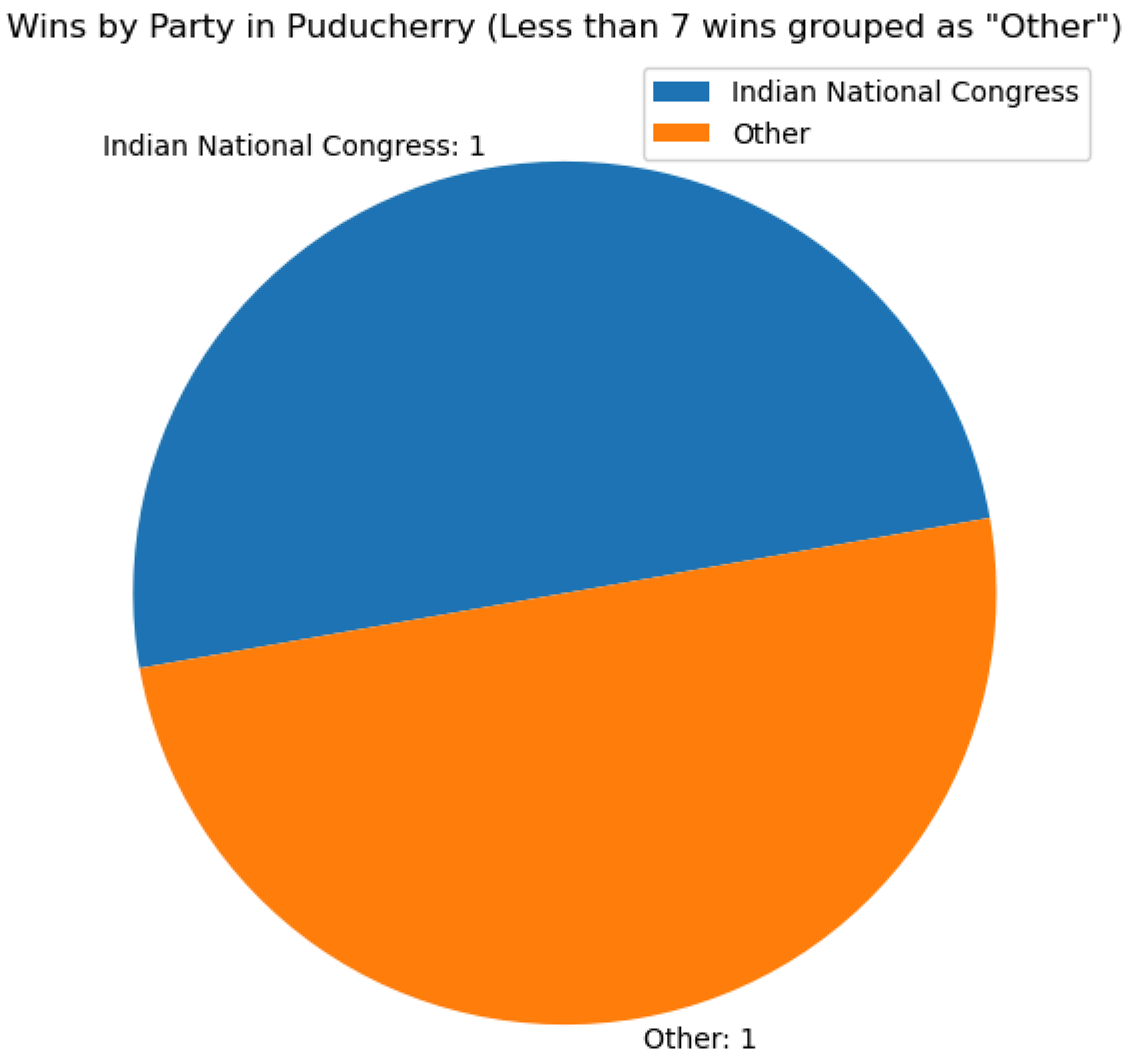

Wins by Party in Meghalaya (Less than 7 wins grouped as "Other")
Other: 2
Voice of the People Party: 1
Indian National Congress: 1
Other
Indian National Congress
Voice of the People Party

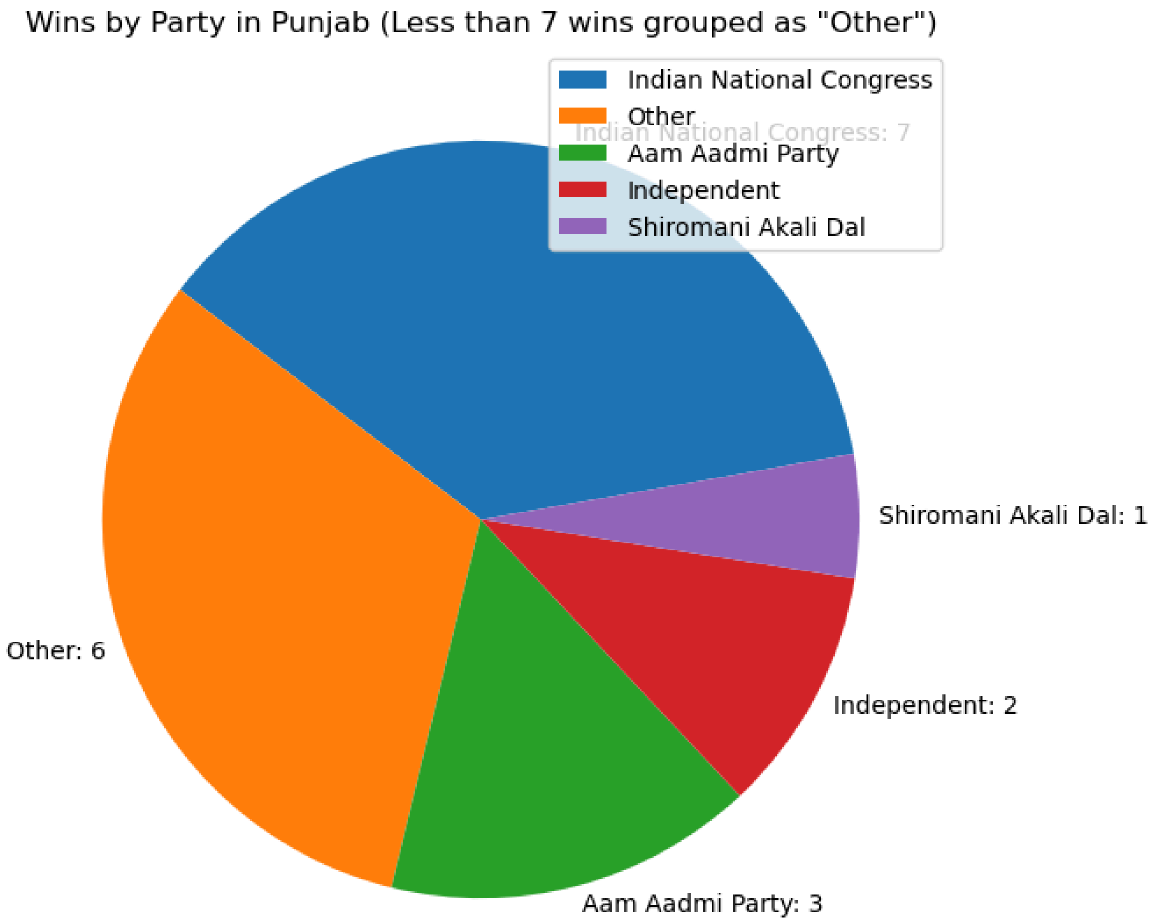

Wins by Party in Mizoram (Less than 7 wins grouped as "Other")
Zoram People’s Movement
Other
Zoram People’s Movement: 1
Other: 1

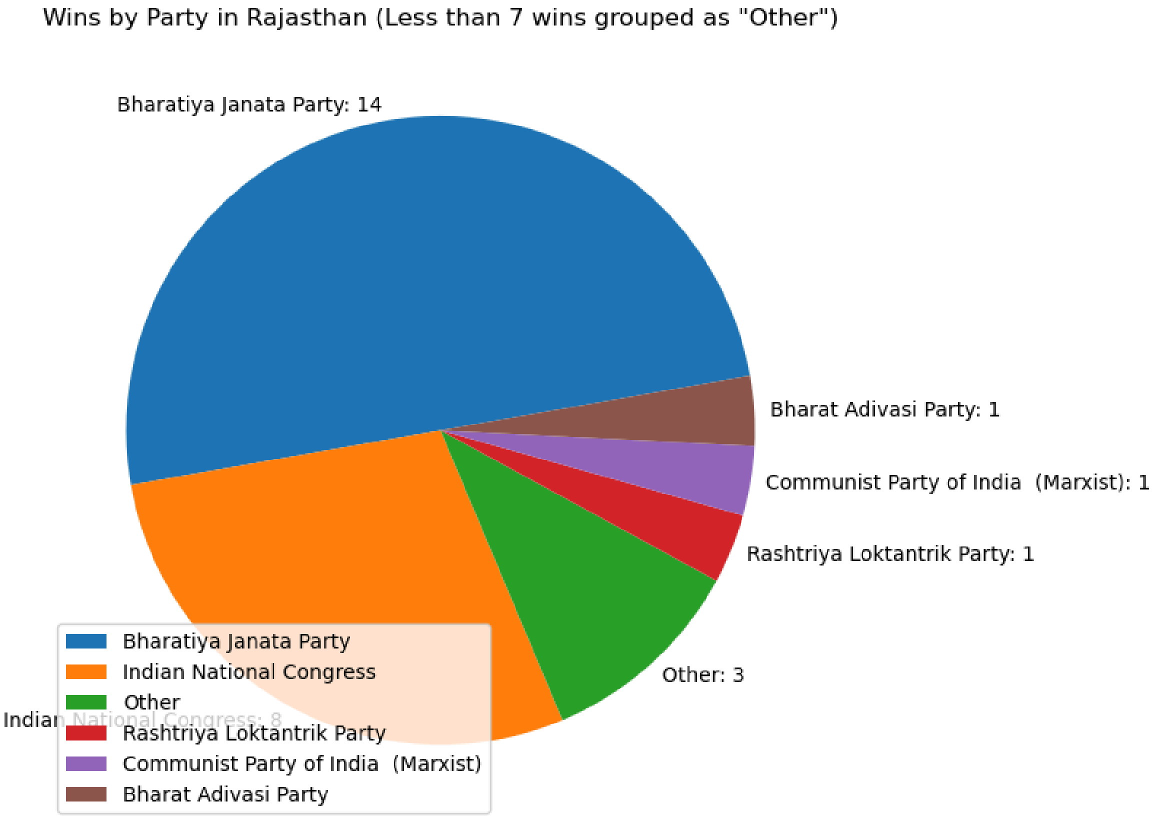

Wins by Party in NCT OF Delhi (Less than 7 wins grouped as "Other")
🟦 Bharatiya Janata Party
🟧 Other
Other: 0
Bharatiya Janata Party: 7

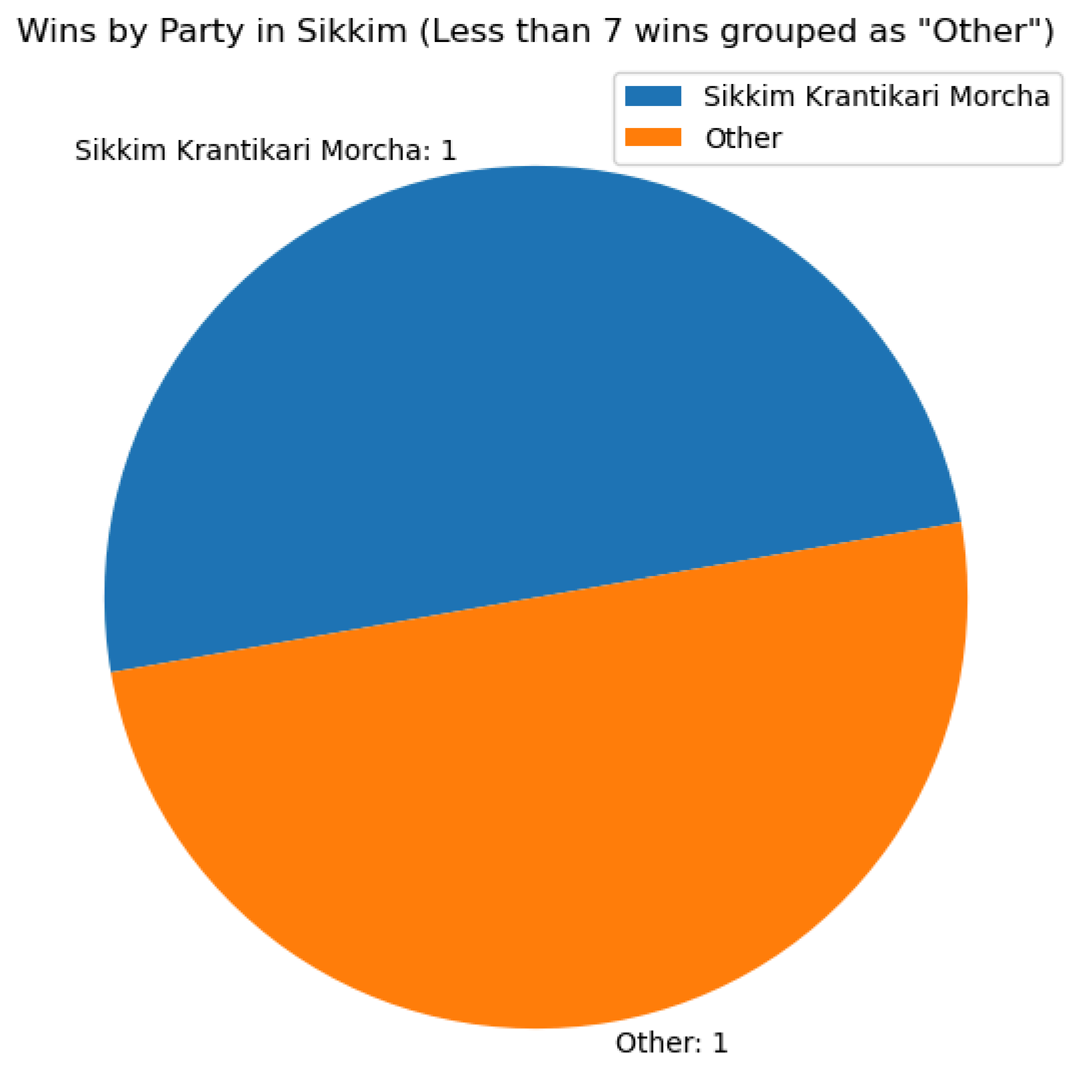

Wins by Party in Nagaland (Less than 7 wins grouped as "Other")
Indian National Congress
Other
Indian National Congress: 1
Other: 1

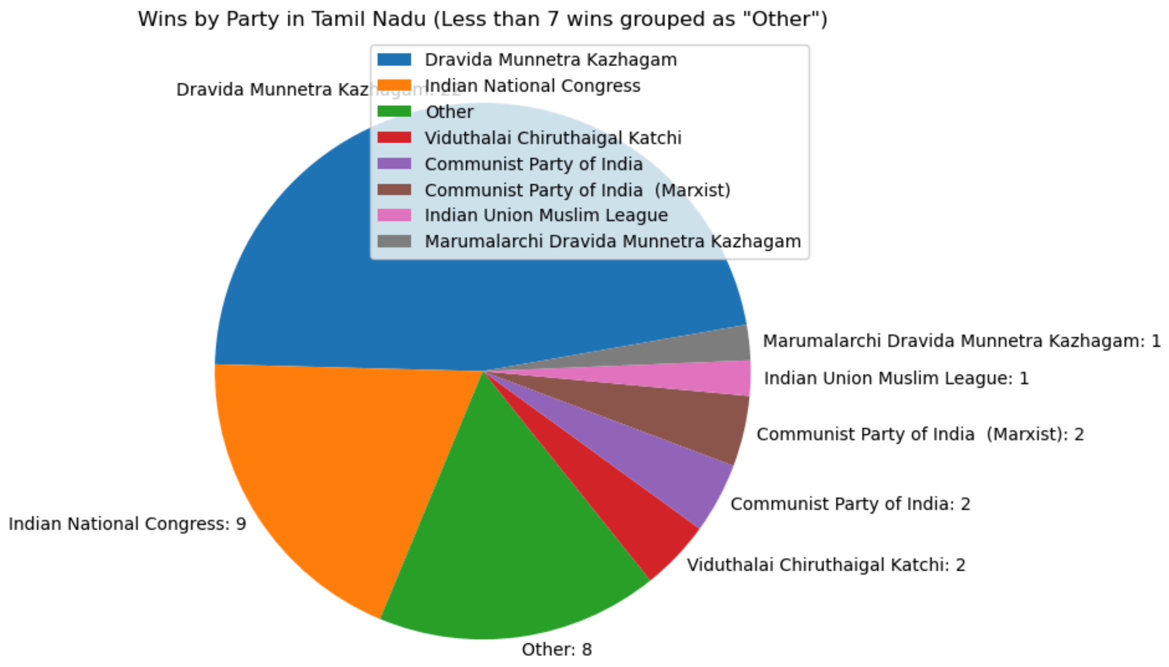

Wins by Party in Odisha (Less than 7 wins grouped as "Other")
Bharatiya Janata Party
Indian National Congress
Other
Bharatiya Janata Party: 20
Other: 1
Indian National Congress: 1


Wins by Party in Puducherry (Less than 7 wins grouped as "Other")
Indian National Congress
Other
Indian National Congress: 1
Other: 1

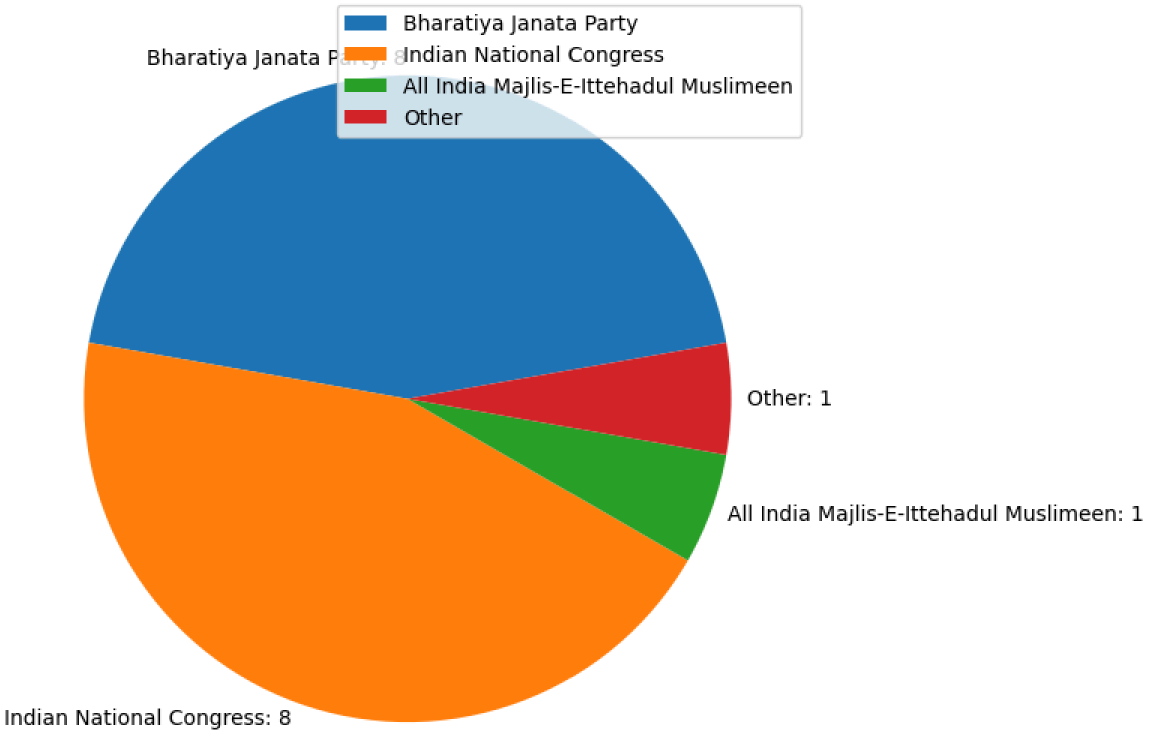

Wins by Party in Punjab (Less than 7 wins grouped as "Other")


Wins by Party in Rajasthan (Less than 7 wins grouped as "Other")

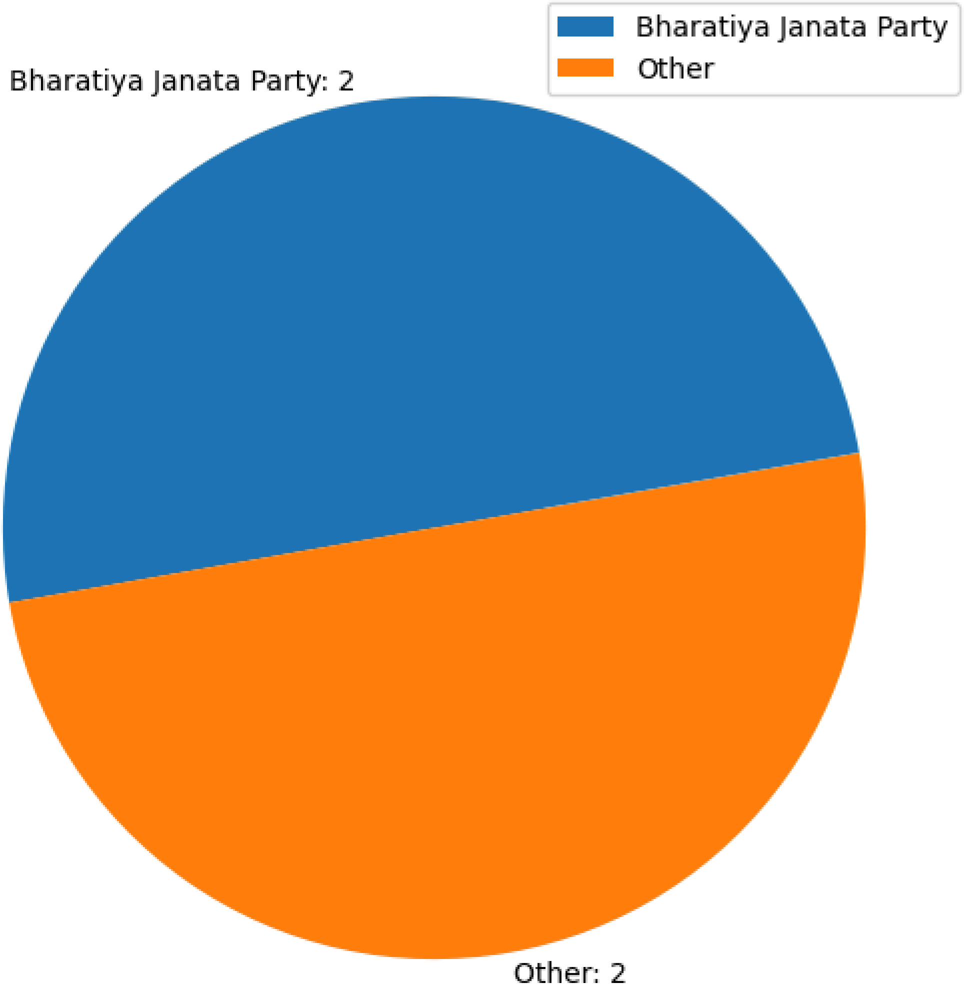

Wins by Party in Sikkim (Less than 7 wins grouped as "Other")
Sikkim Krantikari Morcha
Other
Sikkim Krantikari Morcha: 1
Other: 1


Wins by Party in Tamil Nadu (Less than 7 wins grouped as "Other")
Dravida Munnetra Kazhagam
Indian National Congress
Other
Viduthalai Chiruthaigal Katchi
Communist Party of India
Communist Party of India (Marxist)
Indian Union Muslim League
Marumalarchi Dravida Munnetra Kazhagam
Dravida Munnetra Kazhagam: 22
Marumalarchi Dravida Munnetra Kazhagam: 1
Indian Union Muslim League: 1
Communist Party of India (Marxist): 2
Communist Party of India: 2
Viduthalai Chiruthaigal Katchi: 2
Indian National Congress: 9
Other: 8

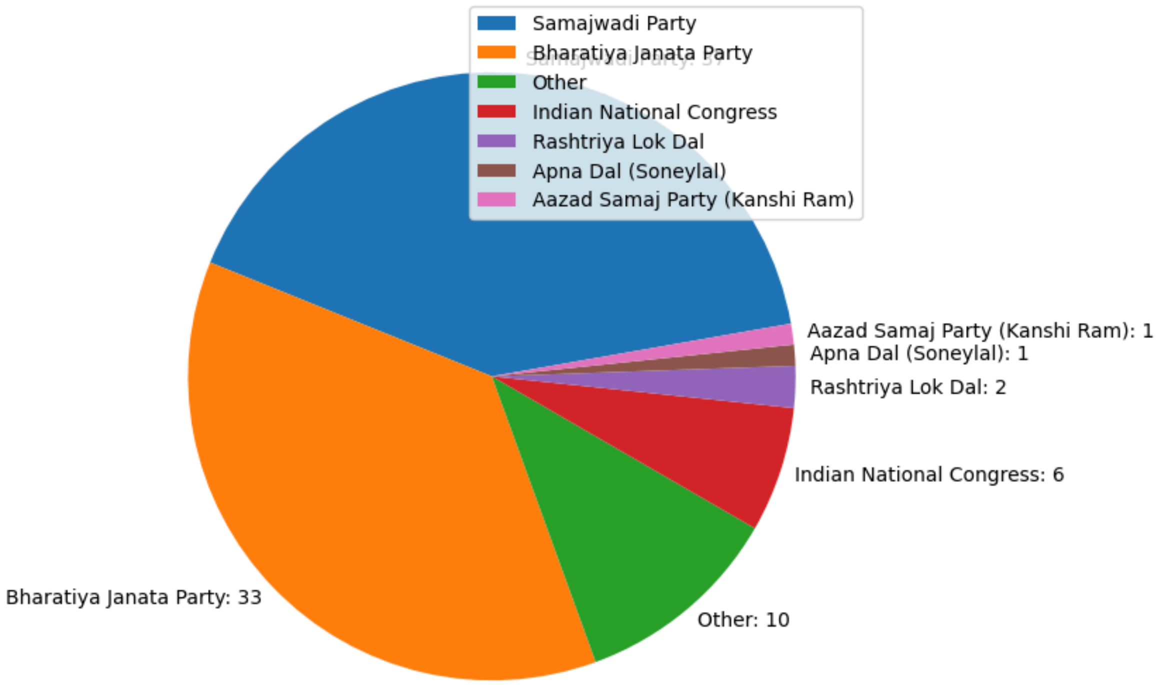
Wins by Party in Telangana (Less than 7 wins grouped as "Other")



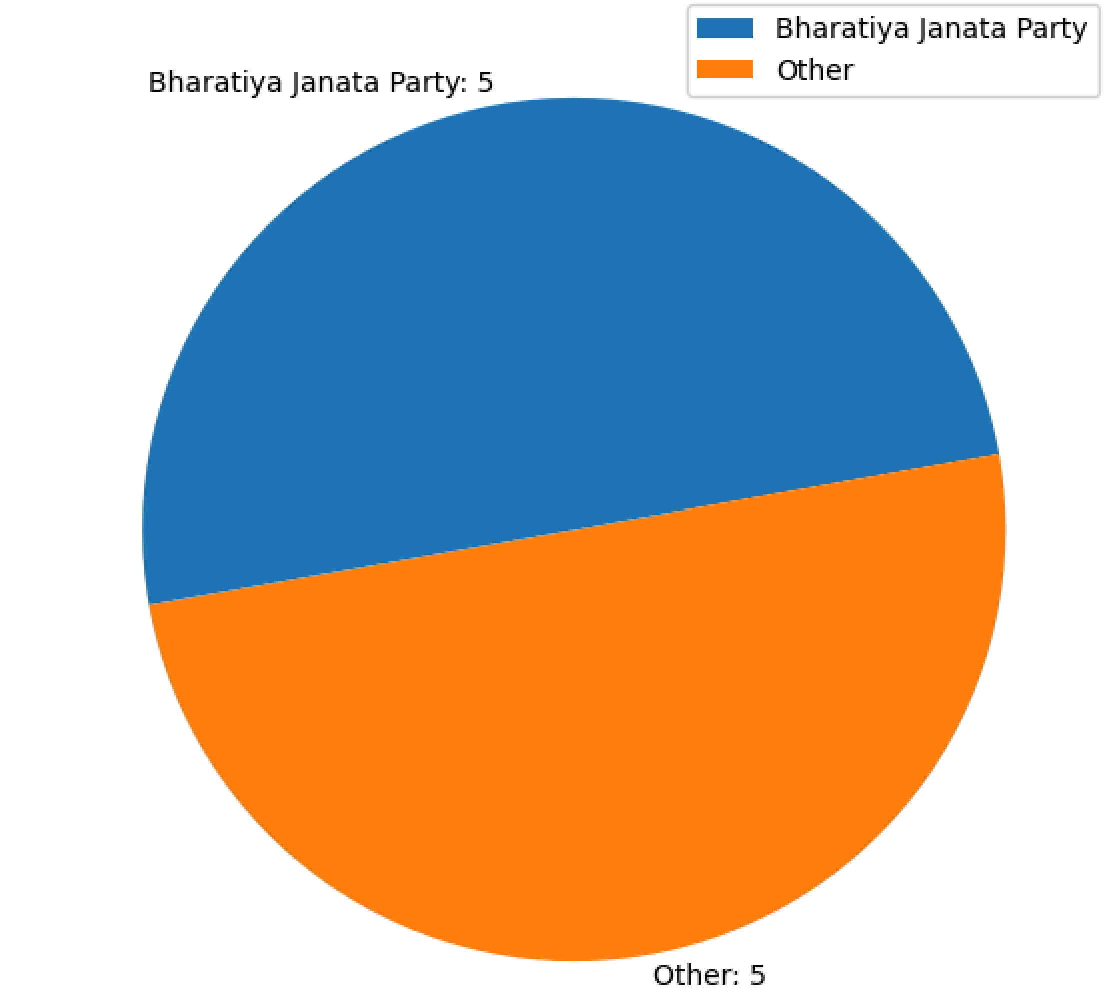
Wins by Party in Tripura (Less than 7 wins grouped as "Other")

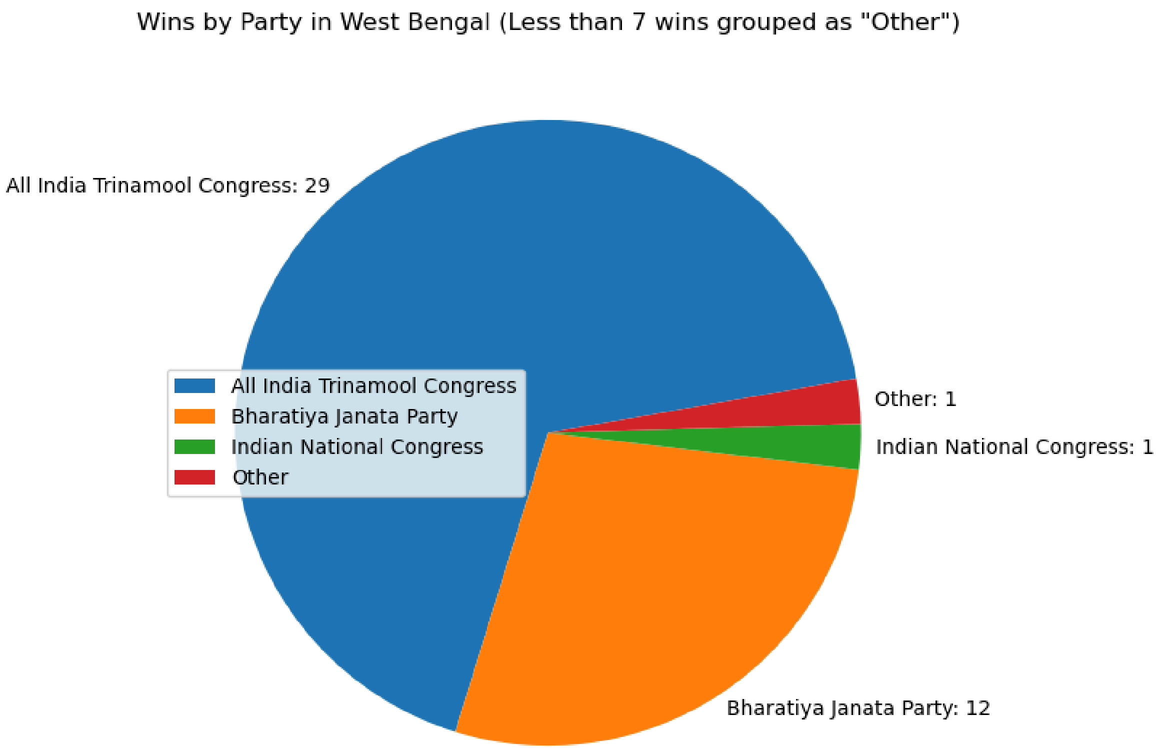


Wins by Party in Uttar Pradesh (Less than 7 wins grouped as "Other")



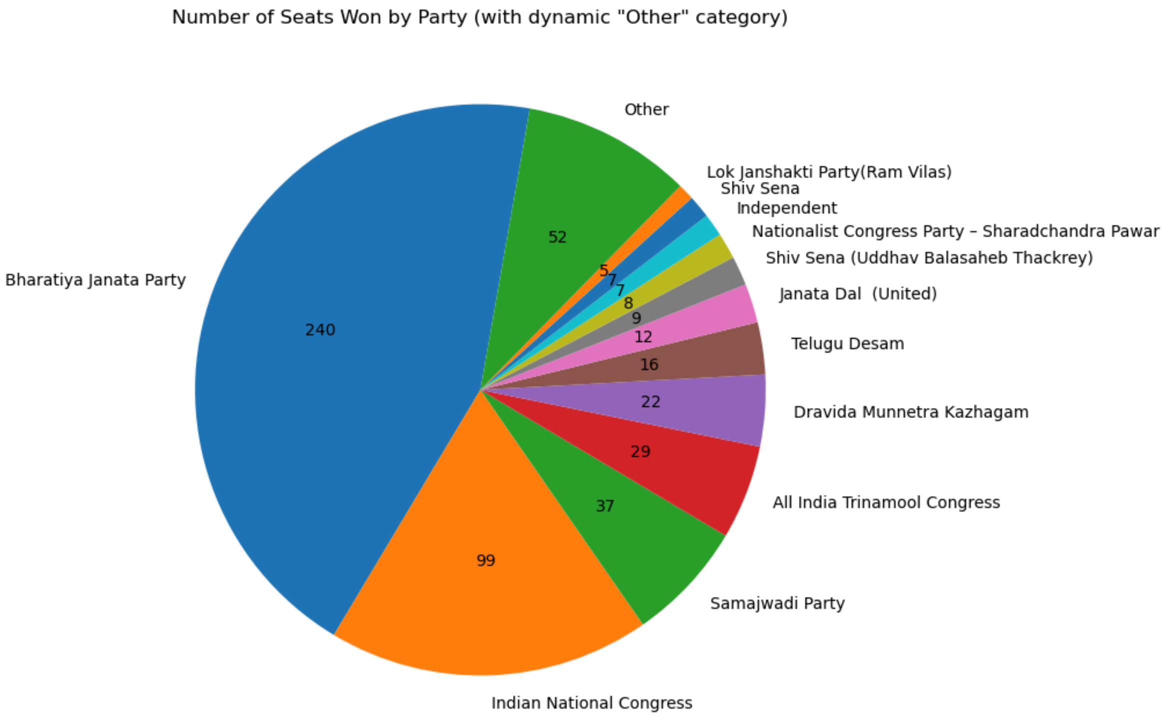
Wins by Party in Uttarakhand (Less than 7 wins grouped as "Other")





In [ ]:

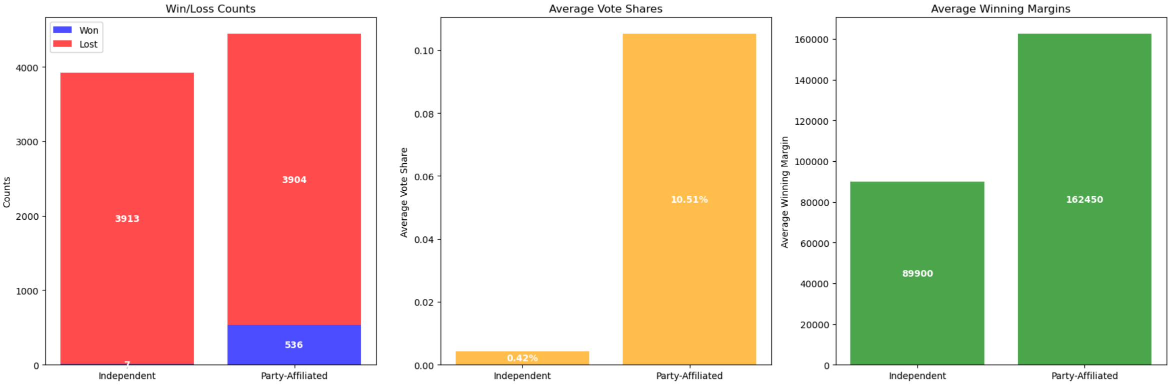
```
# Plotting the number of seats won by party
seats_won_by_party = df[df['result'] == 'won']['party'].value_counts()

threshold = 5
parties_below_threshold = seats_won_by_party[seats_won_by_party < thresho

other_category_count = parties_below_threshold.sum()

seats_won_by_party = seats_won_by_party.drop(parties_below_threshold.inde

if other_category_count > 0:
    seats_won_by_party['Other'] = other_category_count

plt.figure(figsize=(10, 8))
plt.pie(seats_won_by_party, startangle=80, labels=seats_won_by_party.inde
plt.title('Number of Seats Won by Party (with dynamic "Other" category)')
plt.show()
```



In [ ]:

```
# Analyze the performance of independent candidates vs party-affiliated c
df['candidate_type'] = df['party'].apply(lambda x: 'Independent' if x ==

# Win/Loss counts
win_loss_counts = df.groupby(['candidate_type', 'result']).size().unstack

# Vote shares
df['total_votes'] = df.groupby('constituency')['votes'].transform('sum')
df['vote_share'] = df['votes'] / df['total_votes']
average_vote_share = df.groupby('candidate_type')['vote_share'].mean().re

# Winning margins
df_sorted = df.sort_values(by=['constituency', 'votes'], ascending=[True,
df_sorted['next_votes'] = df_sorted.groupby('constituency')['votes'].shif
df_sorted['winning_margin'] = df_sorted['votes'] - df_sorted['next_votes'
average_winning_margin = df_sorted[df_sorted['result'] == 'won'].groupby(

# Merge results into a performance summary
performance_summary = pd.merge(win_loss_counts, average_vote_share, on='c
performance_summary = pd.merge(performance_summary, average_winning_margi

# Rename columns for clarity
performance_summary.columns = ['Candidate Type', 'Lost', 'Won', 'Average

# Print performance summary
print(performance_summary)

# Plotting
plt.figure(figsize=(18, 6))

# Plotting Win/Loss Counts
plt.subplot(1, 3, 1)
plt.bar(performance_summary['Candidate Type'], performance_summary['Won']
plt.bar(performance_summary['Candidate Type'], performance_summary['Lost'
plt.ylabel('Counts')
plt.title('Win/Loss Counts')
plt.legend()

# Adding labels to bars
for i in range(len(performance_summary)):
    plt.text(i, performance_summary['Won'][i]/2, performance_summary['Won
    plt.text(i, performance_summary['Won'][i] + performance_summary['Lost

# Plotting Average Vote Shares
plt.subplot(1, 3, 2)
plt.bar(performance_summary['Candidate Type'], performance_summary['Avera
plt.ylabel('Average Vote Share')
plt.title('Average Vote Shares')

# Adding labels to bars
for i in range(len(performance_summary)):
    plt.text(i, performance_summary['Average Vote Share'][i]/2, f"{perfor

# Plotting Average Winning Margins
plt.subplot(1, 3, 3)
plt.bar(performance_summary['Candidate Type'], performance_summary['Avera
plt.ylabel('Average Winning Margin')
plt.title('Average Winning Margins')

# Adding labels to bars
for i in range(len(performance_summary)):
    plt.text(i, performance_summary['Average Winning Margin'][i]/2, int(p

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

```
     Candidate Type  Lost  Won  Average Vote Share  Average Winning Margin
0       Independent  3913    7            0.004155            89900.285714
1  Party-Affiliated  3904  536            0.105142           162450.263060
```



In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
from prettytable import PrettyTable

# Load the dataset
file_path = 'CLEANED_DATA.CSV'
df = pd.read_csv(file_path)

# Ensure 'votes' column is correctly converted to integers, handling non-
df['votes'] = pd.to_numeric(df['votes'], errors='coerce').fillna(0).astyp

# Calculate voter turnout for each constituency
voter_turnout = df.groupby('constituency')['votes'].sum().reset_index()

# Display the top 10 constituencies by voter turnout using PrettyTable
```

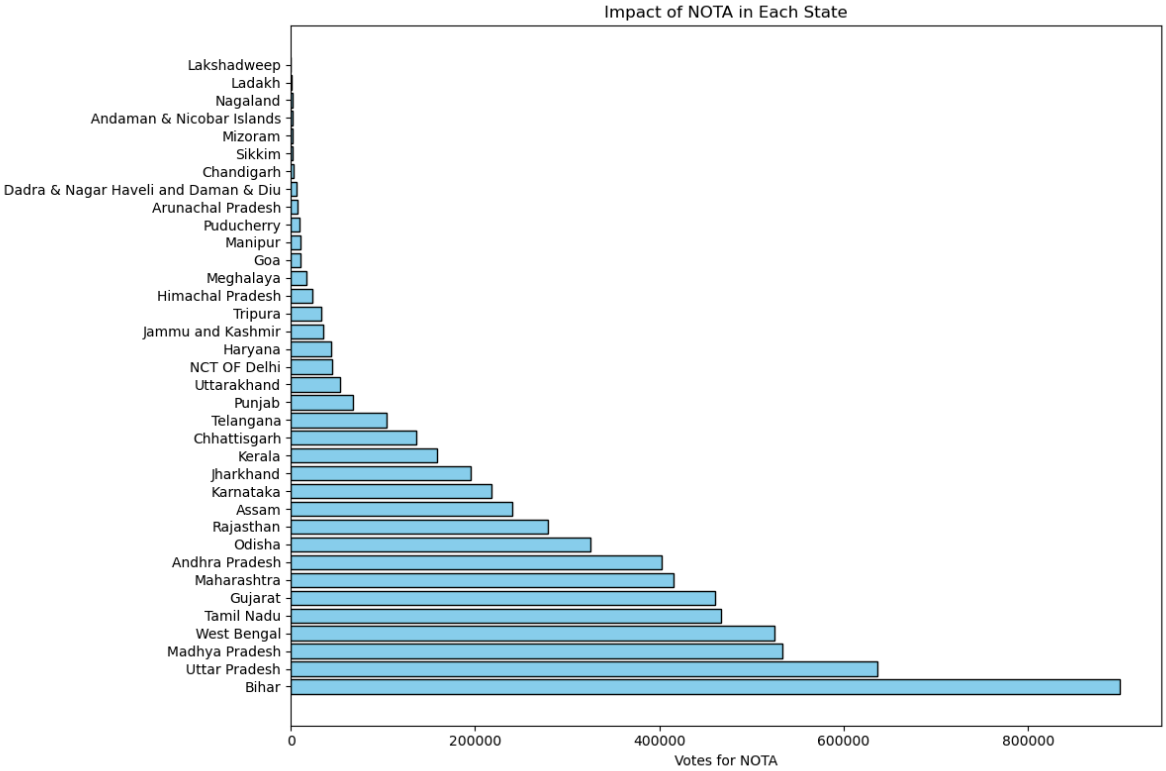
```
top_n = 10
top_voter_turnout = voter_turnout.nlargest(top_n, 'votes')
bottom_voter_turnout = voter_turnout.nsmallest(top_n, 'votes')

table = PrettyTable()
table.field_names = ["Constituency", "Total Votes"]

for _, row in top_voter_turnout.iterrows():
    table.add_row([row['constituency'], f"{row['votes']:,}"])

print("Top 10 Constituencies by Voter Turnout")
print(table)

# Display the bottom 10 constituencies by voter turnout using PrettyTable
table.clear_rows()
for _, row in bottom_voter_turnout.iterrows():
    table.add_row([row['constituency'], f"{row['votes']:,}"])

print("\nBottom 10 Constituencies by Voter Turnout")
print(table)
print("0 votes means that the candidate was uncontested in that constitue
```

```
Top 10 Constituencies by Voter Turnout
+------------------+-------------+
|   Constituency   | Total Votes |
+------------------+-------------+
|      Dhubri      |  2,453,608  |
|    Aurangabad    |  2,248,077  |
|   Maharajganj    |  2,224,560  |
|    Malkajgiri    |  1,933,843  |
| Bangalore Rural  |  1,919,540  |
| Darrang-Udalguri |  1,811,200  |
| Bangalore North  |  1,752,504  |
|      BARMER      |  1,688,051  |
|     Barpeta      |  1,685,943  |
|     Chevella     |  1,675,354  |
+------------------+-------------+

Bottom 10 Constituencies by Voter Turnout
+---------------------------+-------------+
|        Constituency       | Total Votes |
+---------------------------+-------------+
|           Surat           |      0      |
|        Lakshadweep        |    49,200   |
|        Daman & Diu        |    92,410   |
|           Ladakh          |   135,524   |
| Andaman & Nicobar Islands |   202,514   |
|    Dadar & Nagar Haveli   |   205,588   |
|       Arunachal East      |   323,443   |
|           Sikkim          |   384,893   |
|       Arunachal West      |   399,804   |
|         Chandigarh        |   449,275   |
+---------------------------+-------------+
0 votes means that the candidate was uncontested in that constituency.
```

In [ ]:

```
# Filter NOTA votes
nota_df = df[df['party'] == 'None of the Above']

# Group by state and sum votes
nota_votes_by_state = nota_df.groupby('state')['votes'].sum().reset_index
```

```python
# Add total votes for each state to calculate the percentage of NOTA vote
total_votes_by_state = df.groupby('state')['votes'].sum().reset_index()
nota_votes_by_state = pd.merge(nota_votes_by_state, total_votes_by_state,

# Calculate percentage of NOTA votes
nota_votes_by_state['percentage'] = (nota_votes_by_state['votes_nota'] /

# Sort by NOTA votes
nota_votes_by_state = nota_votes_by_state.sort_values(by='votes_nota', as

# Plot the impact of NOTA in each state
plt.figure(figsize=(12, 8))
bars = plt.barh(nota_votes_by_state['state'], nota_votes_by_state['votes_

# Add title and labels
plt.title('Impact of NOTA in Each State')
plt.xlabel('Votes for NOTA')
# plt.text(width, bar.get_y() + bar.get_height()/2, f'{width:,}', va='cen

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



In [ ]:

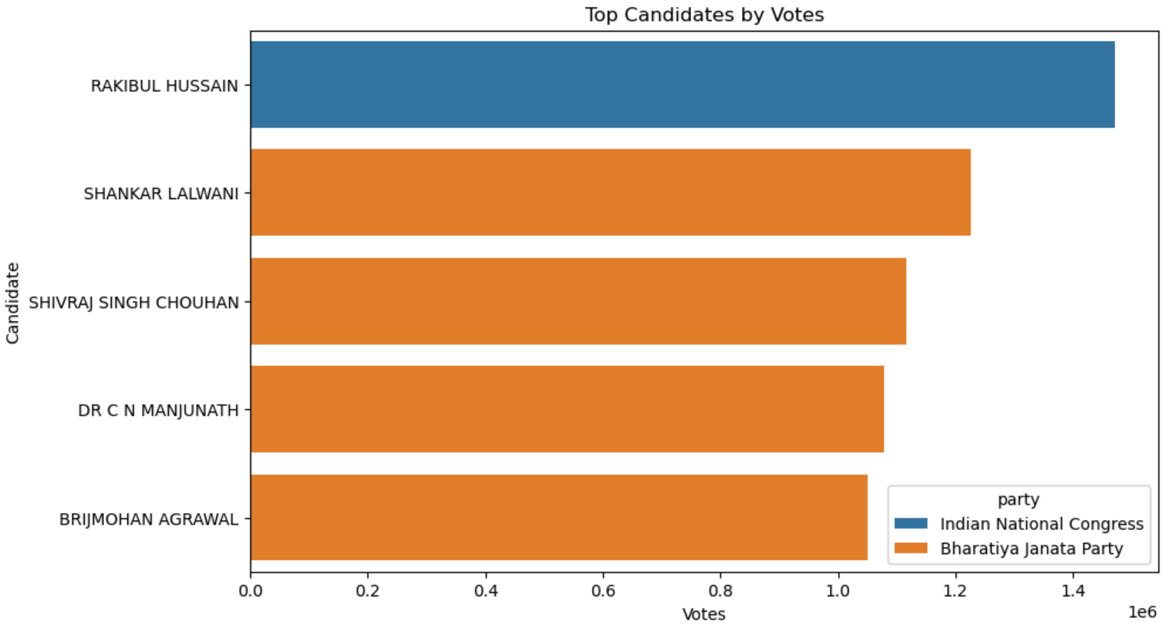
```python
# Analyse highest and lowest votes received by a party
import csv

highest_votes = 0
lowest_votes = float('inf')
highest_votes_state = ""
lowest_votes_state = ""

with open('CLEANED_DATA.CSV', mode='r') as file:
    csv_reader = csv.DictReader(file)
    for row in csv_reader:
        votes = int(row['votes']) if row['votes'] else 0
        state = row['state']
        if votes > highest_votes:
            highest_votes = votes
            highest_votes_state = state
        if 0 < votes < lowest_votes:
            lowest_votes = votes
            lowest_votes_state = state

print(f"Highest votes received by a party: {highest_votes_state} with {hi
print(f"Lowest votes received by a party: {lowest_votes_state} with {lowe
```

```
Highest votes received by a party: Assam with 1471885 votes
Lowest votes received by a party: Lakshadweep with 61 votes
```

In [ ]:

```
# Analyse the top candidates by votes
df['votes'] = pd.to_numeric(df['votes'], errors='coerce')

df_sorted = df.sort_values(by='votes', ascending=False)

top_candidates = df_sorted.head(5)

print(top_candidates[['name', 'party', 'votes']])
plt.figure(figsize=(10, 6))
sns.barplot(x='votes', y='name', data=top_candidates, hue='party', dodge=
plt.title('Top Candidates by Votes')
plt.xlabel('Votes')
plt.ylabel('Candidate')
plt.show()
```

```
                   name                     party    votes
596         RAKIBUL HUSSAIN  Indian National Congress  1471885
3459        SHANKAR LALWANI    Bharatiya Janata Party  1226751
3276   SHIVRAJ SINGH CHOUHAN    Bharatiya Janata Party  1116460
2685        DR C N MANJUNATH    Bharatiya Janata Party  1079002
1239      BRIJMOHAN AGRAWAL    Bharatiya Janata Party  1050351
```

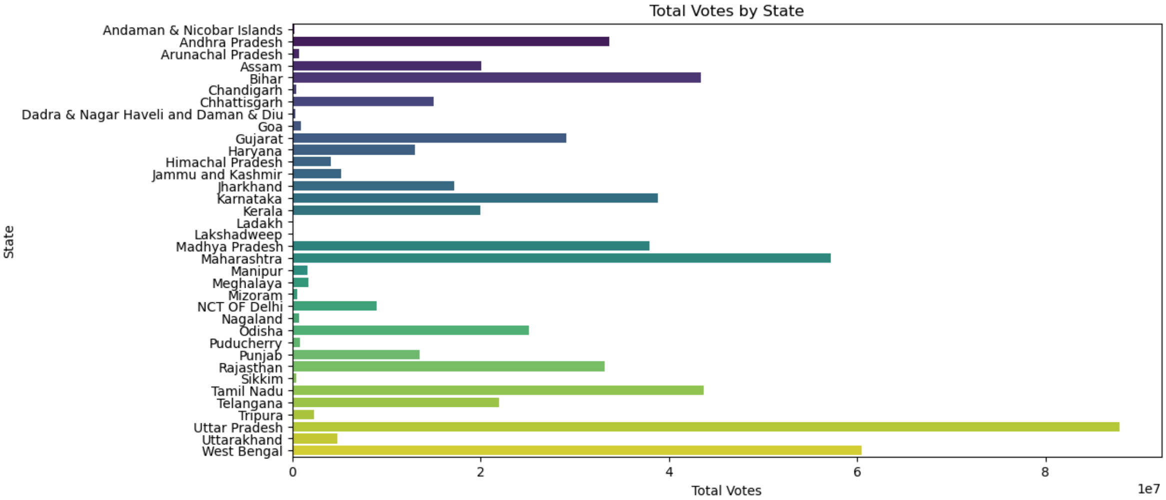


In [ ]:

```
# top state in terms of voter turnout
top_state = df.groupby('state')['votes'].sum().idxmax()
top_state_votes = df.groupby('state')['votes'].sum().max()

# bottom state in terms of voter turnout
bottom_state = df.groupby('state')['votes'].sum().idxmin()
bottom_state_votes = df.groupby('state')['votes'].sum().min()
```

```
print(f'Top State in Voter Turnout: {top_state} with {top_state_votes} vo
print(f'Bottom State in Voter Turnout: {bottom_state} with {bottom_state_

# Plotting the results
plt.figure(figsize=(12, 6))
sns.barplot(x='votes', y='state', data=df, estimator=sum, errorbar=None,
plt.title('Total Votes by State')
plt.xlabel('Total Votes')
plt.ylabel('State')
plt.show()
```

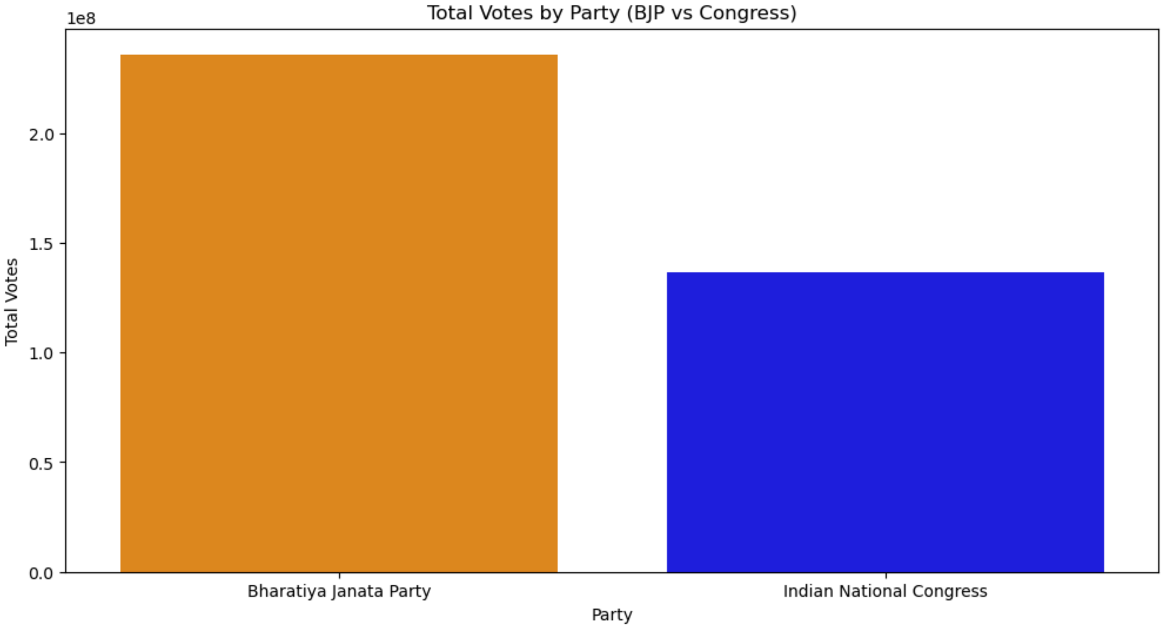
```
Top State in Voter Turnout: Uttar Pradesh with 87911642 votes
Bottom State in Voter Turnout: Lakshadweep with 49200 votes
```



In [ ]:

```
# Filter data for BJP and Congress and explicitly create a copy
bjp_congress_data = df[df['party'].isin(['Bharatiya Janata Party', 'India

# The rest of your code remains the same
bjp_wins = (bjp_congress_data['party'] == 'Bharatiya Janata Party') & (bj
congress_wins = (bjp_congress_data['party'] == 'Indian National Congress'

bjp_votes = bjp_congress_data[bjp_congress_data['party'] == 'Bharatiya Ja
congress_votes = bjp_congress_data[bjp_congress_data['party'] == 'Indian

bjp_congress_data['abs_margin'] = bjp_congress_data['margin'].apply(lambd
bjp_avg_margin = bjp_congress_data[bjp_congress_data['party'] == 'Bharati
congress_avg_margin = bjp_congress_data[bjp_congress_data['party'] == 'In

print(f"BJP Wins: {bjp_wins.sum()}, Congress Wins: {congress_wins.sum()}"
print(f"BJP Vote Share: {bjp_votes}, Congress Vote Share: {congress_votes
print(f"BJP Average Margin: {bjp_avg_margin}, Congress Average Margin: {c

plt.figure(figsize=(12, 6))
sns.barplot(x='party', y='votes', data=bjp_congress_data, estimator=sum,
plt.title('Total Votes by Party (BJP vs Congress)')
plt.xlabel('Party')
plt.ylabel('Total Votes')
plt.show()
```

```
BJP Wins: 240, Congress Wins: 99
BJP Vote Share: 235973935, Congress Vote Share: 136759064
BJP Average Margin: 104114.56009070294, Congress Average Margin: 39652.201
21951219
```



Total Votes by Party (BJP vs Congress)
1e8
2.0
1.5
1.0
0.5
0.0
Total Votes
Bharatiya Janata Party
Indian National Congress
Party