```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt

# Function to fetch and parse data from the general election results
page
def fetch_general_election_results(url):
    print(f"Fetching data from {url}...")

    # Fetch the HTML content of the page
    response = requests.get(url)
    if response.status_code == 200:
        print(f"Successfully accessed the page: {url}")
    else:
        print(f"Failed to access the page: {url}. Status code:
{response.status_code}")
        return

    # Parse HTML using BeautifulSoup
    soup = BeautifulSoup(response.content, 'html.parser')
    print("HTML content fetched and parsed.")

    # Extract data from the first table in the page
    results_table = soup.find('table', class_='table')
    if results_table is None:
        print("No results table found.")
        return

    print("Results table found.")

    # Initialize lists to store extracted data
    parties = []
    won = []
    leading = []
    total = []

    for row in results_table.find('tbody').find_all('tr'):
        cells = row.find_all('td')
        if len(cells) < 4:
            print("Skipping row with insufficient columns.")
            continue  # Skip rows that do not have enough columns

        parties.append(cells[0].text.strip())  # Party name
        won.append(cells[1].text.strip())     # Number of seats won
        leading.append(cells[2].text.strip()) # Number of seats
leading
        total.append(cells[3].text.strip())   # Total seats contested

    print(f"Extracted data for {len(parties)} parties.")
```

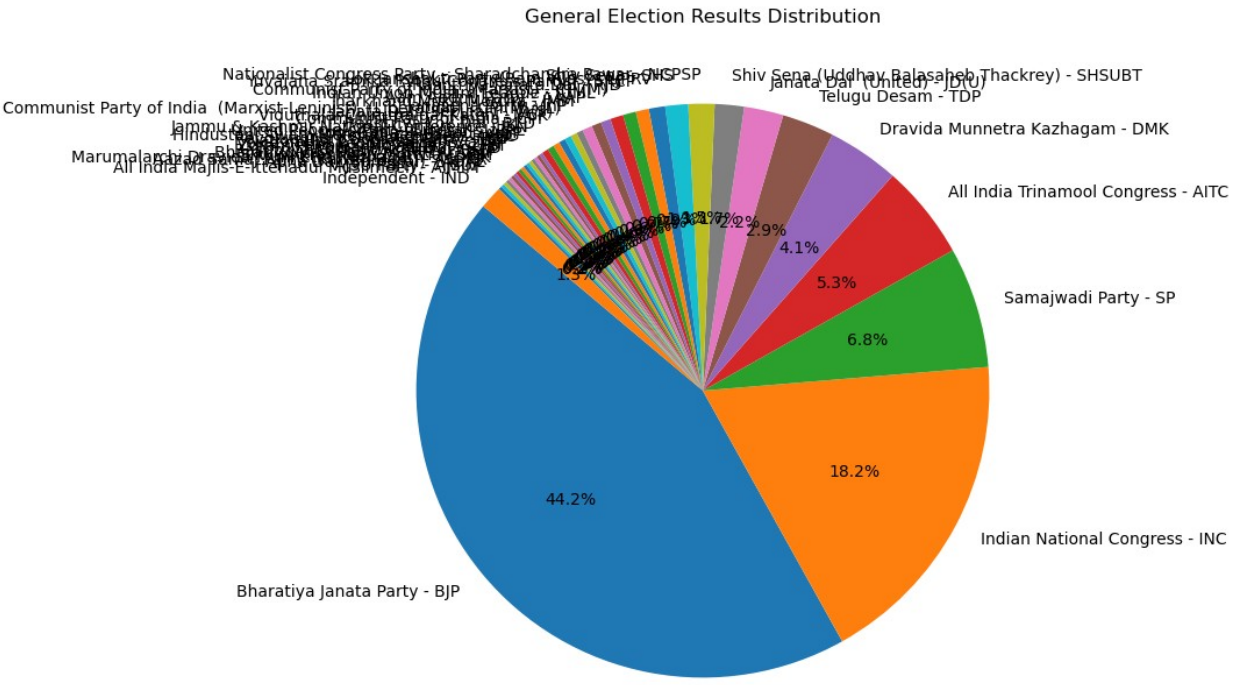
```
    # Create a DataFrame to store the extracted data
    df = pd.DataFrame({
        'Party': parties,
        'Seats Won': won,
        'Seats Leading': leading,
        'Total Seats': total
    })

    # Convert columns to appropriate data types
    df['Seats Won'] = df['Seats Won'].astype(int)
    df['Seats Leading'] = df['Seats Leading'].astype(int)
    df['Total Seats'] = df['Total Seats'].astype(int)

    # Save the data to a CSV file
    csv_filename = 'general_election_results.csv'
    df.to_csv(csv_filename, index=False)
    print(f"Data saved to {csv_filename}.")

    # Display the DataFrame
    print(df.head())  # Print the first few rows of the DataFrame to
check the data

    # Plot the data
    plot_general_election_results(df)

    return df

# Function to plot pie chart and bar chart for the general election
results
def plot_general_election_results(df):
    df_filtered = df[['Party', 'Seats Won']].copy()
    df_filtered['Percent'] = (df_filtered['Seats Won'] /
df_filtered['Seats Won'].sum()) * 100

    # Pie Chart
    plt.figure(figsize=(8, 8))
    plt.pie(df_filtered['Seats Won'], labels=df_filtered['Party'],
autopct='%1.1f%%', startangle=140)
    plt.title('General Election Results Distribution')
    plt.savefig('general_pie_chart.png')
    plt.show()

    # Bar Chart
    plt.figure(figsize=(12, 6))
    plt.bar(df_filtered['Party'], df_filtered['Seats Won'])
    plt.title('Number of Seats Won by Each Party')
    plt.xlabel('Party')
    plt.ylabel('Number of Seats')
    plt.xticks(rotation=90)
```

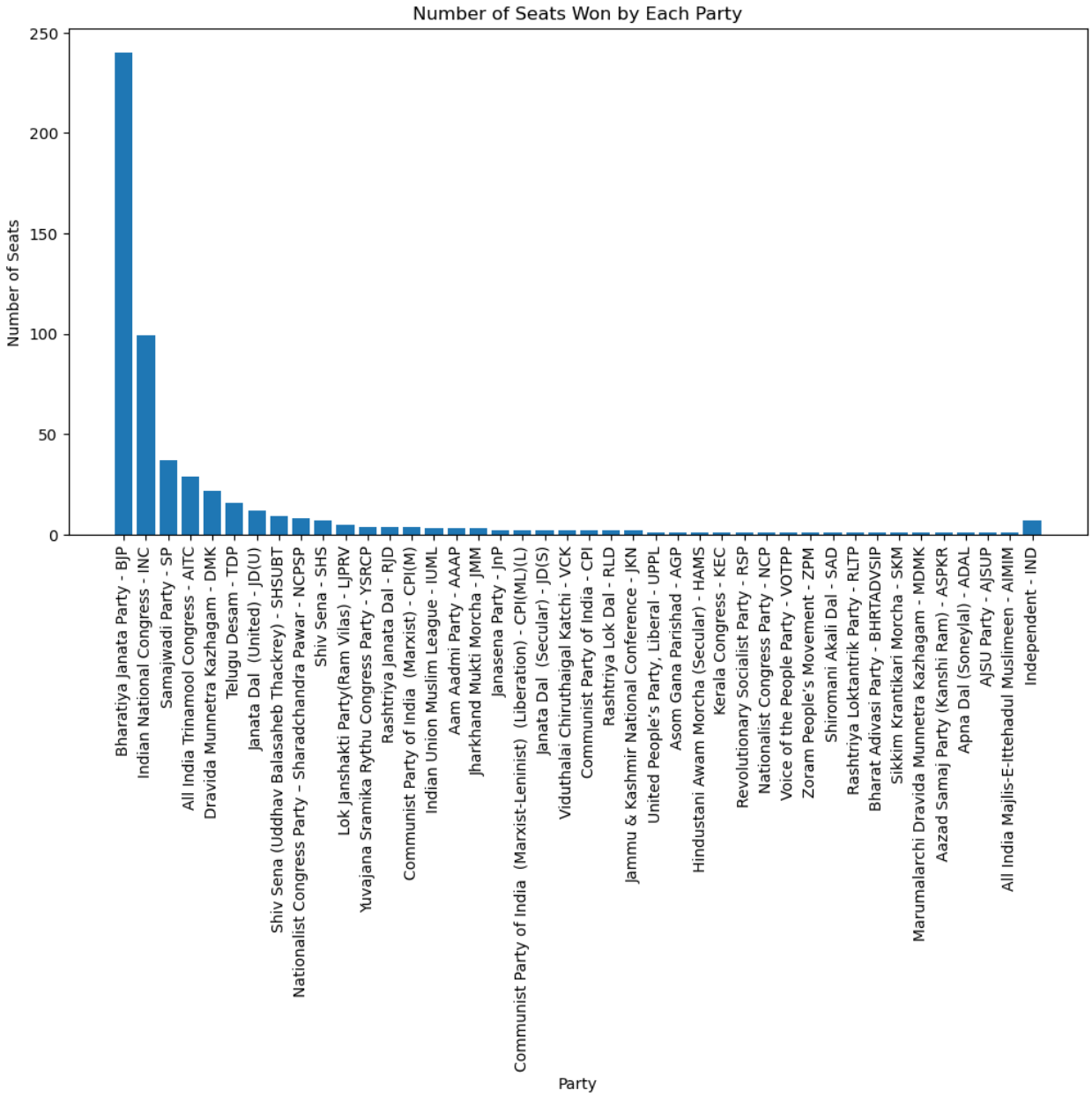
```
    plt.savefig('general_bar_chart.png')
    plt.show()

# Function to fetch and parse data from the bye-election results page
def fetch_by_election_results(url):
    print(f"Fetching data from {url}...")

    # Fetch the HTML content of the page
    response = requests.get(url)
    if response.status_code == 200:
        print(f"Successfully accessed the page: {url}")
    else:
        print(f"Failed to access the page: {url}. Status code:
{response.status_code}")
        return None  # Return None if page access fails

    # Parse HTML using BeautifulSoup
    soup = BeautifulSoup(response.content, 'html.parser')
    print("HTML content fetched and parsed.")

    # Extract data from the div elements with class 'const-box'
    const_boxes = soup.find_all('div', class_='const-box')

    if not const_boxes:
        print("No constituency boxes found.")
        return None

    print(f"Found {len(const_boxes)} constituency boxes.")

    # Initialize lists to store extracted data
    constituencies = []
    states = []
    results = []
    candidates = []
    parties = []

    for box in const_boxes:
        constituency = box.find('h3').text.strip()
        state = box.find('h4').text.strip()
        result = box.find('h2').text.strip()
        candidate = box.find('h5').text.strip()
        party = box.find('h6').text.strip()

        constituencies.append(constituency)
        states.append(state)
        results.append(result)
        candidates.append(candidate)
        parties.append(party)

    print(f"Extracted data for {len(constituencies)} constituencies.")
```

```
    df = pd.DataFrame({
        'Constituency': constituencies,
        'State': states,
        'Result': results,
        'Candidate': candidates,
        'Party': parties
    })

    # Save the data to a CSV file
    csv_filename = url.split('/')[-2] + '_results.csv'
    df.to_csv(csv_filename, index=False)
    print(f"Data saved to {csv_filename}.")

    # Display the DataFrame
    print(df.head())  # Print the first few rows of the DataFrame to
check the data

    # Plot the data
    plot_bye_election_results(df)

    return df

# Function to plot pie chart and bar chart for the bye-election
results
def plot_bye_election_results(df):
    # Pie Chart
    result_counts = df['Result'].value_counts()
    plt.figure(figsize=(8, 8))
    plt.pie(result_counts, labels=result_counts.index, autopct='%1.1f%
%', startangle=140)
    plt.title('Bye-Election Results Distribution')
    plt.savefig('bye_pie_chart.png')
    plt.show()

    # Bar Chart
    party_counts = df['Party'].value_counts()
    plt.figure(figsize=(10, 6))
    party_counts.plot(kind='bar')
    plt.title('Number of Seats Won by Each Party')
    plt.xlabel('Party')
    plt.ylabel('Number of Seats')
    plt.xticks(rotation=90)
    plt.savefig('bye_bar_chart.png')
    plt.show()

# Function to fetch data from multiple pages
def fetch_data_from_multiple_urls(urls, fetch_function):
    for url in urls:
        df = fetch_function(url)
```

```
        if df is not None:  # Ensure we only save if df is valid
            # Save data to CSV
            csv_filename = url.split('/')[-2] + '_results.csv'
            df.to_csv(csv_filename, index=False)
            print(f"Data saved to {csv_filename}.")

# Define the URLs
general_election_url =
'https://results.eci.gov.in/PcResultGenJune2024/index.htm'

# Fetch data from the general election results page
fetch_general_election_results(general_election_url)

# Define the URLs for bye-elections
bye_election_urls = [
    'https://results.eci.gov.in/AcResultByeJune2024/'

]

# Fetch data from the bye-election results pages
fetch_data_from_multiple_urls(bye_election_urls,
fetch_by_election_results)
```

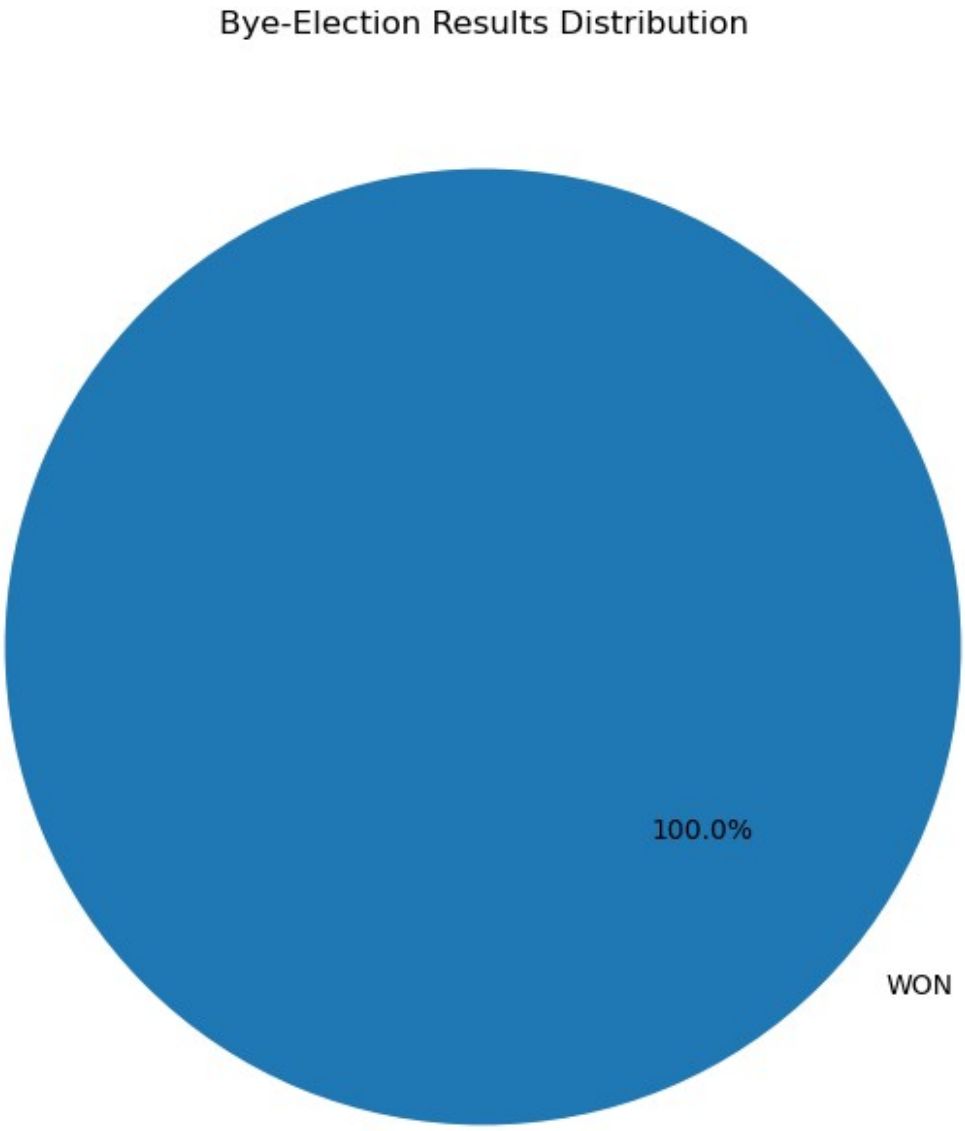
```
Fetching data from
https://results.eci.gov.in/PcResultGenJune2024/index.htm...
Successfully accessed the page:
https://results.eci.gov.in/PcResultGenJune2024/index.htm
HTML content fetched and parsed.
Results table found.
Extracted data for 42 parties.
Data saved to general_election_results.csv.
                                Party  Seats Won  Seats Leading
Total Seats
0         Bharatiya Janata Party - BJP        240              0
240
1       Indian National Congress - INC         99              0
99
2                 Samajwadi Party - SP         37              0
37
3  All India Trinamool Congress - AITC         29              0
29
4      Dravida Munnetra Kazhagam - DMK         22              0
22
```

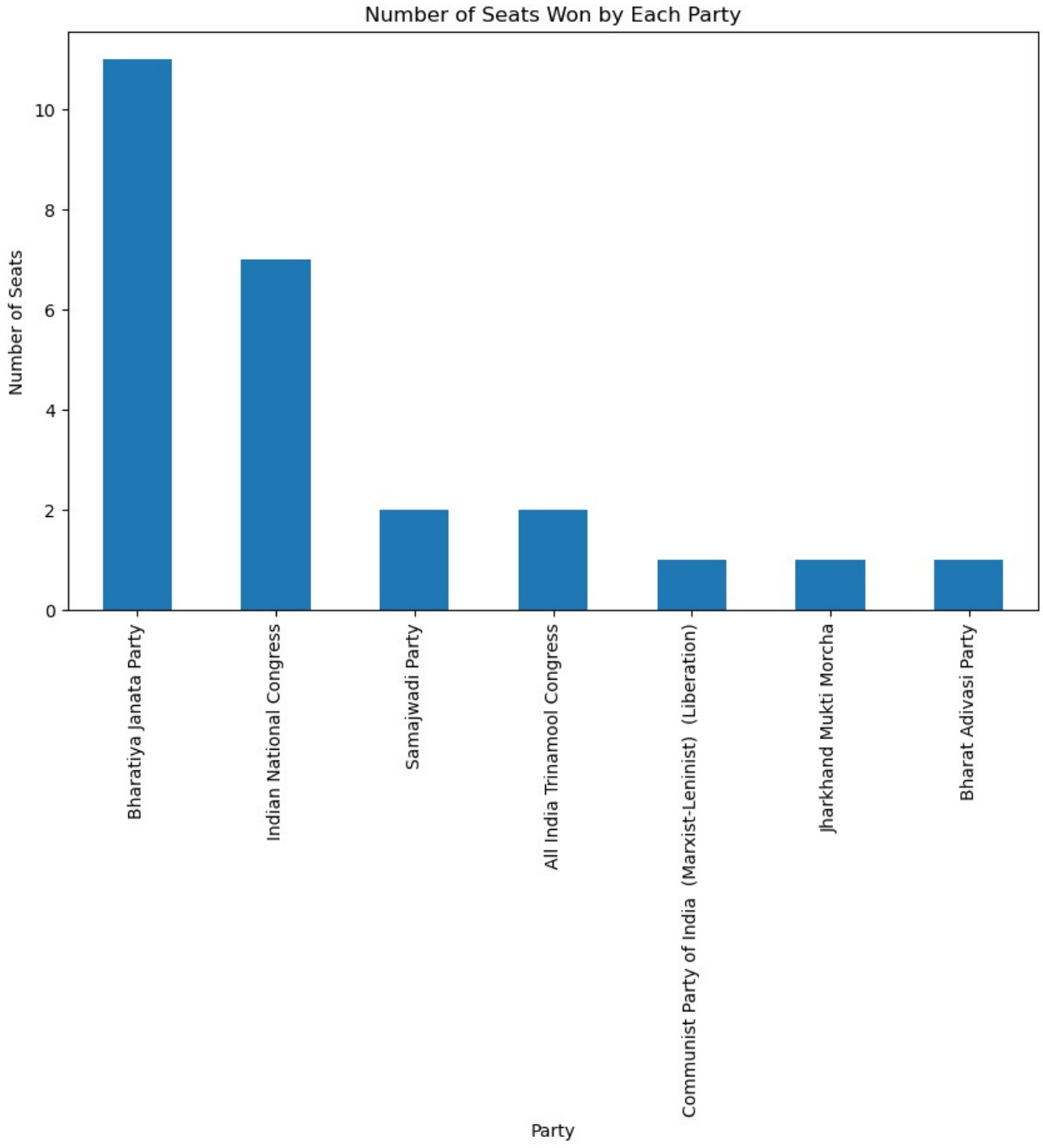

General Election Results Distribution
Shiv Sena (Uddhav Balasaheb Thackrey) - SHSUBT
Janata Dal (United) - JDU
Telugu Desam - TDP
Dravida Munnetra Kazhagam - DMK
All India Trinamool Congress - AITC
Samajwadi Party - SP
Indian National Congress - INC
Bharatiya Janata Party - BJP
Communist Party of India (Marxist-Leninist)
Independent - IND
44.2%
18.2%
6.8%
5.3%
4.1%
2.9%
2.2%




```
Fetching data from https://results.eci.gov.in/AcResultByeJune2024/...
Successfully accessed the page:
https://results.eci.gov.in/AcResultByeJune2024/
HTML content fetched and parsed.
Found 25 constituency boxes.
Extracted data for 25 constituencies.
Data saved to AcResultByeJune2024_results.csv.
     Constituency    State Result                    Candidate  \
0    Agiaon (195)    Bihar    WON          SHIV PRAKASH RANJAN
1    Vijapur (26)  Gujarat    WON             DR. C. J. CHAVDA
2  Porbandar (83)  Gujarat    WON     ARJUN DEVABHAI MODHWADIA
3  Manavadar (85)  Gujarat    WON   ARVINDBHAI JINABHAI LADANI
```

```
4  Khambhat (108)  Gujarat    WON  CHIRAGKUMAR ARVINDBHAI PATEL

                                                  Party
0  Communist Party of India  (Marxist-Leninist)  ...
1                              Bharatiya Janata Party
2                              Bharatiya Janata Party
3                              Bharatiya Janata Party
4                              Bharatiya Janata Party
```






```
Data saved to AcResultByeJune2024_results.csv.
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data from the CSV file
csv_filename = 'general_election_results.csv'
df = pd.read_csv(csv_filename)
```

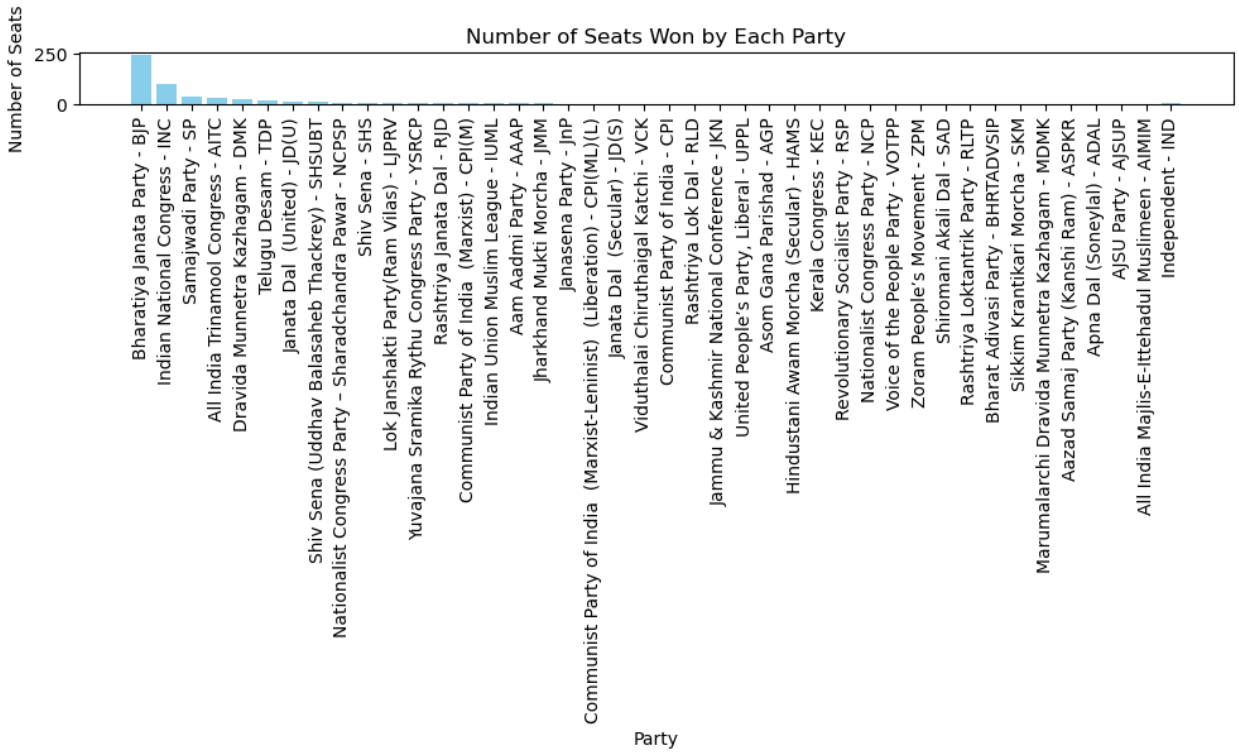
```python
# Display the first few rows of the DataFrame
print("First few rows of the DataFrame:")
print(df.head())

# Summary statistics
print("\nSummary statistics:")
print(df.describe())

# Total seats won by each party
print("\nTotal seats won by each party:")
print(df[['Party', 'Seats Won']])

# Plot the number of seats won by each party
def plot_seats_won(df):
    plt.figure(figsize=(10, 6))
    plt.bar(df['Party'], df['Seats Won'], color='skyblue')
    plt.title('Number of Seats Won by Each Party')
    plt.xlabel('Party')
    plt.ylabel('Number of Seats')
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.savefig('seats_won_by_party.png')
    plt.show()

# Call the plot function
plot_seats_won(df)

# Additional insights
total_seats = df['Seats Won'].sum()
print(f"\nTotal seats contested: {total_seats}")

# Party with the most seats won
max_seats = df['Seats Won'].max()
party_max_seats = df[df['Seats Won'] == max_seats]['Party'].values[0]
print(f"The party with the most seats won is {party_max_seats} with
{max_seats} seats.")

# Seats leading by each party
print("\nSeats leading by each party:")
print(df[['Party', 'Seats Leading']])

# Plot the number of seats leading by each party
def plot_seats_leading(df):
    plt.figure(figsize=(10, 6))
    plt.bar(df['Party'], df['Seats Leading'], color='lightcoral')
    plt.title('Number of Seats Leading by Each Party')
    plt.xlabel('Party')
    plt.ylabel('Number of Seats Leading')
    plt.xticks(rotation=90)
    plt.tight_layout()
```

```
    plt.savefig('seats_leading_by_party.png')
    plt.show()

# Call the plot function
plot_seats_leading(df)

# Percentage of total seats won by each party
df['Percentage of Total Seats'] = (df['Seats Won'] / total_seats) *
100
print("\nPercentage of total seats won by each party:")
print(df[['Party', 'Percentage of Total Seats']])

# Plot the percentage of total seats won by each party
def plot_percentage_seats_won(df):
    plt.figure(figsize=(10, 6))
    plt.pie(df['Percentage of Total Seats'], labels=df['Party'],
autopct='%1.1f%%', startangle=140)
    plt.title('Percentage of Total Seats Won by Each Party')
    plt.tight_layout()
    plt.savefig('percentage_seats_won_by_party.png')
    plt.show()

# Call the plot function
plot_percentage_seats_won(df)
```

```
First few rows of the DataFrame:
                                  Party  Seats Won  Seats Leading
Total Seats
0         Bharatiya Janata Party - BJP        240              0
240
1       Indian National Congress - INC         99              0
99
2                 Samajwadi Party - SP         37              0
37
3  All India Trinamool Congress - AITC         29              0
29
4      Dravida Munnetra Kazhagam - DMK         22              0
22

Summary statistics:
        Seats Won  Seats Leading  Total Seats
count   42.000000           42.0    42.000000
mean    12.928571            0.0    12.928571
std     39.466808            0.0    39.466808
min      1.000000            0.0     1.000000
25%      1.000000            0.0     1.000000
50%      2.000000            0.0     2.000000
75%      6.500000            0.0     6.500000
max    240.000000            0.0   240.000000
```

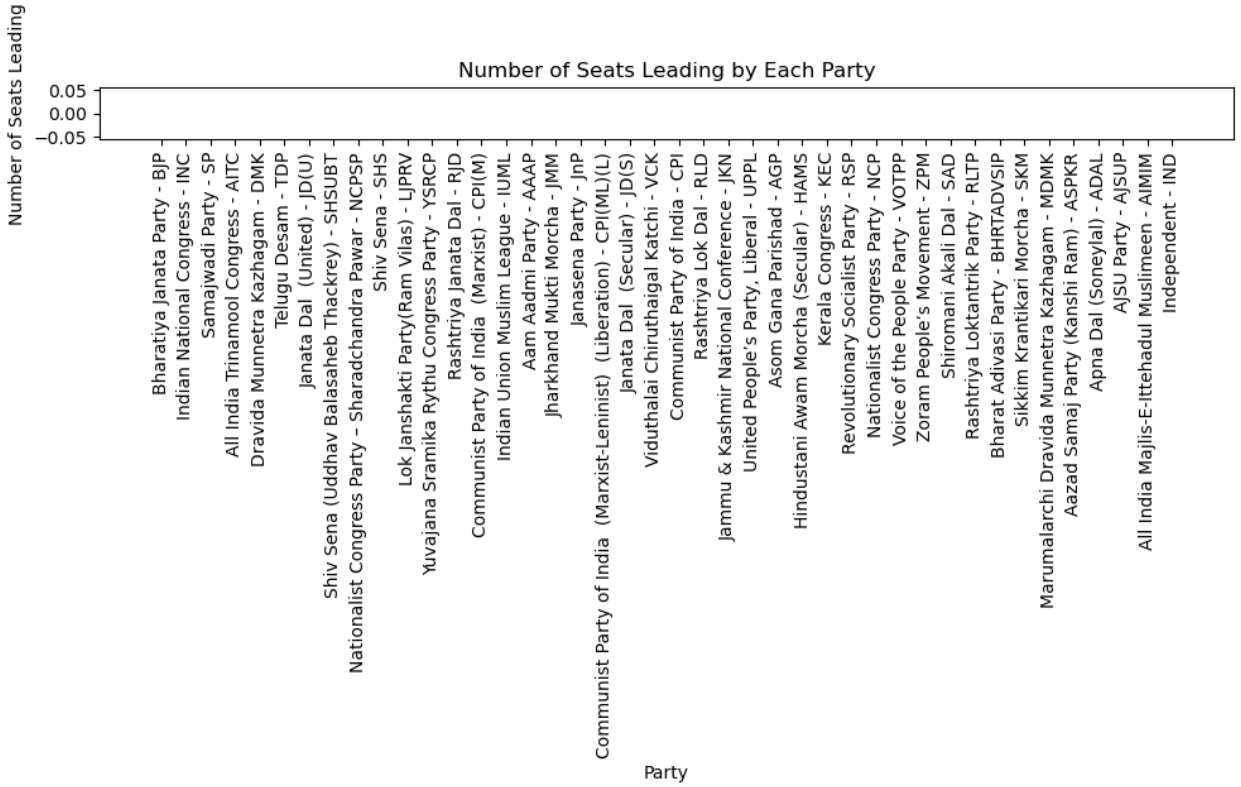
Total seats won by each party:

	Party	Seats Won
0	Bharatiya Janata Party - BJP	240
1	Indian National Congress - INC	99
2	Samajwadi Party - SP	37
3	All India Trinamool Congress - AITC	29
4	Dravida Munnetra Kazhagam - DMK	22
5	Telugu Desam - TDP	16
6	Janata Dal (United) - JD(U)	12
7	Shiv Sena (Uddhav Balasaheb Thackrey) - SHSUBT	9
8	Nationalist Congress Party – Sharadchandra Paw...	8
9	Shiv Sena - SHS	7
10	Lok Janshakti Party(Ram Vilas) - LJPRV	5
11	Yuvajana Sramika Rythu Congress Party - YSRCP	4
12	Rashtriya Janata Dal - RJD	4
13	Communist Party of India (Marxist) - CPI(M)	4
14	Indian Union Muslim League - IUML	3
15	Aam Aadmi Party - AAAP	3
16	Jharkhand Mukti Morcha - JMM	3
17	Janasena Party - JnP	2
18	Communist Party of India (Marxist-Leninist) ...	2
19	Janata Dal (Secular) - JD(S)	2
20	Viduthalai Chiruthaigal Katchi - VCK	2
21	Communist Party of India - CPI	2
22	Rashtriya Lok Dal - RLD	2
23	Jammu & Kashmir National Conference - JKN	2
24	United People’s Party, Liberal - UPPL	1
25	Asom Gana Parishad - AGP	1
26	Hindustani Awam Morcha (Secular) - HAMS	1
27	Kerala Congress - KEC	1
28	Revolutionary Socialist Party - RSP	1
29	Nationalist Congress Party - NCP	1
30	Voice of the People Party - VOTPP	1
31	Zoram People’s Movement - ZPM	1
32	Shiromani Akali Dal - SAD	1
33	Rashtriya Loktantrik Party - RLTP	1
34	Bharat Adivasi Party - BHRTADVSIP	1
35	Sikkim Krantikari Morcha - SKM	1
36	Marumalarchi Dravida Munnetra Kazhagam - MDMK	1
37	Aazad Samaj Party (Kanshi Ram) - ASPKR	1
38	Apna Dal (Soneylal) - ADAL	1
39	AJSU Party - AJSUP	1
40	All India Majlis-E-Ittehadul Muslimeen - AIMIM	1
41	Independent - IND	7




```
Total seats contested: 543
The party with the most seats won is Bharatiya Janata Party - BJP with
240 seats.

Seats leading by each party:
                                                Party  Seats Leading
0                         Bharatiya Janata Party - BJP              0
1                       Indian National Congress - INC              0
2                                Samajwadi Party - SP               0
3                   All India Trinamool Congress - AITC              0
4                       Dravida Munnetra Kazhagam - DMK              0
5                                    Telugu Desam - TDP              0
6                          Janata Dal  (United) - JD(U)              0
7       Shiv Sena (Uddhav Balasaheb Thackrey) - SHSUBT              0
8   Nationalist Congress Party – Sharadchandra Paw...              0
9                                      Shiv Sena - SHS              0
10              Lok Janshakti Party(Ram Vilas) - LJPRV              0
11       Yuvajana Sramika Rythu Congress Party - YSRCP              0
12                         Rashtriya Janata Dal - RJD              0
13        Communist Party of India  (Marxist) - CPI(M)              0
14                  Indian Union Muslim League - IUML              0
15                               Aam Aadmi Party - AAAP              0
16                         Jharkhand Mukti Morcha - JMM              0
17                                Janasena Party - JnP               0
18  Communist Party of India  (Marxist-Leninist)  ...              0
```

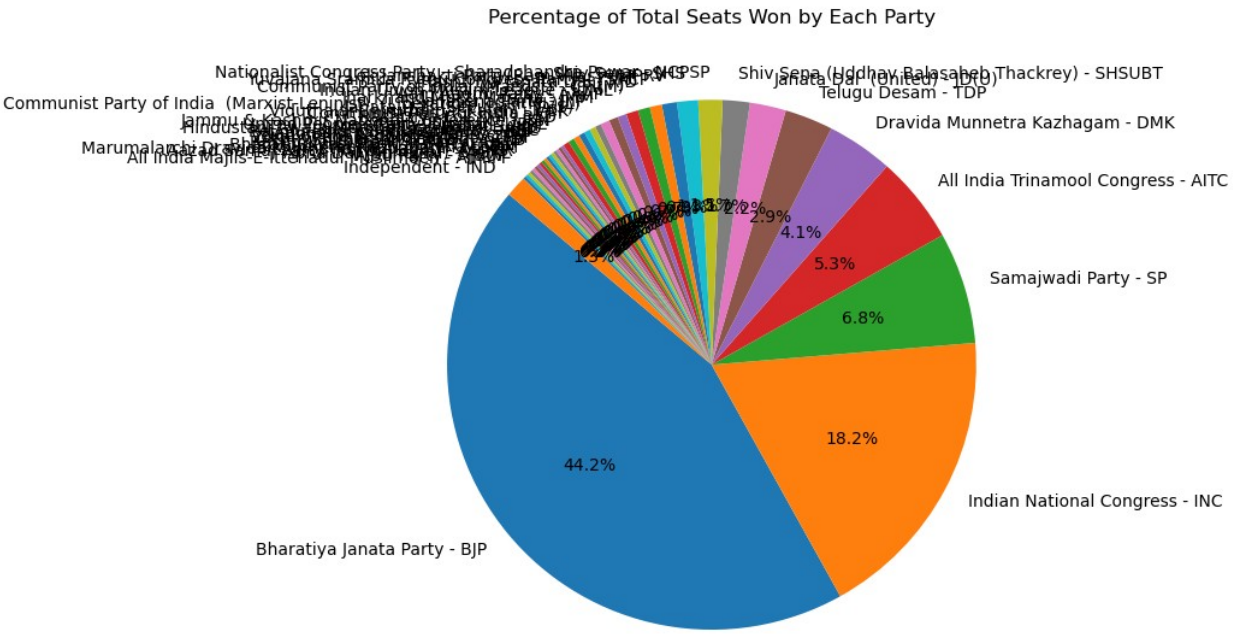
19	Janata Dal (Secular) - JD(S)	0
20	Viduthalai Chiruthaigal Katchi - VCK	0
21	Communist Party of India - CPI	0
22	Rashtriya Lok Dal - RLD	0
23	Jammu & Kashmir National Conference - JKN	0
24	United People’s Party, Liberal - UPPL	0
25	Asom Gana Parishad - AGP	0
26	Hindustani Awam Morcha (Secular) - HAMS	0
27	Kerala Congress - KEC	0
28	Revolutionary Socialist Party - RSP	0
29	Nationalist Congress Party - NCP	0
30	Voice of the People Party - VOTPP	0
31	Zoram People’s Movement - ZPM	0
32	Shiromani Akali Dal - SAD	0
33	Rashtriya Loktantrik Party - RLTP	0
34	Bharat Adivasi Party - BHRTADVSIP	0
35	Sikkim Krantikari Morcha - SKM	0
36	Marumalarchi Dravida Munnetra Kazhagam - MDMK	0
37	Aazad Samaj Party (Kanshi Ram) - ASPKR	0
38	Apna Dal (Soneylal) - ADAL	0
39	AJSU Party - AJSUP	0
40	All India Majlis-E-Ittehadul Muslimeen - AIMIM	0
41	Independent - IND	0



Percentage of total seats won by each party:

	Party	Percentage of Total Seats
0	Bharatiya Janata Party - BJP	44.198895
1	Indian National Congress - INC	18.232044
2	Samajwadi Party - SP	6.813996
3	All India Trinamool Congress - AITC	
4	Dravida Munnetra Kazhagam - DMK	
5	Telugu Desam - TDP	
6	Janata Dal (United) - JD(U)	
7	Shiv Sena (Uddhav Balasaheb Thackrey) - SHSUBT	
8	Nationalist Congress Party – Sharadchandra Paw...	
9	Shiv Sena - SHS	
10	Lok Janshakti Party(Ram Vilas) - LJPRV	
11	Yuvajana Sramika Rythu Congress Party - YSRCP	
12	Rashtriya Janata Dal - RJD	
13	Communist Party of India (Marxist) - CPI(M)	
14	Indian Union Muslim League - IUML	
15	Aam Aadmi Party - AAAP	
16	Jharkhand Mukti Morcha - JMM	
17	Janasena Party - JnP	
18	Communist Party of India (Marxist-Leninist) ...	
19	Janata Dal (Secular) - JD(S)	
20	Viduthalai Chiruthaigal Katchi - VCK	
21	Communist Party of India - CPI	
22	Rashtriya Lok Dal - RLD	
23	Jammu & Kashmir National Conference - JKN	
24	United People’s Party, Liberal - UPPL	
25	Asom Gana Parishad - AGP	
26	Hindustani Awam Morcha (Secular) - HAMS	
27	Kerala Congress - KEC	
28	Revolutionary Socialist Party - RSP	
29	Nationalist Congress Party - NCP	
30	Voice of the People Party - VOTPP	
31	Zoram People’s Movement - ZPM	
32	Shiromani Akali Dal - SAD	
33	Rashtriya Loktantrik Party - RLTP	
34	Bharat Adivasi Party - BHRTADVSIP	
35	Sikkim Krantikari Morcha - SKM	
36	Marumalarchi Dravida Munnetra Kazhagam - MDMK	
37	Aazad Samaj Party (Kanshi Ram) - ASPKR	
38	Apna Dal (Soneylal) - ADAL	
39	AJSU Party - AJSUP	
40	All India Majlis-E-Ittehadul Muslimeen - AIMIM	
41	Independent - IND	

3	5.340700
4	4.051565
5	2.946593
6	2.209945
7	1.657459
8	1.473297
9	1.289134
10	0.920810
11	0.736648
12	0.736648
13	0.736648
14	0.552486
15	0.552486
16	0.552486
17	0.368324
18	0.368324
19	0.368324
20	0.368324
21	0.368324
22	0.368324
23	0.368324
24	0.184162
25	0.184162
26	0.184162
27	0.184162
28	0.184162
29	0.184162
30	0.184162
31	0.184162
32	0.184162
33	0.184162
34	0.184162
35	0.184162
36	0.184162
37	0.184162
38	0.184162
39	0.184162
40	0.184162
41	1.289134



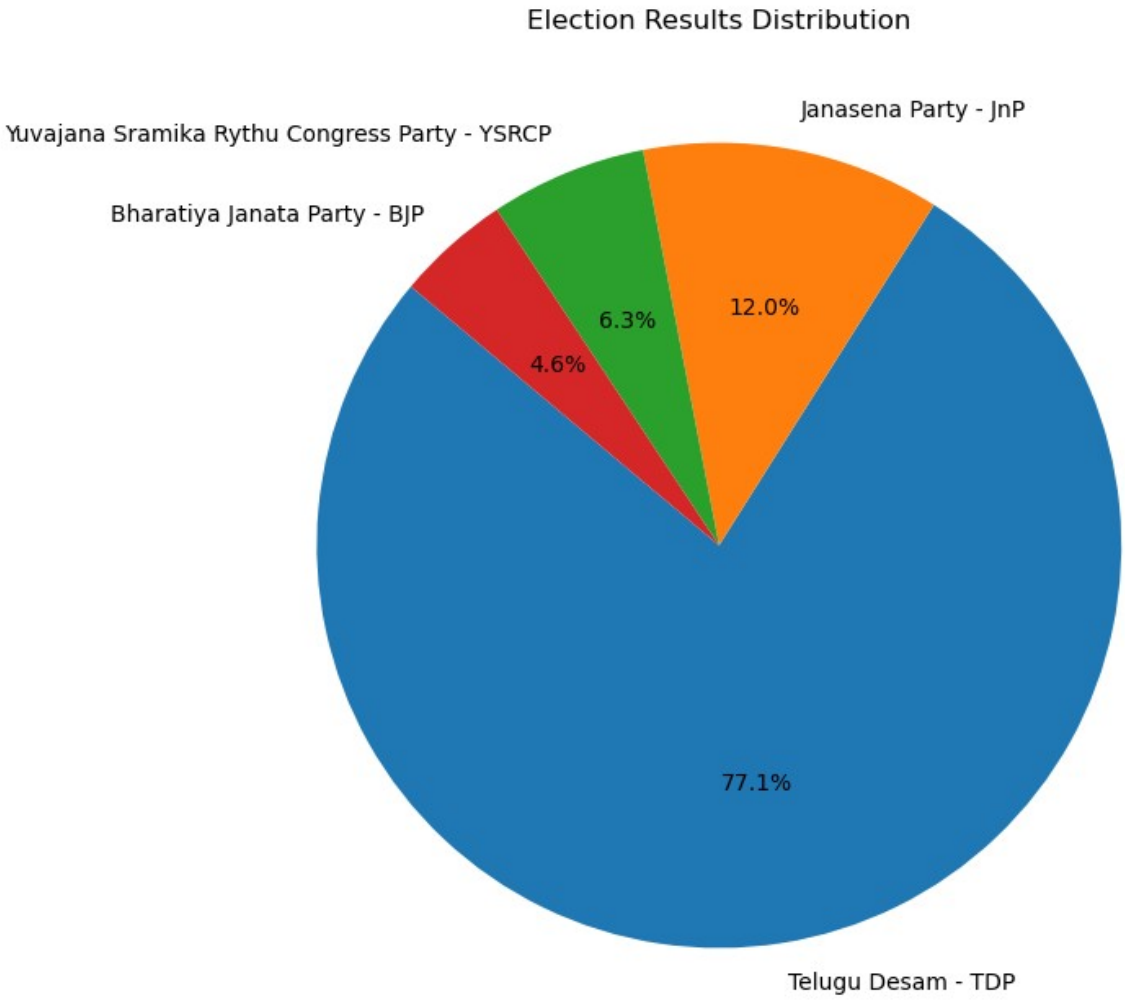

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt

# Function to fetch and parse data from the election results page
def fetch_election_results(url):
    print(f"Fetching data from {url}...")

    # Fetch the HTML content of the page
    response = requests.get(url)
    if response.status_code == 200:
        print(f"Successfully accessed the page: {url}")
    else:
        print(f"Failed to access the page: {url}. Status code:
{response.status_code}")
        return

    # Parse HTML using BeautifulSoup
    soup = BeautifulSoup(response.content, 'html.parser')
    print("HTML content fetched and parsed.")

    # Extract data from the table inside the nested div elements
    results_table = soup.find('div', class_='rslt-
table').find('table', class_='table')
    if results_table is None:
        print("No results table found.")
        return
```

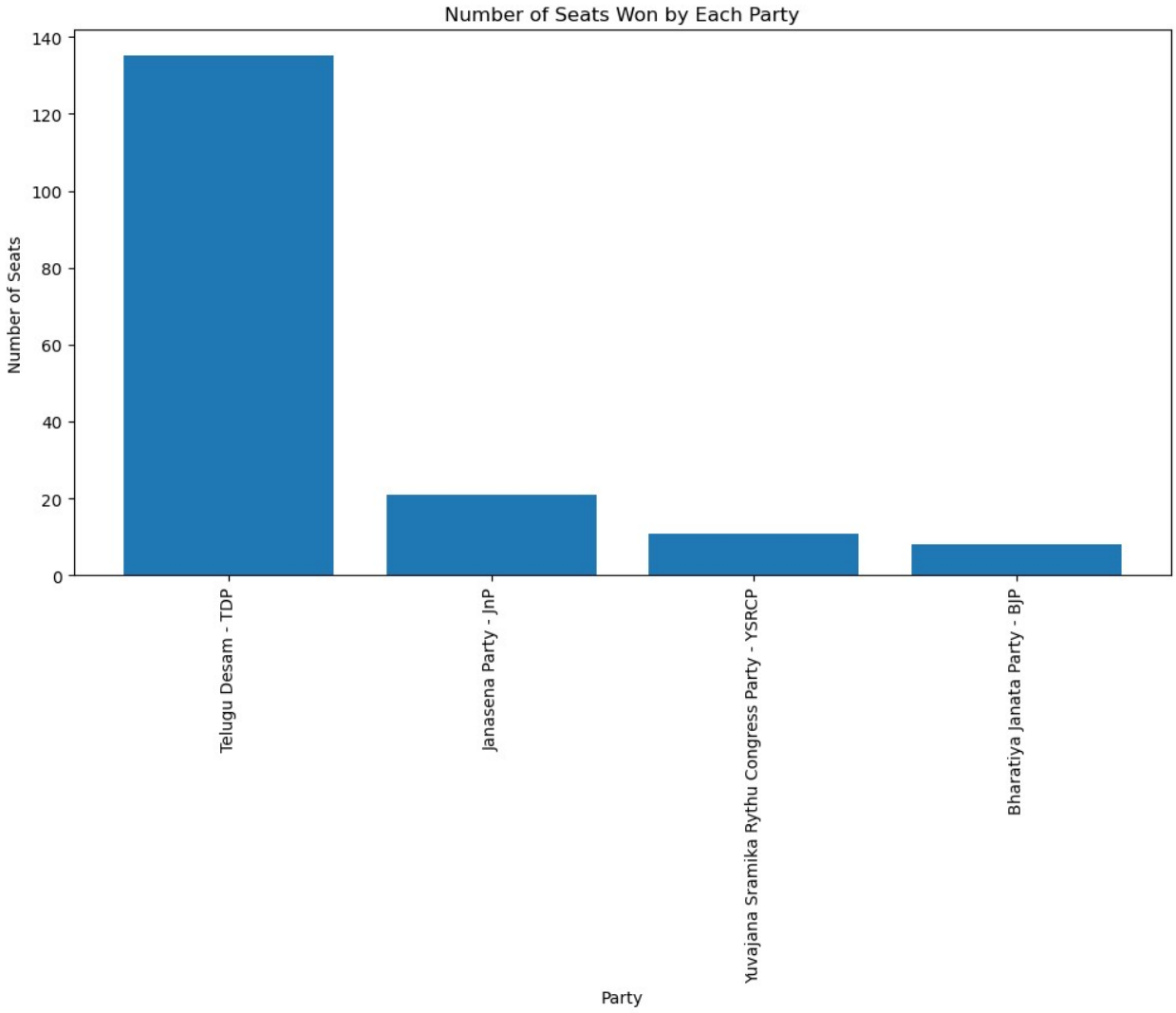
```
    print("Results table found.")

    # Initialize lists to store extracted data
    parties = []
    won = []
    leading = []
    total = []

    for row in results_table.find('tbody').find_all('tr'):
        cells = row.find_all('td')
        if len(cells) < 4:
            print("Skipping row with insufficient columns.")
            continue  # Skip rows that do not have enough columns

        party_name = cells[0].text.strip()  # Party name
        won_seats = cells[1].text.strip()  # Number of seats won
        leading_seats = cells[2].text.strip()  # Number of seats
leading
        total_seats = cells[3].text.strip()  # Total seats contested

        parties.append(party_name)
        won.append(int(won_seats))  # Convert to integer
        leading.append(int(leading_seats))  # Convert to integer
        total.append(int(total_seats))  # Convert to integer

    print(f"Extracted data for {len(parties)} parties.")

    # Create a DataFrame to store the extracted data
    df = pd.DataFrame({
        'Party': parties,
        'Seats Won': won,
        'Seats Leading': leading,
        'Total Seats': total
    })

    # Save the data to a CSV file
    csv_filename = 'party_wise_election_results.csv'
    df.to_csv(csv_filename, index=False)
    print(f"Data saved to {csv_filename}.")

    # Display the DataFrame
    print(df.head())  # Print the first few rows of the DataFrame to
check the data

    # Plot the data
    plot_election_results(df)

    return df
```

```
# Function to plot pie chart and bar chart for the election results
def plot_election_results(df):
    df_filtered = df[['Party', 'Seats Won']].copy()
    df_filtered['Percent'] = (df_filtered['Seats Won'] /
df_filtered['Seats Won'].sum()) * 100

    # Pie Chart
    plt.figure(figsize=(8, 8))
    plt.pie(df_filtered['Seats Won'], labels=df_filtered['Party'],
autopct='%1.1f%%', startangle=140)
    plt.title('Election Results Distribution')
    plt.savefig('party_wise_pie_chart.png')
    plt.show()

    # Bar Chart
    plt.figure(figsize=(12, 6))
    plt.bar(df_filtered['Party'], df_filtered['Seats Won'])
    plt.title('Number of Seats Won by Each Party')
    plt.xlabel('Party')
    plt.ylabel('Number of Seats')
    plt.xticks(rotation=90)
    plt.savefig('party_wise_bar_chart.png')
    plt.show()

# Fetch data from the election results page
url = 'https://results.eci.gov.in/AcResultGenJune2024/partywiseresult-
S01.htm'
df = fetch_election_results(url)

# Analyze the fetched data
if df is not None:
    print("Data analysis:")
    # Display the first few rows of the DataFrame
    print("First few rows of the DataFrame:")
    print(df.head())

    # Summary statistics
    print("\nSummary statistics:")
    print(df.describe())

    # Total seats won by each party
    print("\nTotal seats won by each party:")
    print(df[['Party', 'Seats Won']])

    # Additional insights
    total_seats = df['Seats Won'].sum()
    print(f"\nTotal seats contested: {total_seats}")

    # Party with the most seats won
    max_seats = df['Seats Won'].max()
```

```
    party_max_seats = df[df['Seats Won'] == max_seats]
['Party'].values[0]
    print(f"The party with the most seats won is {party_max_seats}
with {max_seats} seats.")

    # Seats leading by each party
    print("\nSeats leading by each party:")
    print(df[['Party', 'Seats Leading']])

    # Percentage of total seats won by each party
    df['Percentage of Total Seats'] = (df['Seats Won'] / total_seats)
* 100
    print("\nPercentage of total seats won by each party:")
    print(df[['Party', 'Percentage of Total Seats']])
```

```
Fetching data from
https://results.eci.gov.in/AcResultGenJune2024/partywiseresult-
S01.htm...
Successfully accessed the page:
https://results.eci.gov.in/AcResultGenJune2024/partywiseresult-S01.htm
HTML content fetched and parsed.
Results table found.
Extracted data for 4 parties.
Data saved to party_wise_election_results.csv.
                                            Party  Seats Won  Seats
Leading  \
0                              Telugu Desam - TDP        135
0
1                            Janasena Party - JnP         21
0
2  Yuvajana Sramika Rythu Congress Party - YSRCP         11
0
3                    Bharatiya Janata Party - BJP          8
0

   Total Seats
0          135
1           21
2           11
3            8
```


Election Results Distribution
Janasena Party - JnP
Yuvajana Sramika Rythu Congress Party - YSRCP
Bharatiya Janata Party - BJP
12.0%
6.3%
4.6%
77.1%
Telugu Desam - TDP




```
Data analysis:
First few rows of the DataFrame:
                                          Party  Seats Won  Seats
Leading  \
0                            Telugu Desam - TDP        135
0
1                          Janasena Party - JnP         21
0
2  Yuvajana Sramika Rythu Congress Party - YSRCP         11
0
3                   Bharatiya Janata Party - BJP          8
0

   Total Seats
0          135
1           21
2           11
```

```
3            8

Summary statistics:
        Seats Won  Seats Leading  Total Seats
count    4.000000            4.0     4.000000
mean    43.750000            0.0    43.750000
std     61.086687            0.0    61.086687
min      8.000000            0.0     8.000000
25%     10.250000            0.0    10.250000
50%     16.000000            0.0    16.000000
75%     49.500000            0.0    49.500000
max    135.000000            0.0   135.000000

Total seats won by each party:
                                            Party  Seats Won
0                            Telugu Desam - TDP         135
1                          Janasena Party - JnP          21
2  Yuvajana Sramika Rythu Congress Party - YSRCP          11
3                   Bharatiya Janata Party - BJP           8

Total seats contested: 175
The party with the most seats won is Telugu Desam - TDP with 135
seats.

Seats leading by each party:
                                            Party  Seats Leading
0                            Telugu Desam - TDP              0
1                          Janasena Party - JnP              0
2  Yuvajana Sramika Rythu Congress Party - YSRCP              0
3                   Bharatiya Janata Party - BJP              0

Percentage of total seats won by each party:
                                            Party  Percentage of Total
Seats
0                            Telugu Desam - TDP
77.142857
1                          Janasena Party - JnP
12.000000
2  Yuvajana Sramika Rythu Congress Party - YSRCP
6.285714
3                   Bharatiya Janata Party - BJP
4.571429
```

```python
import pandas as pd

# Load the CSV file into a DataFrame
df = pd.read_csv('party_wise_election_results.csv')
print(df.head())
```

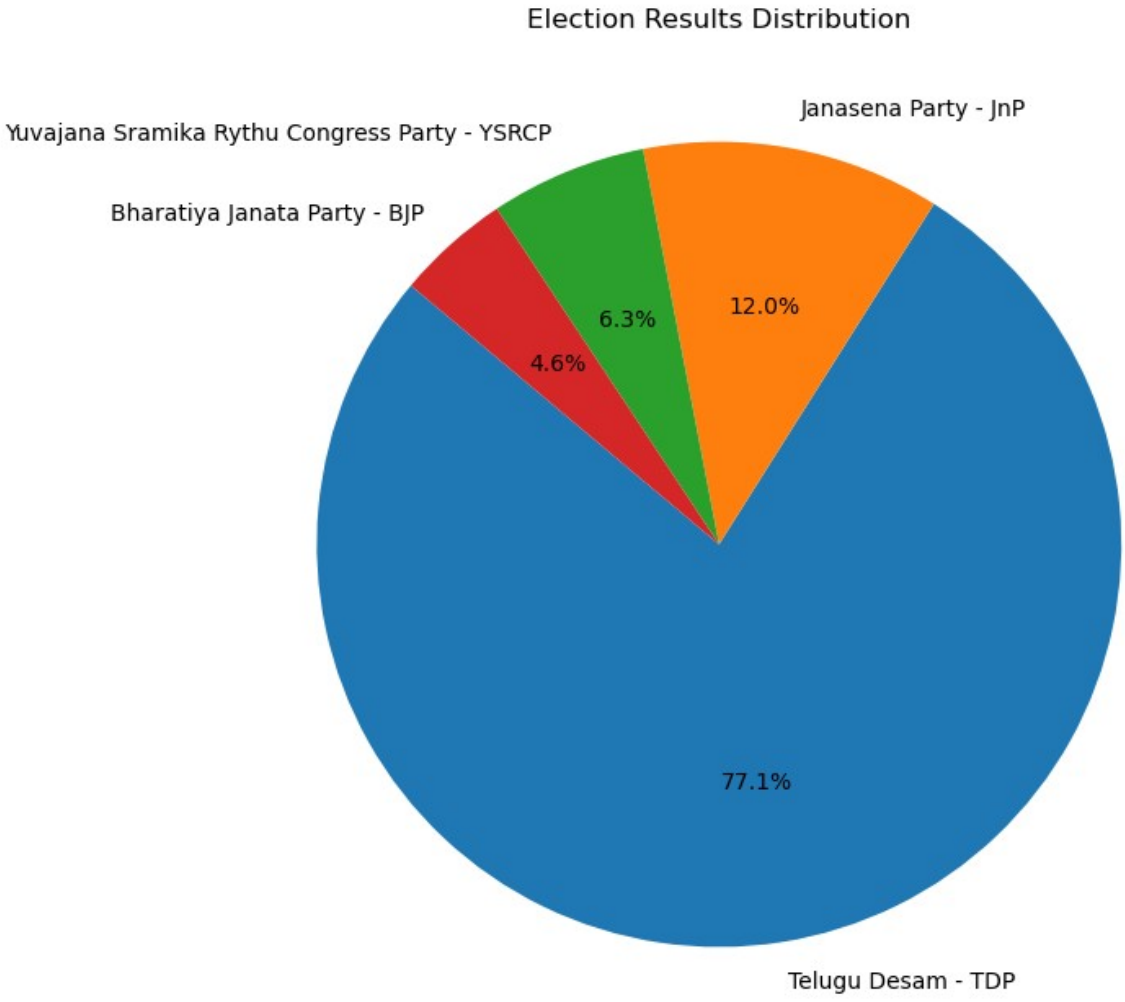
```
                                          Party  Seats Won  Seats
Leading  \
0                            Telugu Desam - TDP        135
0
1                          Janasena Party - JnP         21
0
2  Yuvajana Sramika Rythu Congress Party - YSRCP         11
0
3                   Bharatiya Janata Party - BJP          8
0

   Total Seats
0          135
1           21
2           11
3            8
```

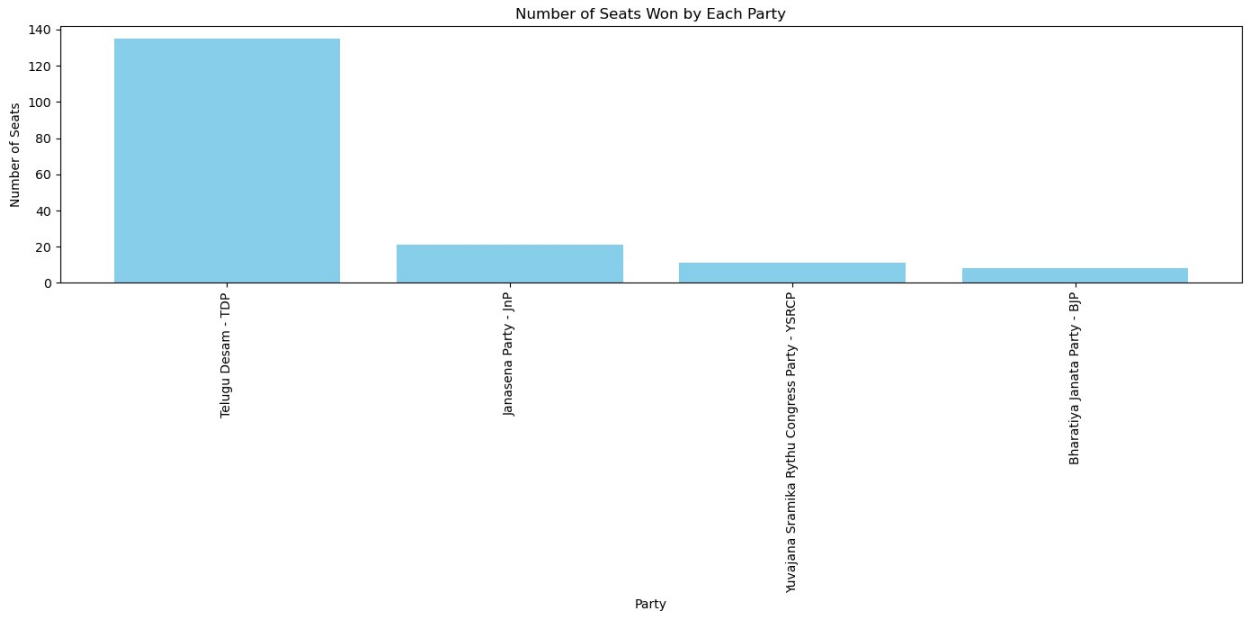
```python
# Summary statistics
print("Summary statistics:")
print(df.describe())

# Total seats won
total_seats_won = df['Seats Won'].sum()
print(f"\nTotal seats won by all parties: {total_seats_won}")

# Party with the maximum seats won
max_seats = df['Seats Won'].max()
party_max_seats = df[df['Seats Won'] == max_seats]['Party'].values[0]
print(f"The party with the most seats won is {party_max_seats} with
{max_seats} seats.")

# Party with the least seats won
min_seats = df['Seats Won'].min()
party_min_seats = df[df['Seats Won'] == min_seats]['Party'].values[0]
print(f"The party with the least seats won is {party_min_seats} with
{min_seats} seats.")
```

```
Summary statistics:
        Seats Won  Seats Leading  Total Seats
count    4.000000            4.0     4.000000
mean    43.750000            0.0    43.750000
std     61.086687            0.0    61.086687
min      8.000000            0.0     8.000000
25%     10.250000            0.0    10.250000
50%     16.000000            0.0    16.000000
75%     49.500000            0.0    49.500000
max    135.000000            0.0   135.000000

Total seats won by all parties: 175
The party with the most seats won is Telugu Desam - TDP with 135
```

```
seats.
The party with the least seats won is Bharatiya Janata Party - BJP
with 8 seats.

# Add percentage of total seats
df['Percentage of Total Seats'] = (df['Seats Won'] / total_seats_won)
* 100

print("\nPercentage of total seats won by each party:")
print(df[['Party', 'Percentage of Total Seats']])

# Highest percentage of seats won
max_percentage = df['Percentage of Total Seats'].max()
party_max_percentage = df[df['Percentage of Total Seats'] ==
max_percentage]['Party'].values[0]
print(f"The party with the highest percentage of total seats won is
{party_max_percentage} with {max_percentage:.2f}% of the seats.")


Percentage of total seats won by each party:
                                            Party  Percentage of Total
Seats
0                              Telugu Desam - TDP
77.142857
1                            Janasena Party - JnP
12.000000
2  Yuvajana Sramika Rythu Congress Party - YSRCP
6.285714
3                    Bharatiya Janata Party - BJP
4.571429
The party with the highest percentage of total seats won is Telugu
Desam - TDP with 77.14% of the seats.

import matplotlib.pyplot as plt

# Pie Chart for Seats Won
plt.figure(figsize=(10, 8))
plt.pie(df['Seats Won'], labels=df['Party'], autopct='%1.1f%%',
startangle=140)
plt.title('Election Results Distribution')
plt.savefig('party_wise_pie_chart.png')
plt.show()

# Bar Chart for Number of Seats Won
plt.figure(figsize=(14, 7))
plt.bar(df['Party'], df['Seats Won'], color='skyblue')
plt.title('Number of Seats Won by Each Party')
plt.xlabel('Party')
plt.ylabel('Number of Seats')
plt.xticks(rotation=90)
```

```
plt.tight_layout()
plt.savefig('party_wise_bar_chart.png')
plt.show()
```






```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt

# Function to fetch and parse data from the party-wise results page
def fetch_party_wise_results(url):
    print(f"Fetching data from {url}...")

    # Fetch the HTML content of the page
    response = requests.get(url)
    if response.status_code == 200:
        print(f"Successfully accessed the page: {url}")
    else:
        print(f"Failed to access the page: {url}. Status code:
{response.status_code}")
        return

    # Parse HTML using BeautifulSoup
    soup = BeautifulSoup(response.content, 'html.parser')
    print("HTML content fetched and parsed.")

    # Extract data from the first table in the page
    results_table = soup.find('table', class_='table')
    if results_table is None:
        print("No results table found.")
        return

    print("Results table found.")

    # Initialize lists to store extracted data
```

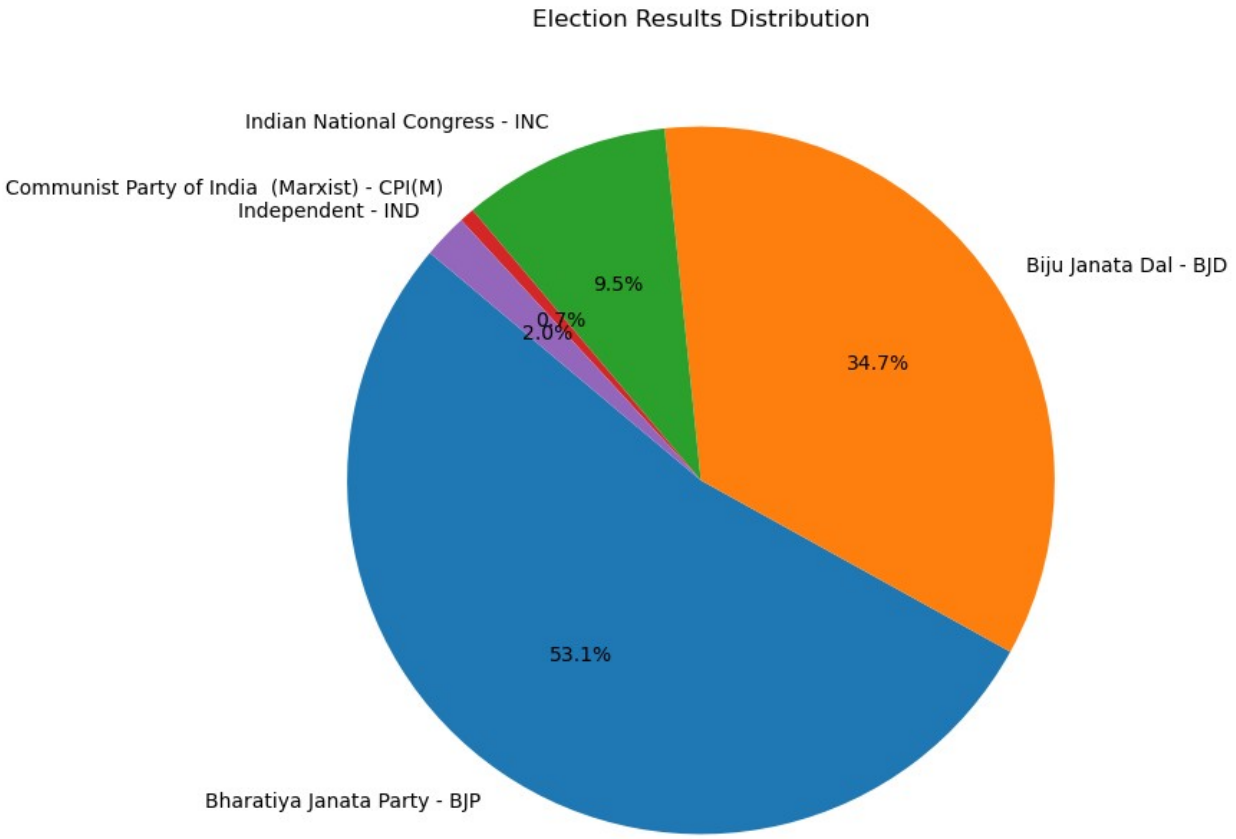
```
    parties = []
    won = []
    leading = []
    total = []

    for row in results_table.find('tbody').find_all('tr'):
        cells = row.find_all('td')
        if len(cells) < 4:
            print("Skipping row with insufficient columns.")
            continue  # Skip rows that do not have enough columns

        parties.append(cells[0].text.strip())  # Party name
        won.append(cells[1].text.strip())     # Number of seats won
        leading.append(cells[2].text.strip()) # Number of seats
leading
        total.append(cells[3].text.strip())   # Total seats contested

    print(f"Extracted data for {len(parties)} parties.")

    # Create a DataFrame to store the extracted data
    df = pd.DataFrame({
        'Party': parties,
        'Seats Won': won,
        'Seats Leading': leading,
        'Total Seats': total
    })

    # Convert columns to appropriate data types
    df['Seats Won'] = df['Seats Won'].astype(int)
    df['Seats Leading'] = df['Seats Leading'].astype(int)
    df['Total Seats'] = df['Total Seats'].astype(int)

    # Save the data to a CSV file
    csv_filename = 'party_wise_election_results_S18.csv'
    df.to_csv(csv_filename, index=False)
    print(f"Data saved to {csv_filename}.")

    # Display the DataFrame
    print(df.head())  # Print the first few rows of the DataFrame to
check the data

    # Perform analysis
    perform_analysis(df)

    return df

# Function to perform analysis on the DataFrame
def perform_analysis(df):
    print("\nBasic Statistics")
    print(df.describe())
```

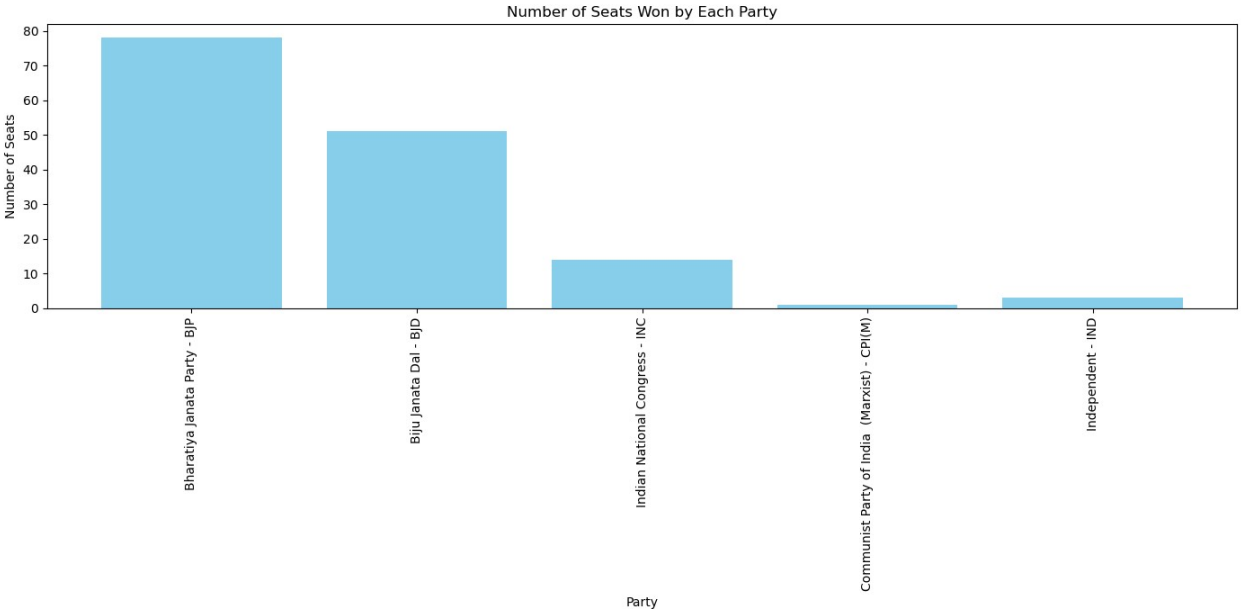
```
    total_seats_won = df['Seats Won'].sum()
    print(f"\nTotal seats won by all parties: {total_seats_won}")

    # Party with the most seats won
    max_seats = df['Seats Won'].max()
    party_max_seats = df[df['Seats Won'] == max_seats]
['Party'].values[0]
    print(f"The party with the most seats won is {party_max_seats}
with {max_seats} seats.")

    # Party with the least seats won
    min_seats = df['Seats Won'].min()
    party_min_seats = df[df['Seats Won'] == min_seats]
['Party'].values[0]
    print(f"The party with the least seats won is {party_min_seats}
with {min_seats} seats.")

    # Add percentage of total seats
    df['Percentage of Total Seats'] = (df['Seats Won'] /
total_seats_won) * 100

    print("\nPercentage of total seats won by each party:")
    print(df[['Party', 'Percentage of Total Seats']])

    # Highest percentage of seats won
    max_percentage = df['Percentage of Total Seats'].max()
    party_max_percentage = df[df['Percentage of Total Seats'] ==
max_percentage]['Party'].values[0]
    print(f"\nThe party with the highest percentage of total seats won
is {party_max_percentage} with {max_percentage:.2f}% of the seats.")

    # Analysis of seats leading
    total_leading_seats = df['Seats Leading'].sum()
    print(f"\nTotal seats leading by all parties:
{total_leading_seats}")

    # Party with the maximum seats leading
    max_leading_seats = df['Seats Leading'].max()
    party_max_leading = df[df['Seats Leading'] == max_leading_seats]
['Party'].values[0]
    print(f"The party with the most seats leading is
{party_max_leading} with {max_leading_seats} seats.")

    # Party with the least seats leading
    min_leading_seats = df['Seats Leading'].min()
    party_min_leading = df[df['Seats Leading'] == min_leading_seats]
['Party'].values[0]
    print(f"The party with the least seats leading is
{party_min_leading} with {min_leading_seats} seats.")
```

```
    # Pie Chart for Seats Won
    plt.figure(figsize=(10, 8))
    plt.pie(df['Seats Won'], labels=df['Party'], autopct='%1.1f%%',
startangle=140)
    plt.title('Election Results Distribution')
    plt.savefig('party_wise_pie_chart_S18.png')
    plt.show()

    # Bar Chart for Number of Seats Won
    plt.figure(figsize=(14, 7))
    plt.bar(df['Party'], df['Seats Won'], color='skyblue')
    plt.title('Number of Seats Won by Each Party')
    plt.xlabel('Party')
    plt.ylabel('Number of Seats')
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.savefig('party_wise_bar_chart_S18.png')
    plt.show()

# Define the URL
url = 'https://results.eci.gov.in/AcResultGenJune2024/partywiseresult-
S18.htm'

# Fetch data from the party-wise results page
fetch_party_wise_results(url)
```

```
Fetching data from
https://results.eci.gov.in/AcResultGenJune2024/partywiseresult-
S18.htm...
Successfully accessed the page:
https://results.eci.gov.in/AcResultGenJune2024/partywiseresult-S18.htm
HTML content fetched and parsed.
Results table found.
Extracted data for 5 parties.
Data saved to party_wise_election_results_S18.csv.
                                        Party  Seats Won  Seats
Leading  \
0                  Bharatiya Janata Party - BJP         78
0
1                        Biju Janata Dal - BJD         51
0
2                Indian National Congress - INC         14
0
3  Communist Party of India  (Marxist) - CPI(M)          1
0
4                            Independent - IND          3
0

   Total Seats
```

```
0           78
1           51
2           14
3            1
4            3

Basic Statistics
       Seats Won  Seats Leading  Total Seats
count   5.000000            5.0     5.000000
mean   29.400000            0.0    29.400000
std    33.797929            0.0    33.797929
min     1.000000            0.0     1.000000
25%     3.000000            0.0     3.000000
50%    14.000000            0.0    14.000000
75%    51.000000            0.0    51.000000
max    78.000000            0.0    78.000000

Total seats won by all parties: 147
The party with the most seats won is Bharatiya Janata Party - BJP with
78 seats.
The party with the least seats won is Communist Party of India
(Marxist) - CPI(M) with 1 seats.

Percentage of total seats won by each party:
                                          Party  Percentage of Total
Seats
0                  Bharatiya Janata Party - BJP
53.061224
1                         Biju Janata Dal - BJD
34.693878
2                Indian National Congress - INC
9.523810
3  Communist Party of India  (Marxist) - CPI(M)
0.680272
4                             Independent - IND
2.040816

The party with the highest percentage of total seats won is Bharatiya
Janata Party - BJP with 53.06% of the seats.

Total seats leading by all parties: 0
The party with the most seats leading is Bharatiya Janata Party - BJP
with 0 seats.
The party with the least seats leading is Bharatiya Janata Party - BJP
with 0 seats.
```






```
                                                Party  Seats Won  Seats
Leading  \
```

```
0                   Bharatiya Janata Party - BJP          78
0
1                         Biju Janata Dal - BJD          51
0
2                 Indian National Congress - INC          14
0
3  Communist Party of India  (Marxist) - CPI(M)           1
0
4                              Independent - IND           3
0

   Total Seats  Percentage of Total Seats
0           78                  53.061224
1           51                  34.693878
2           14                   9.523810
3            1                   0.680272
4            3                   2.040816
```

```python
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt

def fetch_by_election_results(url):
    print(f"Fetching data from {url}...")

    # Fetch the HTML content of the page
    response = requests.get(url)
    if response.status_code == 200:
        print(f"Successfully accessed the page: {url}")
    else:
        print(f"Failed to access the page: {url}. Status code:
{response.status_code}")
        return None  # Return None if page access fails

    # Parse HTML using BeautifulSoup
    soup = BeautifulSoup(response.content, 'html.parser')
    print("HTML content fetched and parsed.")

    # Extract data from the appropriate elements
    const_boxes = soup.find_all('div', class_='const-box')

    if not const_boxes:
        print("No constituency boxes found.")
        return None

    print(f"Found {len(const_boxes)} constituency boxes.")

    # Initialize lists to store extracted data
    constituencies = []
```

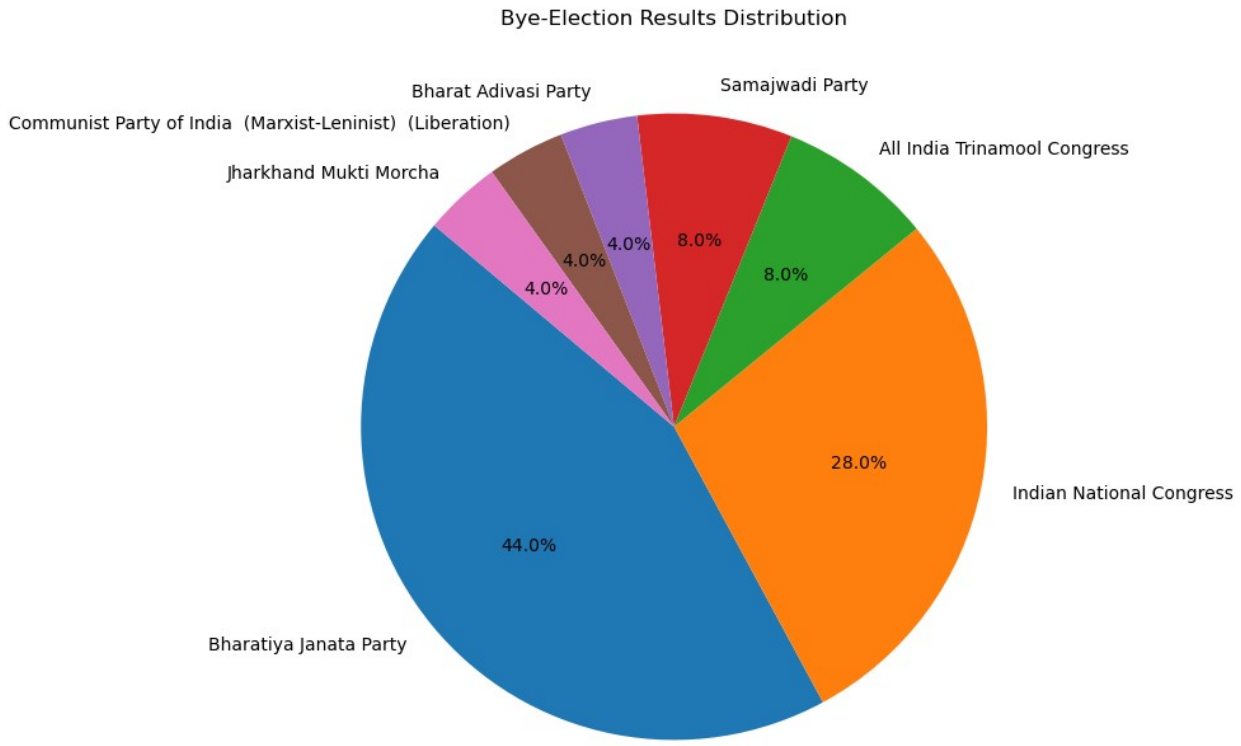
```
    states = []
    results = []
    candidates = []
    parties = []

    for box in const_boxes:
        try:
            # Extracting data based on the HTML structure
            constituency = box.find('div', class_='box-
content').find('h3').text.strip()
            state = box.find('div', class_='box-
content').find('h4').text.strip()
            result = box.find('div', class_='box-
content').find('h2').text.strip()
            candidate = box.find('div', class_='box-
content').find('h5').text.strip()
            party = box.find('div', class_='box-
content').find('h6').text.strip()

            constituencies.append(constituency)
            states.append(state)
            results.append(result)
            candidates.append(candidate)
            parties.append(party)
        except AttributeError:
            print("Some elements are missing in the div box.")
            continue

    # Check if data was extracted
    if not constituencies:
        print("No data extracted from the page.")
        return None

    print(f"Extracted data for {len(constituencies)} constituencies.")

    # Create a DataFrame to store the extracted data
    df = pd.DataFrame({
        'Constituency': constituencies,
        'State': states,
        'Result': results,
        'Candidate': candidates,
        'Party': parties
    })

    # Check if the DataFrame is populated correctly
    print("Data sample:")
    print(df.head())

    # Save the data to a CSV file
    csv_filename = 'bye_election_results_June2024.csv'
```

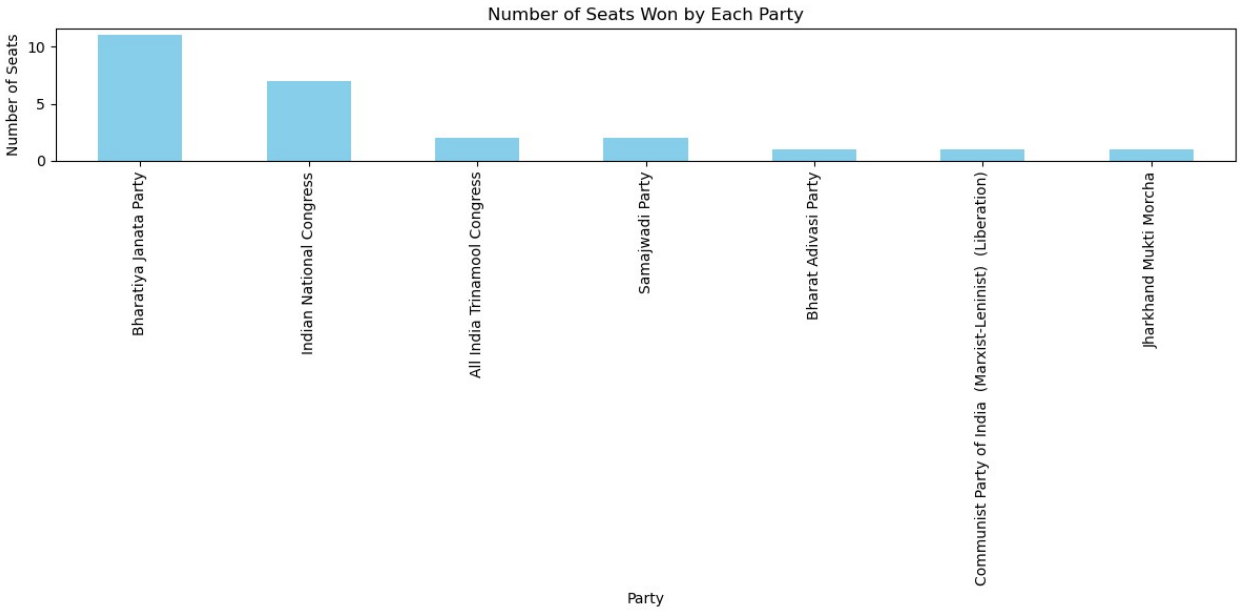
```
    df.to_csv(csv_filename, index=False)
    print(f"Data saved to {csv_filename}.")

    # Perform analysis
    perform_by_election_analysis(df)

    return df

def perform_by_election_analysis(df):
    print("\nBasic Statistics")
    print(df.describe(include='object'))

    # Analyze the distribution of results
    result_counts = df['Result'].value_counts()
    print("\nDistribution of Election Results:")
    print(result_counts)

    # Analyze the number of seats won by each party
    party_counts = df['Party'].value_counts()
    print("\nNumber of Seats Won by Each Party:")
    print(party_counts)

    # Add a column for counting the number of seats won by each party
    df['Seats Won'] = df['Result'].apply(lambda x: 1 if x == 'WON'
else 0)
    seats_by_party = df.groupby('Party')['Seats
Won'].sum().sort_values(ascending=False)

    # Total seats won by all parties
    total_seats_won = seats_by_party.sum()
    print(f"\nTotal seats won by all parties: {total_seats_won}")

    if seats_by_party.empty or total_seats_won == 0:
        print("Not enough data for plotting.")
        return  # Exit the function if there is no data to plot

    # Party with the most seats won
    max_seats = seats_by_party.max()
    party_max_seats = seats_by_party[seats_by_party ==
max_seats].index[0]
    print(f"\nThe party with the most seats won is {party_max_seats}
with {max_seats} seats.")

    # Party with the least seats won
    min_seats = seats_by_party.min()
    party_min_seats = seats_by_party[seats_by_party ==
min_seats].index[0]
    print(f"\nThe party with the least seats won is {party_min_seats}
with {min_seats} seats.")
```

```
    # Pie Chart for Seats Won
    plt.figure(figsize=(8, 8))
    plt.pie(seats_by_party, labels=seats_by_party.index,
autopct='%1.1f%%', startangle=140)
    plt.title('Bye-Election Results Distribution')
    plt.savefig('bye_election_pie_chart_June2024.png')
    plt.show()

    # Bar Chart for Number of Seats Won by Each Party
    plt.figure(figsize=(12, 6))
    seats_by_party.plot(kind='bar', color='skyblue')
    plt.title('Number of Seats Won by Each Party')
    plt.xlabel('Party')
    plt.ylabel('Number of Seats')
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.savefig('bye_election_bar_chart_June2024.png')
    plt.show()

def fetch_data_from_multiple_urls(urls, fetch_function):
    for url in urls:
        df = fetch_function(url)
        if df is not None:  # Ensure we only save if df is valid
            # Save data to CSV
            csv_filename = url.split('/')[-2] + '_results.csv'
            df.to_csv(csv_filename, index=False)
            print(f"Data saved to {csv_filename}.")

# Define the URLs for bye-elections
bye_election_urls = [
    'https://results.eci.gov.in/AcResultByeJune2024/'
    # Add other valid URLs as needed
]

# Fetch data from the bye-election results pages
fetch_data_from_multiple_urls(bye_election_urls,
fetch_by_election_results)
```

```
Fetching data from https://results.eci.gov.in/AcResultByeJune2024/...
Successfully accessed the page:
https://results.eci.gov.in/AcResultByeJune2024/
HTML content fetched and parsed.
Found 25 constituency boxes.
Extracted data for 25 constituencies.
Data sample:
     Constituency    State Result                    Candidate  \
0    Agiaon (195)    Bihar    WON          SHIV PRAKASH RANJAN
1    Vijapur (26)  Gujarat    WON             DR. C. J. CHAVDA
2  Porbandar (83)  Gujarat    WON     ARJUN DEVABHAI MODHWADIA
3  Manavadar (85)  Gujarat    WON   ARVINDBHAI JINABHAI LADANI
```

```
4  Khambhat (108)  Gujarat    WON  CHIRAGKUMAR ARVINDBHAI PATEL

                                                  Party
0  Communist Party of India  (Marxist-Leninist)  ...
1                              Bharatiya Janata Party
2                              Bharatiya Janata Party
3                              Bharatiya Janata Party
4                              Bharatiya Janata Party
Data saved to bye_election_results_June2024.csv.

Basic Statistics
        Constituency             State Result            Candidate  \
count             25                   25     25                   25
unique            25                   12      1                   25
top     Agiaon (195)  Himachal Pradesh    WON  SHIV PRAKASH RANJAN
freq               1                    6     25                    1

                           Party
count                         25
unique                         7
top     Bharatiya Janata Party
freq                          11

Distribution of Election Results:
WON    25
Name: Result, dtype: int64

Number of Seats Won by Each Party:
Bharatiya Janata Party                                    11
Indian National Congress                                   7
Samajwadi Party                                            2
All India Trinamool Congress                               2
Communist Party of India  (Marxist-Leninist)  (Liberation)     1
Jharkhand Mukti Morcha                                     1
Bharat Adivasi Party                                       1
Name: Party, dtype: int64

Total seats won by all parties: 25

The party with the most seats won is Bharatiya Janata Party with 11
seats.

The party with the least seats won is Bharat Adivasi Party with 1
seats.
```






```
Data saved to AcResultByeJune2024_results.csv.
```

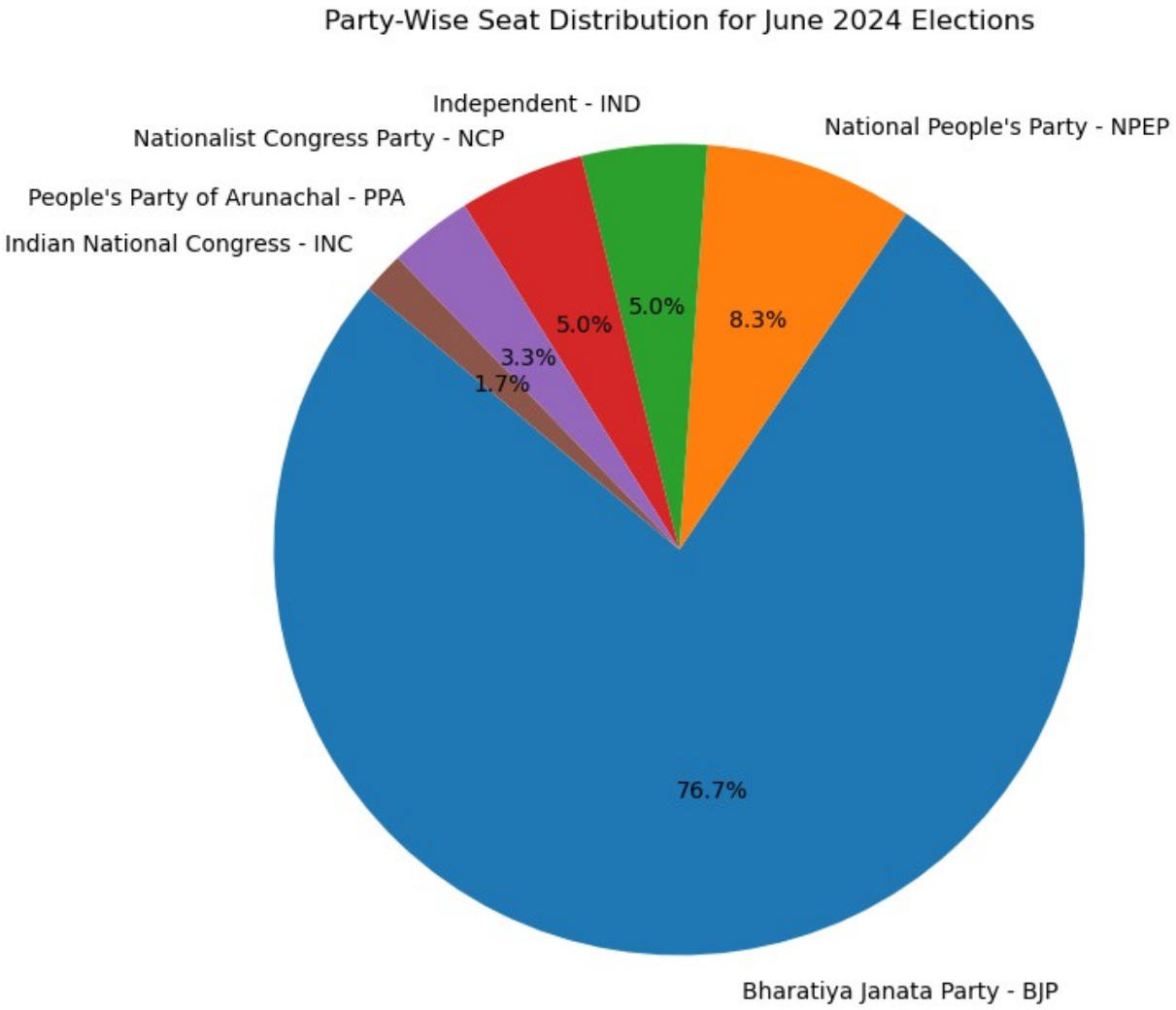
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt
```

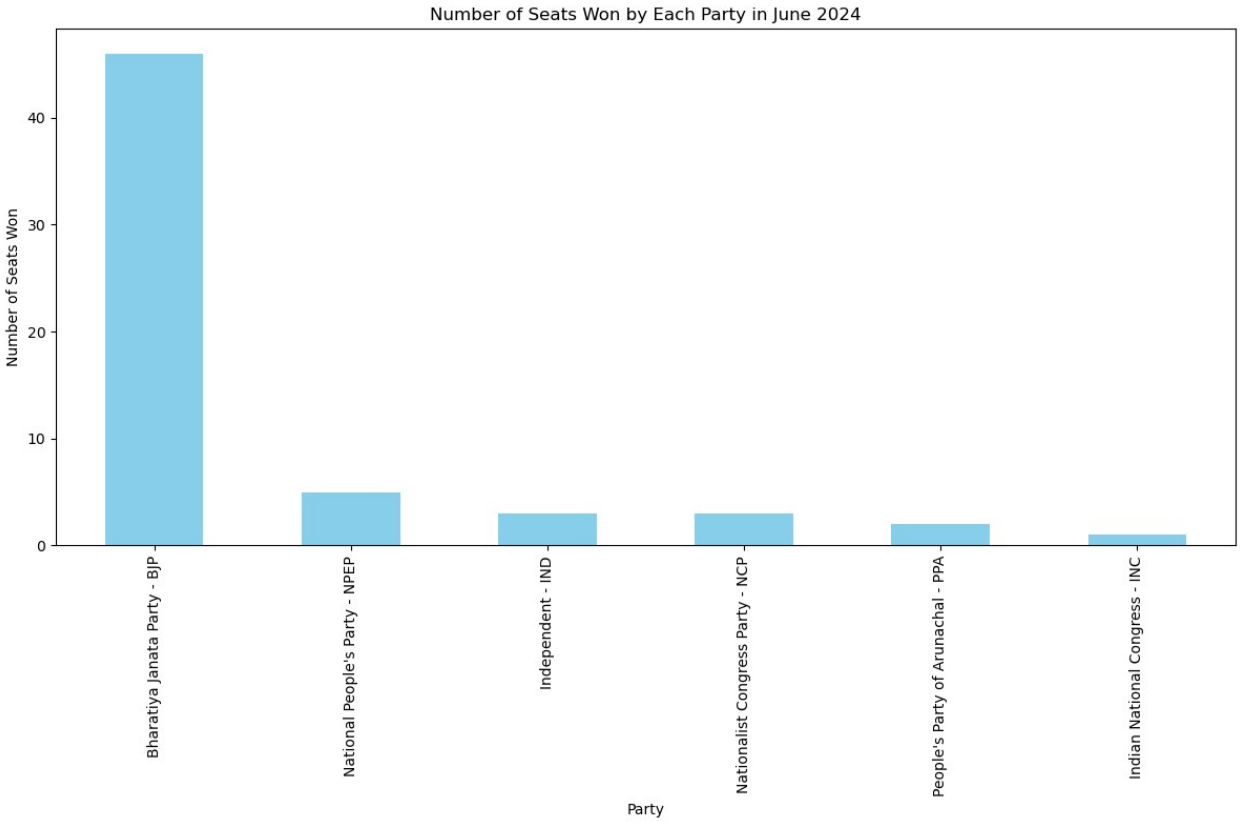
```
def fetch_party_wise_results(url):
    print(f"Fetching data from {url}...")

    # Fetch the HTML content of the page
    response = requests.get(url)
    if response.status_code == 200:
        print(f"Successfully accessed the page: {url}")
    else:
        print(f"Failed to access the page: {url}. Status code:
{response.status_code}")
        return None  # Return None if page access fails

    # Parse HTML using BeautifulSoup
    soup = BeautifulSoup(response.content, 'html.parser')
    print("HTML content fetched and parsed.")

    # Extract data from the table
    table = soup.find('table', class_='table')

    if not table:
        print("No table found on the page.")
        return None

    print("Table found.")

    # Extract table rows
    rows = table.find('tbody').find_all('tr')

    # Initialize lists to store extracted data
    parties = []
    seats_won = []
    leading_counts = []
    total_counts = []

    for row in rows:
        cols = row.find_all('td')
        if len(cols) >= 4:
            party = cols[0].text.strip()
            won = int(cols[1].text.strip())
            leading = int(cols[2].text.strip())
            total = int(cols[3].text.strip())

            parties.append(party)
            seats_won.append(won)
            leading_counts.append(leading)
            total_counts.append(total)

    # Check if data was extracted
```

```
    if not parties:
        print("No data extracted from the table.")
        return None

    print(f"Extracted data for {len(parties)} parties.")

    # Create a DataFrame to store the extracted data
    df = pd.DataFrame({
        'Party': parties,
        'Seats Won': seats_won,
        'Leading': leading_counts,
        'Total Seats': total_counts
    })

    # Check if the DataFrame is populated correctly
    print("Data sample:")
    print(df.head())

    # Save the data to a CSV file
    csv_filename = 'party_wise_results_June2024.csv'
    df.to_csv(csv_filename, index=False)
    print(f"Data saved to {csv_filename}.")

    # Perform analysis
    perform_party_wise_analysis(df)

    return df

def perform_party_wise_analysis(df):
    print("\nBasic Statistics")
    print(df.describe(include='object'))

    # Analyze the distribution of seats won
    seats_distribution = df['Seats Won'].value_counts()
    print("\nDistribution of Seats Won:")
    print(seats_distribution)

    # Analyze the total number of seats won by each party
    party_seat_counts = df.groupby('Party')['Seats
Won'].sum().sort_values(ascending=False)

    # Total seats won by all parties
    total_seats_won = party_seat_counts.sum()
    print(f"\nTotal seats won by all parties: {total_seats_won}")

    if party_seat_counts.empty or total_seats_won == 0:
        print("Not enough data for plotting.")
        return  # Exit the function if there is no data to plot

    # Party with the most seats won
```

```
    max_seats = party_seat_counts.max()
    party_max_seats = party_seat_counts[party_seat_counts ==
max_seats].index[0]
    print(f"\nThe party with the most seats won is {party_max_seats}
with {max_seats} seats.")

    # Party with the least seats won
    min_seats = party_seat_counts.min()
    party_min_seats = party_seat_counts[party_seat_counts ==
min_seats].index[0]
    print(f"\nThe party with the least seats won is {party_min_seats}
with {min_seats} seats.")

    # Pie Chart for Seats Won by Parties
    plt.figure(figsize=(10, 8))
    plt.pie(party_seat_counts, labels=party_seat_counts.index,
autopct='%1.1f%%', startangle=140)
    plt.title('Party-Wise Seat Distribution for June 2024 Elections')
    plt.savefig('party_wise_seat_distribution_June2024.png')
    plt.show()

    # Bar Chart for Number of Seats Won by Each Party
    plt.figure(figsize=(12, 8))
    party_seat_counts.plot(kind='bar', color='skyblue')
    plt.title('Number of Seats Won by Each Party in June 2024')
    plt.xlabel('Party')
    plt.ylabel('Number of Seats Won')
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.savefig('party_wise_seats_bar_chart_June2024.png')
    plt.show()

def fetch_data_from_multiple_urls(urls, fetch_function):
    for url in urls:
        df = fetch_function(url)
        if df is not None:  # Ensure we only save if df is valid
            # Save data to CSV
            csv_filename = url.split('/')[-2] + '_results.csv'
            df.to_csv(csv_filename, index=False)
            print(f"Data saved to {csv_filename}.")

# Define the URL for the party-wise results
party_wise_results_url = [

'https://results.eci.gov.in/AcResultGen2ndJune2024/partywiseresult-
S02.htm'
]

# Fetch data from the party-wise results page
```

```
fetch_data_from_multiple_urls(party_wise_results_url,
fetch_party_wise_results)

Fetching data from
https://results.eci.gov.in/AcResultGen2ndJune2024/partywiseresult-
S02.htm...
Successfully accessed the page:
https://results.eci.gov.in/AcResultGen2ndJune2024/partywiseresult-
S02.htm
HTML content fetched and parsed.
Table found.
Extracted data for 6 parties.
Data sample:
                               Party  Seats Won  Leading  Total Seats
0       Bharatiya Janata Party - BJP         46        0           46
1     National People's Party - NPEP          5        0            5
2   Nationalist Congress Party - NCP          3        0            3
3  People's Party of Arunachal - PPA          2        0            2
4     Indian National Congress - INC          1        0            1
Data saved to party_wise_results_June2024.csv.

Basic Statistics
                               Party
count                              6
unique                             6
top     Bharatiya Janata Party - BJP
freq                               1

Distribution of Seats Won:
3     2
46    1
5     1
2     1
1     1
Name: Seats Won, dtype: int64

Total seats won by all parties: 60

The party with the most seats won is Bharatiya Janata Party - BJP with
46 seats.

The party with the least seats won is Indian National Congress - INC
with 1 seats.
```


Party-Wise Seat Distribution for June 2024 Elections
Independent - IND
National People's Party - NPEP
Nationalist Congress Party - NCP
People's Party of Arunachal - PPA
Indian National Congress - INC
5.0%
5.0%
8.3%
3.3%
1.7%
76.7%
Bharatiya Janata Party - BJP




```
Data saved to AcResultGen2ndJune2024_results.csv.
```

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt

def fetch_party_wise_results(url):
    print(f"Fetching data from {url}...")

    # Fetch the HTML content of the page
    response = requests.get(url)
    if response.status_code == 200:
        print(f"Successfully accessed the page: {url}")
    else:
        print(f"Failed to access the page: {url}. Status code:
{response.status_code}")
        return None  # Return None if page access fails

    # Parse HTML using BeautifulSoup
    soup = BeautifulSoup(response.content, 'html.parser')
    print("HTML content fetched and parsed.")

    # Extract data from the table
    table = soup.find('table', class_='table')
```

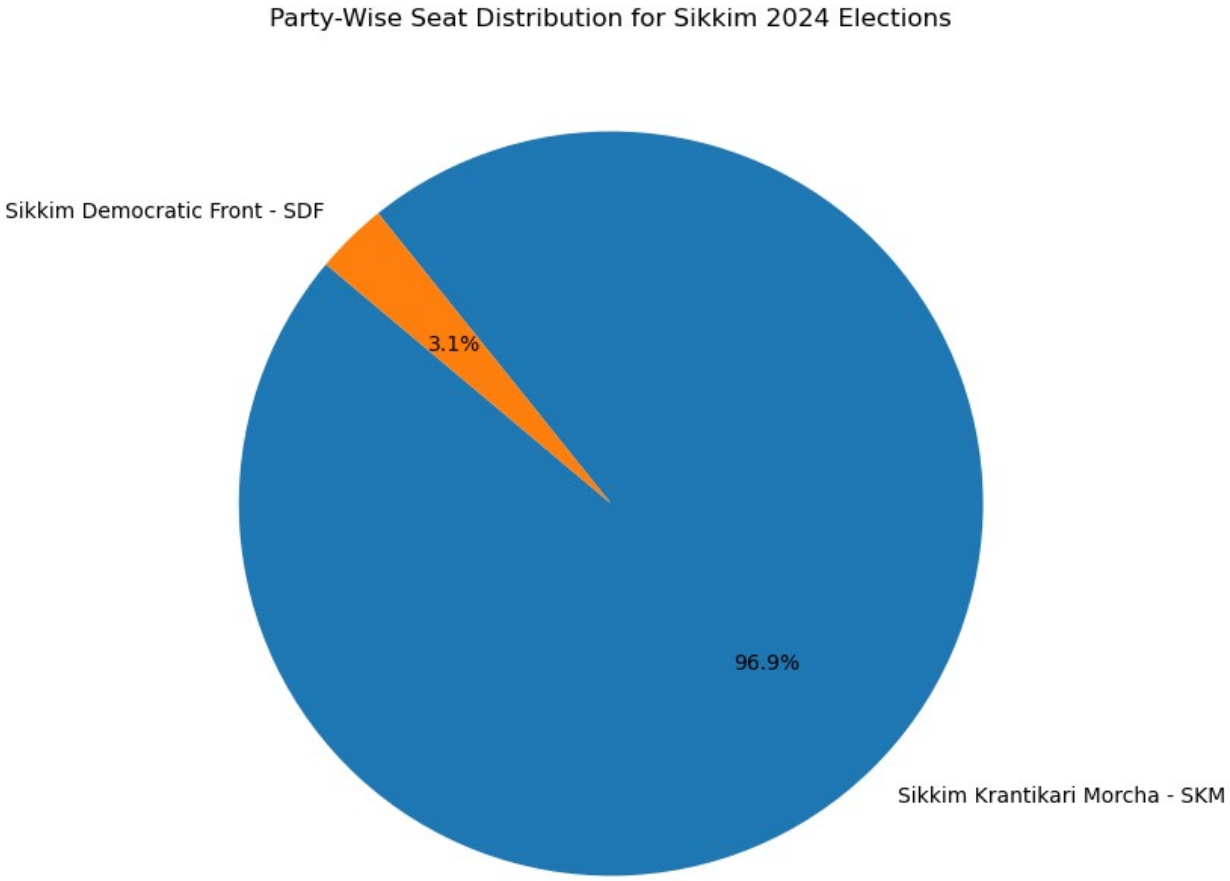
```
    if not table:
        print("No table found on the page.")
        return None

    print("Table found.")

    # Extract table rows
    rows = table.find('tbody').find_all('tr')

    # Initialize lists to store extracted data
    parties = []
    seats_won = []
    leading_counts = []
    total_counts = []

    for row in rows:
        cols = row.find_all('td')
        if len(cols) >= 4:
            party = cols[0].text.strip()
            won = int(cols[1].text.strip())
            leading = int(cols[2].text.strip())
            total = int(cols[3].text.strip())

            parties.append(party)
            seats_won.append(won)
            leading_counts.append(leading)
            total_counts.append(total)

    # Check if data was extracted
    if not parties:
        print("No data extracted from the table.")
        return None

    print(f"Extracted data for {len(parties)} parties.")

    # Create a DataFrame to store the extracted data
    df = pd.DataFrame({
        'Party': parties,
        'Seats Won': seats_won,
        'Leading': leading_counts,
        'Total Seats': total_counts
    })

    # Check if the DataFrame is populated correctly
    print("Data sample:")
    print(df.head())

    # Save the data to a CSV file
```

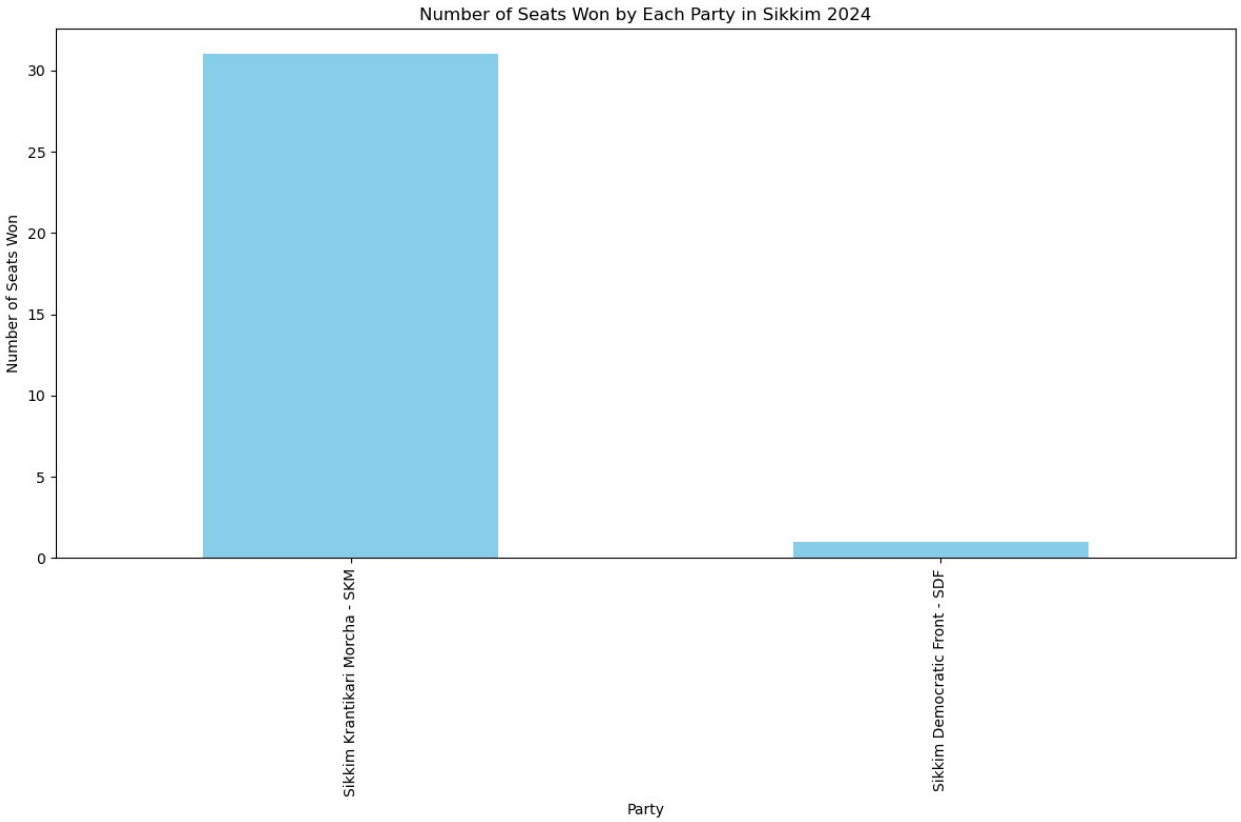
```
    csv_filename = 'party_wise_results_Sikkim_2024.csv'
    df.to_csv(csv_filename, index=False)
    print(f"Data saved to {csv_filename}.")

    # Perform analysis
    perform_party_wise_analysis(df)

    return df

def perform_party_wise_analysis(df):
    print("\nBasic Statistics")
    print(df.describe(include='object'))

    # Analyze the distribution of seats won
    seats_distribution = df['Seats Won'].value_counts()
    print("\nDistribution of Seats Won:")
    print(seats_distribution)

    # Analyze the total number of seats won by each party
    party_seat_counts = df.groupby('Party')['Seats
Won'].sum().sort_values(ascending=False)

    # Total seats won by all parties
    total_seats_won = party_seat_counts.sum()
    print(f"\nTotal seats won by all parties: {total_seats_won}")

    if party_seat_counts.empty or total_seats_won == 0:
        print("Not enough data for plotting.")
        return  # Exit the function if there is no data to plot

    # Party with the most seats won
    max_seats = party_seat_counts.max()
    party_max_seats = party_seat_counts[party_seat_counts ==
max_seats].index[0]
    print(f"\nThe party with the most seats won is {party_max_seats}
with {max_seats} seats.")

    # Party with the least seats won
    min_seats = party_seat_counts.min()
    party_min_seats = party_seat_counts[party_seat_counts ==
min_seats].index[0]
    print(f"\nThe party with the least seats won is {party_min_seats}
with {min_seats} seats.")

    # Pie Chart for Seats Won by Parties
    plt.figure(figsize=(10, 8))
    plt.pie(party_seat_counts, labels=party_seat_counts.index,
autopct='%1.1f%%', startangle=140)
    plt.title('Party-Wise Seat Distribution for Sikkim 2024
Elections')
```

```
    plt.savefig('party_wise_seat_distribution_Sikkim_2024.png')
    plt.show()

    # Bar Chart for Number of Seats Won by Each Party
    plt.figure(figsize=(12, 8))
    party_seat_counts.plot(kind='bar', color='skyblue')
    plt.title('Number of Seats Won by Each Party in Sikkim 2024')
    plt.xlabel('Party')
    plt.ylabel('Number of Seats Won')
    plt.xticks(rotation=90)
    plt.tight_layout()
    plt.savefig('party_wise_seats_bar_chart_Sikkim_2024.png')
    plt.show()

def fetch_data_from_multiple_urls(urls, fetch_function):
    for url in urls:
        df = fetch_function(url)
        if df is not None:  # Ensure we only save if df is valid
            # Save data to CSV
            csv_filename = url.split('/')[-2] + '_results.csv'
            df.to_csv(csv_filename, index=False)
            print(f"Data saved to {csv_filename}.")

# Define the URL for the party-wise results
party_wise_results_url = [

'https://results.eci.gov.in/AcResultGen2ndJune2024/partywiseresult-
S21.htm'
]

# Fetch data from the party-wise results page
fetch_data_from_multiple_urls(party_wise_results_url,
fetch_party_wise_results)
```

```
Fetching data from
https://results.eci.gov.in/AcResultGen2ndJune2024/partywiseresult-
S21.htm...
Successfully accessed the page:
https://results.eci.gov.in/AcResultGen2ndJune2024/partywiseresult-
S21.htm
HTML content fetched and parsed.
Table found.
Extracted data for 2 parties.
Data sample:
                          Party  Seats Won  Leading  Total Seats
0  Sikkim Krantikari Morcha - SKM         31        0           31
1   Sikkim Democratic Front - SDF          1        0            1
Data saved to party_wise_results_Sikkim_2024.csv.

Basic Statistics
```

```
                                    Party
count                                   2
unique                                  2
top     Sikkim Krantikari Morcha - SKM
freq                                    1

Distribution of Seats Won:
31    1
1     1
Name: Seats Won, dtype: int64

Total seats won by all parties: 32

The party with the most seats won is Sikkim Krantikari Morcha - SKM
with 31 seats.

The party with the least seats won is Sikkim Democratic Front - SDF
with 1 seats.
```






```
Data saved to AcResultGen2ndJune2024_results.csv.
```