

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)

← → ↺

en.wikipedia.org/wiki/Tiger_reserves_of_India

WIKIPEDIA

The Free Encyclopedia

Search Wikipedia

Search

Contents

hide

(Top)

Goal

Tiger population

Tiger reserves

References

External links

Tiger reserves of India

5 languages

Article

Talk

Read

Edit

View history

Tools

From Wikipedia, the free encyclopedia

The tiger reserves in [India](#) were set up as a part of [Project Tiger](#) initiated in 1973 and are administered by the [National Tiger Conservation Authority](#) of [Government of India](#). As of December 2024, there are 57 [protected areas](#) that have been designated as tiger reserves. As of 2023, there were 3,682 wild tigers in India, which is almost 75% of the world's wild tiger population.

Goal

[edit]

As per the section 38 of the [Wild Life \(Protection\) Act, 1972](#), the state governments are responsible preparing a Tiger Conservation Plan which includes planning and management of notified areas and maintaining the requisite competent staff to ensure the protection of the tiger reserve and providing inputs for maintaining a viable population of tigers, co-predators and prey animals.^{[1][2]}

Tiger population

[edit]

See also: [List of Indian states by wildlife population](#)

In 2006, it was estimated that there were 1,411 tigers living in the wild, the lowest ever recorded.^[3] The 2010 National Tiger Assessment estimated the total population of wild tigers in India at 1,706. As per [Ministry of Environment and Forests](#), the wild tiger population in India stood at 2,226 in 2014 with an increase of 30.5% since the 2010 estimate.^[4]

In 2018, according to the National Tiger Conservation Authority, there were an estimated 2,967 wild tigers in existence in India. The wild tiger population increased to 3,682 as of 2022.^[5] As India is home to majority of the global wild tiger population, the increase in population of tigers in India

Statewise Bengal Tiger Population

2006

2011

2014

2018

2022

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=([^\s]+)/);if(cookie){cookie[1].split('%2C').forEach(function(pref){className=className.replace(new RegExp('(\\s| )'+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 India"}
```

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-Srisaillam	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	Simlipal	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-Srisaillam	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.

Tiger_Reserves_of_India - Excel

File Home Insert Page Layout Formulas Data Review View Developer Help Tell me what you want to do

Clipboard Font Alignment Number Styles Cells Editing

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		No.	Name	Inclusion	Last Notified	State	Tiger population (2023)[5]	Core area (km2)	Buffer area (km2)	Total area (km2)						
2	0	1	Bandipur	1973-74	2007	Karnataka	150	872.24	584.06	1,456.30						
3	1	2	Corbett	1973-74	2010	Uttarakhand	260	821.99	466.32	1,288.31						
4	2	3	Kanha	1973-74	2007	Madhya Pradesh	105	917.43	1,134.36	2,051.79						
5	3	4	Manas	1973-74	2008	Assam	58	526.22	2,310.88	2,837.10						
6	4	5	Melghat	1973-74	2007	Maharashtra	57	1,500.49	1,268.03	2,768.52						
7	5	6	Palamau	1973-74	2012	Jharkhand	5	414.08	715.85	1,129.93						
8	6	7	Ranthambore	1973-74	2007	Rajasthan	57	1,113.36	297.92	1,411.29						
9	7	8	Simlipal	1973-74	2007	Odisha	16	1,194.75	1,555.25	2,750						
10	8	9	Sunderbans	1973-74	2007	West Bengal	100	1,699.62	885.27	2,584.89						
11	9	10	Periyar	1978-79	2007	Kerala	30	881	44	925						
12	10	11	Sariska	1978-79	2007	Rajasthan	19	881.11	332.23	1,213.34						
13	11	12	Buxa	1982-83	2009	West Bengal	1	390.58	367.32	757.9						
14	12	13	Indravati	1982-83	2009	Chhattisgarh	1	1,258.37	1,540.70	2,799.07						
15	13	14	Namdapha	1982-83	1987	Arunachal Pradesh	1	1,807.82	245	2,052.82						
16	14	15	Nagarjunsagar-Srisaileam	1982-83	2007	Andhra Pradesh	58	2,595.72	700.59	3,296.31						
17	15	16	Dudhwa	1987-88	2010	Uttar Pradesh	135	1,093.79	1,107.98	2,201.77						
18	16	17	Kalakad-Mundanthurai	1988-89	2007	Tamil Nadu	5	895	706.54	1,601.54						
19	17	18	Valmiki	1989-90	2012	Bihar	54	598.45	300.93	899.38						
20	18	19	Pench	1992-93	2007	Madhya Pradesh	77	411.33	768.3	1,179.63						
21	19	20	Tadoba-Andhari	1993-94	2007	Maharashtra	97	625.82	1,101.77	1,727.59						
22	20	21	Bandhavgarh	1993-94	2007	Madhya Pradesh	135	716.9	820.03	1,536.93						
23	21	22	Panna	1994-95	2007	Madhya Pradesh	55	576.13	1,021.97	1,598.10						
24	22	23	Dampa	1994-95	2007	Mizoram	0	500	488	988						
25	23	24	Bhadra	1998-99	2007	Karnataka	28	492.46	571.83	1,064.29						
26	24	25	Pench (MH)	1998-99	2007	Maharashtra	48	257.26	483.96	741.22						
27	25	26	Pakke	1999-2000	2012	Arunachal Pradesh	6	683.45	515	1,198.45						
28	26	27	Nameri	1999-2000	2000	Assam	3	320	144	464						
29	27	28	Satrphu	1999-2000	2007	Madhya Pradesh	50	1,130.26	704.04	2,133.31						

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.