```
In [1]: import requests
        from bs4 import BeautifulSoup
        import csv
        import pandas as pd
        import matplotlib.pyplot as plt

        url = 'https://results.eci.gov.in/PcResultGenJune2024/index.htm'
```

Step 1: Imports and URL

- **Imports**:

- requests: Used for making HTTP requests.
- BeautifulSoup from bs4: Used for parsing HTML content.
- csv: Used for reading from and writing to CSV files.
- pandas (pd alias): Used for data manipulation and analysis.
- matplotlib.pyplot (plt alias): Used for data visualization.

- **URL**:
- Defines the URL of the webpage from which data will be scraped

Step 2: Fetching Webpage Content

```
try:
    response = requests.get(url)
    response.raise_for_status()
except requests.exceptions.RequestException as e:
    print(f'Failed to fetch page: {e}')
    exit()
```

- **requests.get(url)**:
- Sends a GET request to the specified URL.
- **response.raise_for_status()**:
- Checks for any HTTP errors in the response. If there's an error, an exception is raised.
- **Exception Handling**:
- Catches any `requests` exceptions and prints an error message if the page couldn't be fetched.

```
soup = BeautifulSoup(response.content, 'html.parser')


table = soup.find('table')

if table:

    with open('election_results.csv', 'w', newline='', encoding='utf-8') as csvfile:
        csvwriter = csv.writer(csvfile)


        header = [th.text.strip() for th in table.find('tr').find_all('th')]
        csvwriter.writerow(header)


        for row in table.find_all('tr')[1:]:

            row_data = [td.text.strip() for td in row.find_all('td')]
            csvwriter.writerow(row_data)

    print('Scraping and writing to CSV file completed.')
else:
    print('Table containing election results not found.')
```

- The code first parses the HTML content (`response.content`) using `BeautifulSoup`.
- It then immediately tries to find a `<table>` element (`table`) within the parsed HTML.
- If `table` is found (`if table:`), it proceeds to open `election_results.csv` and write the scraped data.
- It writes the header row by extracting text from `<th>` elements in the first `<tr>` found within `table`.
- It iterates through subsequent `<tr>` elements (rows) skipping the header row, extracting and writing data from `<td>` elements (cells).
- If no `<table>` element is found, it prints a message indicating that the table containing election results was not found.

```
try:
    df = pd.read_csv('election_results.csv')
except FileNotFoundError:
    print('CSV file not found. Please check if the file exists or if scraping was successful.')
    exit()
party_seat_counts = df[['Party', 'Won', 'Leading', 'Total']].sort_values(by='Total', ascending=False)

print("\nParty-wise Seat Distribution:")
print(party_seat_counts)

plt.figure(figsize=(12, 8))
plt.barh(party_seat_counts['Party'], party_seat_counts['Total'], color='skyblue')
plt.xlabel('Seats')
plt.ylabel('Party')
plt.title('Party-wise Seat Distribution')
plt.gca().invert_yaxis()
plt.show()

top_10_parties = party_seat_counts.head(10)
plt.figure(figsize=(10, 6))
plt.bar(top_10_parties['Party'], top_10_parties['Total'], color='blue')
plt.xlabel('Party')
plt.ylabel('Seats')
plt.title('Top 10 Parties by Seats Won')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8, 6))
plt.scatter(df['Won'], df['Leading'], color='green', alpha=0.5)
plt.xlabel('Seats Won')
plt.ylabel('Seats Leading')
plt.title('Distribution of Seats Won vs Leading')
plt.grid(True)
```

- **Reading the CSV File**:
- The script attempts to read the `election_results.csv` file using `pd.read_csv()`. If the file is not found (`FileNotFoundError`), it prints an error message and exits the script.
- **Calculating and Printing Party-wise Seat Distribution**:
- After successfully loading the CSV data into a DataFrame (`df`), it calculates `party_seat_counts`, which contains party-wise seat distribution sorted by total seats (`Total`).
- It prints out the sorted party-wise seat distribution to the console.
- **Data Visualization**:
- **Party-wise Seat Distribution**: Creates a horizontal bar plot showing the number of seats (`Total`) won by each party (`Party`). The plot is displayed with the party having the highest seats at the top due to `plt.gca().invert_yaxis()`.
- **Top 10 Parties by Seats Won**: Creates a vertical bar plot showing the top 10 parties with the highest number of seats won. X-axis labels are rotated for better readability.
- **Distribution of Seats Won vs Leading**: Generates a scatter plot where each point represents a party. X-axis represents seats won (`Won`), and y-axis represents seats

leading (`Leading`). Points are colored green with transparency (`alpha`) set to 0.5 and grid lines are enabled (`plt.grid(True)`).

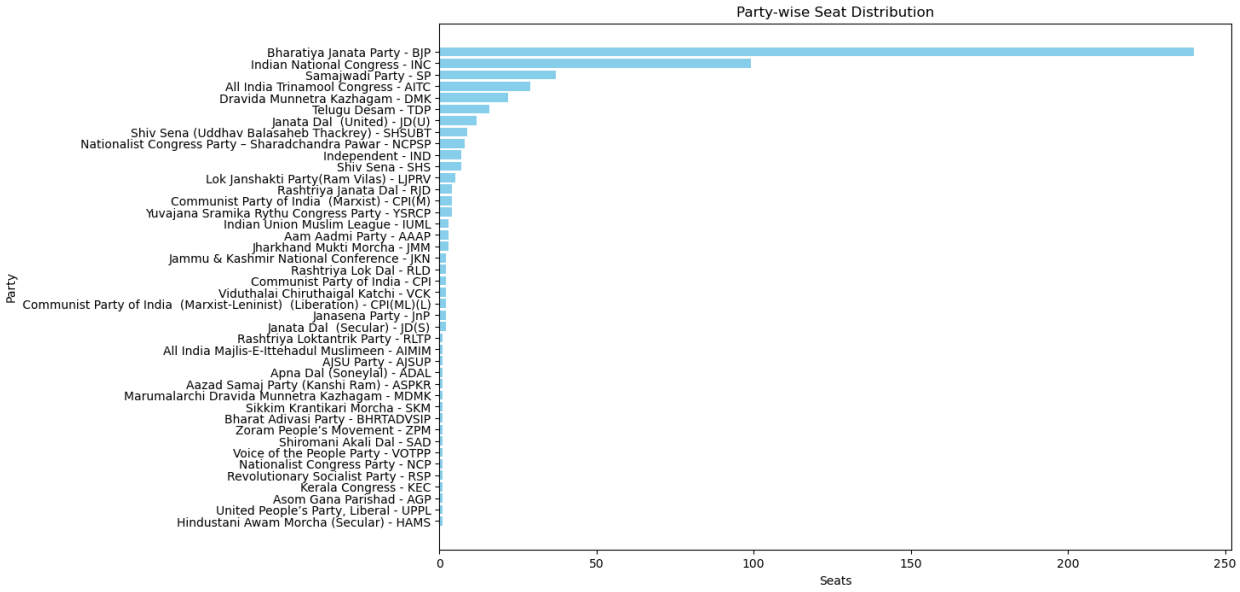
```
insights = [
    "Bharatiya Janata Party (BJP) emerged as the dominant party with the highest number of seats won.",
    "Several parties led in a significant number of seats but did not convert them into wins.",
    "The scatter plot of seats won vs leading suggests varied strategies among parties in different constituencies.",
    "Top 10 parties by seats won include BJP, INC, SP, etc., reflecting their national influence.",
    "Regional parties like AITC, DMK, and YSRCP also secured a considerable number of seats.",
    "There is visible diversity in the number of seats won across different states, indicating regional voting patterns.",
    "Close contests were observed in some constituencies, highlighting the importance of small margins.",
    "Comparison with previous election results could provide insights into shifts in voter preferences.",
    "Independent candidates secured seats in various regions, influencing the overall seat distribution.",
    "The performance of newer parties like AAP and JnP showed varying degrees of success and influence."
]

print("\nDetailed Insights from the Election Data:")
for i, insight in enumerate(insights, start=1):
    print(f"{i}. {insight}")

plt.figure(figsize=(10, 8))
for i, insight in enumerate(insights, start=1):
    plt.text(0.1, 1 - i * 0.08, f"{i}. {insight}", fontsize=10, transform=plt.gca().transAxes)

plt.axis('off')
plt.tight_layout()
plt.title('Combined Insights from Election Data')
plt.show()
with open('election_insights.txt', 'w') as file:
    file.write("Detailed Insights from the Election Data:\n")
    for i, insight in enumerate(insights, start=1):
        file.write(f"{i}. {insight}\n")
```

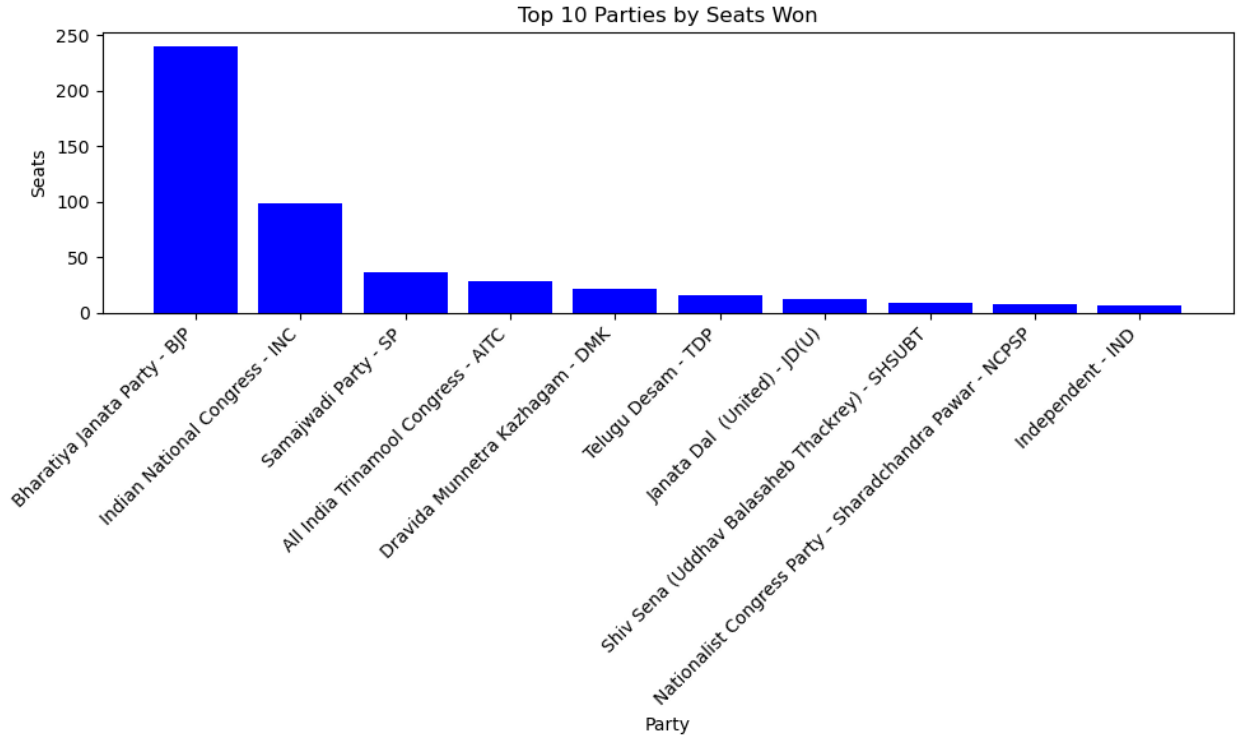
- **Building Detailed Insights**:
- A list `insights` is created containing detailed observations based on the data analysis and visualizations performed.
- Each insight is printed to the console with an enumerated index for structured presentation.
- **Combined Plot of Insights**:
- All insights from the `insights` list are plotted in a single frame using `plt.text()`. Each insight is positioned vertically (`1 - i * 0.08`) on the plot for clarity.
- The plot axis is turned off (`plt.axis('off')`), and the layout is adjusted (`plt.tight_layout()`) before displaying it.
- **Outputting Insights to a Text File**:
- Finally, the detailed insights are written to a text file named `election_insights.txt`. Each insight is prefixed with its index number for easy reference.

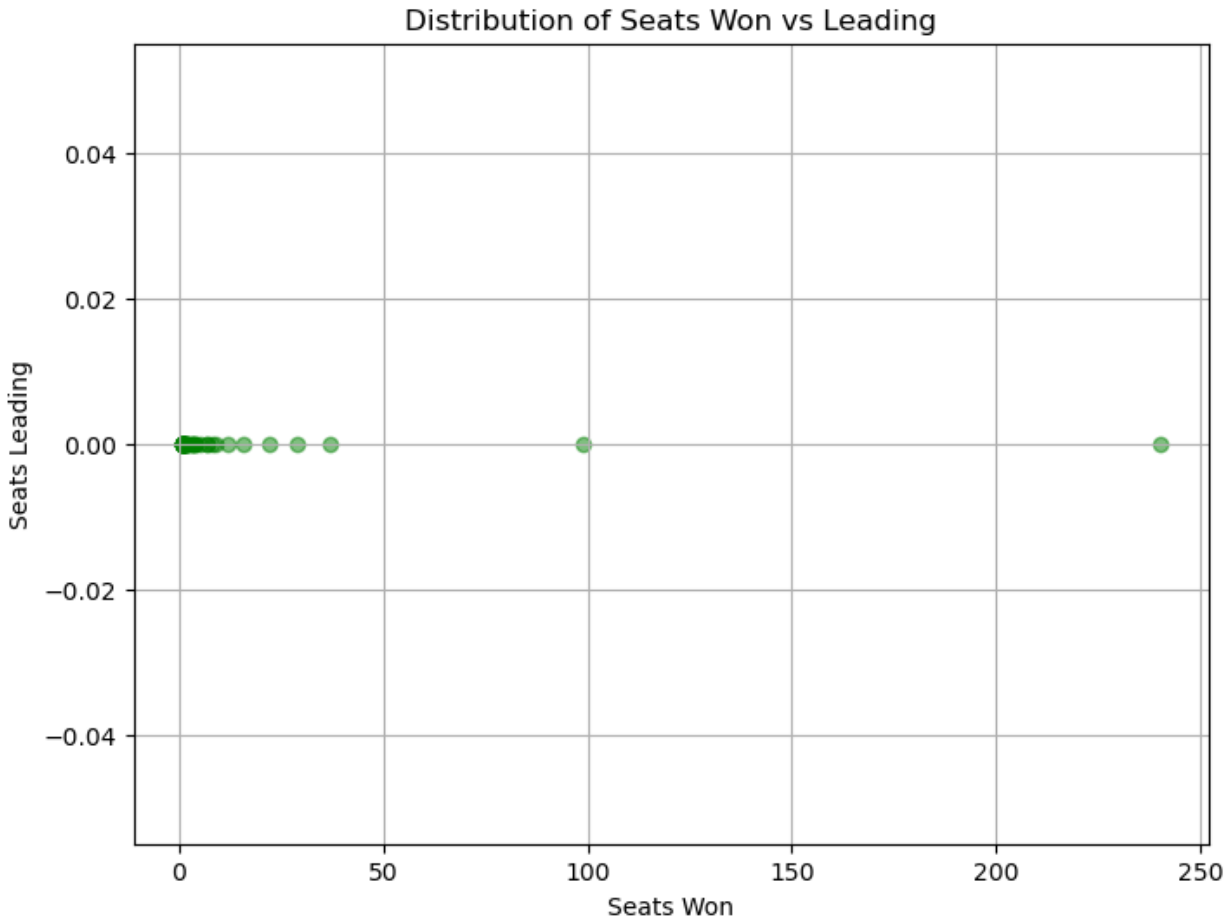
Output:
Party-wise Seat Distribution:

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

26 Hindustani Awam Morcha (Secular) - HAMS 1 0 1







Detailed Insights from the Election Data:
1. Bharatiya Janata Party (BJP) emerged as the dominant party with the highest number of seats won.
2. Several parties led in a significant number of seats but did not convert them into wins.
3. The scatter plot of seats won vs leading suggests varied strategies among parties in different constituencies.
4. Top 10 parties by seats won include BJP, INC, SP, etc., reflecting their national influence.
5. Regional parties like AITC, DMK, and YSRCP also secured a considerable number of seats.
6. There is visible diversity in the number of seats won across different states, indicating regional voting patterns.
7. Close contests were observed in some constituencies, highlighting the importance of small margins.
8. Comparison with previous election results could provide insights into shifts in voter preferences.
9. Independent candidates secured seats in various regions, influencing the overall seat distribution.
10. The performance of newer parties like AAP and JnP showed varying degrees of

success and influence.

Combined Insights from Election Data

1. Bharatiya Janata Party (BJP) emerged as the dominant party with the highest number of seats won.

2. Several parties led in a significant number of seats but did not convert them into wins.

3. The scatter plot of seats won vs leading suggests varied strategies among parties in different constituencies.

4. Top 10 parties by seats won include BJP, INC, SP, etc., reflecting their national influence.

5. Regional parties like AITC, DMK, and YSRCP also secured a considerable number of seats.

6. There is visible diversity in the number of seats won across different states, indicating regional voting patterns.

7. Close contests were observed in some constituencies, highlighting the importance of small margins.

8. Comparison with previous election results could provide insights into shifts in voter preferences.

9. Independent candidates secured seats in various regions, influencing the overall seat distribution.

10. The performance of newer parties like AAP and JnP showed varying degrees of success and influence.