

In [1]:	#WEB SCRAPING - ASSIGNMENT 4																																																																																																																																																																																																																																													
In [2]:	<pre>#r => Remember the problem statements, notes carefully and scrape the required data using any web scraping tool of your choice. #r -> You have to handle commonly occurring EXCEPTIONS by using exception handling programming. To get information about selenium Exceptions, You may visit following links: #1. https://selenium-python.readthedocs.io/api.html #2. https://www.guru99.com/exception-handling-selenium.html #3. https://stackoverflow.com/questions/38022658/selenium-python-handling-no-such-elementexception/38023345</pre>																																																																																																																																																																																																																																													
In [3]:	<pre>pip install selenium Requirement already satisfied: selenium in c:\users\vasus\anaconda3\lib\site-packages (4.8.0) Requirement already satisfied: trio-websocket<=0.9 in c:\users\vasus\anaconda3\lib\site-packages (from selenium) (0.9.2) Requirement already satisfied: trio<=0.17 in c:\users\vasus\anaconda3\lib\site-packages (from selenium) (0.22.0) Requirement already satisfied: urllib3[socks]<=1.26 in c:\users\vasus\anaconda3\lib\site-packages (from selenium) (1.26.0) Requirement already satisfied: certifi<=2021.10.8 in c:\users\vasus\anaconda3\lib\site-packages (from selenium) (2021.10.8) Requirement already satisfied: idna in c:\users\vasus\anaconda3\lib\site-packages (from trio<=0.17->selenium) (3.3) Requirement already satisfied: cffi<=1.14 in c:\users\vasus\anaconda3\lib\site-packages (from trio<=0.17->selenium) (1.15.0) Requirement already satisfied: sniffio in c:\users\vasus\anaconda3\lib\site-packages (from trio<=0.17->selenium) (1.2.0) Requirement already satisfied: attrs<=19.2.0 in c:\users\vasus\anaconda3\lib\site-packages (from trio<=0.17->selenium) (21.4.0) Requirement already satisfied: exceptiongroup<=1.0.0rc8 in c:\users\vasus\anaconda3\lib\site-packages (from trio<=0.17->selenium) (1.1.0) Requirement already satisfied: async-generator<=1.9 in c:\users\vasus\anaconda3\lib\site-packages (from trio<=0.17->selenium) (1.10) Requirement already satisfied: sortedcontainers in c:\users\vasus\anaconda3\lib\site-packages (from trio<=0.17->selenium) (2.4.0) Requirement already satisfied: wsproto<=0.14 in c:\users\vasus\anaconda3\lib\site-packages (from trio-websocket<=0.9->selenium) (1.2.0) Requirement already satisfied: h1<=1,>=0.9.0 in c:\users\vasus\anaconda3\lib\site-packages (from wsproto<=0.14->trio-websocket<=0.9->selenium) (0.14.0)</pre>																																																																																																																																																																																																																																													
In [2]:	<pre>import pandas as pd import selenium from selenium.webdriver import webdriver import warnings warnings.filterwarnings('ignore') import requests import time from selenium.common.exceptions import NoSuchElementException import re from bs4 import BeautifulSoup from selenium.webdriver.common.keys import Keys import Keys from selenium.webdriver.common.by import By</pre>																																																																																																																																																																																																																																													
In [5]:	<pre>#1. Scrape the details of most viewed videos on YouTube from Wikipedia. url = https://en.wikipedia.org/wiki/List_of_most-viewed_Youtube_videos #you need to find following details: #A) Rank #B) Name #C) Artist #D) Upload date #E) Views</pre>																																																																																																																																																																																																																																													
In [6]:	<pre>driver=webdriver.Chrome("chromedriver.exe") driver.get("https://en.wikipedia.org/wiki/List_of_most-viewed_Youtube_videos")</pre>																																																																																																																																																																																																																																													
In [20]:	<pre>Rank=[] driver.find_elements(By.XPATH, '//table[@class="wikitable sortable jquery-tablesorter"]//tbody/tr/td[1]') for i in rank[:30]: Rank.append(ranks)</pre>																																																																																																																																																																																																																																													
In [23]:	<pre>Name=[] name=driver.find_elements(By.XPATH, '//table[@class="wikitable sortable jquery-tablesorter"]//tbody/tr/td[2]') for i in name[:10]: names=i.text Name.append(names)</pre>																																																																																																																																																																																																																																													
In [26]:	<pre>Artist=[] artist=driver.find_elements(By.XPATH, '//table[@class="wikitable sortable jquery-tablesorter"]//tbody/tr/td[3]') for i in artist[:20]: artists=i.text Artist.append(artists)</pre>																																																																																																																																																																																																																																													
In [31]:	<pre>Upload_date=[] upload_date=driver.find_elements(By.XPATH, '//table[@class="wikitable sortable jquery-tablesorter"]//tbody/tr/td[5]') for i in upload_date[:10]: Upload_dates=i.text Upload_date.append(upload_dates)</pre>																																																																																																																																																																																																																																													
In [38]:	<pre>Views=[] view=driver.find_elements(By.XPATH, '//table[@class="wikitable sortable jquery-tablesorter"]//tbody/tr/td[4]') for i in view[:10]: Views=i.text Views.append(views)</pre>																																																																																																																																																																																																																																													
In [40]:	<pre>print(len(Rank),len(Name),len(Artist),len(Upload_date),len(Views)) 30 30 30 30 30</pre>																																																																																																																																																																																																																																													
In [42]:	<pre>df=pd.DataFrame({"RANK":Rank,"NAME":Name,"ARTIST_NAME":Artist,"UPLOAD_DATE":Upload_dates,"VIEWS(in billion)":Views}) df</pre>																																																																																																																																																																																																																																													
Out[42]:	<table><tr><th>RANK</th><th>NAME</th><th>ARTIST NAME</th><th>UPLOAD_DATE</th><th>VIEWS(in billion)</th></tr><tr><td>0</td><td>1.</td><td>"Baby Shark Dance"[4]</td><td>Pinkfong Baby Shark - Kids' Songs & Stories</td><td>June 17, 2016</td><td>12.37</td></tr><tr><td>1</td><td>2.</td><td>"Despacito"[7]</td><td>Luis Fonsi</td><td>January 12, 2017</td><td>8.09</td></tr><tr><td>2</td><td>3.</td><td>"Johnny Johnny Yes Papa"[4]</td><td>Loooloo Kids</td><td>October 8, 2016</td><td>6.63</td></tr><tr><td>3</td><td>4.</td><td>"Bath Song"[15]</td><td>Cocomelon - Nursery Rhymes</td><td>May 2, 2018</td><td>6.03</td></tr><tr><td>4</td><td>5.</td><td>"Shape of You"[16]</td><td>Ed Sheeran</td><td>January 30, 2017</td><td>5.92</td></tr><tr><td>5</td><td>6.</td><td>"See You Again"[18]</td><td>Wiz Khalifa</td><td>April 6, 2015</td><td>5.79</td></tr><tr><td>6</td><td>7.</td><td>"Phonics Song with Two Words"[2]</td><td>ChuChuTV</td><td>March 6, 2014</td><td>5.17</td></tr><tr><td>7</td><td>8.</td><td>"Wheels on the Bus"[4]</td><td>Cocomelon - Nursery Rhymes</td><td>May 24, 2018</td><td>4.95</td></tr><tr><td>8</td><td>9.</td><td>"Uptown Funk"[25]</td><td>Mark Ronson</td><td>November 19, 2014</td><td>4.84</td></tr><tr><td>9</td><td>10.</td><td>"Learning Colors - Colorful Eggs on a Farm"[2]</td><td>Minsikha TV</td><td>February 27, 2018</td><td>4.83</td></tr><tr><td>10</td><td>11.</td><td>"Gangnam Style"[27]</td><td>Psy</td><td>July 15, 2012</td><td>4.70</td></tr><tr><td>11</td><td>12.</td><td>"Masha and the Bear - Recipe for Disaster"[32]</td><td>Get Movies</td><td>January 31, 2012</td><td>4.53</td></tr><tr><td>12</td><td>13.</td><td>"Dance Tu Dena"[35]</td><td>EI Chhomba</td><td>April 5, 2018</td><td>4.24</td></tr><tr><td>13</td><td>14.</td><td>"Sugar"[34]</td><td>Maroon 5</td><td>January 14, 2015</td><td>3.82</td></tr><tr><td>14</td><td>15.</td><td>"Avei F[36]</td><td>Crazy Frog</td><td>June 16, 2009</td><td>3.76</td></tr><tr><td>15</td><td>16.</td><td>"Roar"[36]</td><td>Katy Perry</td><td>September 5, 2013</td><td>3.73</td></tr><tr><td>16</td><td>17.</td><td>"Counting Stars"[37]</td><td>OneRepublic</td><td>May 31, 2013</td><td>3.73</td></tr><tr><td>17</td><td>18.</td><td>"Sorry"[38]</td><td>Justin Bieber</td><td>October 22, 2015</td><td>3.63</td></tr><tr><td>18</td><td>19.</td><td>"Thinking Out Loud"[39]</td><td>Ed Sheeran</td><td>October 7, 2014</td><td>3.55</td></tr><tr><td>19</td><td>20.</td><td>"Baas Baas Black Sheep"[40]</td><td>Cocomelon - Nursery Rhymes</td><td>June 25, 2018</td><td>3.51</td></tr><tr><td>20</td><td>21.</td><td>"Waka Waka (This Time for Africa)"[41]</td><td>Shakira</td><td>June 4, 2010</td><td>3.48</td></tr><tr><td>21</td><td>22.</td><td>"Dark Horse"[42]</td><td>Katy Perry</td><td>February 30, 2014</td><td>3.45</td></tr><tr><td>22</td><td>23.</td><td>"Faded"[43]</td><td>Alan Walker</td><td>December 3, 2015</td><td>3.41</td></tr><tr><td>23</td><td>24.</td><td>"Let Her Go"[44]</td><td>Passenger</td><td>July 25, 2012</td><td>3.38</td></tr><tr><td>24</td><td>25.</td><td>"Gifs Like You"[45]</td><td>Maroon 5</td><td>May 31, 2018</td><td>3.37</td></tr><tr><td>25</td><td>26.</td><td>"Perfect"[46]</td><td>Ed Sheeran</td><td>November 9, 2017</td><td>3.37</td></tr><tr><td>26</td><td>27.</td><td>"Bailando"[47]</td><td>Enrique Iglesias</td><td>April 11, 2014</td><td>3.34</td></tr><tr><td>27</td><td>28.</td><td>"Lean On"[48]</td><td>Major Lazer</td><td>March 22, 2015</td><td>3.32</td></tr><tr><td>28</td><td>29.</td><td>"Lalali Ki Kahni"[49]</td><td>Jango Toons</td><td>June 14, 2018</td><td>3.32</td></tr><tr><td>29</td><td>30.</td><td>"Humpty the train on a flats ride"[50]</td><td>Kiddesoft Hindi - Nursery Rhymes & Kids Songs</td><td>January 26, 2018</td><td>3.31</td></tr></table>	RANK	NAME	ARTIST NAME	UPLOAD_DATE	VIEWS(in billion)	0	1.	"Baby Shark Dance"[4]	Pinkfong Baby Shark - Kids' Songs & Stories	June 17, 2016	12.37	1	2.	"Despacito"[7]	Luis Fonsi	January 12, 2017	8.09	2	3.	"Johnny Johnny Yes Papa"[4]	Loooloo Kids	October 8, 2016	6.63	3	4.	"Bath Song"[15]	Cocomelon - Nursery Rhymes	May 2, 2018	6.03	4	5.	"Shape of You"[16]	Ed Sheeran	January 30, 2017	5.92	5	6.	"See You Again"[18]	Wiz Khalifa	April 6, 2015	5.79	6	7.	"Phonics Song with Two Words"[2]	ChuChuTV	March 6, 2014	5.17	7	8.	"Wheels on the Bus"[4]	Cocomelon - Nursery Rhymes	May 24, 2018	4.95	8	9.	"Uptown Funk"[25]	Mark Ronson	November 19, 2014	4.84	9	10.	"Learning Colors - Colorful Eggs on a Farm"[2]	Minsikha TV	February 27, 2018	4.83	10	11.	"Gangnam Style"[27]	Psy	July 15, 2012	4.70	11	12.	"Masha and the Bear - Recipe for Disaster"[32]	Get Movies	January 31, 2012	4.53	12	13.	"Dance Tu Dena"[35]	EI Chhomba	April 5, 2018	4.24	13	14.	"Sugar"[34]	Maroon 5	January 14, 2015	3.82	14	15.	"Avei F[36]	Crazy Frog	June 16, 2009	3.76	15	16.	"Roar"[36]	Katy Perry	September 5, 2013	3.73	16	17.	"Counting Stars"[37]	OneRepublic	May 31, 2013	3.73	17	18.	"Sorry"[38]	Justin Bieber	October 22, 2015	3.63	18	19.	"Thinking Out Loud"[39]	Ed Sheeran	October 7, 2014	3.55	19	20.	"Baas Baas Black Sheep"[40]	Cocomelon - Nursery Rhymes	June 25, 2018	3.51	20	21.	"Waka Waka (This Time for Africa)"[41]	Shakira	June 4, 2010	3.48	21	22.	"Dark Horse"[42]	Katy Perry	February 30, 2014	3.45	22	23.	"Faded"[43]	Alan Walker	December 3, 2015	3.41	23	24.	"Let Her Go"[44]	Passenger	July 25, 2012	3.38	24	25.	"Gifs Like You"[45]	Maroon 5	May 31, 2018	3.37	25	26.	"Perfect"[46]	Ed Sheeran	November 9, 2017	3.37	26	27.	"Bailando"[47]	Enrique Iglesias	April 11, 2014	3.34	27	28.	"Lean On"[48]	Major Lazer	March 22, 2015	3.32	28	29.	"Lalali Ki Kahni"[49]	Jango Toons	June 14, 2018	3.32	29	30.	"Humpty the train on a flats ride"[50]	Kiddesoft Hindi - Nursery Rhymes & Kids Songs	January 26, 2018	3.31																																																				
RANK	NAME	ARTIST NAME	UPLOAD_DATE	VIEWS(in billion)																																																																																																																																																																																																																																										
0	1.	"Baby Shark Dance"[4]	Pinkfong Baby Shark - Kids' Songs & Stories	June 17, 2016	12.37																																																																																																																																																																																																																																									
1	2.	"Despacito"[7]	Luis Fonsi	January 12, 2017	8.09																																																																																																																																																																																																																																									
2	3.	"Johnny Johnny Yes Papa"[4]	Loooloo Kids	October 8, 2016	6.63																																																																																																																																																																																																																																									
3	4.	"Bath Song"[15]	Cocomelon - Nursery Rhymes	May 2, 2018	6.03																																																																																																																																																																																																																																									
4	5.	"Shape of You"[16]	Ed Sheeran	January 30, 2017	5.92																																																																																																																																																																																																																																									
5	6.	"See You Again"[18]	Wiz Khalifa	April 6, 2015	5.79																																																																																																																																																																																																																																									
6	7.	"Phonics Song with Two Words"[2]	ChuChuTV	March 6, 2014	5.17																																																																																																																																																																																																																																									
7	8.	"Wheels on the Bus"[4]	Cocomelon - Nursery Rhymes	May 24, 2018	4.95																																																																																																																																																																																																																																									
8	9.	"Uptown Funk"[25]	Mark Ronson	November 19, 2014	4.84																																																																																																																																																																																																																																									
9	10.	"Learning Colors - Colorful Eggs on a Farm"[2]	Minsikha TV	February 27, 2018	4.83																																																																																																																																																																																																																																									
10	11.	"Gangnam Style"[27]	Psy	July 15, 2012	4.70																																																																																																																																																																																																																																									
11	12.	"Masha and the Bear - Recipe for Disaster"[32]	Get Movies	January 31, 2012	4.53																																																																																																																																																																																																																																									
12	13.	"Dance Tu Dena"[35]	EI Chhomba	April 5, 2018	4.24																																																																																																																																																																																																																																									
13	14.	"Sugar"[34]	Maroon 5	January 14, 2015	3.82																																																																																																																																																																																																																																									
14	15.	"Avei F[36]	Crazy Frog	June 16, 2009	3.76																																																																																																																																																																																																																																									
15	16.	"Roar"[36]	Katy Perry	September 5, 2013	3.73																																																																																																																																																																																																																																									
16	17.	"Counting Stars"[37]	OneRepublic	May 31, 2013	3.73																																																																																																																																																																																																																																									
17	18.	"Sorry"[38]	Justin Bieber	October 22, 2015	3.63																																																																																																																																																																																																																																									
18	19.	"Thinking Out Loud"[39]	Ed Sheeran	October 7, 2014	3.55																																																																																																																																																																																																																																									
19	20.	"Baas Baas Black Sheep"[40]	Cocomelon - Nursery Rhymes	June 25, 2018	3.51																																																																																																																																																																																																																																									
20	21.	"Waka Waka (This Time for Africa)"[41]	Shakira	June 4, 2010	3.48																																																																																																																																																																																																																																									
21	22.	"Dark Horse"[42]	Katy Perry	February 30, 2014	3.45																																																																																																																																																																																																																																									
22	23.	"Faded"[43]	Alan Walker	December 3, 2015	3.41																																																																																																																																																																																																																																									
23	24.	"Let Her Go"[44]	Passenger	July 25, 2012	3.38																																																																																																																																																																																																																																									
24	25.	"Gifs Like You"[45]	Maroon 5	May 31, 2018	3.37																																																																																																																																																																																																																																									
25	26.	"Perfect"[46]	Ed Sheeran	November 9, 2017	3.37																																																																																																																																																																																																																																									
26	27.	"Bailando"[47]	Enrique Iglesias	April 11, 2014	3.34																																																																																																																																																																																																																																									
27	28.	"Lean On"[48]	Major Lazer	March 22, 2015	3.32																																																																																																																																																																																																																																									
28	29.	"Lalali Ki Kahni"[49]	Jango Toons	June 14, 2018	3.32																																																																																																																																																																																																																																									
29	30.	"Humpty the train on a flats ride"[50]	Kiddesoft Hindi - Nursery Rhymes & Kids Songs	January 26, 2018	3.31																																																																																																																																																																																																																																									
In [43]:	<pre>#2. Scrape the details teamIndia's international fixtures from bcci.tv. url = https://www.bcci.tv/ #you need to find following details: #A) Match title (i.e. int000) #B) Series #C) Place #D) Date #E) Time #Note: - from bcci.tv home page you have reach to the international fixture page through code</pre>																																																																																																																																																																																																																																													
In [13]:	<pre>driver=webdriver.Chrome("chromedriver.exe") driver.get("https://www.bcci.tv/")</pre>																																																																																																																																																																																																																																													
In [14]:	<pre>IF=driver.find_element(By.XPATH, '//li[@class="nav-item"]') IF.click()</pre>																																																																																																																																																																																																																																													
In [18]:	<pre>Place=[] places=driver.find_elements(By.XPATH, '//span[@class="ng-binding"]') for i in places: p=i.text place.append(p)</pre>																																																																																																																																																																																																																																													
In [27]:	<pre>Time=[] times=driver.find_elements(By.XPATH, '//div[@class="match-time ng-noargin ng-binding"]') for i in times: t=i.text time.append(t)</pre>																																																																																																																																																																																																																																													
In [30]:	<pre>date=[] dates=driver.find_elements(By.XPATH, '//div[@class="match-dates ng-binding"]') for i in dates: d=i.text date.append(d)</pre>																																																																																																																																																																																																																																													
In [33]:	<pre>series=[] series=driver.find_elements(By.XPATH, '//h5[@class="match-tournament-name ng-binding"]') for i in series: s=i.text series.append(s)</pre>																																																																																																																																																																																																																																													
In [36]:	<pre>title=[] titles=driver.find_elements(By.XPATH, '//span[@class="matchOrderText ng-binding ng-scope"]') for i in titles: tli=i.text title.append(tli)</pre>																																																																																																																																																																																																																																													
In [39]:	<pre>print(len(title),len(series),len(date),len(time),len(place)) 3 3 3 3 3</pre>																																																																																																																																																																																																																																													
In [41]:	<pre>df=pd.DataFrame({"MATCH_TITLE":title,"SERIES":series,"PLACE":place,"DATE":date,"TIME":time}) df</pre>																																																																																																																																																																																																																																													
Out[41]:	<table><tr><th></th><th>MATCH_TITLE</th><th>SERIES</th><th>PLACE</th><th>DATE</th><th>TIME</th></tr><tr><td>0</td><td>1st Codi</td><td>AUSTRALIA TOUR OF INDIA 2023</td><td>Mumbai</td><td>17 MAR 2023</td><td>1:30 PM IST</td></tr><tr><td>1</td><td>2nd Codi</td><td>AUSTRALIA TOUR OF INDIA 2023</td><td>Vishakhapatnam</td><td>19 MAR 2023</td><td>1:30 PM IST</td></tr><tr><td>2</td><td>3rd Codi</td><td>AUSTRALIA TOUR OF INDIA 2023</td><td>Chennai</td><td>22 MAR 2023</td><td>1:30 PM IST</td></tr></table>		MATCH_TITLE	SERIES	PLACE	DATE	TIME	0	1st Codi	AUSTRALIA TOUR OF INDIA 2023	Mumbai	17 MAR 2023	1:30 PM IST	1	2nd Codi	AUSTRALIA TOUR OF INDIA 2023	Vishakhapatnam	19 MAR 2023	1:30 PM IST	2	3rd Codi	AUSTRALIA TOUR OF INDIA 2023	Chennai	22 MAR 2023	1:30 PM IST																																																																																																																																																																																																																					
	MATCH_TITLE	SERIES	PLACE	DATE	TIME																																																																																																																																																																																																																																									
0	1st Codi	AUSTRALIA TOUR OF INDIA 2023	Mumbai	17 MAR 2023	1:30 PM IST																																																																																																																																																																																																																																									
1	2nd Codi	AUSTRALIA TOUR OF INDIA 2023	Vishakhapatnam	19 MAR 2023	1:30 PM IST																																																																																																																																																																																																																																									
2	3rd Codi	AUSTRALIA TOUR OF INDIA 2023	Chennai	22 MAR 2023	1:30 PM IST																																																																																																																																																																																																																																									
In [45]:	<pre>#3. Scrape the details of State-wise GDP of India from statistictime.com. url = http://statisticstimes.com/ #you have to find following details: #A) Rank #B) State #C) GDP(10-19) : at current prices #D) GDP(19-20) : at current prices #E) Share(18-19) #F) Share(20) (in \$ billion) #Note: - From statistictimes home page you have to reach to economy page through code.</pre>																																																																																																																																																																																																																																													
In [46]:	<pre>driver=webdriver.Chrome("chromedriver.exe") driver.get("http://statisticstimes.com/")</pre>																																																																																																																																																																																																																																													
In [49]:	<pre>Economy=driver.find_element(By.XPATH, '//html/body/div[2]/div[1]/div[2]/div[2]/button') Economy.click()</pre>																																																																																																																																																																																																																																													
In [54]:	<pre>India=driver.find_elements(By.XPATH, '//html/body/div[2]/div[1]/div[2]/div[2]/div/a[3]')</pre>																																																																																																																																																																																																																																													
In [56]:	<pre>GDP=driver.find_element(By.XPATH, '//html/body/div[2]/div[2]/div[2]/div[2]/ul/li[1]/a') GDP.click()</pre>																																																																																																																																																																																																																																													
In [79]:	<pre>Rank=[] rank=driver.find_elements(By.XPATH, '//table[@class="display dataTable"]//tbody/tr/td[1]') for i in rank[:30]: ranks=i.text Rank.append(ranks)</pre>																																																																																																																																																																																																																																													
In [80]:	<pre>State=[] state=driver.find_elements(By.XPATH, '//table[@class="display dataTable"]//tbody/tr/td[2]') for i in state[:10]: states=i.text State.append(states)</pre>																																																																																																																																																																																																																																													
In [81]:	<pre>GSDP_1=[] gsdp_1=driver.find_elements(By.XPATH, '//table[@class="display dataTable"]//tbody/tr/td[4]') for i in gsdp_1[:10]: gsdp_1=i.text GSDP_1.append(gsdp_1)</pre>																																																																																																																																																																																																																																													
In [82]:	<pre>GSDP_2=[] gsdp_2=driver.find_elements(By.XPATH, '//table[@class="display dataTable"]//tbody/tr/td[7]') for i in gsdp_2[:10]: gsdp_2=i.text GSDP_2.append(gsdp_2)</pre>																																																																																																																																																																																																																																													
In [83]:	<pre>Share=[] share=driver.find_elements(By.XPATH, '//table[@class="display dataTable"]//tbody/tr/td[5]') for i in share[:10]: shares=i.text Share.append(shares)</pre>																																																																																																																																																																																																																																													
In [84]:	<pre>GDPs=[] gdp=driver.find_elements(By.XPATH, '//table[@class="display dataTable"]//tbody/tr/td[6]') for i in gdp[:10]: gdp=i.text GDPs.append(gdp)</pre>																																																																																																																																																																																																																																													
In [85]:	<pre>print(len(Rank),len(State),len(GSDP_1),len(GSDP_2),len(Share),len(GDPs)) 33 33 33 33 33</pre>																																																																																																																																																																																																																																													
In [86]:	<pre>df=pd.DataFrame({"RANK":Rank,"NAME_OF_STATES":State,"GSDP(1)":GSDP_1,"GSDP(2)":GSDP_2,"SHARE(in %)":Share,"GDP(in billion)":GDPs}) df</pre>																																																																																																																																																																																																																																													
Out[86]:	<table><tr><th>RANK</th><th>NAME_OF_STATES</th><th>GSDP(1)</th><th>GSDP(2)</th><th>SHARE(in %)</th><th>GDP(in billion)</th></tr><tr><td>0</td><td>1</td><td>Maharashtra</td><td>2,632,792</td><td>-</td><td>13.94%</td><td>399,021</td></tr><tr><td>1</td><td>2</td><td>Tamil Nadu</td><td>1,630,208</td><td>1,312,929</td><td>8.63%</td><td>247,629</td></tr><tr><td>2</td><td>3</td><td>Uttar Pradesh</td><td>1,584,764</td><td>1,166,817</td><td>8.39%</td><td>240,726</td></tr><tr><td>3</td><td>4</td><td>Gujarat</td><td>1,502,899</td><td>-</td><td>7.96%</td><td>228,290</td></tr><tr><td>4</td><td>5</td><td>Karnataka</td><td>1,493,127</td><td>1,156,039</td><td>7.91%</td><td>226,806</td></tr><tr><td>5</td><td>6</td><td>West Bengal</td><td>1,089,898</td><td>793,223</td><td>5.77%</td><td>165,556</td></tr><tr><td>6</td><td>7</td><td>Rajasthan</td><td>942,586</td><td>711,627</td><td>4.99%</td><td>143,179</td></tr><tr><td>7</td><td>8</td><td>Andhra Pradesh</td><td>862,957</td><td>672,018</td><td>4.57%</td><td>131,083</td></tr><tr><td>8</td><td>9</td><td>Telangana</td><td>861,031</td><td>663,258</td><td>4.56%</td><td>130,791</td></tr><tr><td>9</td><td>10</td><td>Madhya Pradesh</td><td>809,592</td><td>561,801</td><td>4.29%</td><td>122,977</td></tr><tr><td>10</td><td>11</td><td>Kerala</td><td>781,653</td><td>-</td><td>4.14%</td><td>118,733</td></tr><tr><td>11</td><td>12</td><td>Delhi</td><td>774,870</td><td>634,408</td><td>4.10%</td><td>117,703</td></tr><tr><td>12</td><td>13</td><td>Haryana</td><td>734,163</td><td>572,240</td><td>3.89%</td><td>111,519</td></tr><tr><td>13</td><td>14</td><td>Bihar</td><td>530,363</td><td>414,977</td><td>2.81%</td><td>80,562</td></tr><tr><td>14</td><td>15</td><td>Punjab</td><td>526,376</td><td>418,868</td><td>2.79%</td><td>79,957</td></tr><tr><td>15</td><td>16</td><td>Odisha</td><td>487,805</td><td>396,499</td><td>2.58%</td><td>74,098</td></tr><tr><td>16</td><td>17</td><td>Assam</td><td>315,881</td><td>-</td><td>1.67%</td><td>47,982</td></tr><tr><td>17</td><td>18</td><td>Chhattisgarh</td><td>304,063</td><td>243,477</td><td>1.61%</td><td>46,187</td></tr><tr><td>18</td><td>19</td><td>Jharkhand</td><td>297,204</td><td>240,036</td><td>1.57%</td><td>45,145</td></tr><tr><td>19</td><td>20</td><td>Uttarakhand</td><td>245,895</td><td>-</td><td>1.30%</td><td>37,351</td></tr><tr><td>20</td><td>21</td><td>Jammu & Kashmir</td><td>155,956</td><td>-</td><td>0.83%</td><td>23,690</td></tr><tr><td>21</td><td>22</td><td>Himachal Pradesh</td><td>153,845</td><td>124,403</td><td>0.81%</td><td>23,369</td></tr><tr><td>22</td><td>23</td><td>Goa</td><td>73,170</td><td>63,408</td><td>0.39%</td><td>11,115</td></tr><tr><td>23</td><td>24</td><td>Tripura</td><td>49,845</td><td>40,583</td><td>0.26%</td><td>7,571</td></tr><tr><td>24</td><td>25</td><td>Chandigarh</td><td>42,114</td><td>-</td><td>0.22%</td><td>6,397</td></tr><tr><td>25</td><td>26</td><td>Puducherry</td><td>34,433</td><td>25,083</td><td>0.18%</td><td>5,230</td></tr><tr><td>26</td><td>27</td><td>Meghalaya</td><td>33,481</td><td>26,695</td><td>0.18%</td><td>5,096</td></tr><tr><td>27</td><td>28</td><td>Sikkim</td><td>28,723</td><td>20,017</td><td>0.15%</td><td>4,363</td></tr><tr><td>28</td><td>29</td><td>Manipur</td><td>27,870</td><td>20,673</td><td>0.15%</td><td>4,233</td></tr><tr><td>29</td><td>30</td><td>Nagaland</td><td>27,283</td><td>-</td><td>0.14%</td><td>4,144</td></tr><tr><td>30</td><td>31</td><td>Anarachal Pradesh</td><td>24,603</td><td>-</td><td>0.13%</td><td>3,737</td></tr><tr><td>31</td><td>32</td><td>Mizoram</td><td>22,287</td><td>18,787</td><td>0.12%</td><td>3,385</td></tr><tr><td>32</td><td>33</td><td>Andaman & Nicobar Islands</td><td>-</td><td>-</td><td>-</td><td>-</td></tr></table>	RANK	NAME_OF_STATES	GSDP(1)	GSDP(2)	SHARE(in %)	GDP(in billion)	0	1	Maharashtra	2,632,792	-	13.94%	399,021	1	2	Tamil Nadu	1,630,208	1,312,929	8.63%	247,629	2	3	Uttar Pradesh	1,584,764	1,166,817	8.39%	240,726	3	4	Gujarat	1,502,899	-	7.96%	228,290	4	5	Karnataka	1,493,127	1,156,039	7.91%	226,806	5	6	West Bengal	1,089,898	793,223	5.77%	165,556	6	7	Rajasthan	942,586	711,627	4.99%	143,179	7	8	Andhra Pradesh	862,957	672,018	4.57%	131,083	8	9	Telangana	861,031	663,258	4.56%	130,791	9	10	Madhya Pradesh	809,592	561,801	4.29%	122,977	10	11	Kerala	781,653	-	4.14%	118,733	11	12	Delhi	774,870	634,408	4.10%	117,703	12	13	Haryana	734,163	572,240	3.89%	111,519	13	14	Bihar	530,363	414,977	2.81%	80,562	14	15	Punjab	526,376	418,868	2.79%	79,957	15	16	Odisha	487,805	396,499	2.58%	74,098	16	17	Assam	315,881	-	1.67%	47,982	17	18	Chhattisgarh	304,063	243,477	1.61%	46,187	18	19	Jharkhand	297,204	240,036	1.57%	45,145	19	20	Uttarakhand	245,895	-	1.30%	37,351	20	21	Jammu & Kashmir	155,956	-	0.83%	23,690	21	22	Himachal Pradesh	153,845	124,403	0.81%	23,369	22	23	Goa	73,170	63,408	0.39%	11,115	23	24	Tripura	49,845	40,583	0.26%	7,571	24	25	Chandigarh	42,114	-	0.22%	6,397	25	26	Puducherry	34,433	25,083	0.18%	5,230	26	27	Meghalaya	33,481	26,695	0.18%	5,096	27	28	Sikkim	28,723	20,017	0.15%	4,363	28	29	Manipur	27,870	20,673	0.15%	4,233	29	30	Nagaland	27,283	-	0.14%	4,144	30	31	Anarachal Pradesh	24,603	-	0.13%	3,737	31	32	Mizoram	22,287	18,787	0.12%	3,385	32	33	Andaman & Nicobar Islands	-	-	-	-
RANK	NAME_OF_STATES	GSDP(1)	GSDP(2)	SHARE(in %)	GDP(in billion)																																																																																																																																																																																																																																									
0	1	Maharashtra	2,632,792	-	13.94%	399,021																																																																																																																																																																																																																																								
1	2	Tamil Nadu	1,630,208	1,312,929	8.63%	247,629																																																																																																																																																																																																																																								
2	3	Uttar Pradesh	1,584,764	1,166,817	8.39%	240,726																																																																																																																																																																																																																																								
3	4	Gujarat	1,502,899	-	7.96%	228,290																																																																																																																																																																																																																																								
4	5	Karnataka	1,493,127	1,156,039	7.91%	226,806																																																																																																																																																																																																																																								
5	6	West Bengal	1,089,898	793,223	5.77%	165,556																																																																																																																																																																																																																																								
6	7	Rajasthan	942,586	711,627	4.99%	143,179																																																																																																																																																																																																																																								
7	8	Andhra Pradesh	862,957	672,018	4.57%	131,083																																																																																																																																																																																																																																								
8	9	Telangana	861,031	663,258	4.56%	130,791																																																																																																																																																																																																																																								
9	10	Madhya Pradesh	809,592	561,801	4.29%	122,977																																																																																																																																																																																																																																								
10	11	Kerala	781,653	-	4.14%	118,733																																																																																																																																																																																																																																								
11	12	Delhi	774,870	634,408	4.10%	117,703																																																																																																																																																																																																																																								
12	13	Haryana	734,163	572,240	3.89%	111,519																																																																																																																																																																																																																																								
13	14	Bihar	530,363	414,977	2.81%	80,562																																																																																																																																																																																																																																								
14	15	Punjab	526,376	418,868	2.79%	79,957																																																																																																																																																																																																																																								
15	16	Odisha	487,805	396,499	2.58%	74,098																																																																																																																																																																																																																																								
16	17	Assam	315,881	-	1.67%	47,982																																																																																																																																																																																																																																								
17	18	Chhattisgarh	304,063	243,477	1.61%	46,187																																																																																																																																																																																																																																								
18	19	Jharkhand	297,204	240,036	1.57%	45,145																																																																																																																																																																																																																																								
19	20	Uttarakhand	245,895	-	1.30%	37,351																																																																																																																																																																																																																																								
20	21	Jammu & Kashmir	155,956	-	0.83%	23,690																																																																																																																																																																																																																																								
21	22	Himachal Pradesh	153,845	124,403	0.81%	23,369																																																																																																																																																																																																																																								
22	23	Goa	73,170	63,408	0.39%	11,115																																																																																																																																																																																																																																								
23	24	Tripura	49,845	40,583	0.26%	7,571																																																																																																																																																																																																																																								
24	25	Chandigarh	42,114	-	0.22%	6,397																																																																																																																																																																																																																																								
25	26	Puducherry	34,433	25,083	0.18%	5,230																																																																																																																																																																																																																																								
26	27	Meghalaya	33,481	26,695	0.18%	5,096																																																																																																																																																																																																																																								
27	28	Sikkim	28,723	20,017	0.15%	4,363																																																																																																																																																																																																																																								
28	29	Manipur	27,870	20,673	0.15%	4,233																																																																																																																																																																																																																																								
29	30	Nagaland	27,283	-	0.14%	4,144																																																																																																																																																																																																																																								
30	31	Anarachal Pradesh	24,603	-	0.13%	3,737																																																																																																																																																																																																																																								
31	32	Mizoram	22,287	18,787	0.12%	3,385																																																																																																																																																																																																																																								
32	33	Andaman & Nicobar Islands	-	-	-	-																																																																																																																																																																																																																																								
In []:	<pre>#6. Scrape the details of Highest selling novels. url = https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-greycompare #you have to find the following details: #A) Book name #B) Author name #C) Volumes sold #D) Publisher #E) Genre</pre>																																																																																																																																																																																																																																													
In [21]:	<pre>driver=webdriver.Chrome("chromedriver.exe") driver.get("https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-grey-compare")</pre>																																																																																																																																																																																																																																													
In [32]:	<pre>Title=[] title=driver.find_elements(By.XPATH, '//table[@class="in-article sortable"]//tbody/tr/td[2]') for i in title: Titles=i.text Title.append(titles)</pre>																																																																																																																																																																																																																																													
In [35]:	<pre>Writers=[] writers=driver.find_elements(By.XPATH, '//table[@class="in-article sortable"]//tbody/tr/td[3]') for i in writer: writers=i.text Writer.append(writers)</pre>																																																																																																																																																																																																																																													
In [42]:	<pre>Vol=[] Vols=driver.find_elements(By.XPATH, '//table[@class="in-article sortable"]//tbody/tr/td[4]') for i in Vols: Vol=i.text Vol.append(vol)</pre>																																																																																																																																																																																																																																													
In [45]:	<pre>Publisher=[] publisher=driver.find_elements(By.XPATH, '//table[@class="in-article sortable"]//tbody/tr/td[5]') for i in publisher: publi=i.text Publisher.append(pub)</pre>																																																																																																																																																																																																																																													
In [49]:	<pre>Genre=[] genre=driver.find_elements(By.XPATH, '//table[@class="in-article sortable"]//tbody/tr/td[6]') for i in genre: gens=i.text Genre.append(gens)</pre>																																																																																																																																																																																																																																													
In [52]:	<pre>print(len(Title),len(Writer),len(Vol),len(Publisher),len(Genre)) 100 100 100 100 100</pre>																																																																																																																																																																																																																																													
In [53]:	<pre>df=pd.DataFrame({"BOOK":Title,"AUTHOR":Writer,"VOLUMES":Vol,"PUBLISHER":Publisher,"GENRE":Genre}) df</pre>																																																																																																																																																																																																																																													
Out[53]:	<table><tr><th></th><th>BOOK</th><th>AUTHOR</th><th>VOLUMES</th><th>PUBLISHER</th><th>GENRE</th></tr><tr><td>0</td><td>Da Vinci Code:The</td><td>Brown, Dan</td><td>5,094,805</td><td>Transworld</td><td>Crime, Thriller & Adventure</td></tr><tr><td>1</td><td>Harry Potter and the Deathly Hallows</td><td>Rowling, J.K.</td><td>4,475,152</td><td>Bloomsbury</td><td>Children's Fiction</td></tr><tr><td>2</td><td>Harry Potter and the Philosopher's Stone</td><td>Rowling, J.K.</td><td>4,200,654</td><td>Bloomsbury</td><td>Children's Fiction</td></tr><tr><td>3</td><td>Harry Potter and the Order of the Phoenix</td><td>Rowling, J.K.</td><td>4,179,479</td><td>Bloomsbury</td><td>Children's Fiction</td></tr><tr><td>4</td><td>Fifty Shades of Grey</td><td>James, E.L.</td><td>3,758,936</td><td>Random House</td><td>Romance & Sagas</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>95</td><td>Ghost,The</td><td>Harris, Robert</td><td>807,311</td><td>Random House</td><td>General & Literary Fiction</td></tr><tr><td>96</td><td>Happy Days with the Naked Chef</td><td>Oliver, Jamie</td><td>794,201</td><td>Penguin</td><td>Food & Drink; General</td></tr><tr><td>97</td><td>Hunger Games,The-Hunger Games Trilogy</td><td>Collins, Suzanne</td><td>792,187</td><td>Scholastic Ltd.</td><td>Young Adult Fiction</td></tr><tr><td>98</td><td>Lost Boy,The-A Foster Child's Search for the L...</td><td>Pelzer, Dave</td><td>791,507</td><td>Orion</td><td>Biography; General</td></tr><tr><td>99</td><td>James's Ministry of Food Anyone Can Learn to C...</td><td>Oliver, Jamie</td><td>791,095</td><td>Penguin</td><td>Food & Drink; General</td></tr></table> <p>100 rows x 5 columns</p>		BOOK	AUTHOR	VOLUMES	PUBLISHER	GENRE	0	Da Vinci Code:The	Brown, Dan	5,094,805	Transworld	Crime, Thriller & Adventure	1	Harry Potter and the Deathly Hallows	Rowling, J.K.	4,475,152	Bloomsbury	Children's Fiction	2	Harry Potter and the Philosopher's Stone	Rowling, J.K.	4,200,654	Bloomsbury	Children's Fiction	3	Harry Potter and the Order of the Phoenix	Rowling, J.K.	4,179,479	Bloomsbury	Children's Fiction	4	Fifty Shades of Grey	James, E.L.	3,758,936	Random House	Romance & Sagas	95	Ghost,The	Harris, Robert	807,311	Random House	General & Literary Fiction	96	Happy Days with the Naked Chef	Oliver, Jamie	794,201	Penguin	Food & Drink; General	97	Hunger Games,The-Hunger Games Trilogy	Collins, Suzanne	792,187	Scholastic Ltd.	Young Adult Fiction	98	Lost Boy,The-A Foster Child's Search for the L...	Pelzer, Dave	791,507	Orion	Biography; General	99	James's Ministry of Food Anyone Can Learn to C...	Oliver, Jamie	791,095	Penguin	Food & Drink; General																																																																																																																																																																					
	BOOK	AUTHOR	VOLUMES	PUBLISHER	GENRE																																																																																																																																																																																																																																									
0	Da Vinci Code:The	Brown, Dan	5,094,805	Transworld	Crime, Thriller & Adventure																																																																																																																																																																																																																																									
1	Harry Potter and the Deathly Hallows	Rowling, J.K.	4,475,152	Bloomsbury	Children's Fiction																																																																																																																																																																																																																																									
2	Harry Potter and the Philosopher's Stone	Rowling, J.K.	4,200,654	Bloomsbury	Children's Fiction																																																																																																																																																																																																																																									
3	Harry Potter and the Order of the Phoenix	Rowling, J.K.	4,179,479	Bloomsbury	Children's Fiction																																																																																																																																																																																																																																									
4	Fifty Shades of Grey	James, E.L.	3,758,936	Random House	Romance & Sagas																																																																																																																																																																																																																																									
...																																																																																																																																																																																																																																									
95	Ghost,The	Harris, Robert	807,311	Random House	General & Literary Fiction																																																																																																																																																																																																																																									
96	Happy Days with the Naked Chef	Oliver, Jamie	794,201	Penguin	Food & Drink; General																																																																																																																																																																																																																																									
97	Hunger Games,The-Hunger Games Trilogy	Collins, Suzanne	792,187	Scholastic Ltd.	Young Adult Fiction																																																																																																																																																																																																																																									
98	Lost Boy,The-A Foster Child's Search for the L...	Pelzer, Dave	791,507	Orion	Biography; General																																																																																																																																																																																																																																									
99	James's Ministry of Food Anyone Can Learn to C...	Oliver, Jamie	791,095	Penguin	Food & Drink; General																																																																																																																																																																																																																																									
In []:	<pre>#7. Scrape the details most watched tv series of all time from imdb.com. url = https://www.imdb.com/list/ls095964455/ #you have to find the following details: #A) Name #B) Year span #C) Genre #D) Run time #E) Ratings #F) Votes</pre>																																																																																																																																																																																																																																													
In [54]:	<pre>driver=webdriver.Chrome("chromedriver") driver.get("https://www.imdb.com/list/ls095964455/")</pre>																																																																																																																																																																																																																																													
In [69]:	<pre>Name=[] name=driver.find_elements(By.XPATH, '//h3[@class="listser-item-header"]//a') for i in name: names=i.text.split('(')[0].strip() Name.append(names)</pre>																																																																																																																																																																																																																																													
In [80]:	<pre>Year=[] year=driver.find_elements(By.XPATH, '//span[@class="listser-item-year listser-item-year-unbold"]') for i in year: Year=i.text Year.append(years)</pre>																																																																																																																																																																																																																																													
In [83]:	<pre>Genre=[] genre=driver.find_elements(By.XPATH, '//span[@class="genre"]') for i in genre: g=i.text Genre.append(g)</pre>																																																																																																																																																																																																																																													
In [86]:	<pre>Time=[] times=driver.find_elements(By.XPATH, '//span[@class="runtime"]') for i in times: t=i.text time.append(t)</pre>																																																																																																																																																																																																																																													
In [93]:	<pre>rating=[] ratings=driver.find_elements(By.XPATH, '//div[@class="ipl-rating-star small"]') for i in ratings: r=i.text rating.append(r)</pre>																																																																																																																																																																																																																																													
In [98]:	<pre>voting=[] votes=driver.find_elements(By.NAME, "nv") for i in voting: v=i.text voting.append(v)</pre>																																																																																																																																																																																																																																													
In [103]:	<pre>print(len(Name),len(Year),len(Genre),len(Time),len(rating),len(voting)) 100 100 100 100 100 100</pre>																																																																																																																																																																																																																																													
In [104]:	<pre>df=pd.DataFrame({"NAME":Name,"YEAR_SPAN":Year,"GENRE":Genre,"RUN_TIME":Time,"RATINGS":rating,"VOTES":voting}) df</pre>																																																																																																																																																																																																																																													
Out[104]:	<table><tr><th></th><th>NAME</th><th>YEAR_SPAN</th><th>GENRE</th><th>RUN_TIME</th><th>RATINGS</th><th>VOTES</th></tr><tr><td>0</td><td>Game of Thrones</td><td>(2011–2019)</td><td>Action, Adventure, Drama</td><td>57 min</td><td>9.2</td><td>2,137,870</td></tr><tr><td>1</td><td>Stranger Things</td><td>(2016–2024)</td><td>Drama, Fantasy, Horror</td><td>51 min</td><td>8.7</td><td>1,222,829</td></tr><tr><td>2</td><td>The Walking Dead</td><td>(2010–2022)</td><td>Drama, Horror, Thriller</td><td>44 min</td><td>8.1</td><td>1,014,905</td></tr><tr><td>3</td><td>13 Reasons Why</td><td>(2017–2020)</td><td>Drama, Mystery, Thriller</td><td>60 min</td><td>7.5</td><td>296,960</td></tr><tr><td>4</td><td>The 100</td><td>(2014–2020)</td><td>Drama, Mystery, Sci-Fi</td><td>43 min</td><td>7.6</td><td>257,836</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>95</td><td>Influence</td><td>(2013–2017)</td><td>Documentary</td><td>40 min</td><td>7.4</td><td>51,005</td></tr><tr><td>96</td><td>A Series of Unfortunate Events</td><td>(2017–2019)</td><td>Adventure, Comedy, Drama</td><td>52 min</td><td>7.8</td><td>63,027</td></tr><tr><td>97</td><td>Criminal Minds</td><td>(2005–)</td><td>Crime, Drama, Mystery</td><td>42 min</td><td>8.1</td><td>204,883</td></tr><tr><td>98</td><td>Scream: The TV Series</td><td>(2015–2019)</td><td>Comedy, Crime, Drama</td><td>45 min</td><td>7.1</td><td>42,437</td></tr><tr><td>99</td><td>The Haunting of Hill House</td><td>(2018)</td><td>Drama, Horror, Mystery</td><td>572 min</td><td>8.6</td><td>253,524</td></tr></table> <p>100 rows x 7 columns</p>		NAME	YEAR_SPAN	GENRE	RUN_TIME	RATINGS	VOTES	0	Game of Thrones	(2011–2019)	Action, Adventure, Drama	57 min	9.2	2,137,870	1	Stranger Things	(2016–2024)	Drama, Fantasy, Horror	51 min	8.7	1,222,829	2	The Walking Dead	(2010–2022)	Drama, Horror, Thriller	44 min	8.1	1,014,905	3	13 Reasons Why	(2017–2020)	Drama, Mystery, Thriller	60 min	7.5	296,960	4	The 100	(2014–2020)	Drama, Mystery, Sci-Fi	43 min	7.6	257,836	95	Influence	(2013–2017)	Documentary	40 min	7.4	51,005	96	A Series of Unfortunate Events	(2017–2019)	Adventure, Comedy, Drama	52 min	7.8	63,027	97	Criminal Minds	(2005–)	Crime, Drama, Mystery	42 min	8.1	204,883	98	Scream: The TV Series	(2015–2019)	Comedy, Crime, Drama	45 min	7.1	42,437	99	The Haunting of Hill House	(2018)	Drama, Horror, Mystery	572 min	8.6	253,524																																																																																																																																																									
	NAME	YEAR_SPAN	GENRE	RUN_TIME	RATINGS	VOTES																																																																																																																																																																																																																																								
0	Game of Thrones	(2011–2019)	Action, Adventure, Drama	57 min	9.2	2,137,870																																																																																																																																																																																																																																								
1	Stranger Things	(2016–2024)	Drama, Fantasy, Horror	51 min	8.7	1,222,829																																																																																																																																																																																																																																								
2	The Walking Dead	(2010–2022)	Drama, Horror, Thriller	44 min	8.1	1,014,905																																																																																																																																																																																																																																								
3	13 Reasons Why	(2017–2020)	Drama, Mystery, Thriller	60 min	7.5	296,960																																																																																																																																																																																																																																								
4	The 100	(2014–2020)	Drama, Mystery, Sci-Fi	43 min	7.6	257,836																																																																																																																																																																																																																																								
...																																																																																																																																																																																																																																								
95	Influence	(2013–2017)	Documentary	40 min	7.4	51,005																																																																																																																																																																																																																																								
96	A Series of Unfortunate Events	(2017–2019)	Adventure, Comedy, Drama	52 min	7.8	63,027																																																																																																																																																																																																																																								
97	Criminal Minds	(2005–)	Crime, Drama, Mystery	42 min	8.1	204,883																																																																																																																																																																																																																																								
98	Scream: The TV Series	(2015–2019)	Comedy, Crime, Drama	45 min	7.1	42,437																																																																																																																																																																																																																																								
99	The Haunting of Hill House	(2018)	Drama, Horror, Mystery	572 min	8.6	253,524																																																																																																																																																																																																																																								
In []:	<pre>#8. Details of Datasets from UCI machine learning repositories. url = https://archive.ics.uci.edu/ #you have to find the following details: #A) Dataset name #B) Data type #C) Task #D) Attribute type #E) No of instances #F) No of attribute #G) Year #Note: - from the home page you have to go to the ShowAllDataset page through code.</pre>																																																																																																																																																																																																																																													
In [4]:	<pre>driver=webdriver.Chrome("chromedriver.exe") driver.get("https://archive.ics.uci.edu/")</pre>																																																																																																																																																																																																																																													
In [5]:	<pre>link=driver.find_elements(By.XPATH, '//html/body/table[2]//tbody/tr/td/span/a/1')</pre>																																																																																																																																																																																																																																													
In [43]:	<pre>names=driver.find_elements(By.XPATH, '//table[@border="1"]//tbody/tr/td[1]') for i in names[:1623]: name=i.text names.append(n)</pre>																																																																																																																																																																																																																																													
In [37]:	<pre>types=driver.find_elements(By.XPATH, '//table[@border="1"]//tbody/tr/td[2]') for i in types[:1623]: t=i.text types.append(t)</pre>																																																																																																																																																																																																																																													
In [49]:	<pre>task=[] tasks=driver.find_elements(By.XPATH, '//table[@border="1"]//tbody/tr/td[3]') for i in tasks[:1623]: t=i.text task.append(tas)</pre>																																																																																																																																																																																																																																													
In [56]:	<pre>a_types=driver.find_elements(By.XPATH, '//table[@border="1"]//tbody/tr/td[4]') for i in a_types[:1623]: a=i.text a_type.append(a)</pre>																																																																																																																																																																																																																																													
In [61]:	<pre>instances=[] inst=driver.find_elements(By.XPATH, '//table[@border="1"]//tbody/tr/td[5]') for i in inst[:1623]: i=i.text instance.append(ins)</pre>																																																																																																																																																																																																																																													
In [64]:	<pre>attri=driver.find_elements(By.XPATH, '//table[@border="1"]//tbody/tr/td[6]') for i in attri[:1623]: a=i.text attribute.append(attr)</pre>																																																																																																																																																																																																																																													
In [70]:	<pre>years=driver.find_elements(By.XPATH, '//table[@border="1"]//tbody/tr/td[7]') for i in years[:1623]: y=i.text year.append(y)</pre>																																																																																																																																																																																																																																													
In [73]:	<pre>print(len(name),len(types),len(task),len(a_type),len(instance),len(attribute),len(year)) 622 622 622 622 622 622 622</pre>																																																																																																																																																																																																																																													
In [4]:	<pre>df=pd.DataFrame({"DATASET_NAME":name,"DATASET_TYPE":types,"TASK":task,"ATTRIBUTE_TYPE":a_type,"INSTANCES":instance,"ATTRIBUTES":attribute,"YEAR":year}) df</pre>																																																																																																																																																																																																																																													
Out[74]:	<table><tr><th></th><th>DATASET_NAME</th><th>DATASET_TYPE</th><th>TASK</th><th>ATTRIBUTE_TYPE</th><th>INSTANCES</th><th>ATTRIBUTES</th><th>YEAR</th></tr><tr><td>0</td><td>Abalone</td><td>Multivariate</td><td>Classification</td><td>Categorical, Integer, Real</td><td>4177</td><td>8</td><td>1995</td></tr><tr><td>1</td><td>Adult</td><td>Multivariate</td><td>Classification</td><td>Categorical, Integer</td><td>48842</td><td>14</td><td>1998</td></tr><tr><td>2</td><td>Annexing</td><td>Multivariate</td><td>Classification</td><td>Categorical, Integer, Real</td><td>786</td><td>38</td><td>1998</td></tr><tr><td>3</td><td>Anonymous Microsoft Web Data</td><td>Multivariate</td><td>Recommendation-Systems</td><td>Categorical</td><td>37711</td><td>294</td><td>1998</td></tr><tr><td>4</td><td>Arrhythmia</td><td>Multivariate</td><td>Classification</td><td>Categorical, Integer, Real</td><td>452</td><td>279</td><td>1998</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>617</td><td>Influenza outbreak event prediction via Twi...</td><td>Multivariate</td><td>Classification</td><td>Integer, Real</td><td>75840</td><td>525</td><td>2020</td></tr><tr><td>618</td><td>Indian Music Emotion Dataset</td><td>Multivariate</td><td>Classification</td><td>Integer, Real</td><td>400</td><td>50</td><td>2020</td></tr><tr><td>619</td><td>Maternal Health Risk Data Set</td><td>Multivariate</td><td>Classification</td><td>Integer, Real</td><td>1014</td><td>7</td><td>2020</td></tr><tr><td>620</td><td>Room Occupancy Estimation</td><td>Multivariate, Time-Series</td><td>Classification</td><td>Real</td><td>10129</td><td>18</td><td>2021</td></tr><tr><td>621</td><td>Image Recognition Task Execution Times in Mo...</td><td>Univariate</td><td>Regression</td><td>Real</td><td>4000</td><td>2</td><td>2021</td></tr></table> <p>622 rows x 7 columns</p>		DATASET_NAME	DATASET_TYPE	TASK	ATTRIBUTE_TYPE	INSTANCES	ATTRIBUTES	YEAR	0	Abalone	Multivariate	Classification	Categorical, Integer, Real	4177	8	1995	1	Adult	Multivariate	Classification	Categorical, Integer	48842	14	1998	2	Annexing	Multivariate	Classification	Categorical, Integer, Real	786	38	1998	3	Anonymous Microsoft Web Data	Multivariate	Recommendation-Systems	Categorical	37711	294	1998	4	Arrhythmia	Multivariate	Classification	Categorical, Integer, Real	452	279	1998	617	Influenza outbreak event prediction via Twi...	Multivariate	Classification	Integer, Real	75840	525	2020	618	Indian Music Emotion Dataset	Multivariate	Classification	Integer, Real	400	50	2020	619	Maternal Health Risk Data Set	Multivariate	Classification	Integer, Real	1014	7	2020	620	Room Occupancy Estimation	Multivariate, Time-Series	Classification	Real	10129	18	2021	621	Image Recognition Task Execution Times in Mo...	Univariate	Regression	Real	4000	2	2021																																																																																																																																													
	DATASET_NAME	DATASET_TYPE	TASK	ATTRIBUTE_TYPE	INSTANCES	ATTRIBUTES	YEAR																																																																																																																																																																																																																																							
0	Abalone	Multivariate	Classification	Categorical, Integer, Real	4177	8	1995																																																																																																																																																																																																																																							
1	Adult	Multivariate	Classification	Categorical, Integer	48842	14	1998																																																																																																																																																																																																																																							
2	Annexing	Multivariate	Classification	Categorical, Integer, Real	786	38	1998																																																																																																																																																																																																																																							
3	Anonymous Microsoft Web Data	Multivariate	Recommendation-Systems	Categorical	37711	294	1998																																																																																																																																																																																																																																							
4	Arrhythmia	Multivariate	Classification	Categorical, Integer, Real	452	279	1998																																																																																																																																																																																																																																							
...																																																																																																																																																																																																																																							
617	Influenza outbreak event prediction via Twi...	Multivariate	Classification	Integer, Real	75840	525	2020																																																																																																																																																																																																																																							
618	Indian Music Emotion Dataset	Multivariate	Classification	Integer, Real	400	50	2020																																																																																																																																																																																																																																							
619	Maternal Health Risk Data Set	Multivariate	Classification	Integer, Real	1014	7	2020																																																																																																																																																																																																																																							
620	Room Occupancy Estimation	Multivariate, Time-Series	Classification	Real	10129	18	2021																																																																																																																																																																																																																																							
621	Image Recognition Task Execution Times in Mo...	Univariate	Regression	Real	4000	2	2021																																																																																																																																																																																																																																							
In []:	<pre>#9. Scrape the details of Data science recruiters url = https://www.naukri.com/hr-recruiters-consultants #you have to find the following details: #A) Location #B) Designation #C) Company #D)Skills they hire for #E) Location #Note: - From naukri.com homepage click on the recruiters option and the on the search pane type data science and click on search. All this should be done through code.</pre>																																																																																																																																																																																																																																													
In [5]:	<pre>driver=webdriver.Chrome("chromedriver.exe") driver.get("https://www.naukri.com/hr-recruiters-consultants")</pre>																																																																																																																																																																																																																																													
In [40]:	<pre>names=driver.find_elements(By.XPATH, '//span[@class="f1 ellipsis"]') for i in names[:049]: n=i.text name.append(n)</pre>																																																																																																																																																																																																																																													
In [41]:	<pre>roles=driver.find_elements(By.XPATH, '//div[@class="hireSec highlightable"]') for</pre>																																																																																																																																																																																																																																													