TIGER RESERVES OF INDIA

- Rutuja Prakash Bhosale

Batch- T330 / DS

INTRODUCTION:

Tiger reserves in India play a crucial role in wildlife conservation, protecting the country's tiger population and their natural habitats. With over 50 reserves spread across different states, it is important to analyze their distribution, area, and establishment trends.

OBJECTIVES:

1. Gather information on tiger reserves in India from Wikipedia.
2. Extract data such as name, state, area, and establishment year.
3. Store the scraped data in a structured format (CSV).

TOOLS & LIBRARIES:

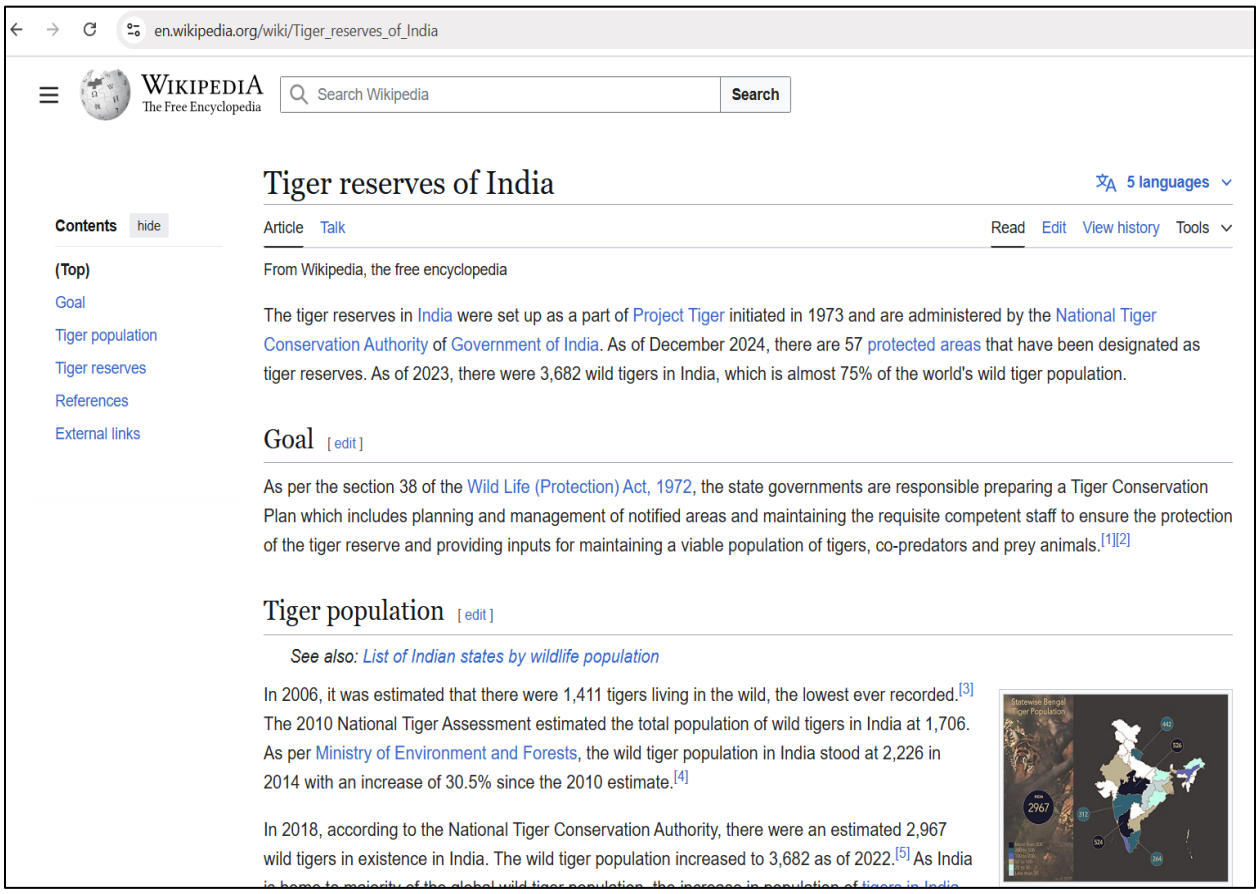
- Python
- BeautifulSoup (for web scraping)
- Requests (for fetching web pages)
- Pandas (for data manipulation)

METHODOLOGY:

1. Identify Source Website: Use the Wikipedia page: Tiger Reserves of 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: Process the extracted data and store it in a CSV file for further analysis.

SELECT THE TARGET WEBSITE(S):

Choose websites that provide the necessary data (e.g., Tiger Reserves of India)



IMPORT THE ESSENTIAL LIBRARIES :

- **Python** for data scraping and processing.
- **BeautifulSoup** and **requests** for web scraping.
- **Pandas** for data manipulation

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

FETCH AND PARSE WIKIPEDIA PAGE :

```
[2]: url="https://en.wikipedia.org/wiki/Tiger_reserves_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-pinned-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-feature-limited-width-content-enabled vector-feature-custom-font-size-clientpref-1 vector-feature-appearance-pinned-clientpref-1 vector-feature-night-mode-enabled skin-theme-clientpref-day vector-sticky-header-enabled vector-toc-available" dir="ltr" lang="en">
<head>
<meta charset="utf-8"/>
<title>Tiger reserves 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-feature-page-tools-pinned-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-feature-limited-width-content-enabled vector-feature-custom-font-size-clientpref-1 vector-feature-appearance-pinned-clientpref-1 vector-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.replace(/-clientpref-\w+$|[^\w-]+/g,'')+'-clientpref-\\w+( |$)'),'$1'+pref+'$2');});}document.documentElement.className=className;}());RLCONF={"wgBreakFrames":false,"wgSeparatorTransformTable":["",""],"wgDigitTransformTable":["",""],"wgDefaultDateFormat":"dmy",
"wgMonthNames":["","January","February","March","April","May","June","July","August","September","October","November","December"],"wgRequestId":"fec6a8c9-547f-4717-81a4-c83ce5a8e382","wgCanonicalNamespace":"","wgCanonicalSpecialPageName":false,"wgNamespaceNumber":0,"wgPageName":"Tiger_reserves_of_Indi
```

EXTRACT HEADERS FROM TABLE :

Extracting and cleaning table headers (<th> elements) from an HTML page using BeautifulSoup

Removing newline characters (\n) using REGEX (sub)

```
[4]: list_of_table_header=soup.find_all("th")  # to find table headers

     list_headers=[]
     for i in range(0,len(list_of_table_header)):
         a=list_of_table_header[i].get_text()
         list_headers.append(a)


     import re
     final_headers_list=[]
     for i in range(0,len(list_headers)):
         a=re.sub("[\n]","",list_headers[i])
         final_headers_list.append(a)

     final_headers_list
```

OUTPUT:

```
[4]: ['Name',
      'Inclusion',
      'Last Notified',
      'State',
      'Tiger population (2023)[5]',
      'Core area (km2)',
      'Buffer area (km2)',
      'Total area (km2)',
      'vteTiger reserves of India',
      'Assam',
      'Arunachal Pradesh',
      'Andhra Pradesh',
      'Bihar',
      'Chhattisgarh',
      'Jharkhand',
      'Karnataka',
      'Kerala',
      'Madhya Pradesh',
      'Maharashtra',
      'Mizoram',
      'Odisha',
      'Rajasthan',
      'Tamil Nadu',
      'Telangana',
      'Uttar Pradesh',
      'Uttarakhand',
      'West Bengal']
```

To extract specific table headers from final_headers_list using indexes.

Step 1 : Using index to find the index of required table headers

```
[5]: stop_index=final_headers_list.index("Total area (km2)")
     stop_index

[5]: 7
```

Step 2: Creating new_final_headers_list with the required values from final_headers_list

```
[6]: new_final_headers_list=[]

     for i in range(0,stop_index+1):
         a=final_headers_list[i]
         new_final_headers_list.append(a)

     new_final_headers_list

[6]: ['Name',
      'Inclusion',
      'Last Notified',
      'State',
      'Tiger population (2023)[5]',
      'Core area (km2)',
      'Buffer area (km2)',
      'Total area (km2)']
```

Step 3: Adding column(No.) in new_final_headers_list

```
[7]: new_final_headers_list.insert(0,"No.")
     new_final_headers_list
```

```
[7]: ['No.',
      'Name',
      'Inclusion',
      'Last Notified',
      'State',
      'Tiger population (2023)[5]',
      'Core area (km2)',
      'Buffer area (km2)',
      'Total area (km2)']
```

EXTRACT DATA FROM TABLE :

Extracting and cleaning table data (<td> elements) from an HTML page using BeautifulSoup

Removing newline characters (\n) using REGEX (sub)

```
[8]: list_table_data=soup.find_all("td")  # to find table data

     data_list=[]
     for i in range(0,len(list_table_data)):
         a=re.sub("[\n]","",list_table_data[i].get_text())
         data_list.append(a)

     data_list
```

```
[8]: ['Bandipur',
      '1973-74',
      '2007',
      'Karnataka',
      '150',
      '872.24',
      '584.06',
      '1,456.3',
      'Corbett',
      '1973-74',
      '2010',
      'Uttarakhand',
      '260',
      '821.99',
      '466.32',
      '1,288.31',
      'Kanha',
      '1973-74',
```

To extract specific table data from data_list using indexes.

Step 1 : Using index to find the index of required table data

```
[9]: stop_index=data_list.index("1,271.47")
     stop_index

[9]: 455
```

Step 2: Creating new_data_list with the required values from data_list

```
[10]: new_data_list=[]

      for i in range(0,stop_index+1):
          a=data_list[i]
          new_data_list.append(a)

      new_data_list

[10]: ['Bandipur',
       '1973-74',
       '2007',
       'Karnataka',
       '150',
       '872.24',
       '584.06',
       '1,456.3',
       'Corbett',
       '1973-74',
       '2010',
       'Uttarakhand',
       '260',
       '821.99',
       '466.32',
       '1,288.31',
       'Kanha',
       '1973-74',
```

Implementing a loop function to extract table content from the given list efficiently.

Content to be extracted are

- **No**
- **Name**
- **Inclusion**
- **Last_Notified**
- **State**
- **Tiger_Population**
- **Core_Area**
- **Buffer_Area**
- **Total_Area**

Adding additional Details in (“No” column)

[11]:
```
No=[]
for i in range(1,58):
    No.append(i)

No
```

[12]:
```
Name=[]
for i in range(0,len(new_data_list),8):
    a=new_data_list[i]
    Name.append(a)

Name
```

[13]:
```
Inclusion=[]
for i in range(1,len(new_data_list),8):
    a=new_data_list[i]
    Inclusion.append(a)

Inclusion
```

[14]:
```
Last_Notified=[]
for i in range(2,len(new_data_list),8):
    a=new_data_list[i]
    Last_Notified.append(a)

Last_Notified
```

[15]:
```
State=[]
for i in range(3,len(new_data_list),8):
    a=new_data_list[i]
    State.append(a)

State
```

[16]:
```
Tiger_Population=[]
for i in range(4,len(new_data_list),8):
    a=new_data_list[i]
    Tiger_Population.append(a)

Tiger_Population
```

[17]:
```
Core_Area=[]
for i in range(5,len(new_data_list),8):
    a=new_data_list[i]
    Core_Area.append(a)

Core_Area
```

```
[18]: Buffer_Area=[]
      for i in range(6,len(new_data_list),8):
          a=new_data_list[i]
          Buffer_Area.append(a)

      Buffer_Area
```

```
[19]: Total_Area=[]
      for i in range(7,len(new_data_list),8):
          a=new_data_list[i]
          Total_Area.append(a)

      Total_Area
```

Implementing a loop function to extract table headers and table data in the form of dictionary.

```
[20]: list_cols=[No,Name,Inclusion,Last_Notified,State,Tiger_Population,Core_Area,Buffer_Area,Total_Area]

      data={}
      for i in range(0,len(new_final_headers_list)):
          data[new_final_headers_list[i]]=list_cols[i]

      data
```

```
[20]: {'No.': [1,
        2,
        3,
        4,
        5,
        6,
        7,
        8,
        9,
        10,
        11,
        12,
        13,
        14,
        15,
        16,
        17,
        18,
```

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

```
[21]: file=pd.DataFrame(data)
      file
```

[21]:

	No.	Name	Inclusion	Last Notified	State	Tiger population (2023)[5]	Core area (km2)	Buffer area (km2)	Total area (km2)
0	1	Bandipur	1973–74	2007	Karnataka	150	872.24	584.06	1,456.3
1	2	Corbett	1973–74	2010	Uttarakhand	260	821.99	466.32	1,288.31
2	3	Kanha	1973–74	2007	Madhya Pradesh	105	917.43	1,134.36	2,051.79
3	4	Manas	1973–74	2008	Assam	58	526.22	2,310.88	2,837.10
4	5	Melghat	1973–74	2007	Maharashtra	57	1,500.49	1,268.03	2,768.52
5	6	Palamau	1973–74	2012	Jharkhand	5	414.08	715.85	1,129.93
6	7	Ranthambore	1973–74	2007	Rajasthan	57	1,113.36	297.92	1,411.29
7	8	Similipal	1973–74	2007	Odisha	16	1,194.75	1,555.25	2,750
8	9	Sunderbans	1973–74	2007	West Bengal	100	1,699.62	885.27	2,584.89
9	10	Periyar	1978–79	2007	Kerala	30	881	44	925
10	11	Sariska	1978–79	2007	Rajasthan	19	881.11	332.23	1,213.34
11	12	Buxa	1982–83	2009	West Bengal	1	390.58	367.32	757.90
12	13	Indravati	1982–83	2009	Chhattisgarh	1	1,258.37	1,540.70	2,799.07
13	14	Namdapha	1982–83	1987	Arunachal Pradesh	1	1,807.82	245.00	2,052.82
14	15	Nagarjunsagar-Srisailam	1982–83	2007	Andhra Pradesh	58	2,595.72	700.59	3,296.31
15	16	Dudhwa	1987–88	2010	Uttar Pradesh	135	1,093.79	1,107.98	2,201.77

Exporting the data into csv file

```
[22]: file.to_csv("Tiger_Reserves_of_India.csv")
```

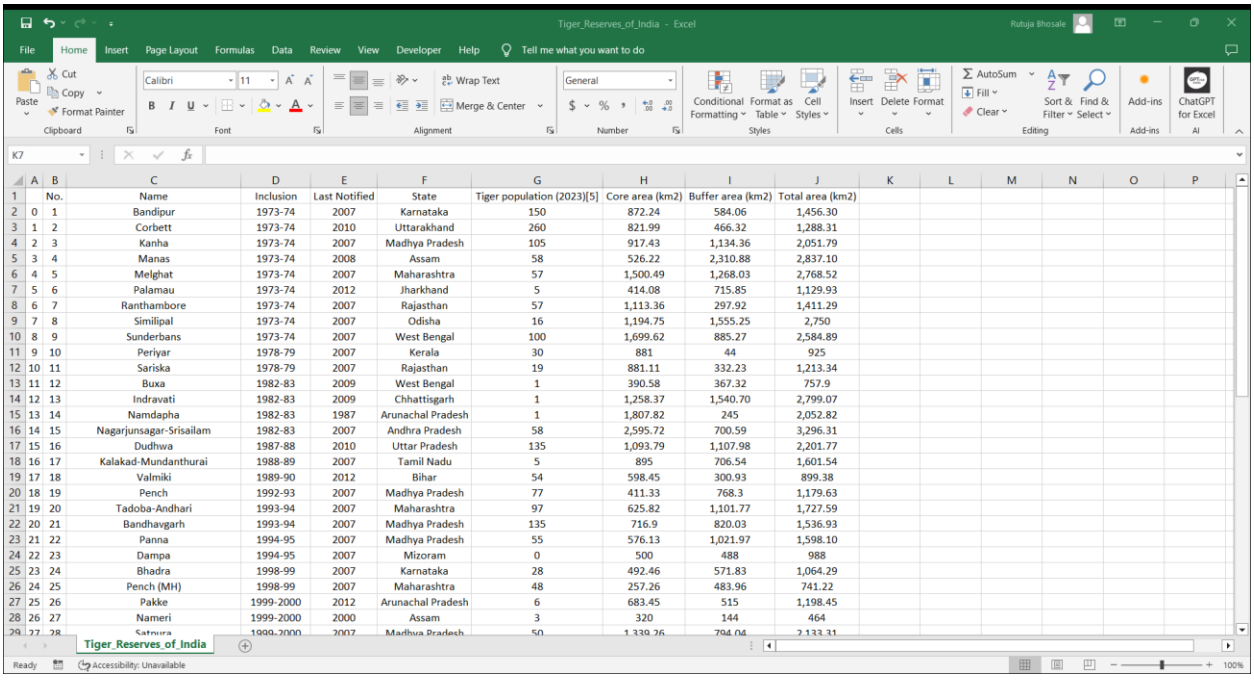
Opening the CSV file in read mode for reading

```
[23]: file=pd.read_csv("Tiger_Reserves_of_India.csv")
      file.drop(["Unnamed: 0"],axis=1,inplace=True)
      file
```

[23]:

	No.	Name	Inclusion	Last Notified	State	Tiger population (2023)[5]	Core area (km2)	Buffer area (km2)	Total area (km2)
0	1	Bandipur	1973–74	2007	Karnataka	150.0	872.24	584.06	1,456.3
1	2	Corbett	1973–74	2010	Uttarakhand	260.0	821.99	466.32	1,288.31
2	3	Kanha	1973–74	2007	Madhya Pradesh	105.0	917.43	1,134.36	2,051.79
3	4	Manas	1973–74	2008	Assam	58.0	526.22	2,310.88	2,837.10
4	5	Melghat	1973–74	2007	Maharashtra	57.0	1,500.49	1,268.03	2,768.52
5	6	Palamau	1973–74	2012	Jharkhand	5.0	414.08	715.85	1,129.93
6	7	Ranthambore	1973–74	2007	Rajasthan	57.0	1,113.36	297.92	1,411.29
7	8	Similipal	1973–74	2007	Odisha	16.0	1,194.75	1,555.25	2,750
8	9	Sunderbans	1973–74	2007	West Bengal	100.0	1,699.62	885.27	2,584.89
9	10	Periyar	1978–79	2007	Kerala	30.0	881	44	925
10	11	Sariska	1978–79	2007	Rajasthan	19.0	881.11	332.23	1,213.34
11	12	Buxa	1982–83	2009	West Bengal	1.0	390.58	367.32	757.90
12	13	Indravati	1982–83	2009	Chhattisgarh	1.0	1,258.37	1,540.70	2,799.07
13	14	Namdapha	1982–83	1987	Arunachal Pradesh	1.0	1,807.82	245.00	2,052.82
14	15	Nagarjunsagar-Srisailam	1982–83	2007	Andhra Pradesh	58.0	2,595.72	700.59	3,296.31

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



CONCLUSION :

This project helps collect and analyze data on tiger reserves in India using web scraping. It provides insights into their locations, sizes, and conservation trends. The data can be used for better understanding and visualization, making it easier to track and study tiger reserves.