In [173]:
```
import selenium
import pandas as pd
from selenium import webdriver
import warnings
warnings.filterwarnings('ignore')
from selenium.webdriver.common.by import By
import time
```

In [2]:
```
#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
driver=webdriver.Chrome(r"chromedriver.exe")
driver.get("https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos")
time.sleep(5)
```

In [3]:
```
rank=[]
name=[]
artist=[]
dates=[]
views=[]
```

In [5]:
```
rank_tags=driver.find_elements(By.XPATH,'//td[@align="leftt"]')
for i in rank_tags:
    r=i.text
    rank.append(r)

name_tags=driver.find_elements(By.XPATH,'//a[@class="mw-redirect"]')
for i in name_tags:
    n=i.text
    name.append(n)

artist_tags=driver.find_elements(By.XPATH,'//a[@title="Pinkfong"]')
for i in artist_tags:
    art=i.text
    artist.append(art)

date_tags=driver.find_elements(By.XPATH,'//td[@align="right"]')
for i in date_tags:
    date=i.text
    dates.append(date)

views_tags=driver.find_elements(By.XPATH,'//td[@align="center"]')
for i in views_tags:
    view=i.text
    views.append(view)
```

In [6]:
```
print(len(rank), len(name), len(artist), len(views))
```

0 19 4 129

In [7]:
```
#2. Scrape the details team India's international fixtures from bcci.tv. Url = https://www.bcci.tv/.You need to find following
#details: A) Match title (I.E. 1st ODI) B) Series C) Place D) Date E) Time
#Note: - From bcci.tv home page you have reach to the international fixture page through code
```

In [9]:
```
driver=webdriver.Chrome(r"chromedriver.exe")
driver.get("https://www.bcci.tv/")
time.sleep(5)
```

In [10]:
```
element = driver.find_element(By.CLASS_NAME,'nav-link ')
element.click()
```

In [11]:
```
title=[]
series=[]
place=[]
date=[]
time=[]
```

In [12]:
```
element=driver.find_element(By.XPATH,'//button[@class="match-btn btn-red d-flex align-items-center justify-content-center mx-auto mt-3"]')
element.click()
```

In [15]:
```
title_tags=driver.find_elements(By.XPATH,'//span[@class="matchOrderText ng-binding ng-scope"]')
for i in title_tags:
    titl=i.text
    title.append(titl)

series_tags=driver.find_elements(By.XPATH,'//span[@class="ng-binding"]')
for i in series_tags:
    ser=i.text
    series.append(ser)

place_tags=driver.find_elements(By.XPATH,'//span[@class="ng-binding ng-scope"]')
for i in place_tags:
    pla=i.text
    place.append(pla)

date_tags=driver.find_elements(By.XPATH,'//div[@class="match-card-left match-schedule"]')
for i in date_tags:
    dat=i.text
    date.append(dat)

time_tags=driver.find_elements(By.XPATH,'//div[@class="match-card-right match-schedule "]')
for i in time_tags:
    tim=i.text
    time.append(tim)
```

In [17]:
```
print(len(title), len(series), len(place), len(date), len(time))
```

13 13 13 13 13

In [21]:
```
df=pd.DataFrame({'Title': title, 'Series': series, 'Place': place, 'Date': date, 'Time': time})
df
```

Out[21]:

	Title	Series	Place	Date	Time
0	3rd T20I -	INDIA WOMEN TOUR OF ENGLAND T20 SERIES 2022	County Ground,	15 SEP 2022	11:00 PM IST
1	1st ODI -	INDIA WOMEN TOUR OF ENGLAND ODI SERIES 2022	County Ground,	18 SEP 2022	3:30 PM IST
2	1st T20I -	AUSTRALIA TOUR OF INDIA T20 SERIES 2022	Punjab Cricket Association IS Bindra Stadium,	20 SEP 2022	7:30 PM IST
3	2nd ODI -	INDIA WOMEN TOUR OF ENGLAND ODI SERIES 2022	St Lawrence Ground,	21 SEP 2022	5:30 PM IST
4	2nd T20I -	AUSTRALIA TOUR OF INDIA T20 SERIES 2022	Vidarbha Cricket Association Stadium,	23 SEP 2022	7:30 PM IST
5	3rd ODI -	INDIA WOMEN TOUR OF ENGLAND ODI SERIES 2022	Lord's Cricket Ground,	24 SEP 2022	3:30 PM IST
6	3rd T20I -	AUSTRALIA TOUR OF INDIA T20 SERIES 2022	Rajiv Gandhi International Stadium,	25 SEP 2022	7:30 PM IST
7	1st T20I -	SOUTH AFRICA TOUR OF INDIA T20 SERIES 2022-23	Greenfield International Stadium,	28 SEP 2022	7:30 PM IST
8	2nd T20I -	SOUTH AFRICA TOUR OF INDIA T20 SERIES 2022-23	Barsapara Cricket Stadium,	2 OCT 2022	7:30 PM IST
9	3rd T20I -	SOUTH AFRICA TOUR OF INDIA T20 SERIES 2022-23	Holkar Cricket Stadium,	4 OCT 2022	7:30 PM IST
10	1st ODI -	SOUTH AFRICA TOUR OF INDIA ODI SERIES 2022-23	Bharat Ratna Shri Atal Bihari Vajpayee Ekana C...	6 OCT 2022	1:30 PM IST
11	2nd ODI -	SOUTH AFRICA TOUR OF INDIA ODI SERIES 2022-23	JSCA International Stadium Complex,	9 OCT 2022	1:30 PM IST
12	3rd ODI -	SOUTH AFRICA TOUR OF INDIA ODI SERIES 2022-23	Arun Jaitley Stadium,	11 OCT 2022	1:30 PM IST

In [67]:
```
#3. Scrape the details of selenium exception from guru99.com. Url = https://www.guru99.com/ You need to find following details:
#A) Name B) Description Note: - From guru99 home page you have to reach to selenium exception handling page through code.
```

In [ ]:
```

```

In [66]:
```
#4. Scrape the details of State-wise GDP of India from statisticstime.com. Url = http://statisticstimes.com/
#You have to find following details: A) Rank B) State C) GSDP(18-19) D) GSDP(17-18) E) Share(2017) F) GDP($ billion)
#Note: - From statisticstimes home page you have to reach to economy page through code
```

In [61]:
```
driver=webdriver.Chrome(r"chromedriver.exe")
driver.get("https://www.statisticstimes.com/economy/india/indian-states-gdp.php")
time.sleep(5)
```

In [62]:
```
rank=[]
state=[]
gsdp1=[]
gsdp2=[]
share=[]
gdp=[]
```

In [63]:
```
rank_tags=driver.find_elements(By.CLASS_NAME,'data1')
for i in rank_tags[:34]:
    ran=i.text
    rank.append(ran)

state_tags=driver.find_elements(By.XPATH,'//td[@class="name"]')
for i in state_tags[:34]:
    st=i.text
    state.append(st)

gsdp1_tags=driver.find_elements(By.XPATH,'//td[@class="data"][1]')
for i in gsdp1_tags[:34]:
    gsd1=i.text
    gsdp1.append(gsd1)

gsdp2_tags=driver.find_elements(By.XPATH,'//td[@class="data"][2]')
for i in gsdp2_tags[:34]:
    gsd2=i.text
    gsdp2.append(gsd2)

share_tags=driver.find_elements(By.XPATH,'//td[@class="data"][3]')
for i in share_tags[:34]:
    sh=i.text
    share.append(sh)

gdp_tags=driver.find_elements(By.XPATH,'//td[@class="data sorting_1"]')
for i in gdp_tags[:34]:
    gd=i.text
    gdp.append(gd)
```

In [64]:
```
print(len(rank), len(state), len(gsdp1), len(gsdp2), len(share), len(gdp))
```

34 34 34 34 34 34

In [65]:
```
df=pd.DataFrame({'Rank':rank, 'State': state, 'GSDP(19-20)':gsdp1, 'GSDP(18-19)':gsdp2, 'Share':share, 'GDP(18-19)':gdp})
df
```

Out[65]:

	Rank	State	GSDP(19-20)	GSDP(18-19)	Share	GDP(18-19)
0	1	Maharashtra	-	13.94%	399.921	2,632,792
1	2	Tamil Nadu	1,845,853	8.63%	247.629	1,630,208
2	3	Uttar Pradesh	1,687,818	8.39%	240.726	1,584,764
3	4	Gujarat	-	7.96%	228.290	1,502,899
4	5	Karnataka	1,631,977	7.91%	226.806	1,493,127
5	6	West Bengal	1,253,832	5.77%	165.556	1,089,898
6	7	Rajasthan	1,020,989	4.99%	143.179	942,586
7	8	Andhra Pradesh	972,782	4.57%	131.083	862,957
8	9	Telangana	969,604	4.56%	130.791	861,031
9	10	Madhya Pradesh	906,672	4.29%	122.977	809,592
10	11	Kerala	-	4.14%	118.733	781,653
11	12	Delhi	856,112	4.10%	117.703	774,870
12	13	Haryana	831,610	3.89%	111.519	734,163
13	14	Bihar	611,804	2.81%	80.562	530,363
14	15	Punjab	574,760	2.79%	79.957	526,376
15	16	Odisha	521,275	2.58%	74.098	487,805
16	17	Assam	-	1.67%	47.982	315,881
17	18	Chhattisgarh	329,180	1.61%	46.187	304,063
18	19	Jharkhand	328,598	1.57%	45.145	297,204
19	20	Uttarakhand	-	1.30%	37.351	245,895
20	21	Jammu & Kashmir	-	0.83%	23.690	155,956
21	22	Himachal Pradesh	165,472	0.81%	23.369	153,845
22	23	Goa	80,449	0.39%	11.115	73,170
23	24	Tripura	55,984	0.26%	7.571	49,845
24	25	Chandigarh	-	0.22%	6.397	42,114
25	26	Puducherry	38,253	0.18%	5.230	34,433
26	27	Meghalaya	36,572	0.18%	5.086	33,481
27	28	Sikkim	32,496	0.15%	4.363	28,723
28	29	Manipur	31,790	0.15%	4.233	27,870
29	30	Nagaland	-	0.14%	4.144	27,283
30	31	Arunachal Pradesh	-	0.13%	3.737	24,603
31	32	Mizoram	26,503	0.12%	3.385	22,287
32	33	Andaman & Nicobar Islands	-	-	-	-
33		India	20,351,013	18,886,957		2,332,992

In [68]:
```
#5. Scrape the details of trending repositories on Github.com. Url = https://github.com/ You have to find the following details:
#A) Repository title B) Repository description C) Contributors count D) Language used
#Note: - From the home page you have to click on the trending option from Explore menu through code
```

In [88]:
```
driver=webdriver.Chrome(r"chromedriver.exe")
driver.get("https://github.com/")
time.sleep(5)
```

In [89]:
```
element=driver.find_element(By.XPATH,'//li[@class="HeaderMenu-item position-relative flex-wrap flex-justify-between flex-items-center d-block d-lg-flex flex-lg-nowrap flex-lg-items
element.click()
time.sleep(5)
```

In [117]:
```
driver.get("https://github.com/trending")
```

In [118]:
```
title=[]
description=[]
count=[]
language=[]
```

In [119]:
```
title_tags=driver.find_elements(By.XPATH,'//span[@class="text-normal"]')
for i in title_tags[:25]:
    ti=i.text
    title.append(ti)

description_tags=driver.find_elements(By.XPATH,'//p[@class="col-9 color-fg-muted my-1 pr-4"]')
for i in description_tags[:25]:
    des=i.text
    description.append(des)

count_tags=driver.find_elements(By.XPATH,'//a[@class="Link--muted d-inline-block mr-3"][2]')
for i in count_tags[:25]:
    cnt=i.text
    count.append(cnt)

language_tags=driver.find_elements(By.XPATH,'//span[@class="d-inline-block ml-0 mr-3"]')
for i in language_tags:
    lan=i.text
    language.append(lan)
```

In [120]:
```
print(len(title), len(description), len(count), len(language))
```

25 25 25 20

In [121]:
```
df=pd.DataFrame({'Repository_Title': title, 'Repository_Description': description, 'Contributors_Count': count})
df
```

Out[121]:

	Repository_Title	Repository_Description	Contributors_Count
0	danielgindi /	Beautiful charts for iOS/tvOS/OSX! The Apple s...	5,481
1	surrealdb /	A scalable, distributed, collaborative, docume...	132
2	TheAlgorithms /	All Algorithms implemented in Python	36,742
3	SerenityOS /	Ladybird web browser	58
4	divamgupta /	Diffusion Bee is the easiest way to run Stable...	62
5	karpathy /	Neural Networks: Zero to Hero	52
6	EbookFoundation /	📚 Freely available programming books	50,681
7	twitter /	Static checks to aid with a healthy adoption o...	17
8	nlln-io /	Free and source-available fair-code licensed w...	2,943
9	InterviewReady /	These are the best resources for System Design...	892
10	moby /	Moby Project - a collaborative project for the...	18,279
11	Alamofire /	Elegant HTTP Networking in Swift	7,245
12	AykutSarac /	🔮 Seamlessly visualize your JSON data instantl...	537
13	pocketbase /	Open Source realtime backend in 1 file	382
14	ascoders /	前端精读周刊。帮你理解最前沿、实用的技术。	2,544
15	jellyfin /	The Free Software Media System	1,633
16	apptension /	An opinionated guide on how to become a profes...	412
17	vasanthk /	What happens behind the scenes when we type ww...	1,045
18	rust-lang /	Empowering everyone to build reliable and effi...	9,752
19	dotnet /	Home repository for .NET Core	4,608
20	microsoft /	Playwright is a framework for Web Testing and ...	2,001
21	gothinkster /	"The mother of all demo apps" — Exemplary full...	6,411
22	SerenityOS /	The Serenity Operating System 🐞	2,302
23	jwasham /	A complete computer science study plan to beco...	62,338
24	joeyballentine /	A flowchart/node-based image processing GUI ai...	40

In [137]:
```
#6. Scrape the details of top 100 songs on billiboard.com. Url = https://www.billboard.com/ You have to find the following
#details: A) Song name B) Artist name C) Last week rank D) Peak rank E) Weeks on board
#Note: - From the home page you have to click on the charts option then hot 100-page link through code.
```

In [96]:
```
driver=webdriver.Chrome(r"chromedriver.exe")
driver.get("https://www.billboard.com/")
time.sleep(5)
```

In [97]:
```
element=driver.find_element(By.XPATH,'//a[@class="c-link  lrv-a-unstyle-link lrv-a-unstyle-link lrv-u-color-brand-accent-blue:hover lrv-a-hover-effect lrv-u-whitespace-nowrap lrv-u
element.click()
```

In [98]:
```
element=driver.find_element(By.XPATH,'//a[@class="c-link  lrv-a-unstyle-link lrv-u-color-brand-primary:hover lrv-a-hover-effect lrv-u-whitespace-nowrap lrv-u-color-grey-dark"][1]')
element.click()
```

In [99]:
```
name=[]
artist_name=[]
rank=[]
peak_rank=[]
week=[]
```

In [100]:
```
song_tags=driver.find_elements(By.XPATH,'//h3[@class="c-title  a-no-trucate a-font-primary-bold-s u-letter-spacing-0021 u-font-size-23@tablet lrv-u-font-size-16 u-line-height-125 u
for i in song_tags:
    song=i.text
    name.append(song)

song_tags=driver.find_elements(By.XPATH,'//h3[@class="c-title  a-no-trucate a-font-primary-bold-s u-letter-spacing-0021 lrv-u-font-size-18@tablet lrv-u-font-size-16 lrv-u-line-height-1
for i in song_tags:
    song=i.text
    name.append(song)

artist_tags=driver.find_elements(By.XPATH,'//span[@class="c-label  a-no-trucate a-font-primary-s lrv-u-font-size-14@mobile-max u-line-height-normal@mobile-max u-letter-spacing-0021
for i in artist_tags:
    art=i.text
    artist_name.append(art)

artist_tags=driver.find_elements(By.XPATH,'//span[@class="c-label  a-no-trucate a-font-primary-s lrv-u-font-size-14@mobile-max u-line-height-normal@mobile-max u-letter-spacing-0021
for i in artist_tags:
    art=i.text
    artist_name.append(art)

rank_tags=driver.find_elements(By.XPATH,'//li[@class="o-chart-results-list__item // a-chart-color u-width-72 u-width-55@mobile-max u-width-55@tablet-only lrv-u-flex lrv-u-flex-shrin
for i in rank_tags:
    ran=i.text
    rank.append(ran)

peak_tags=driver.find_elements(By.XPATH,'//li[@class="o-chart-results-list__item // a-chart-bg-color a-chart-color u-width-72 u-width-55@mobile-max u-width-55@tablet-only lrv-u-flex
for i in peak_tags:
    peak=i.text
    peak_rank.append(peak)

week_tags=driver.find_elements(By.XPATH,'//li[@class="o-chart-results-list__item // a-chart-color u-width-72 u-width-55@mobile-max u-width-55@tablet-only lrv-u-flex lrv-u-flex-shrin
for i in week_tags:
    we=i.text
    week.append(we)
```

In [101]:
```
print(len(name), len(artist_name), len(rank), len(peak_rank), len(week))
```

100 100 100 100 100

In [103]:
```
df=pd.DataFrame({'Song_Name':name, 'Artist_Name': artist_name, 'Last_Week_Rank': rank, 'Peak_Rank': peak_rank, 'Weeks_On_Board':week})
df
```

Out[103]:

	Song_Name	Artist_Name	Last_Week_Rank	Peak_Rank	Weeks_On_Board
0	As It Was	Harry Styles	1	1	23
1	Bad Habit	Steve Lacy	2	2	10
2	Late Night Talking	Harry Styles	12	3	16
3	Sunroof	Nicky Youre & dazy	5	4	15
4	About Damn Time	Lizzo	3	1	21
...	...	...	...	...	...
95	Thought You Should Know	Morgan Wallen	96	12	18
96	Country On	Luke Bryan	-	76	3
97	Static	Steve Lacy	-	98	1
98	Billie Eilish.	Armani White	-	99	1
99	Sin Fin	Romeo Santos & Justin Timberlake	-	100	1

100 rows × 5 columns

In [52]:
```
#7. Scrape the details of Data science recruiters from naukri.com. Url = https://www.naukri.com/ You have to find the following details:
#A) Name 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
```

In [105]:
```
driver=webdriver.Chrome(r"chromedriver.exe")
driver.get("https://www.naukri.com/")
time.sleep(5)
```

In [ ]:
```
#8. Scrape the details of Highest selling novels. Url = https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-all-time-fifty-shades-grey- compare/
#You have to find the following details:
#A) Book name B) Author name C) Volumes sold D) Publisher E) Genre
```

In [106]:
```
driver=webdriver.Chrome(r"chromedriver.exe")
driver.get("https://www.theguardian.com/international")
time.sleep(5)
```

In [107]:
```
element=driver.find_element(By.XPATH,'//a[@class="pillar-link pillar-link--Culture pillar-link--current-section"]')
element.click()
```

In [122]:
```
search=driver.find_element(By.XPATH,'//a[@class="top-bar__item popup__toggle hide-until-desktop js-search-toggle js-toggle-ready"]')
search.click()
```

In [124]:
```
search=driver.find_element(By.XPATH,'//input[@class="gsc-input"]')
search.send_keys('Highest selling novels')
```

In [125]:
```
button=driver.find_element(By.XPATH,'//button[@class="gsc-search-button gsc-search-button-v2"]')
button.click()
```

In [126]:
```
search=driver.find_element(By.XPATH,'//div[@class="gs-title"][1]')
search.click()
```

In [127]:
```
book_name=[]
author_name=[]
volumes=[]
publisher=[]
genre=[]
```

In [128]:
```
book_tags=driver.find_elements(By.XPATH,'//td[@class="left"][2]')
for i in book_tags:
    name=i.text
    book_name.append(name)

author_tags=driver.find_elements(By.XPATH,'//td[@class="left"][3]')
for i in author_tags:
    auth=i.text
    author_name.append(auth)

volume_tags=driver.find_elements(By.XPATH,'//td[@class="left"][4]')
for i in volume_tags:
    vol=i.text
    volumes.append(vol)

publisher_tags=driver.find_elements(By.XPATH,'//td[@class="left"][5]')
for i in publisher_tags:
    publish=i.text
    publisher.append(publish)

genre_tags=driver.find_elements(By.XPATH,'//td[@class="last left"]')
for i in genre_tags:
    gen=i.text
    genre.append(gen)
```

In [129]:
```
print(len(book_name), len(author_name), len(volumes), len(publisher), len(genre))
```

100 100 100 100 100

In [130]:
```
df=pd.DataFrame({'Book_Name':book_name, 'Author_Name': author_name, 'VOlumes_Sold':volumes, 'Publisher': publisher, 'Genre':genre})
df
```

Out[130]:

	Book_Name	Author_Name	VOlumes_Sold	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	Jamie's Ministry of Food:Anyone Can Learn to C...	Oliver, Jamie	791,095	Penguin	Food & Drink: General

100 rows × 5 columns

In [131]:
```
#9. 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
```

In [132]:
```
driver=webdriver.Chrome(r"chromedriver.exe")
driver.get("https://www.imdb.com/list/ls095964455/")
time.sleep(5)
```

In [133]:
```
name=[]
year=[]
genre=[]
time=[]
ratings=[]
votes=[]
```

In [162]:
```
name_tags=driver.find_elements(By.XPATH,'//h3[@class="lister-item-header"]')
for i in name_tags:
    na=i.text
    name.append(na)

year_tags=driver.find_elements(By.XPATH,'//span[@class="lister-item-year text-muted unbold"]')
for i in year_tags:
    ye=i.text
    year.append(ye)

genre_tags=driver.find_elements(By.XPATH,'//span[@class="genre"]')
for i in genre_tags:
    gen=i.text
    genre.append(gen)

time_tags=driver.find_elements(By.XPATH,'//span[@class="runtime"]')
for i in time_tags:
    tm=i.text
    time.append(tm)

votes_tags=driver.find_elements(By.XPATH,'//p[@class="text-muted text-small"][3]')
for i in votes_tags:
    vote=i.text
    votes.append(vote)

rating_tags=driver.find_elements(By.XPATH,'//div[@class="ipl-rating-star small"]')
for i in rating_tags:
    rate=i.text
    ratings.append(rate)
```

In [163]:
```
print(len(name), len(year), len(genre), len(time), len(votes), len(ratings))
```

100 100 100 100 100 100

In [169]:
```
df=pd.DataFrame({'Name':name, 'Year_Span': year, 'Genre':genre, 'Run_Time': time, 'Votes': votes, 'Ratings':ratings})
df
```

Out[169]:

	Name	Year_Span	Genre	Run_Time	Votes	Ratings
0	1. Game of Thrones (2011–2019)	(2011–2019)	Action, Adventure, Drama	57 min	Votes: 2,049,856	9.2
1	2. Stranger Things (2016–)	(2016–)	Drama, Fantasy, Horror	51 min	Votes: 1,145,620	8.7
2	3. The Walking Dead (2010–2022)	(2010–2022)	Drama, Horror, Thriller	44 min	Votes: 968,416	8.1
3	4. 13 Reasons Why (2017–2020)	(2017–2020)	Drama, Mystery, Thriller	60 min	Votes: 288,863	7.5
4	5. The 100 (2014–2020)	(2014–2020)	Drama, Mystery, Sci-Fi	43 min	Votes: 248,439	7.6
...	...	...	...	...	...	...
95	96. Reign (2013–2017)	(2013–2017)	Drama	42 min	Votes: 49,564	7.4
96	97. A Series of Unfortunate Events (2017–2019)	(2017–2019)	Adventure, Comedy, Drama	50 min	Votes: 60,568	7.8
97	98. Criminal Minds (2005–2020)	(2005–2020)	Crime, Drama, Mystery	42 min	Votes: 194,798	8.1
98	99. Scream: The TV Series (2015–2019)	(2015–2019)	Comedy, Crime, Drama	45 min	Votes: 41,023	7.1
99	100. The Haunting of Hill House (2018)	(2018)	Drama, Horror, Mystery	572 min	Votes: 236,492	8.6

100 rows × 6 columns

In [171]:
```
#10. 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 instancesF) No of attribute G) Year
#Note: - From the home page you have to go to the ShowAllDataset page through code
```

In [174]:
```
driver=webdriver.Chrome(r"chromedriver.exe")
driver.get("https://archive.ics.uci.edu/ml/index.php")
time.sleep(5)
```

In [178]:
```
driver.get("https://archive-beta.ics.uci.edu/ml/datasets")
```

In [179]:
```
element=driver.find_element(By.XPATH,'//button[@class="MuiButton-root MuiButton-text MuiButton-textWarning MuiButton-sizeSmall MuiButton-textSizeSmall MuiButtonBase-root  css-165ej
element.click()
```

In [190]:
```
name=[]
typ=[]
task=[]
attribute_type=[]
instances=[]
number=[]
year=[]
```

In [191]:
```
name_tags=driver.find_elements(By.XPATH,'//a[@class="MuiTypography-root MuiTypography-inherit MuiLink-root MuiLink-underlineHover MuiTypography-root MuiTypography-h6 css-txu841"]')
for i in name_tags:
    na=i.text
    name.append(na)

typ_tags=driver.find_elements(By.XPATH,'//div[@class="MuiGrid-root MuiGrid-item MuiGrid-grid-xs-6 css-4xkoi8"][2]')
for i in typ_tags:
    ty=i.text
    typ.append(ty)

task_tags=driver.find_elements(By.XPATH,'//div[@class="MuiGrid-root MuiGrid-item MuiGrid-grid-xs-6 css-4xkoi8"][1]')
for i in task_tags:
    tsk=i.text
    task.append(tsk)

instances_tags=driver.find_elements(By.XPATH,'//div[@class="MuiGrid-root MuiGrid-item MuiGrid-grid-xs-6 css-4xkoi8"][3]')
for i in instances_tags:
    instant=i.text
    instances.append(instant)

number_tags=driver.find_elements(By.XPATH,'//div[@class="MuiGrid-root MuiGrid-item MuiGrid-grid-xs-6 css-4xkoi8"][4]')
for i in number_tags:
    num=i.text
    number.append(num)
```

In [192]:
```
print(len(name), len(typ), len(task), len(instances), len(number))
```

500 500 500 500 500

In [193]:
```
df=pd.DataFrame({'Dataset_Name': name, 'Data_Type': typ, 'Task':task, 'Instances': instances, 'Number_Of_Attribute':number})
df
```

Out[193]:

	Dataset_Name	Data_Type	Task	Instances	Number_Of_Attribute
0	Iris	Multivariate	Classification	150 instances	5 attributes
1	Dry Bean Dataset	Multivariate	Classification	13.61K instances	0 attributes
2	Diabetes	Multivariate, Time-Series	N/A	0 instances	0 attributes
3	Adult	Multivariate	Classification	48.84K instances	15 attributes
4	Heart Disease	Multivariate	Classification	303 instances	14 attributes
...	...	...	...	...	...
495	Divorce Predictors data set	Multivariate, Univariate	Classification	170 instances	0 attributes
496	CalIt2 Building People Counts	Multivariate, Time-Series	N/A	10.08K instances	4 attributes
497	EMG dataset in Lower Limb	Multivariate, Time-Series	N/A	132 instances	0 attributes
498	Indoor User Movement Prediction from RSS data	Multivariate, Sequential, Time-Series	Classification	13.2K instances	0 attributes
499	StoneFlakes	Multivariate	Classification, Clustering, Causal-Discovery	79 instances	8 attributes

500 rows × 5 columns

In [ ]:
```

```

In [ ]:
```

```