name = Divyanshu singh

batch = DS2306

id = 56

In [2]:
```
import selenium
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import NoSuchElementException , StaleElementReferenceE
from webdriver_manager.chrome import ChromeDriverManager
import pandas as pd
import time
import re

import warnings
warnings.filterwarnings('ignore')
```

.

1

In [17]:
```
driver = webdriver.Chrome()
driver.get("https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos")
```

In [18]:
```
Rank = []
Ra = driver.find_elements(By.XPATH,'//*[@id="mw-content-text"]/div[1]/table[1]/tbody/t
for i in Ra:
    Rank.append(i.text)
```

In [19]:
```
Rank
```

Out[19]:
```
['Baby Shark Dance']
```

In [ ]:
```
A) Rank
B) Name
C) Artist
D) Upload date
E) Views
```

In [ ]:
```
driver.close()
```

.

2

•

In [2]:
```
driver = webdriver.Chrome()
driver.get("https://www.bcci.tv")
```

In [7]:
```
view_all = driver.find_element(By.XPATH,'/html/body/div[5]/div/div[2]/div/div[2]/div/d
view_all.click()
```

In [53]:
```
Match_title = []
start = 0
end = 3
for page in range(start,end):
    Match = driver.find_elements(By.XPATH,'//span[@class="matchOrderText ng-binding ng
    for i in Match:
        Match_title.append(i.text)
    next_button = driver.find_element(By.XPATH,'/html/body/div[2]/div[2]/div/div/div/d
```

In [55]:
```
df = pd.DataFrame({'Match_title':Match_title})
df
```

Out[55]:

	Match_title
0	1st T20I -
1	2nd T20I -
2	3rd T20I -
3	1st ODI -
4	2nd ODI -
5	1st ODI -
6	2nd ODI -
7	3rd ODI -
8	1st T20I -
9	2nd T20I -
10	3rd T20I -
11	1st ODI -
12	2nd ODI -
13	1st ODI -
14	2nd ODI -
15	3rd ODI -
16	1st T20I -
17	2nd T20I -
18	3rd T20I -
19	1st ODI -
20	2nd ODI -
21	1st ODI -
22	2nd ODI -
23	3rd ODI -

In [60]:
```
Series = []
start = 0
end = 3
for page in range(start,end):
    Ser = driver.find_elements(By.XPATH,'//h5[@class="match-tournament-name ng-binding
    for i in Ser:
        Series.append(i.text)
    next_button = driver.find_element(By.XPATH,'/html/body/div[2]/div[2]/div/div/div/d
```

In [77]:
```
Place = []
start = 0
end = 3
for page in range(start,end):
    Pla = driver.find_elements(By.XPATH,'//span[@class="ng-binding"]')
    for i in Pla:
```

```
        Place.append(i.text)
    next_button = driver.find_element(By.XPATH,'/html/body/div[2]/div[2]/div/div/div/d
```

In [75]:
```
Date = []
start = 0
end = 3
for page in range(start,end):
    Da = driver.find_elements(By.XPATH,'//div[@class="match-dates ng-binding"]')
    for i in Da:
        Date.append(i.text)
    next_button = driver.find_element(By.XPATH,'/html/body/div[2]/div[2]/div/div/div/d
```

In [76]:
```
Time = []
start = 0
end = 3
for page in range(start,end):
    Ti = driver.find_elements(By.XPATH,'//div[@class="match-time no-margin ng-binding'
    for i in Ti:
        Time.append(i.text)
    next_button = driver.find_element(By.XPATH,'/html/body/div[2]/div[2]/div/div/div/d
```

In [ ]:
```
A) Match title (I.e. 1 ODI)
B) Series
C) Place
D) Date
E) Time
```

In [78]:
```
df = pd.DataFrame({'Match_title':Match_title,'Series':Series,'Place':Place,'Date':Date
df
```

Out[78]:

	Match_title	Series	Place	Date	Time
0	1st T20I -	INDIA TOUR OF IRELAND 2023	Dublin	18 AUG 2023	7:00 AM PDT
1	2nd T20I -	INDIA TOUR OF IRELAND 2023	Dublin	20 AUG 2023	7:00 AM PDT
2	3rd T20I -	INDIA TOUR OF IRELAND 2023	Dublin	23 AUG 2023	7:00 AM PDT
3	1st ODI -	ASIA CUP 2023	Pallekele	1 SEP 2023	9:30 PM PDT
4	2nd ODI -	ASIA CUP 2023	Pallekele	3 SEP 2023	9:30 PM PDT
5	1st ODI -	AUSTRALIA TOUR OF INDIA 2023-24	Mohali	22 SEP 2023	1:00 AM PDT
6	2nd ODI -	AUSTRALIA TOUR OF INDIA 2023-24	Indore	24 SEP 2023	1:00 AM PDT
7	3rd ODI -	AUSTRALIA TOUR OF INDIA 2023-24	Rajkot	27 SEP 2023	1:00 AM PDT
8	1st T20I -	INDIA TOUR OF IRELAND 2023	Dublin	18 AUG 2023	7:00 AM PDT
9	2nd T20I -	INDIA TOUR OF IRELAND 2023	Dublin	20 AUG 2023	7:00 AM PDT
10	3rd T20I -	INDIA TOUR OF IRELAND 2023	Dublin	23 AUG 2023	7:00 AM PDT
11	1st ODI -	ASIA CUP 2023	Pallekele	1 SEP 2023	9:30 PM PDT
12	2nd ODI -	ASIA CUP 2023	Pallekele	3 SEP 2023	9:30 PM PDT
13	1st ODI -	AUSTRALIA TOUR OF INDIA 2023-24	Mohali	22 SEP 2023	1:00 AM PDT
14	2nd ODI -	AUSTRALIA TOUR OF INDIA 2023-24	Indore	24 SEP 2023	1:00 AM PDT
15	3rd ODI -	AUSTRALIA TOUR OF INDIA 2023-24	Rajkot	27 SEP 2023	1:00 AM PDT
16	1st T20I -	INDIA TOUR OF IRELAND 2023	Dublin	18 AUG 2023	7:00 AM PDT
17	2nd T20I -	INDIA TOUR OF IRELAND 2023	Dublin	20 AUG 2023	7:00 AM PDT
18	3rd T20I -	INDIA TOUR OF IRELAND 2023	Dublin	23 AUG 2023	7:00 AM PDT
19	1st ODI -	ASIA CUP 2023	Pallekele	1 SEP 2023	9:30 PM PDT
20	2nd ODI -	ASIA CUP 2023	Pallekele	3 SEP 2023	9:30 PM PDT
21	1st ODI -	AUSTRALIA TOUR OF INDIA 2023-24	Mohali	22 SEP 2023	1:00 AM PDT
22	2nd ODI -	AUSTRALIA TOUR OF INDIA 2023-24	Indore	24 SEP 2023	1:00 AM PDT
23	3rd ODI -	AUSTRALIA TOUR OF INDIA 2023-24	Rajkot	27 SEP 2023	1:00 AM PDT

In [79]:
```
df.to_csv('team_India's_international_fixtures.csv')
```

In [80]:
```
driver.close()
```

.

3

.

In [2]:
```
driver = webdriver.Chrome()
driver.get('http://statisticstimes.com/')
```

In [3]:
```
economy = driver.find_element(By.XPATH,'/html/body/div[2]/div[1]/div[2]/div[2]/button'
economy.click()
india = driver.find_element(By.XPATH,'/html/body/div[2]/div[1]/div[2]/div[2]/div/a[3]'
india.click()
```

In [5]:
```
gdp_states = driver.find_element(By.XPATH,'/html/body/div[2]/div[2]/div[2]/ul/li[1]/a'
gdp_states.click()
```

In [6]:
```
Rank = []
Ra = driver.find_elements(By.XPATH,'//td[@class="data1"]')
for i in Ra:
    Rank.append(i.text)
```

In [71]:
```
Rank = Rank[:33]
Rank
```

Out[71]:
```
['1',
 '2',
 '3',
 '4',
 '5',
 '6',
 '7',
 '8',
 '9',
 '10',
 '11',
 '12',
 '13',
 '14',
 '15',
 '16',
 '17',
 '18',
 '19',
 '20',
 '21',
 '22',
 '23',
 '24',
 '25',
 '26',
 '27',
 '28',
 '29',
 '30',
 '31',
 '32',
 '33']
```

In [8]:
```
State = []
St = driver.find_elements(By.XPATH,'//td[@class="name"]')
for i in St:
    State.append(i.text)
```

In [72]:
```
State = State[:33]
State
```

Out[72]:
```
['Maharashtra',
 'Tamil Nadu',
 'Uttar Pradesh',
 'Gujarat',
 'Karnataka',
 'West Bengal',
 'Rajasthan',
 'Andhra Pradesh',
 'Telangana',
 'Madhya Pradesh',
 'Kerala',
 'Delhi',
 'Haryana',
 'Bihar',
 'Punjab',
 'Odisha',
 'Assam',
 'Chhattisgarh',
 'Jharkhand',
 'Uttarakhand',
 'Jammu & Kashmir',
 'Himachal Pradesh',
 'Goa',
 'Tripura',
 'Chandigarh',
 'Puducherry',
 'Meghalaya',
 'Sikkim',
 'Manipur',
 'Nagaland',
 'Arunachal Pradesh',
 'Mizoram',
 'Andaman & Nicobar Islands']
```

In [21]:
```
GSDP1819 = []
GS = driver.find_elements(By.XPATH,'//td[@class="data sorting_1"]')
for i in GS:
    GSDP1819.append(i.text)
```

In [73]:
```
GSDP1819 = GSDP1819[:33]
GSDP1819
```

Out[73]:
```
['2,632,792',
 '1,630,208',
 '1,584,764',
 '1,502,899',
 '1,493,127',
 '1,089,898',
 '942,586',
 '862,957',
 '861,031',
 '809,592',
 '781,653',
 '774,870',
 '734,163',
 '530,363',
 '526,376',
 '487,805',
 '315,881',
 '304,063',
 '297,204',
 '245,895',
 '155,956',
 '153,845',
 '73,170',
 '49,845',
 '42,114',
 '34,433',
 '33,481',
 '28,723',
 '27,870',
 '27,283',
 '24,603',
 '22,287',
 '-']
```

In [75]:
```
GSDP1920 = []
DP = driver.find_elements(By.XPATH,'//tr[@class="odd" or@class="even"]/td')
for i in DP:
    GSDP1920.append(i.text)
```

In [76]:
```
GSDP1920 = GSDP1920[:264]
GSDP1920 = GSDP1920[2::8]
GSDP1920
```

Out[76]:
```
['-',
 '1,845,853',
 '1,687,818',
 '-',
 '1,631,977',
 '1,253,832',
 '1,020,989',
 '972,782',
 '969,604',
 '906,672',
 '-',
 '856,112',
 '831,610',
 '611,804',
 '574,760',
 '521,275',
 '-',
 '329,180',
 '328,598',
 '-',
 '-',
 '165,472',
 '80,449',
 '55,984',
 '-',
 '38,253',
 '36,572',
 '32,496',
 '31,790',
 '-',
 '-',
 '26,503',
 '-']
```

In [78]:
```
share1819 = []
sh = driver.find_elements(By.XPATH,'//tr[@class="odd" or@class="even"]/td')
for i in sh:
    share1819.append(i.text)
```

In [79]:
```
share1819 = share1819[:264]
share1819 = share1819[4::8]
share1819
```

Out[79]:
```
['13.94%',
 '8.63%',
 '8.39%',
 '7.96%',
 '7.91%',
 '5.77%',
 '4.99%',
 '4.57%',
 '4.56%',
 '4.29%',
 '4.14%',
 '4.10%',
 '3.89%',
 '2.81%',
 '2.79%',
 '2.58%',
 '1.67%',
 '1.61%',
 '1.57%',
 '1.30%',
 '0.83%',
 '0.81%',
 '0.39%',
 '0.26%',
 '0.22%',
 '0.18%',
 '0.18%',
 '0.15%',
 '0.15%',
 '0.14%',
 '0.13%',
 '0.12%',
 '-']
```

In [80]:
```
GDP =[]
G = driver.find_elements(By.XPATH,'//tr[@class="odd" or@class="even"]/td')
for i in G:
    GDP.append(i.text)
```

In [81]:
```
GDP = GDP[:264]
GDP  = GDP[4::8]
GDP
```

Out[81]:
```
['13.94%',
 '8.63%',
 '8.39%',
 '7.96%',
 '7.91%',
 '5.77%',
 '4.99%',
 '4.57%',
 '4.56%',
 '4.29%',
 '4.14%',
 '4.10%',
 '3.89%',
 '2.81%',
 '2.79%',
 '2.58%',
 '1.67%',
 '1.61%',
 '1.57%',
 '1.30%',
 '0.83%',
 '0.81%',
 '0.39%',
 '0.26%',
 '0.22%',
 '0.18%',
 '0.18%',
 '0.15%',
 '0.15%',
 '0.14%',
 '0.13%',
 '0.12%',
 '-']
```

In [ ]:
```
Rank
B) State
C) GSDP(18-19)- at current prices
D) GSDP(19-20)- at current prices
E) Share(18-19)
F) GDP($ billion)
```

In [84]:
```
df3 = pd.DataFrame({'Rank':Rank,'State':State,'GSDP1819':GSDP1819,'GSDP1920':GSDP1920,
df3
```

Out[84]:

	Rank	State	GSDP1819	GSDP1920	Share1819	GDP
0	1	Maharashtra	2,632,792	-	13.94%	13.94%
1	2	Tamil Nadu	1,630,208	1,845,853	8.63%	8.63%
2	3	Uttar Pradesh	1,584,764	1,687,818	8.39%	8.39%
3	4	Gujarat	1,502,899	-	7.96%	7.96%
4	5	Karnataka	1,493,127	1,631,977	7.91%	7.91%
5	6	West Bengal	1,089,898	1,253,832	5.77%	5.77%
6	7	Rajasthan	942,586	1,020,989	4.99%	4.99%
7	8	Andhra Pradesh	862,957	972,782	4.57%	4.57%
8	9	Telangana	861,031	969,604	4.56%	4.56%
9	10	Madhya Pradesh	809,592	906,672	4.29%	4.29%
10	11	Kerala	781,653	-	4.14%	4.14%
11	12	Delhi	774,870	856,112	4.10%	4.10%
12	13	Haryana	734,163	831,610	3.89%	3.89%
13	14	Bihar	530,363	611,804	2.81%	2.81%
14	15	Punjab	526,376	574,760	2.79%	2.79%
15	16	Odisha	487,805	521,275	2.58%	2.58%
16	17	Assam	315,881	-	1.67%	1.67%
17	18	Chhattisgarh	304,063	329,180	1.61%	1.61%
18	19	Jharkhand	297,204	328,598	1.57%	1.57%
19	20	Uttarakhand	245,895	-	1.30%	1.30%
20	21	Jammu & Kashmir	155,956	-	0.83%	0.83%
21	22	Himachal Pradesh	153,845	165,472	0.81%	0.81%
22	23	Goa	73,170	80,449	0.39%	0.39%
23	24	Tripura	49,845	55,984	0.26%	0.26%
24	25	Chandigarh	42,114	-	0.22%	0.22%
25	26	Puducherry	34,433	38,253	0.18%	0.18%
26	27	Meghalaya	33,481	36,572	0.18%	0.18%
27	28	Sikkim	28,723	32,496	0.15%	0.15%
28	29	Manipur	27,870	31,790	0.15%	0.15%
29	30	Nagaland	27,283	-	0.14%	0.14%
30	31	Arunachal Pradesh	24,603	-	0.13%	0.13%
31	32	Mizoram	22,287	26,503	0.12%	0.12%
32	33	Andaman & Nicobar Islands	-	-	-	-

In [85]:
```
df3.to_csv('State-wise_GDP_of_India.csv')
```

In [107…
```
driver.close()
```

.

4

.

In [27]:
```
driver = webdriver.Chrome()
driver.get('https:/www.github.com')
```

In [34]:
```
option = driver.find_element(By.XPATH,'/html/body/div[1]/div[1]/header/div/div[1]/div[
option.click()
```

In [35]:
```
product = driver.find_element(By.XPATH,'/html/body/div[1]/div[1]/header/div/div[2]/div
product.click()
```

In [36]:
```
search = driver.find_element(By.XPATH,'/html/body/div[1]/div[1]/header/div/div[2]/div/
search.click()
```

In [37]:
```
Repository_title = []
Reposit = driver.find_elements(By.XPATH,'//span[@class="text-normal"]')
for i in Reposit:
    Repository_title.append(i.text)
```

In [46]:
```
opening_url = []
url = driver.find_elements(By.XPATH,'//a[@class="Link"]')
for i in url:
    opening_url.append(i.get_attribute('href'))
```

In [47]:
```
Repository_description = []
for p in opening_url:
    driver.get(p)
    time.sleep(5)
    try:
        Repo= driver.find_element(By.XPATH,'/html/body/div[1]/div[4]/div/main/turbo-fr
        Repository_description.append(Repo.text)
    except NoSuchElementException:
        Repository_description.append('_')
```

In [48]:
```
Repository_description
```

Out[48]:
```
['A MIT-licensed, deployable starter kit for building and customizing your own versio
n of AI town - a virtual town where AI characters live, chat and socialize.',
 "The OpenTF Manifesto expresses concern over HashiCorp's switch of the Terraform lic
ense from open-source to the Business Source License (BSL) and calls for the tool's r
eturn to a truly open-source license.",
 'Focus on prompting and generating',
 'An extensible, easy-to-use, and portable diffusion web UI 👩\u200d🎨',
 'FaceChain is a deep-learning toolchain for generating your Digital-Twin.',
 'A sample app for the Retrieval-Augmented Generation pattern running in Azure, using
Azure Cognitive Search for retrieval and Azure OpenAI large language models to power
ChatGPT-style and Q&A experiences.',
 'Langchain-Chatchat (formerly langchain-ChatGLM), local knowledge based LLM (like Ch
atGLM) QA app with langchain ｜ 基于 Langchain 与 ChatGLM 等语言模型的本地知识库问答',
 'Rift: an AI-native language server for your personal AI software engineer',
 'Official implementation of "Neuralangelo: High-Fidelity Neural Surface Reconstructi
on" (CVPR 2023)',
 '_',
 'A curated list of modern Generative Artificial Intelligence projects and services',
 '2023 HVV情报速递~',
 'AWS zero to hero repo for devops engineers to learn AWS in 30 Days. This repo inclu
des projects, presentations, interview questions and real time examples.',
 'Specify what you want it to build, the AI asks for clarification, and then builds i
t.',
 '</> htmx - high power tools for HTML',
 'The modern web developer’s platform',
 'Official Code for DragGAN (SIGGRAPH 2023)',
 'GoogleTest - Google Testing and Mocking Framework',
 'A well-designed cross-platform ChatGPT UI (Web / PWA / Linux / Win / MacOS). 一键拥
有你自己的跨平台 ChatGPT 应用。',
 'A list of Free Software network services and web applications which can be hosted o
n your own servers',
 'The React Framework',
 '🔥 🔥 🔥 现代化、开源的 Linux 服务器运维管理面板。',
 'GPT based autonomous agent that does online comprehensive research on any given top
ic',
 '.NET MAUI is the .NET Multi-platform App UI, a framework for building native device
applications spanning mobile, tablet, and desktop.',
 "A game where you are a computer's OS and you have to manage processes, memory and
I/O events."]
```

In [58]:
```
Language_used = []
Contributors_count = []
for p in opening_url:
    driver.get(p)
    time.sleep(5)

    try:
        Language= driver.find_element(By.XPATH,'//span[@class="color-fg-default text-b
        Language_used.append(Language.text)
    except NoSuchElementException:
        Language_used.append('_')

    try:
        Contributors= driver.find_element(By.XPATH,'/html/body/div[1]/div[4]/div/main/
        Contributors_count.append(Contributors.text)
    except NoSuchElementException:
        Contributors_count.append('_')
```

In [59]:
```
df4 = pd.DataFrame({'Repository_title':Repository_title,'Repository_description':Repos
df4
```

Out[59]:

	Repository_title	Repository_description	Contributors_count	Language_used
0	a16z-infra /	A MIT-licensed, deployable starter kit for bui...	-	TypeScript
1	opentffoundation /	The OpenTF Manifesto expresses concern over Ha...	239	HTML
2	lllyasviel /	Focus on prompting and generating	-	Python
3	varunshenoy /	An extensible, easy-to-use, and portable diffu...	6	JavaScript
4	modelscope /	FaceChain is a deep-learning toolchain for gen...	7	Python
5	Azure-Samples /	A sample app for the Retrieval-Augmented Gener...	-	Python
6	chatchat-space /	Langchain-Chatchat (formerly langchain-ChatGLM...	74	Python
7	morph-labs /	Rift: an AI-native language server for your pe...	13	Python
8	NVlabs /	Official implementation of "Neuralangelo: High...	-	Python
9	ProfSynapse /	-	-	-
10	steven2358 /	A curated list of modern Generative Artificial...	-	-
11	ibaiw /	2023 HVV情报速递~	-	-
12	iam-veeramalla /	AWS zero to hero repo for devops engineers to ...	-	Python
13	AntonOsika /	Specify what you want it to build, the AI asks...		Python
14	bigskysoftware /	</> htmx - high power tools for HTML		JavaScript
15	angular /	The modern web developer's platform		TypeScript
16	XingangPan /	Official Code for DragGAN (SIGGRAPH 2023)	-	Python
17	google /	GoogleTest - Google Testing and Mocking Framework	406	C++
18	Yidadaa /	A well-designed cross-platform ChatGPT UI (Web...	118	TypeScript
19	awesome-selfhosted /	A list of Free Software network services and w...	1,254	Makefile
20	vercel /	The React Framework	2,824	JavaScript
21	1Panel-dev /	🔥🔥🔥 现代化、开源的 Linux 服务器运维管理面板。	28	Go
22	assafelovic /	GPT based autonomous agent that does online co...	-	Python

	Repository_title	Repository_description	Contributors_count	Language_used
23	dotnet /	.NET MAUI is the .NET Multi-platform App UI, a...	1	C#
24	plbrault /	A game where you are a computer's OS and you h...	2	Python

In [60]:
```
df4.to_csv('trending_repositories.csv')
```

In [62]:
```
'''A) Repository_title
B) Repository_description
C) Contributors_count
D) Language_used'''
```

Out[62]:
```
'A) Repository_title \nB) Repository_description \nC) Contributors_count \nD) Languag
e_used'
```

In [105…
```
driver.close()
```

.

5

.

In [2]:
```
driver = webdriver.Chrome()
driver.get('https:/www.billboard.com/')
```

In [7]:
```
option = driver.find_element(By.XPATH,'/html/body/div[3]/header/div/div[4]/div/div[1]/
option.click()
```

In [8]:
```
chart = driver.find_element(By.XPATH,'/html/body/div[3]/div[9]/div/div/div/ul/li[1]/h3
chart.click()
```

In [9]:
```
hot100 = driver.find_element(By.XPATH,'/html/body/div[3]/main/div[2]/div[1]/div[1]/div
hot100.click()
```

In [71]:
```
Song_name = []
Song = driver.find_elements(By.XPATH,'//li[@class="lrv-u-width-100p"]/ul/li/h3')
for i in Song:
    Song_name.append(i.text)
```

In [90]:
```
Last_week_rank = []
Last_week = driver.find_elements(By.XPATH,'//li[@class="o-chart-results-list__item //
for i in Last_week:
    Last_week_rank.append(i.text)
```

In [91]:
```
Last_week_rank = Last_week_rank[::2]
```

In [94]:
```
Weeks_on_board = []
Weeks = driver.find_elements(By.XPATH,'//li[@class="o-chart-results-list__item // a-ch
```

```
for i in Weeks:
    Weeks_on_board.append(i.text)
```

In [95]:
```
Weeks_on_board = Weeks_on_board[1::2]
```

In [96]:
```
Peak_rank = []
Peak = driver.find_elements(By.XPATH,'//li[@class="o-chart-results-list__item // a-cha
for i in Peak:
    Peak_rank.append(i.text)
```

In [97]:
```
Peak_rank = Peak_rank[1::2]
```

In [ ]:
```
A) Song_name
B) Artist_name
C) Last_week_rank
D) Peak_rank
E) Weeks_on_board
```

In [98]:
```
df5 = pd.DataFrame({'Song_name':Song_name,'Last_week_rank':Last_week_rank,'Peak_rank':
df5
```

Out[98]:

	Song_name	Last_week_rank	Peak_rank	Weeks_on_board
0	Last Night	1	1	28
1	Fast Car	2	2	20
2	Cruel Summer	4	3	14
3	Calm Down	6	3	49
4	Fukumean	7	4	8
...	...	...	...	...
95	Lagunas	-	77	6
96	Overdrive	68	47	3
97	Bzrp Music Sessions, Vol. 55	99	31	10
98	Dawns	-	42	15
99	Rubicon	-	63	6

100 rows × 4 columns

In [99]:
```
df5.to_csv('top_100_songs.csv')
```

In [10]:
```
Artist_name = []
Artist = driver.find_elements(By.XPATH,'//li[@class="lrv-u-width-100p"]/ul/li[1]/span'
for i in Artist:
    Artist_name.append(i.text)
```

In [15]:
```
df5 = pd.read_csv('top_100_songs.csv')
df5
```

Out[15]:

	Unnamed: 0	Song_name	Last_week_rank	Peak_rank	Weeks_on_board
0	0	Last Night	1	1	28
1	1	Fast Car	2	2	20
2	2	Cruel Summer	4	3	14
3	3	Calm Down	6	3	49
4	4	Fukumean	7	4	8
...	...	...	...	...	...
95	95	Lagunas	-	77	6
96	96	Overdrive	68	47	3
97	97	Bzrp Music Sessions, Vol. 55	99	31	10
98	98	Dawns	-	42	15
99	99	Rubicon	-	63	6

100 rows × 5 columns

In [18]:
```
df5['Artist_name']=Artist_name
```

In [19]:
```
df5
```

Out[19]:

	Unnamed: 0	Song_name	Last_week_rank	Peak_rank	Weeks_on_board	Artist_name
0	0	Last Night	1	1	28	Morgan Wallen
1	1	Fast Car	2	2	20	Luke Combs
2	2	Cruel Summer	4	3	14	Taylor Swift
3	3	Calm Down	6	3	49	Rema & Selena Gomez
4	4	Fukumean	7	4	8	Gunna
...	...	...	...	...	...	...
95	95	Lagunas	-	77	6	Peso Pluma & Jasiel Nunez
96	96	Overdrive	68	47	3	Post Malone
97	97	Bzrp Music Sessions, Vol. 55	99	31	10	Bizarrap & Peso Pluma
98	98	Dawns	-	42	15	Zach Bryan Featuring Maggie Rogers
99	99	Rubicon	-	63	6	Peso Pluma

100 rows × 6 columns

In [20]:
```
driver.close()
```

.

6

.

THIS LINK IS NOT OPENING

In [3]:
```
driver = webdriver.Chrome()
driver.get('https://www.theguardian.com/news/datablog/2012/aug/09/best-selling-books-a
```

In [4]:
```
Book_name = []
Book = driver.find_elements(By.XPATH,'//td[@class="left"]')
for i in Book:
    Book_name.append(i.text)
```

In [5]:
```
Book_name = Book_name[1::5]
```

In [6]:
```
Author_name = []
Author = driver.find_elements(By.XPATH,'//td[@class="left"]')
for i in Author:
    Author_name.append(i.text)
```

In [7]:
```
Author_name = Author_name[2::5]
```

In [8]:
```
Volumes_sold = []
Volumes = driver.find_elements(By.XPATH,'//td[@class="left"]')
for i in Volumes:
    Volumes_sold.append(i.text)
```

In [9]:
```
Volumes_sold = Volumes_sold[3::5]
```

In [10]:
```
Publisher = []
Pub = driver.find_elements(By.XPATH,'//td[@class="left"]')
for i in Pub:
    Publisher.append(i.text)
```

In [11]:
```
Publisher = Publisher[4::5]
```

In [12]:
```
Genre = []
Gen = driver.find_elements(By.XPATH,'//td[@class="last left"]')
for i in Gen:
    Genre.append(i.text)
```

In [14]:
```
print(len(Book_name))
print(len(Author_name))
print(len(Volumes_sold))
```

```
print(len(Publisher))
print(len(Genre))
```

```
100
100
100
100
100
```

In [15]:
```
df6 = pd.DataFrame({'Book_name':Book_name,'Author_name':Author_name,'Volumes_sold':Vo
df6
```

Out[15]:

	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 [16]:
```
df6.to_csv('The top 100 bestselling books of all time.csv')
```

In [ ]:
```
driver.close()
```

.

7

.

In [108…
```
driver = webdriver.Chrome()
driver.get('https://www.imdb.com/list/ls095964455/')
```

In [109…
```
Name = []
Na = driver.find_elements(By.XPATH,'//h3[@class="lister-item-header"]')
for i in Na:
    Name.append(i.text)
```

In [110…
```
Year_span = []
Year = driver.find_elements(By.XPATH,'//span[@class="lister-item-year text-muted unbol
for i in Year:
    Year_span.append(i.text)
```

In [111…
```
Genre = []
Gen = driver.find_elements(By.XPATH,'//span[@class="genre"]')
for i in Gen:
    Genre.append(i.text)
```

In [112…
```
Run_time = []
Run = driver.find_elements(By.XPATH,'//span[@class="runtime"]')
for i in Run:
    Run_time.append(i.text)
```

In [113…
```
Ratings = []
Ratin = driver.find_elements(By.XPATH,'//div[@class="ipl-rating-star small"]')
for i in Ratin:
    Ratings.append(i.text)
```

In [114…
```
Votes = []
Vot = driver.find_elements(By.XPATH,'//span[@name="nv"]')
for i in Vot:
    Votes.append(i.text)
```

In [116…
```
'''A) Name
B) Year_span
C) Genre
D) Run_time
E) Ratings
F) Votes '''
```

Out[116]:
```
'A) Name \nB) Year_span \nC) Genre \nD) Run_time \nE) Ratings \nF) Votes '
```

In [117…
```
df7 = pd.DataFrame({'Name':Name,'Year_span':Year_span,'Genre':Genre,'Run_time':Run_tim
df7
```

Out[117]:

	Name	Year_span	Genre	Run_time	Ratings	Votes
0	1. Game of Thrones (2011–2019)	(2011–2019)	Action, Adventure, Drama	57 min	9.2	2,193,366
1	2. Stranger Things (2016–2024)	(2016–2024)	Drama, Fantasy, Horror	51 min	8.7	1,267,099
2	3. The Walking Dead (2010–2022)	(2010–2022)	Drama, Horror, Thriller	44 min	8.1	1,041,100
3	4. 13 Reasons Why (2017–2020)	(2017–2020)	Drama, Mystery, Thriller	60 min	7.5	305,991
4	5. The 100 (2014–2020)	(2014–2020)	Drama, Mystery, Sci-Fi	43 min	7.6	265,074
...	...	...	...	...	...	...
95	96. Reign (2013–2017)	(2013–2017)	Drama	42 min	7.4	52,435
96	97. A Series of Unfortunate Events (2017–2019)	(2017–2019)	Adventure, Comedy, Drama	50 min	7.8	64,462
97	98. Criminal Minds (2005–)	(2005–)	Crime, Drama, Mystery	42 min	8.1	210,123
98	99. Scream: The TV Series (2015–2019)	(2015–2019)	Comedy, Crime, Drama	45 min	7	43,693
99	100. The Haunting of Hill House (2018)	(2018)	Drama, Horror, Mystery	572 min	8.6	263,707

100 rows × 6 columns

In [118...]
```
df7.to_csv('most watched tv series of all time.csv')
```

In [119...]
```
driver.close()
```

.

8

.

In [120...]
```
driver = webdriver.Chrome()
driver.get('https://archive.ics.uci.edu/')
```

In [121...]
```
dataset = driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[1]/d
dataset.click()
```

In [133...]
```
Dataset_name = []
Dataset = driver.find_elements(By.XPATH,'//h2[@class="truncate text-primary"]')
for i in Dataset:
    Dataset_name.append(i.text)
```

In [122…
```
opening_url = []
url = driver.find_elements(By.XPATH,'//a[@class="link-hover link text-xl font-semibold
for i in url:
    opening_url.append(i.get_attribute('href'))
```

In [126…
```
Data_type = []
Task = []
Attribute_type = []
No_of_instances = []
No_of_attribute = []
Year = []

for p in opening_url:
    driver.get(p)
    time.sleep(5)

    try:
        Data= driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[
        Data_type.append(Data.text)
    except NoSuchElementException:
        Data_type.append('_')

    try:
        ta= driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[1]
        Task.append(ta.text)
    except NoSuchElementException:
        Task.append('_')

    try:
        att= driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[1
        Attribute_type.append(att.text)
    except NoSuchElementException:
        Attribute_type.append('_')

    try:
        insta= driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div
        No_of_instances.append(insta.text)
    except NoSuchElementException:
        No_of_instances.append('_')

    try:
        att2= driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[
        No_of_attribute.append(att2.text)
    except NoSuchElementException:
        No_of_attribute.append('_')

    try:
        yea= driver.find_element(By.XPATH,'/html/body/div/div[1]/div[1]/main/div/div[1
        Year.append(yea.text)
    except NoSuchElementException:
        Year.append('_')
```

In [ ]:
```
A) Dataset_name
B) Data_type
C) Task
D) Attribute_type
E) No_of_instances
```

```
F) No_of_attribute
G) Year
```

In [135…
```
df8 = pd.DataFrame({'Dataset_name':Dataset_name,'Data_type':Data_type,'Task':Task,'Att
df8
```

Out[135]:

	Dataset_name	Data_type	Task	Attribute_type	No_of_instances	No_of_attribute	Ye
0	Iris	Multivariate	Classification	Real	150	4	Donat 6/30/19
1	Heart Disease	Multivariate	Classification	Categorical, Integer, Real	303	13	Donat 6/30/19
2	Adult	Multivariate	Classification	Categorical, Integer	48842	14	Donat 4/30/19
3	Dry Bean Dataset	Multivariate	Classification	Integer, Real	13611	16	Donat 9/13/20
4	Diabetes	Multivariate, Time-Series	-	Categorical, Integer	-	20	
5	Wine	Multivariate	Classification	Integer, Real	178	13	Donat 6/30/19
6	Breast Cancer Wisconsin (Diagnostic)	Multivariate	Classification	Real	569	30	Donat 10/31/19
7	Rice (Cammeo and Osmancik)	Multivariate	Classification	Real	3810	8	Donat 10/5/20
8	Car Evaluation	Multivariate	Classification	Categorical	1728	6	Donat 5/31/19
9	Mushroom	Multivariate	Classification	Categorical	8124	22	Donat 4/26/19

In [136…
```
df8.to_csv('dataset.csv')
```

In [137…
```
driver.close()
```