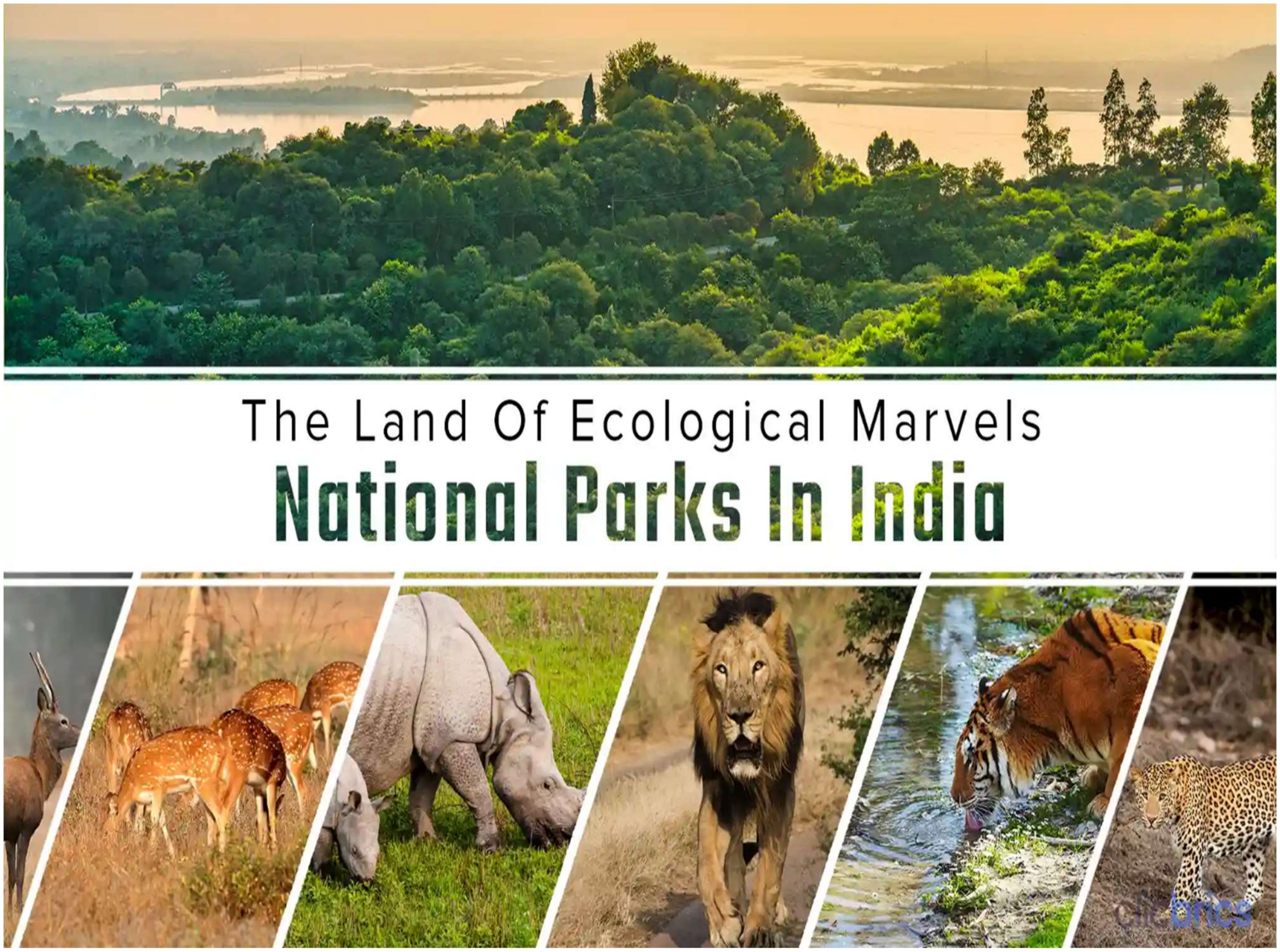
LIST OF NATIONAL PARKS OF INDIA



- Tejal Bharankar

Batch- T335 / DS

INTRODUCTION:

National Parks in India are protected areas that conserve the country's rich biodiversity and natural heritage. With over 100 national parks across various states, and union territories, this project collects, structures, and analyzes data about these parks for better understanding and future applications.

OBJECTIVES:

1. Identify and access a reliable website (Wikipedia) containing the list of National Parks in India.
2. Extract relevant information such as:
    - Name of the park
    - Location (State/UT)
    - Year of Establishment
    - Area (in km²)
3. Store the scraped data in a structured CSV file for further analysis or visualization.

TOOLS & LIBRARIES:

- Python
- BeautifulSoup (for web scraping)
- Requests (for fetching web pages)
- Pandas (for structuring and analyzing tabular data)

METHODOLOGY:

1. Identify The Source Website: Use the Wikipedia page: List of National Parks in India

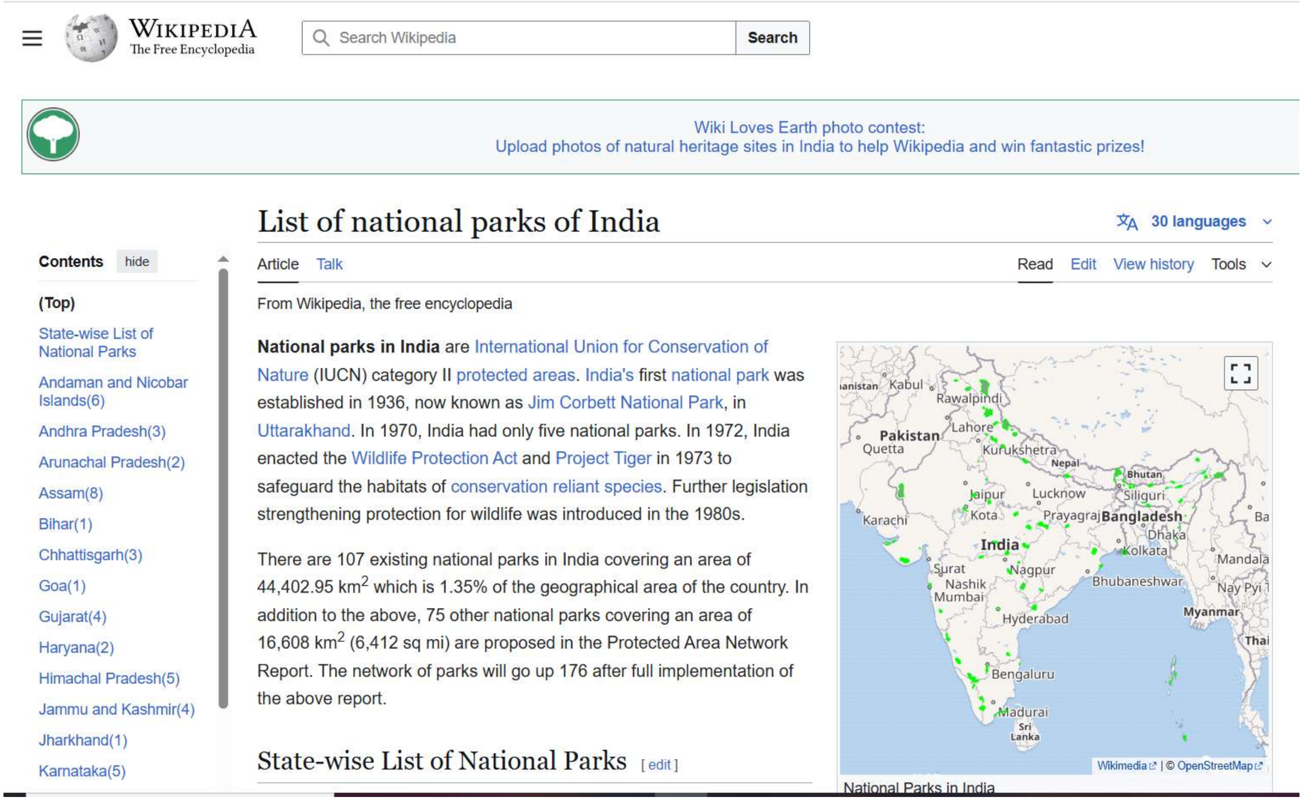
2. Fetch Web Page Data: Use the requests module to retrieve the Wikipedia page content.

3. Parse the HTML Content: Utilize BeautifulSoup to extract structured information from the HTML tables.

4. Data Cleaning & Storage: Clean the extracted data and store it in a CSV file for further analysis

SELECT THE TARGET WEBSITE(S):

Choose websites that provide the required data, such as Wikipedia's 'List of National Parks in India' page



Wiki Loves Earth photo contest:
Upload photos of natural heritage sites in India to help Wikipedia and win fantastic prizes!

List of national parks of India

From Wikipedia, the free encyclopedia

National parks in India are International Union for Conservation of Nature (IUCN) category II protected areas. India's first national park was established in 1936, now known as Jim Corbett National Park, in Uttarakhand. In 1970, India had only five national parks. In 1972, India enacted the Wildlife Protection Act and Project Tiger in 1973 to safeguard the habitats of conservation reliant species. Further legislation strengthening protection for wildlife was introduced in the 1980s.

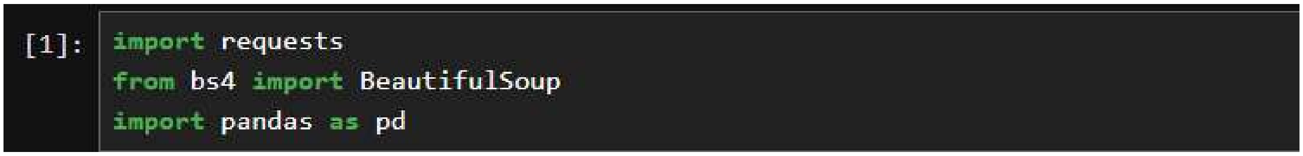
There are 107 existing national parks in India covering an area of 44,402.95 km^2 which is 1.35% of the geographical area of the country. In addition to the above, 75 other national parks covering an area of 16,608 km^2 (6,412 sq mi) are proposed in the Protected Area Network Report. The network of parks will go up 176 after full implementation of the above report.

National Parks in India

State-wise List of National Parks [edit]

IMPORT THE ESSENTIAL LIBRARIES :

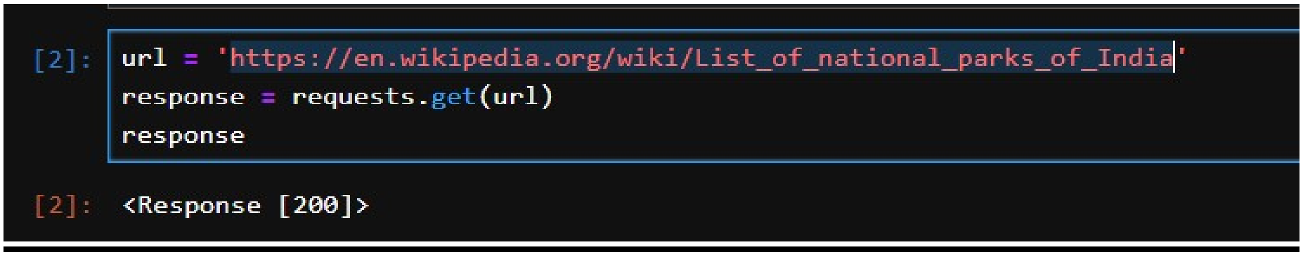
- Python for scripting and processing
- BeautifulSoup and Requests for web scraping
- Pandas for structuring and analyzing data



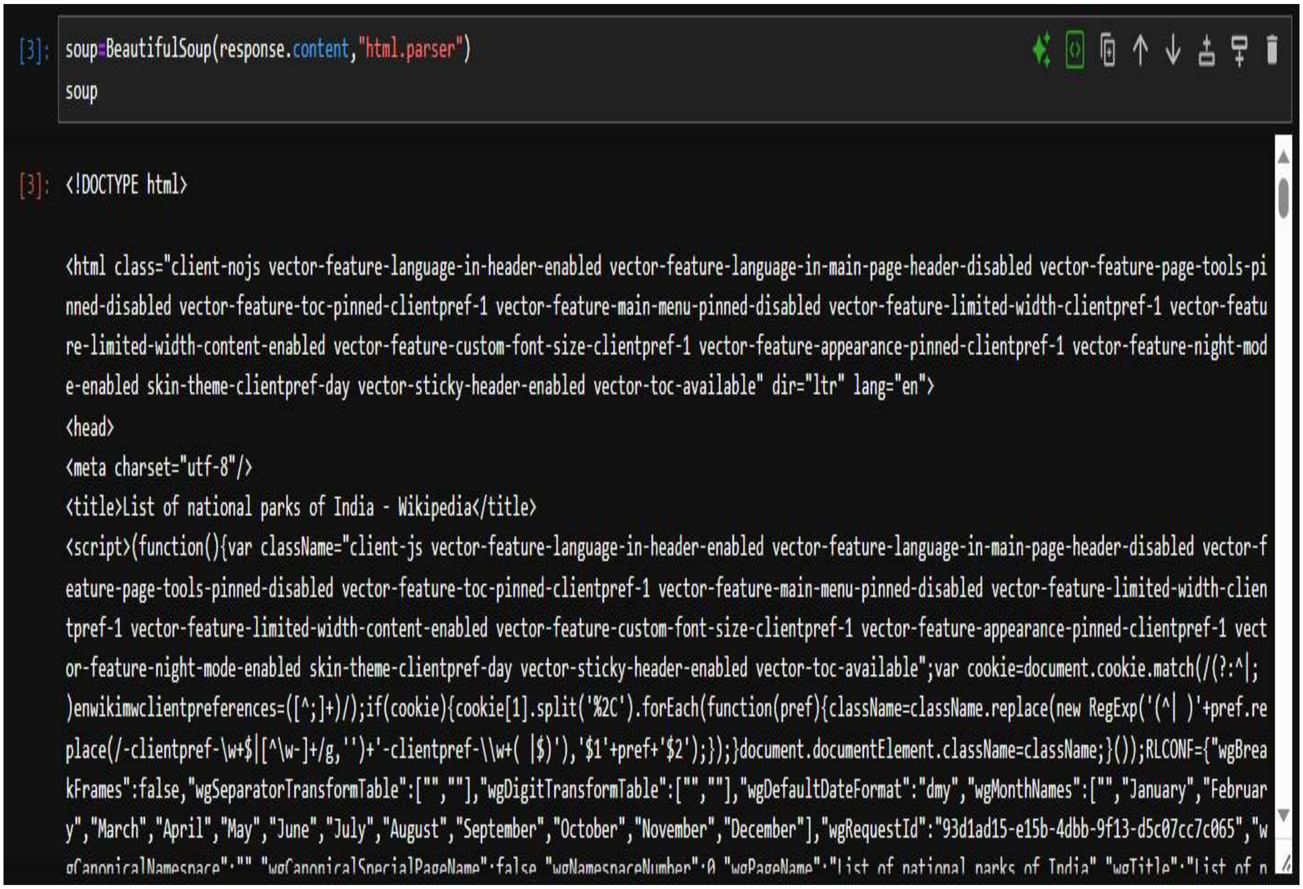
```
[1]: import requests
     from bs4 import BeautifulSoup
     import pandas as pd
```

FETCH AND PARSE WIKIPEDIA PAGE :

Use the Requests module to fetch the HTML content of the Wikipedia page and parse it using BeautifulSoup.



```
[2]: url = 'https://en.wikipedia.org/wiki/List_of_national_parks_of_India'
     response = requests.get(url)
     response

[2]: <Response [200]>
```

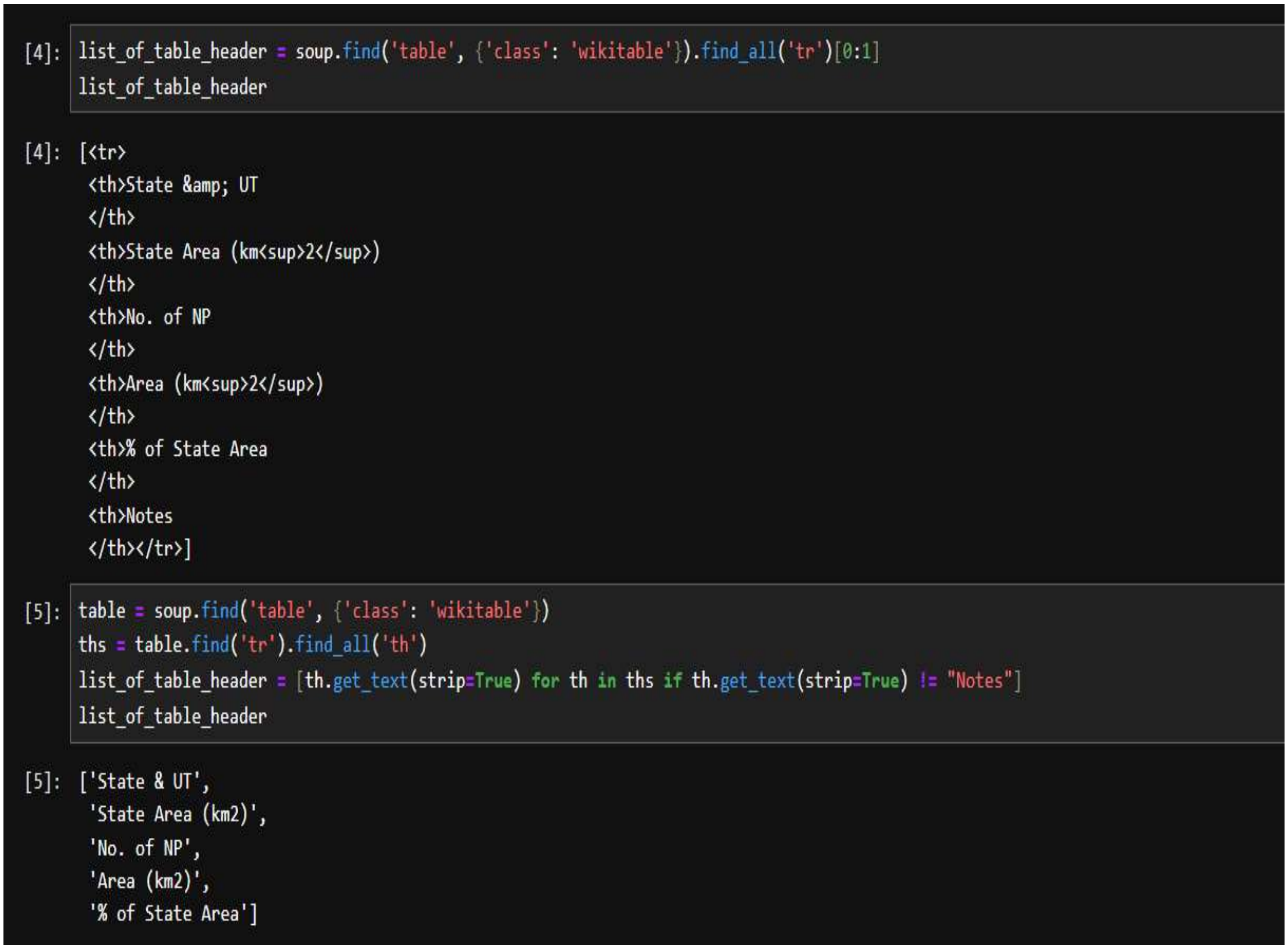


```
[3]: soup=BeautifulSoup(response.content,"html.parser")
     soup
```

```
[3]: <!DOCTYPE html>

<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vector-feature-page-tools-pi
nned-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-featu
re-limited-width-content-enabled vector-feature-custom-font-size-clientpref-1 vector-feature-appearance-pinned-clientpref-1 vector-feature-night-mod
e-enabled skin-theme-clientpref-day vector-sticky-header-enabled vector-toc-available" dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<title>List of national parks of India - Wikipedia</title>
<script>(function(){var className="client-js vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vector-f
eature-page-tools-pinned-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clien
tpref-1 vector-feature-limited-width-content-enabled vector-feature-custom-font-size-clientpref-1 vector-feature-appearance-pinned-clientpref-1 vect
or-feature-night-mode-enabled skin-theme-clientpref-day vector-sticky-header-enabled vector-toc-available";var cookie=document.cookie.match(/(?:^|;
)enwikimwclientpreferences=([^;]+)/);if(cookie){cookie[1].split('%2C').forEach(function(pref){className=className.replace(new RegExp('(^| )'+pref.re
place(/-clientpref-\w+$|[^\w-]+/g,'')+'-clientpref-\\w+( |$)'),'$1'+pref+'$2');});}document.documentElement.className=className;}());RLCONF={"wgBrea
kFrames":false,"wgSeparatorTransformTable":["",""],"wgDigitTransformTable":["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":["","January","Februar
y","March","April","May","June","July","August","September","October","November","December"],"wgRequestId":"93d1ad15-e15b-4dbb-9f13-d5c07cc7c065","w
gCanonicalNamespace":"","wgCanonicalSpecialPageName":false,"wgNamespaceNumber":0,"wgPageName":"List of national parks of India","wgTitle":"List of n
```

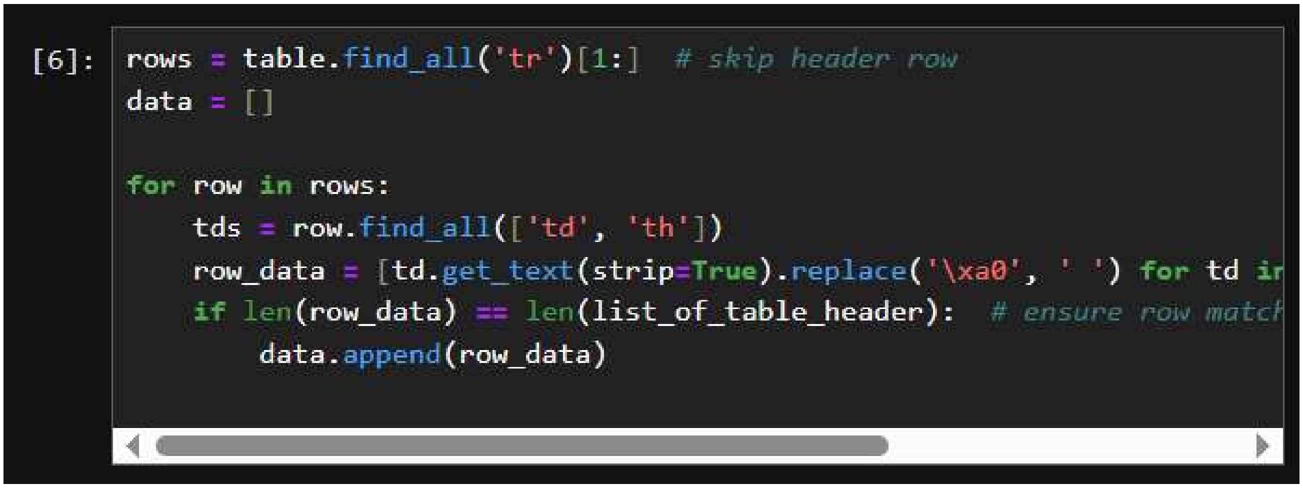
EXTRACT HEADERS FROM TABLE :

Step 1: Use BeautifulSoup to extract <th> tags from the table.
Step 2: Clean the headers using string methods (e.g., strip() or regex) to remove newline characters.
Step 3: Store the headers in a list for further reference



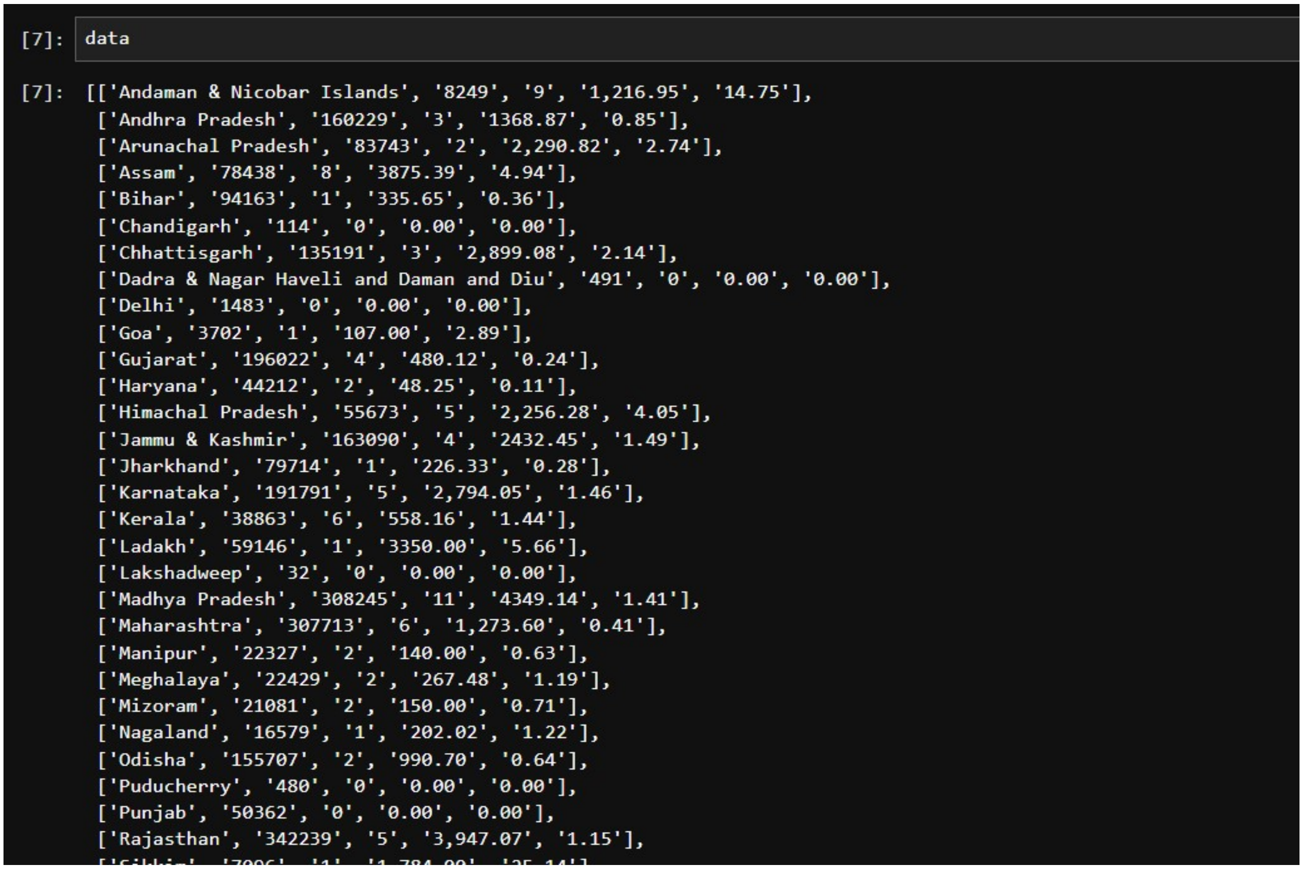
```
[4]: list_of_table_header = soup.find('table', {'class': 'wikitable'}).find_all('tr')[0:1]
     list_of_table_header

[4]: [<tr>
      <th>State &amp; UT
      </th>
      <th>State Area (km<sup>2</sup>)
      </th>
      <th>No. of NP
      </th>
      <th>Area (km<sup>2</sup>)
      </th>
      <th>% of State Area
      </th>
      <th>Notes
      </th></tr>]

[5]: table = soup.find('table', {'class': 'wikitable'})
     ths = table.find('tr').find_all('th')
     list_of_table_header = [th.get_text(strip=True) for th in ths if th.get_text(strip=True) != "Notes"]
     list_of_table_header

[5]: ['State & UT',
      'State Area (km2)',
      'No. of NP',
      'Area (km2)',
      '% of State Area']
```

Extract rows from the HTML table (like the one on Wikipedia), clean the text, and store each row as a list within a larger list.



```
[6]: rows = table.find_all('tr')[1:]  # skip header row
     data = []

     for row in rows:
         tds = row.find_all(['td', 'th'])
         row_data = [td.get_text(strip=True).replace('\xa0', ' ') for td in
         if len(row_data) == len(list_of_table_header):  # ensure row match
             data.append(row_data)
```

This results in a list of rows, with each row containing clean and structured data from the table



```
[7]: data

[7]: [['Andaman & Nicobar Islands', '8249', '9', '1,216.95', '14.75'],
      ['Andhra Pradesh', '160229', '3', '1368.87', '0.85'],
      ['Arunachal Pradesh', '83743', '2', '2,290.82', '2.74'],
      ['Assam', '78438', '8', '3875.39', '4.94'],
      ['Bihar', '94163', '1', '335.65', '0.36'],
      ['Chandigarh', '114', '0', '0.00', '0.00'],
      ['Chhattisgarh', '135191', '3', '2,899.08', '2.14'],
      ['Dadra & Nagar Haveli and Daman and Diu', '491', '0', '0.00', '0.00'],
      ['Delhi', '1483', '0', '0.00', '0.00'],
      ['Goa', '3702', '1', '107.00', '2.89'],
      ['Gujarat', '196022', '4', '480.12', '0.24'],
      ['Haryana', '44212', '2', '48.25', '0.11'],
      ['Himachal Pradesh', '55673', '5', '2,256.28', '4.05'],
      ['Jammu & Kashmir', '163090', '4', '2432.45', '1.49'],
      ['Jharkhand', '79714', '1', '226.33', '0.28'],
      ['Karnataka', '191791', '5', '2,794.05', '1.46'],
      ['Kerala', '38863', '6', '558.16', '1.44'],
      ['Ladakh', '59146', '1', '3350.00', '5.66'],
      ['Lakshadweep', '32', '0', '0.00', '0.00'],
      ['Madhya Pradesh', '308245', '11', '4349.14', '1.41'],
      ['Maharashtra', '307713', '6', '1,273.60', '0.41'],
      ['Manipur', '22327', '2', '140.00', '0.63'],
      ['Meghalaya', '22429', '2', '267.48', '1.19'],
      ['Mizoram', '21081', '2', '150.00', '0.71'],
      ['Nagaland', '16579', '1', '202.02', '1.22'],
      ['Odisha', '155707', '2', '990.70', '0.64'],
      ['Puducherry', '480', '0', '0.00', '0.00'],
      ['Punjab', '50362', '0', '0.00', '0.00'],
      ['Rajasthan', '342239', '5', '3,947.07', '1.15'],
      ['Sikkim', '7096', '1', '1,784.00', '25.14'],
```

EXTRACT DATA FROM TABLE:

This step involves extracting and cleaning table data (<td> elements) from the HTML page using BeautifulSoup.

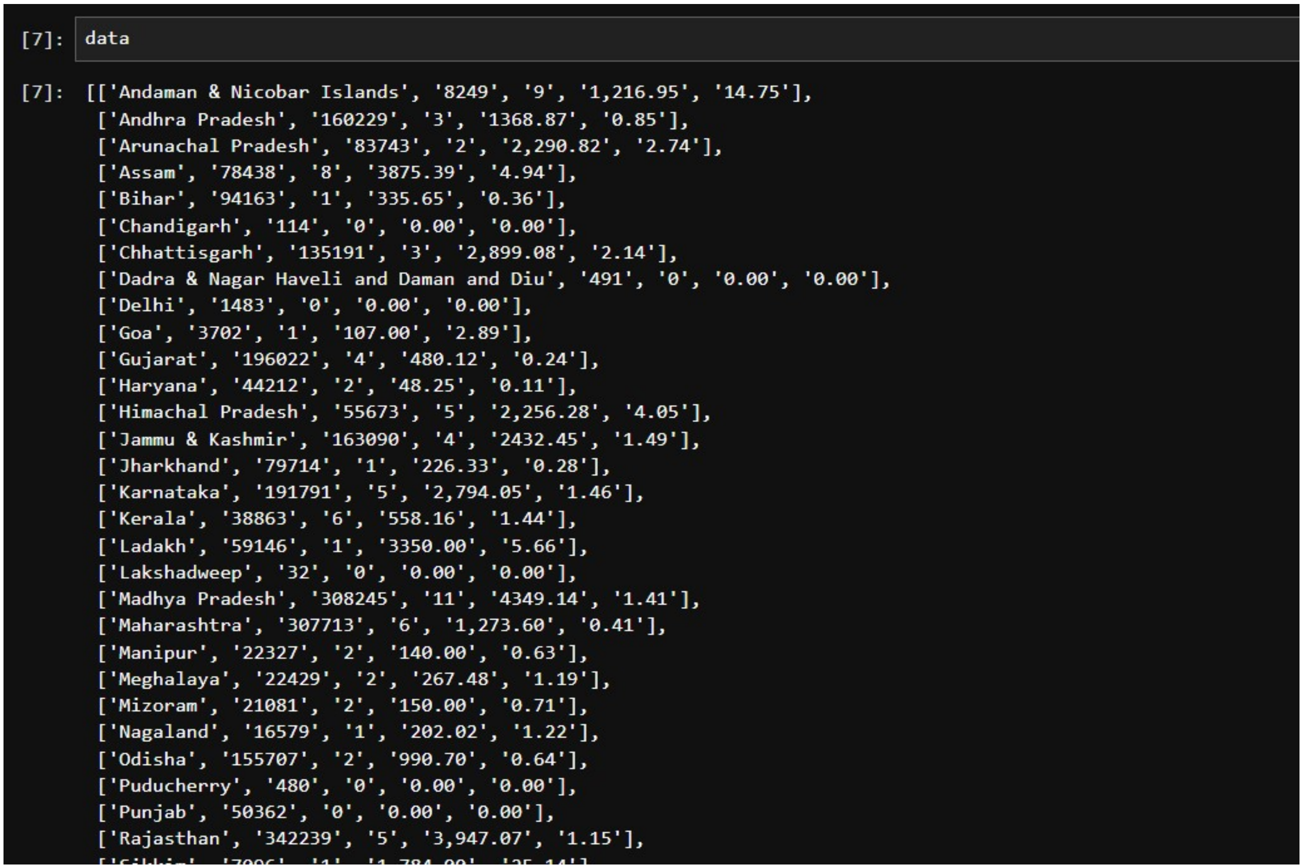
Step 1: Extract rows of the table using <tr> tags, and access each data cell using <td>.

Step 2: Clean the text using .get_text(strip=True) and replace any non-breaking spaces using .replace('\xa0', ' ').

Step 3: Separate the cleaned data into specific lists:

- State/UT
- State Area (in km²)
- Number of National Parks
- Total Area of Parks (in km²)
- Percentage of Area Covered

This structured data can be used to create a DataFrame.



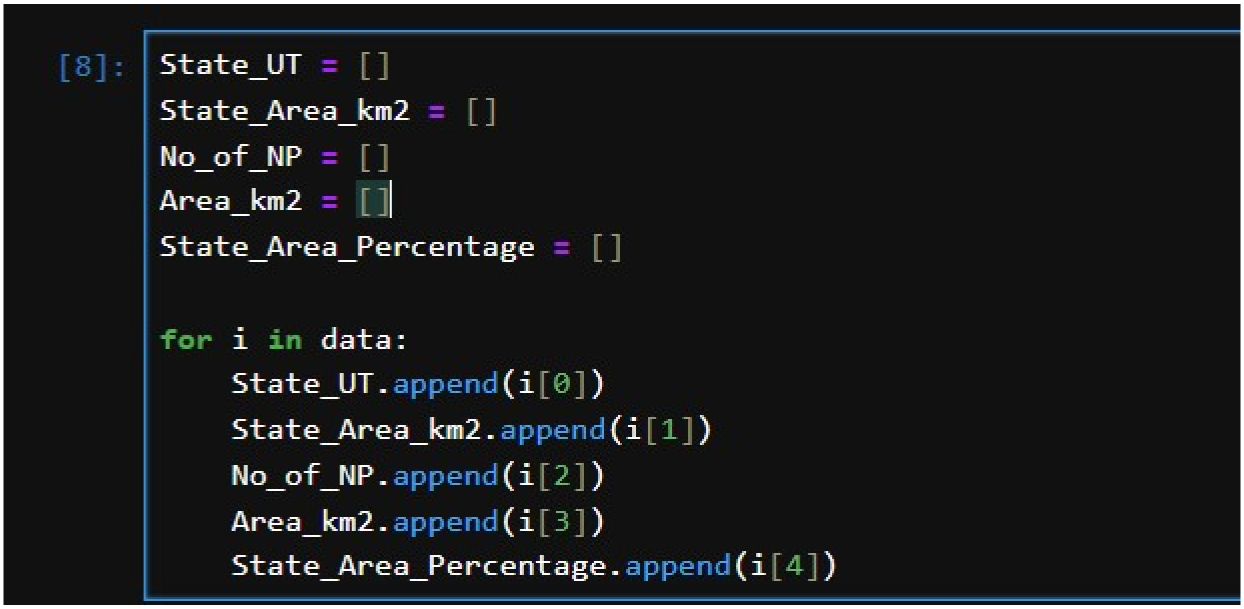
```
[7]: data

[7]: [['Andaman & Nicobar Islands', '8249', '9', '1,216.95', '14.75'],
      ['Andhra Pradesh', '160229', '3', '1368.87', '0.85'],
      ['Arunachal Pradesh', '83743', '2', '2,290.82', '2.74'],
      ['Assam', '78438', '8', '3875.39', '4.94'],
      ['Bihar', '94163', '1', '335.65', '0.36'],
      ['Chandigarh', '114', '0', '0.00', '0.00'],
      ['Chhattisgarh', '135191', '3', '2,899.08', '2.14'],
      ['Dadra & Nagar Haveli and Daman and Diu', '491', '0', '0.00', '0.00'],
      ['Delhi', '1483', '0', '0.00', '0.00'],
      ['Goa', '3702', '1', '107.00', '2.89'],
      ['Gujarat', '196022', '4', '480.12', '0.24'],
      ['Haryana', '44212', '2', '48.25', '0.11'],
      ['Himachal Pradesh', '55673', '5', '2,256.28', '4.05'],
      ['Jammu & Kashmir', '163090', '4', '2432.45', '1.49'],
      ['Jharkhand', '79714', '1', '226.33', '0.28'],
      ['Karnataka', '191791', '5', '2,794.05', '1.46'],
      ['Kerala', '38863', '6', '558.16', '1.44'],
      ['Ladakh', '59146', '1', '3350.00', '5.66'],
      ['Lakshadweep', '32', '0', '0.00', '0.00'],
      ['Madhya Pradesh', '308245', '11', '4349.14', '1.41'],
      ['Maharashtra', '307713', '6', '1,273.60', '0.41'],
      ['Manipur', '22327', '2', '140.00', '0.63'],
      ['Meghalaya', '22429', '2', '267.48', '1.19'],
      ['Mizoram', '21081', '2', '150.00', '0.71'],
      ['Nagaland', '16579', '1', '202.02', '1.22'],
      ['Odisha', '155707', '2', '990.70', '0.64'],
      ['Puducherry', '480', '0', '0.00', '0.00'],
      ['Punjab', '50362', '0', '0.00', '0.00'],
      ['Rajasthan', '342239', '5', '3,947.07', '1.15'],
      ['Sikkim', '7096', '1', '1,784.00', '25.14'],
```

This part of the process involves creating separate lists for each column of the extracted data.

First, empty lists are initialized for: State/UT, State Area, Number of National Parks, Total Area of Parks, and Area Percentage.

Then, for each row in the extracted data, individual values are appended to their respective lists using index-based access (e.g., State_UT.append(i[0])).

As a result, we obtain structured column-wise data, ready for DataFrame creation or visualization.

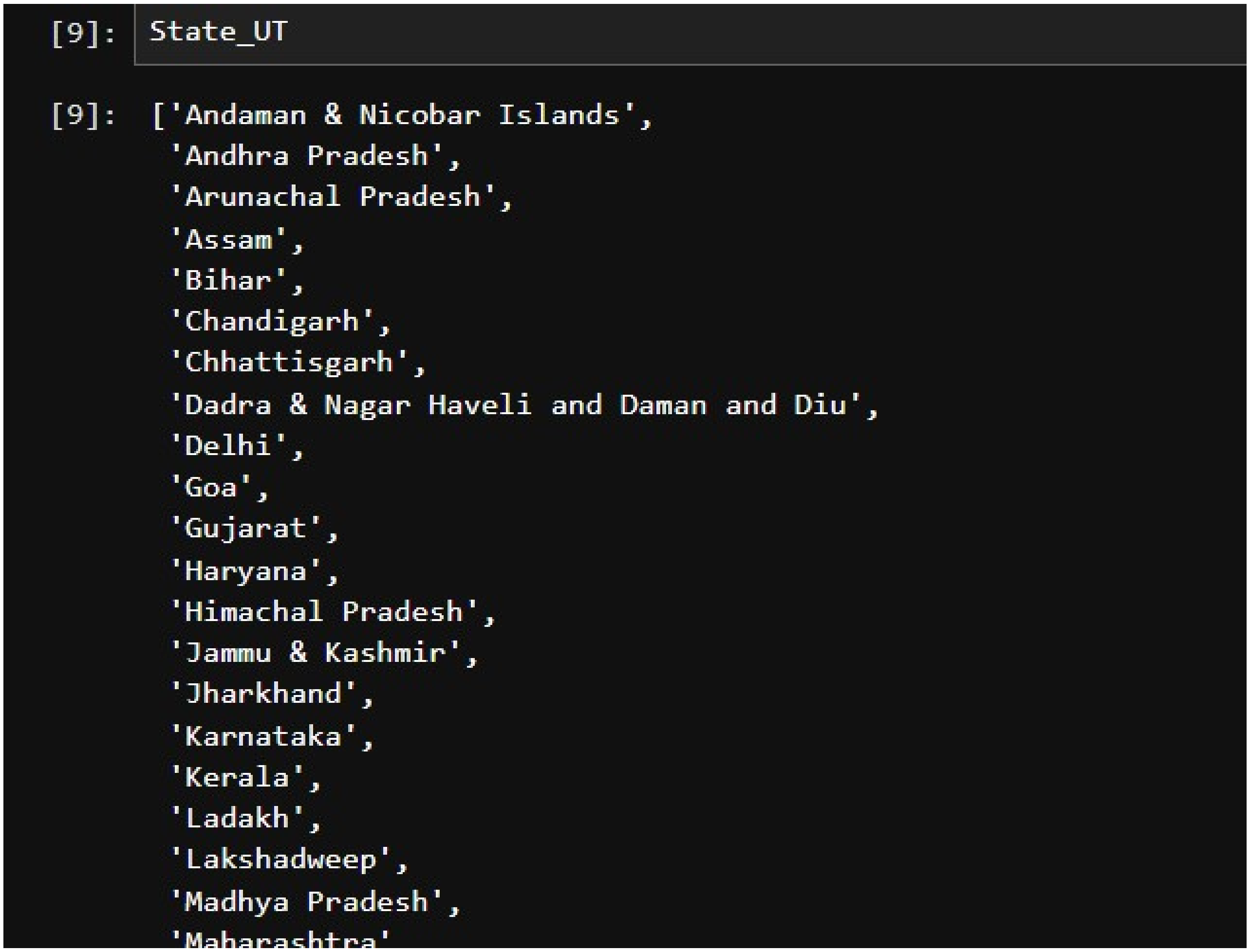


```
[8]: State_UT = []
     State_Area_km2 = []
     No_of_NP = []
     Area_km2 = []
     State_Area_Percentage = []

     for i in data:
         State_UT.append(i[0])
         State_Area_km2.append(i[1])
         No_of_NP.append(i[2])
         Area_km2.append(i[3])
         State_Area_Percentage.append(i[4])
```

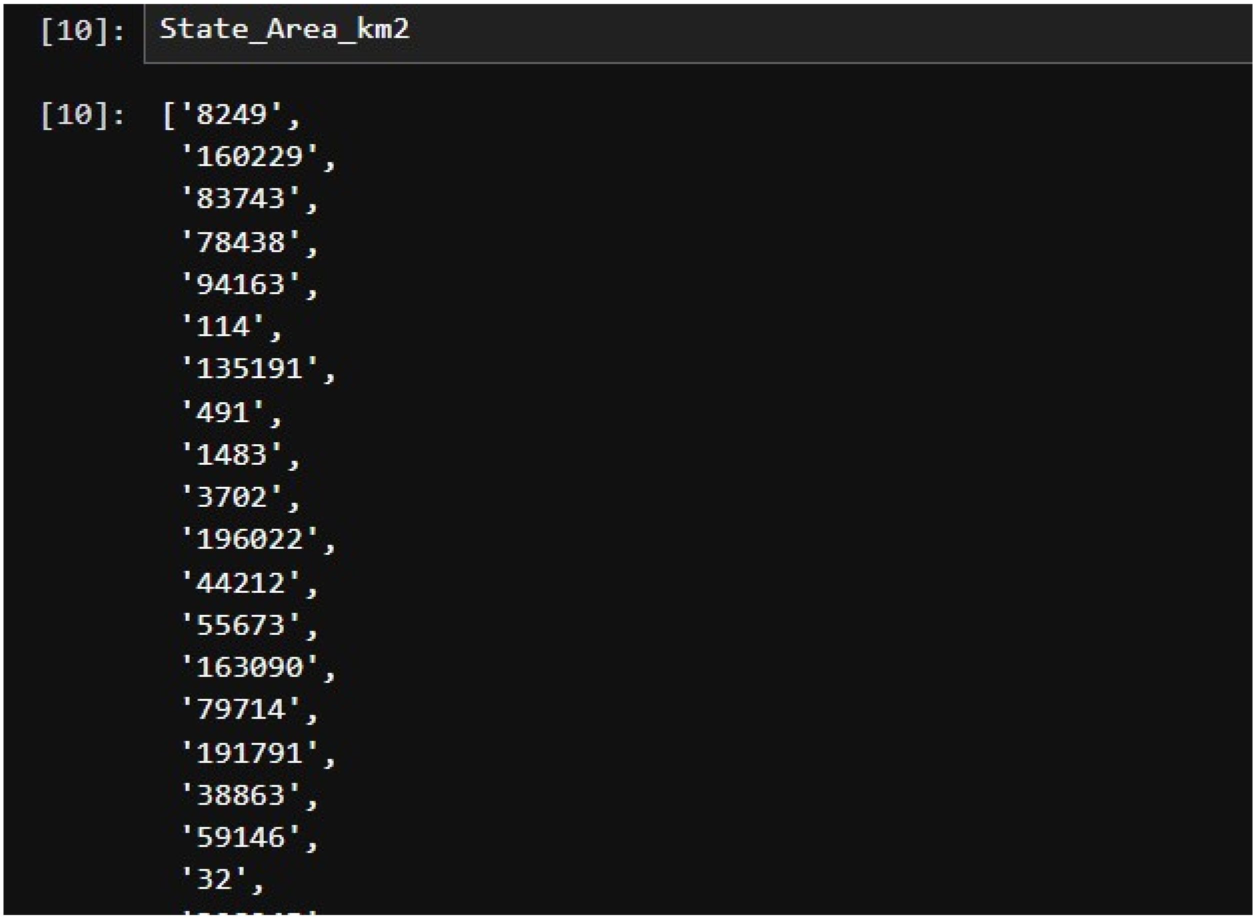
Output:

A structured CSV file containing the following columns:

- **State/UT**
- **State Area**
- **Number of National Parks**
- **Total Area of Parks**
- **Area %**

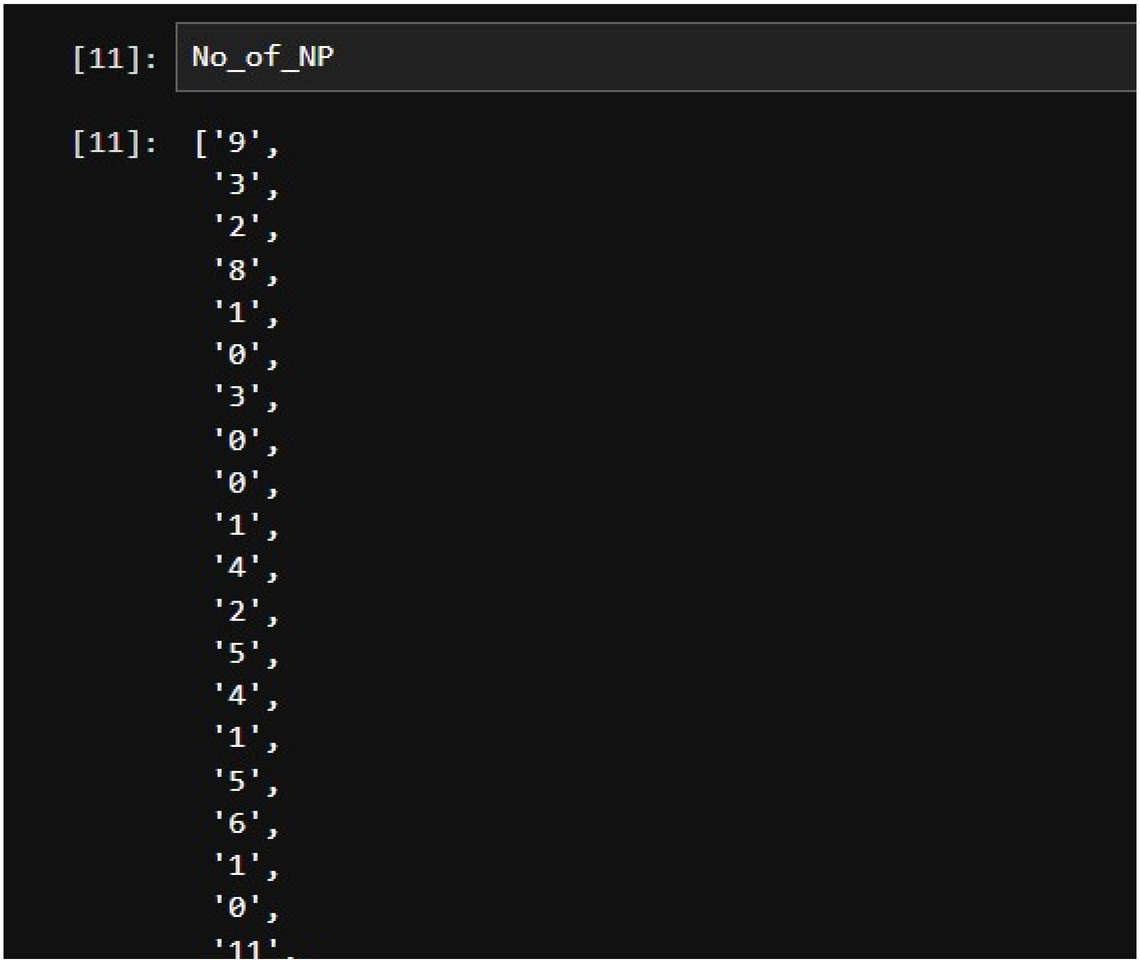


```
[9]:  State_UT

[9]:  ['Andaman & Nicobar Islands',
       'Andhra Pradesh',
       'Arunachal Pradesh',
       'Assam',
       'Bihar',
       'Chandigarh',
       'Chhattisgarh',
       'Dadra & Nagar Haveli and Daman and Diu',
       'Delhi',
       'Goa',
       'Gujarat',
       'Haryana',
       'Himachal Pradesh',
       'Jammu & Kashmir',
       'Jharkhand',
       'Karnataka',
       'Kerala',
       'Ladakh',
       'Lakshadweep',
       'Madhya Pradesh',
       'Maharashtra'
```

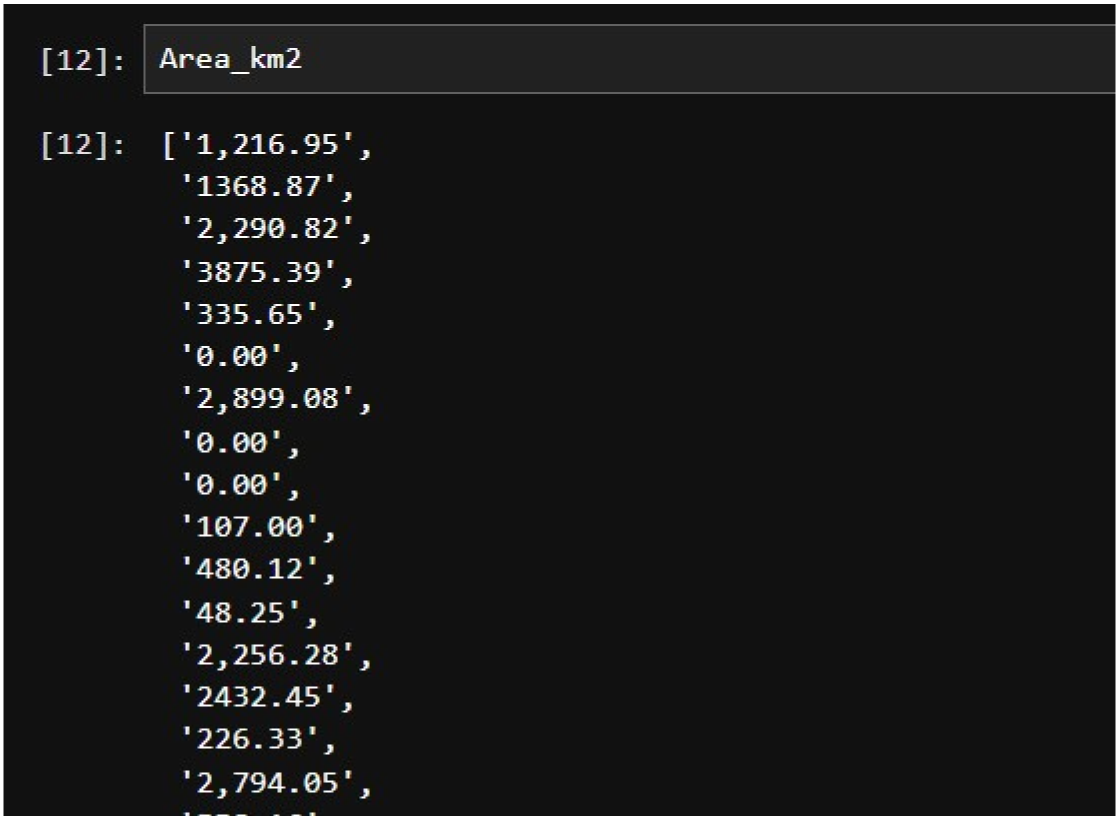


```
[10]: State_Area_km2
```

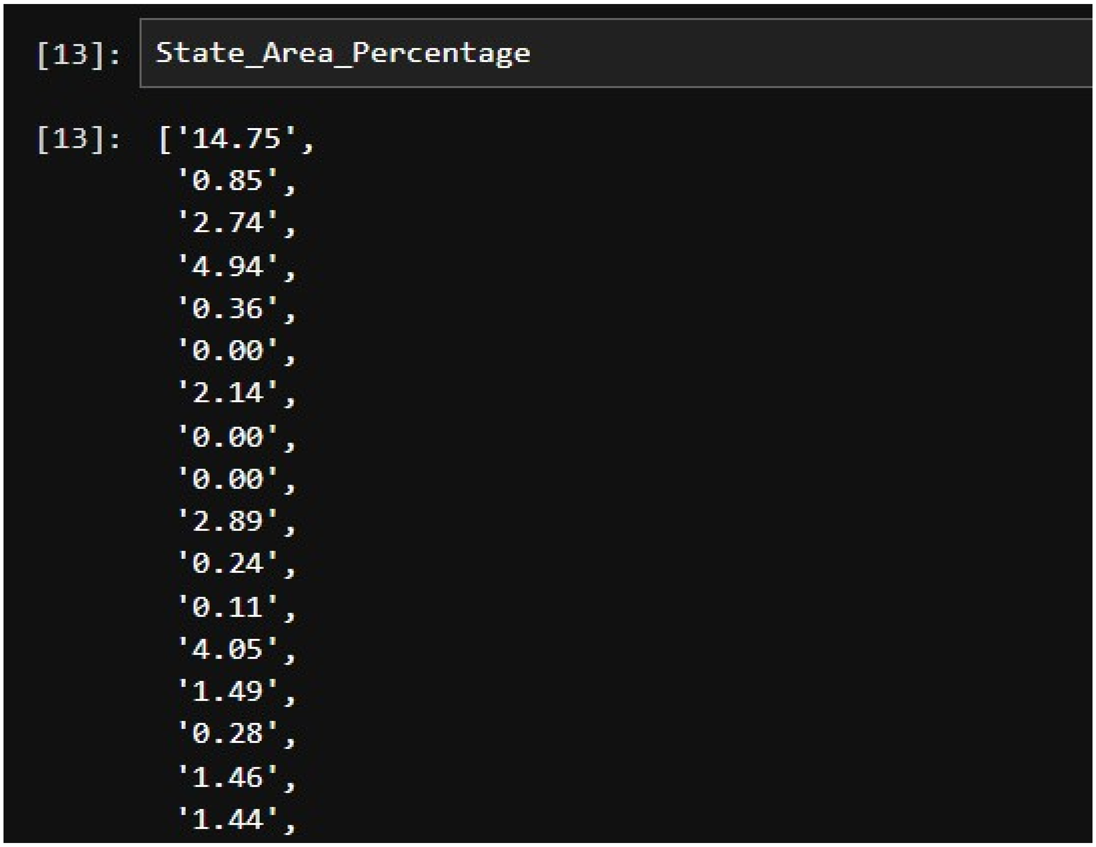
```
[10]: ['8249',
       '160229',
       '83743',
       '78438',
       '94163',
       '114',
       '135191',
       '491',
       '1483',
       '3702',
       '196022',
       '44212',
       '55673',
       '163090',
       '79714',
       '191791',
       '38863',
       '59146',
       '32',
```



```
[11]:  No_of_NP

[11]:  ['9',
        '3',
        '2',
        '8',
        '1',
        '0',
        '3',
        '0',
        '0',
        '1',
        '4',
        '2',
        '5',
        '4',
        '1',
        '5',
        '6',
        '1',
        '0',
        '11',
```

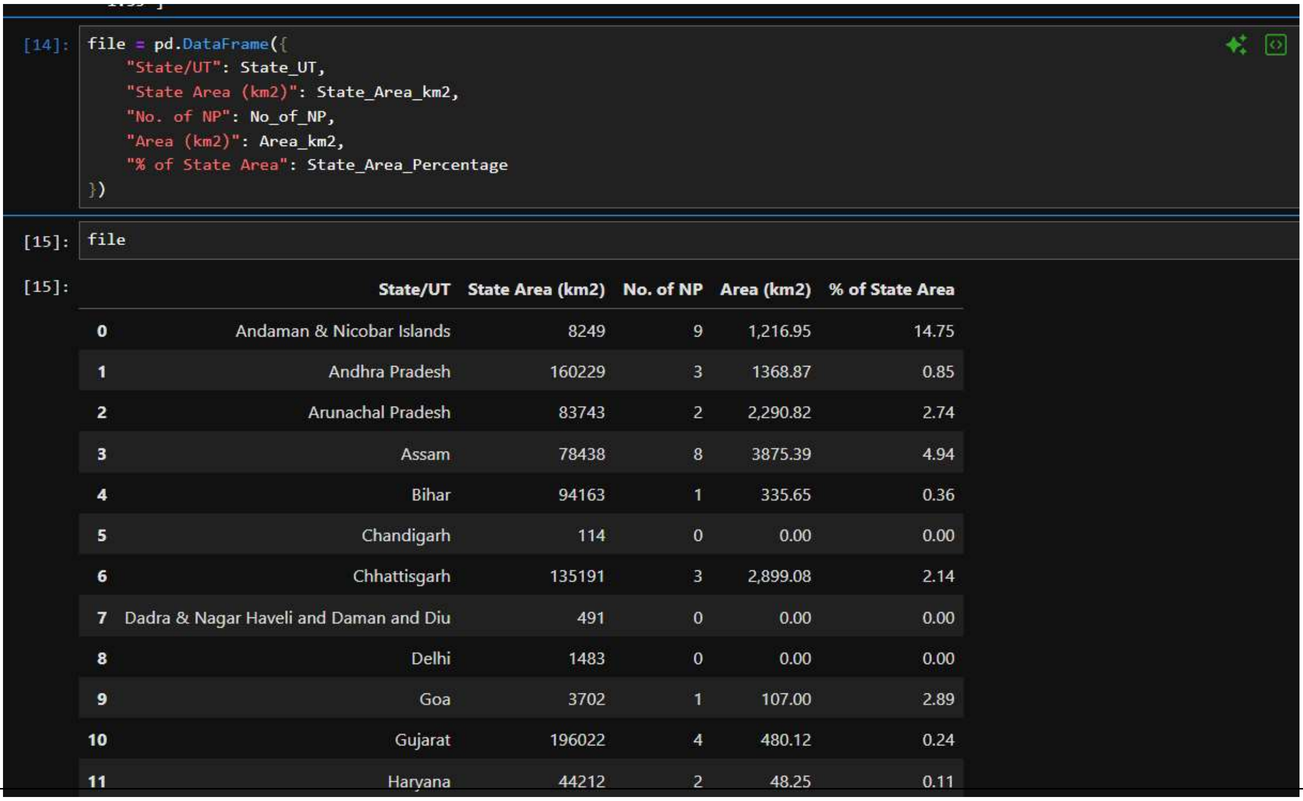


```
[12]:  Area_km2

[12]:  ['1,216.95',
        '1368.87',
        '2,290.82',
        '3875.39',
        '335.65',
        '0.00',
        '2,899.08',
        '0.00',
        '0.00',
        '107.00',
        '480.12',
        '48.25',
        '2,256.28',
        '2432.45',
        '226.33',
        '2,794.05',
```



```
[13]: State_Area_Percentage

[13]: ['14.75',
       '0.85',
       '2.74',
       '4.94',
       '0.36',
       '0.00',
       '2.14',
       '0.00',
       '0.00',
       '2.89',
       '0.24',
       '0.11',
       '4.05',
       '1.49',
       '0.28',
       '1.46',
       '1.44',
```

Transforming the extracted content into a structured table format using DataFrame and Pandas library.



```
[14]: file = pd.DataFrame({
          "State/UT": State_UT,
          "State Area (km2)": State_Area_km2,
          "No. of NP": No_of_NP,
          "Area (km2)": Area_km2,
          "% of State Area": State_Area_Percentage
      })
```

```
[15]: file
```

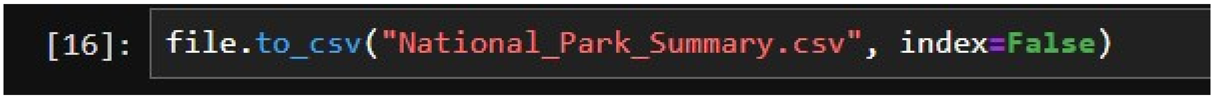
[15]:

	State/UT	State Area (km2)	No. of NP	Area (km2)	% of State Area
0	Andaman & Nicobar Islands	8249	9	1,216.95	14.75
1	Andhra Pradesh	160229	3	1368.87	0.85
2	Arunachal Pradesh	83743	2	2,290.82	2.74
3	Assam	78438	8	3875.39	4.94
4	Bihar	94163	1	335.65	0.36
5	Chandigarh	114	0	0.00	0.00
6	Chhattisgarh	135191	3	2,899.08	2.14
7	Dadra & Nagar Haveli and Daman and Diu	491	0	0.00	0.00
8	Delhi	1483	0	0.00	0.00
9	Goa	3702	1	107.00	2.89
10	Gujarat	196022	4	480.12	0.24
11	Haryana	44212	2	48.25	0.11

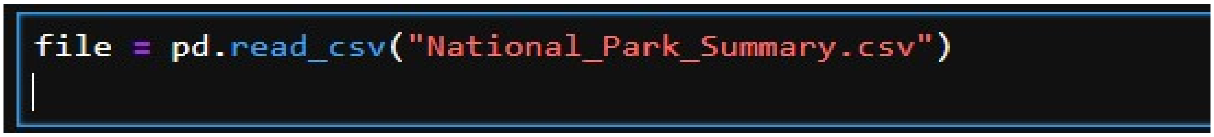
Exporting the data into csv file

Convert the cleaned data into a Pandas DataFrame and export it as a .csv file using df.to_csv().

The CSV file can be opened in Excel for review or visualization.

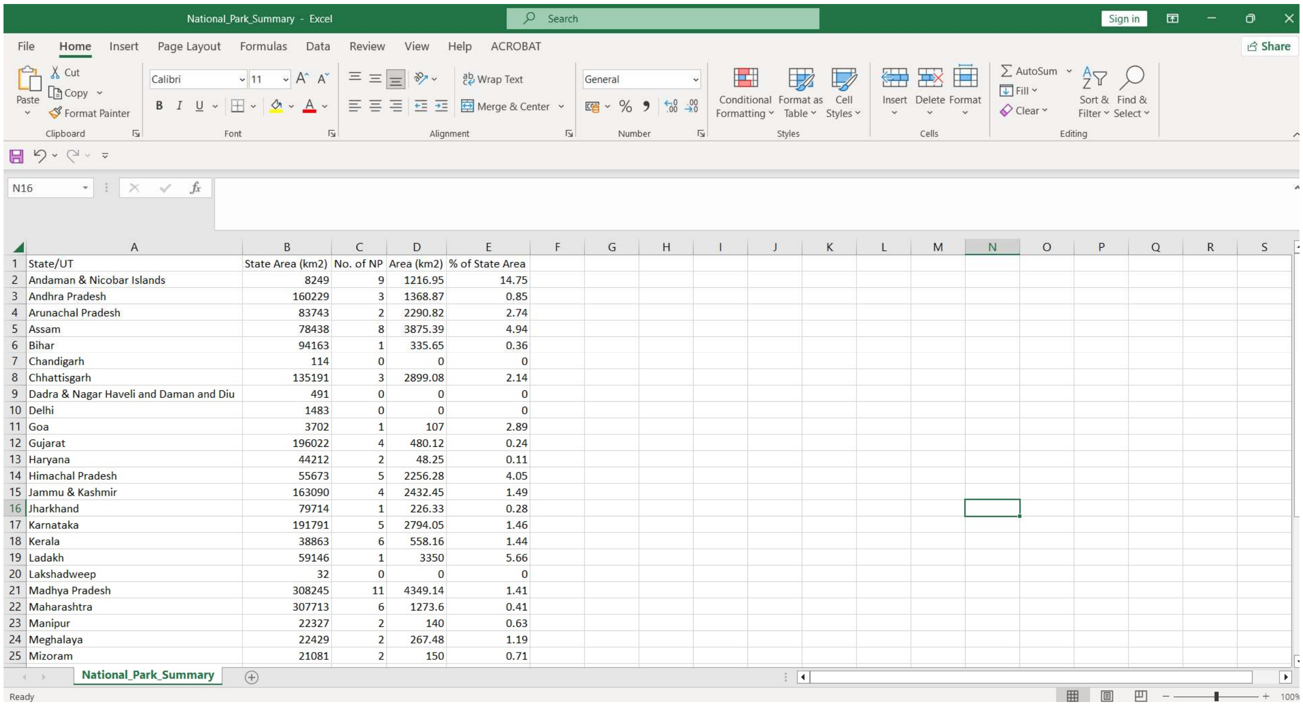


```
[16]: file.to_csv("National_Park_Summary.csv", index=False)
```



```
file = pd.read_csv("National_Park_Summary.csv")
```

Open the CSV file in the Excel application to view its contents.



State/UT	State Area (km2)	No. of NP	Area (km2)	% of State Area
Andaman & Nicobar Islands	8249	9	1216.95	14.75
Andhra Pradesh	160229	3	1368.87	0.85
Arunachal Pradesh	83743	2	2290.82	2.74
Assam	78438	8	3875.39	4.94
Bihar	94163	1	335.65	0.36
Chandigarh	114	0	0	0
Chhattisgarh	135191	3	2899.08	2.14
Dadra & Nagar Haveli and Daman and Diu	491	0	0	0
Delhi	1483	0	0	0
Goa	3702	1	107	2.89
Gujarat	196022	4	480.12	0.24
Haryana	44212	2	48.25	0.11
Himachal Pradesh	55673	5	2256.28	4.05
Jammu & Kashmir	163090	4	2432.45	1.49
Jharkhand	79714	1	226.33	0.28
Karnataka	191791	5	2794.05	1.46
Kerala	38863	6	558.16	1.44
Ladakh	59146	1	3350	5.66
Lakshadweep	32	0	0	0
Madhya Pradesh	308245	11	4349.14	1.41
Maharashtra	307713	6	1273.6	0.41
Manipur	22327	2	140	0.63
Meghalaya	22429	2	267.48	1.19
Mizoram	21081	2	150	0.71

CONCLUSION :

This project demonstrates how web scraping techniques can gather structured data about India’s National Parks. The resulting dataset is a valuable resource for environmental studies, academic research, and policy analysis. It simplifies understanding of park distribution, area coverage, and conservation efforts across Indian states and union territories.