Hands-On Activity 7.2

Webscraping using Beautifulsoup

Performed by: Angelo Luis C. Cu

It should be noted that I do not have a camera available

In [10]:
```
import cv2
key = cv2. waitKey(1)
webcam = cv2.VideoCapture(0)
while True:
    try:
        check, frame = webcam.read()
        print(check) #prints true as long as the webcam is running
        print(frame) #prints matrix values of each framecd
        cv2.imshow("Capturing", frame)
        key = cv2.waitKey(1)
        if key == ord('s'):
            cv2.imwrite(filename='saved_img.jpg', img=frame)
            webcam.release()
            img_new = cv2.imread('saved_img.jpg', cv2.IMREAD_GRAYSCALE)
            img_new = cv2.imshow("Captured Image", img_new)
            cv2.waitKey(1650)
            cv2.destroyAllWindows()
            print("Processing image...")
            img_ = cv2.imread('saved_img.jpg', cv2.IMREAD_ANYCOLOR)
            print("Converting RGB image to grayscale...")
            gray = cv2.cvtColor(img_, cv2.COLOR_BGR2GRAY)
            print("Converted RGB image to grayscale...")
            print("Resizing image to 28x28 scale...")
            img_ = cv2.resize(gray,(28,28))
            print("Resized...")
            img_resized = cv2.imwrite(filename='saved_img-final.jpg', img=img_)
            print("Image saved!")
            break
        elif key == ord('q'):
            print("Turning off camera.")
            webcam.release()
            print("Camera off.")
            print("Program ended.")
            cv2.destroyAllWindows()
            break
    except(KeyboardInterrupt):
        print("Turning off camera.")
        webcam.release()
        print("Camera off.")
        print("Program ended.")
        cv2.destroyAllWindows()
        break
```

```
False
None
---------------------------------------------------------------------------
error                                     Traceback (most recent call last)
Cell In[10], line 9
      7 print(check) #prints true as long as the webcam is running
      8 print(frame) #prints matrix values of each framecd
----> 9 cv2.imshow("Capturing", frame)
     10 key = cv2.waitKey(1)
     11 if key == ord('s'):

error: OpenCV(4.9.0) D:\a\opencv-python\opencv-python\opencv\modules\highgui\src\wind
ow.cpp:971: error: (-215:Assertion failed) size.width>0 && size.height>0 in function
'cv::imshow'
```

In order for the code above to run, I have to download packages such as opencv

In [3]:
```
!pip3 install opencv-python
```

```
Collecting opencv-python
  Downloading opencv_python-4.9.0.80-cp37-abi3-win_amd64.whl.metadata (20 kB)
Requirement already satisfied: numpy>=1.21.2 in c:\users\cu\anaconda3\lib\site-packag
es (from opencv-python) (1.24.3)
Downloading opencv_python-4.9.0.80-cp37-abi3-win_amd64.whl (38.6 MB)
   ---------------------------------------- 0.0/38.6 MB ? eta -:--:--
   ---------------------------------------- 0.1/38.6 MB 5.1 MB/s eta 0:00:08
   - -------------------------------------- 1.0/38.6 MB 16.0 MB/s eta 0:00:03
   -- ------------------------------------- 2.0/38.6 MB 18.1 MB/s eta 0:00:03
   --- ------------------------------------ 3.3/38.6 MB 19.2 MB/s eta 0:00:02
   ---- ----------------------------------- 4.7/38.6 MB 23.1 MB/s eta 0:00:02
   ------ --------------------------------- 6.1/38.6 MB 22.9 MB/s eta 0:00:02
   ------- -------------------------------- 7.6/38.6 MB 24.3 MB/s eta 0:00:02
   --------- ------------------------------ 8.8/38.6 MB 24.6 MB/s eta 0:00:02
   ---------- ----------------------------- 10.3/38.6 MB 26.2 MB/s eta 0:00:02
   ------------ --------------------------- 11.6/38.6 MB 28.5 MB/s eta 0:00:01
   ------------- -------------------------- 12.9/38.6 MB 28.5 MB/s eta 0:00:01
   -------------- ------------------------- 14.1/38.6 MB 28.4 MB/s eta 0:00:01
   --------------- ------------------------ 15.4/38.6 MB 28.4 MB/s eta 0:00:01
   ----------------- ---------------------- 16.6/38.6 MB 27.3 MB/s eta 0:00:01
   ------------------ --------------------- 18.1/38.6 MB 27.3 MB/s eta 0:00:01
   -------------------- ------------------- 19.4/38.6 MB 27.3 MB/s eta 0:00:01
   --------------------- ------------------ 20.6/38.6 MB 27.3 MB/s eta 0:00:01
   ---------------------- ----------------- 21.6/38.6 MB 26.2 MB/s eta 0:00:01
   ----------------------- ---------------- 23.0/38.6 MB 25.1 MB/s eta 0:00:01
   ------------------------ --------------- 24.2/38.6 MB 27.3 MB/s eta 0:00:01
   ------------------------- -------------- 25.5/38.6 MB 26.2 MB/s eta 0:00:01
   --------------------------- ------------ 26.8/38.6 MB 27.3 MB/s eta 0:00:01
   ---------------------------- ----------- 28.0/38.6 MB 25.2 MB/s eta 0:00:01
   ----------------------------- ---------- 29.3/38.6 MB 25.2 MB/s eta 0:00:01
   ------------------------------ --------- 30.6/38.6 MB 27.3 MB/s eta 0:00:01
   -------------------------------- ------- 32.0/38.6 MB 27.3 MB/s eta 0:00:01
   --------------------------------- ------ 33.3/38.6 MB 27.3 MB/s eta 0:00:01
   ---------------------------------- ----- 34.7/38.6 MB 27.3 MB/s eta 0:00:01
   ----------------------------------- ---- 36.0/38.6 MB 27.3 MB/s eta 0:00:01
   ------------------------------------ --- 37.4/38.6 MB 28.5 MB/s eta 0:00:01
   -------------------------------------  38.6/38.6 MB 27.3 MB/s eta 0:00:01
   ---------------------------------------- 38.6/38.6 MB 24.2 MB/s eta 0:00:00
Installing collected packages: opencv-python
Successfully installed opencv-python-4.9.0.80
```

In [11]:
```
!pip3 install sounddevice
```

```
Collecting sounddevice
  Downloading sounddevice-0.4.6-py3-none-win_amd64.whl.metadata (1.4 kB)
Requirement already satisfied: CFFI>=1.0 in c:\users\cu\anaconda3\lib\site-packages
(from sounddevice) (1.15.1)
Requirement already satisfied: pycparser in c:\users\cu\anaconda3\lib\site-packages
(from CFFI>=1.0->sounddevice) (2.21)
Downloading sounddevice-0.4.6-py3-none-win_amd64.whl (199 kB)
   ---------------------------------------- 0.0/199.7 kB ? eta -:--:--
   -- ------------------------------------- 10.2/199.7 kB ? eta -:--:--
   ------------------------------------ - 194.6/199.7 kB 3.0 MB/s eta 0:00:01
   ---------------------------------------- 199.7/199.7 kB 3.1 MB/s eta 0:00:00
Installing collected packages: sounddevice
Successfully installed sounddevice-0.4.6
```

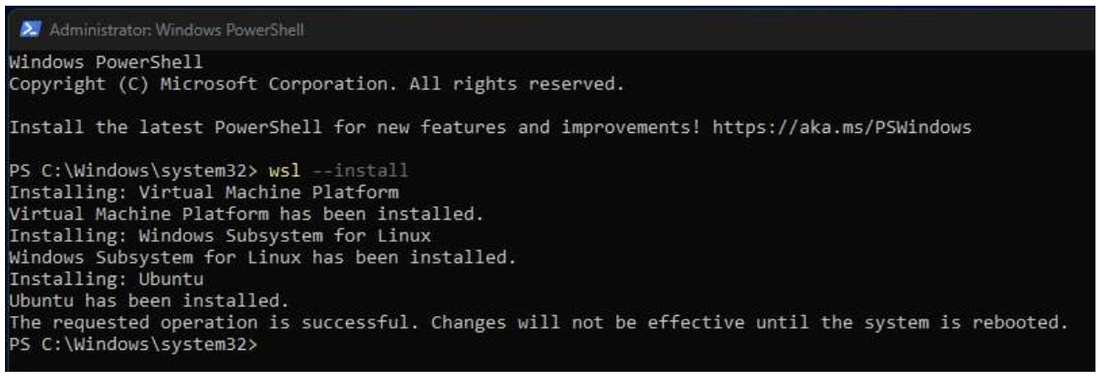
In [12]:
```
!pip3 install wavio
```

```
Collecting wavio
  Downloading wavio-0.0.8-py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: numpy>=1.19.0 in c:\users\cu\anaconda3\lib\site-packag
es (from wavio) (1.24.3)
Downloading wavio-0.0.8-py3-none-any.whl (9.4 kB)
Installing collected packages: wavio
Successfully installed wavio-0.0.8
```

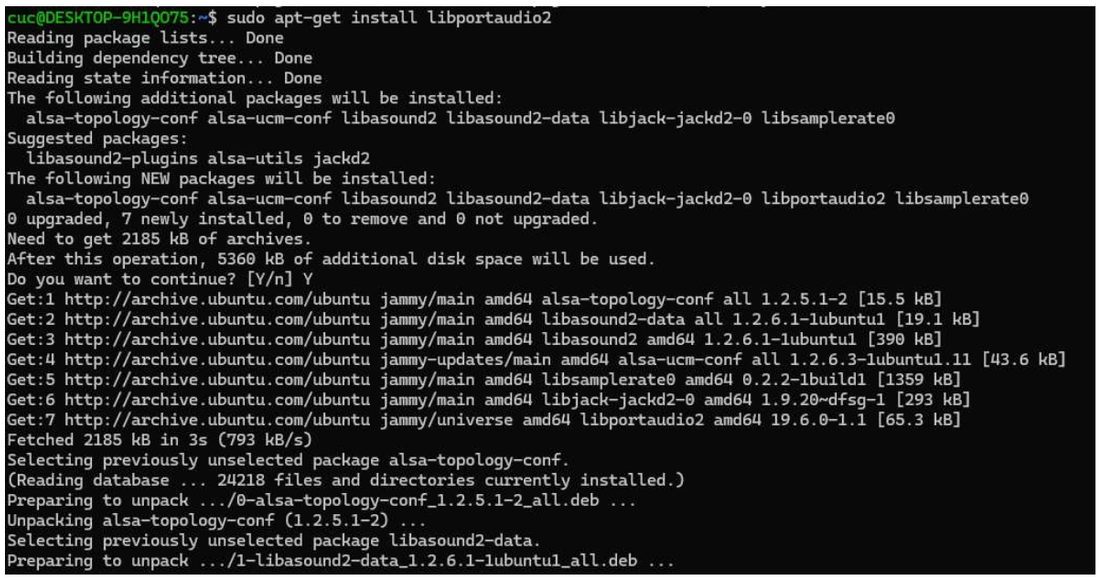
In [13]:
```
!pip3 install scipy
```

```
Requirement already satisfied: scipy in c:\users\cu\anaconda3\lib\site-packages (1.1
1.1)
Requirement already satisfied: numpy<1.28.0,>=1.21.6 in c:\users\cu\anaconda3\lib\sit
e-packages (from scipy) (1.24.3)
```

As I am on windows, the !apt-get command does not work out of the box, I had to install WSL as seen below:



I installed the package on the ubuntu CLI:



In [3]:
```
# import required libraries
import sounddevice as sd
from scipy.io.wavfile import write
import wavio as wv

# Sampling frequency
freq = 44100

# Recording duration
duration = 5

# Start recorder with the given values
# of duration and sample frequency
recording = sd.rec(int(duration * freq),
samplerate=freq, channels=2)

# Record audio for the given number of seconds
sd.wait()

# This will convert the NumPy array to an audio
# file with the given sampling frequency
write("recording0.wav", freq, recording)

# Convert the NumPy array to audio file
wv.write("recording1.wav", recording, freq, sampwidth=2)
```

In [4]:
```
!pip install bs4
```

```
Collecting bs4
  Downloading bs4-0.0.2-py2.py3-none-any.whl.metadata (411 bytes)
Requirement already satisfied: beautifulsoup4 in c:\users\cu\anaconda3\lib\site-packa
ges (from bs4) (4.12.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\cu\anaconda3\lib\site-packag
es (from beautifulsoup4->bs4) (2.4)
Downloading bs4-0.0.2-py2.py3-none-any.whl (1.2 kB)
Installing collected packages: bs4
Successfully installed bs4-0.0.2
```

In [5]:
```
!pip install requests
```

```
Requirement already satisfied: requests in c:\users\cu\anaconda3\lib\site-packages
(2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\cu\anaconda3\lib
\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\cu\anaconda3\lib\site-package
s (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\cu\anaconda3\lib\site-p
ackages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\cu\anaconda3\lib\site-p
ackages (from requests) (2023.11.17)
```

In [6]:

```python
import requests
from bs4 import BeautifulSoup

def getdata(url):
    r = requests.get(url)
    return r.text

htmldata = getdata("https://www.google.com/") # website to get images from
soup = BeautifulSoup(htmldata, 'html.parser')

for item in soup.find_all('img'): # gets all <img> html element
    print(item['src']) # prints the src="" parameter
```

```
/images/branding/googlelogo/1x/googlelogo_white_background_color_272x92dp.png
```

In [8]:

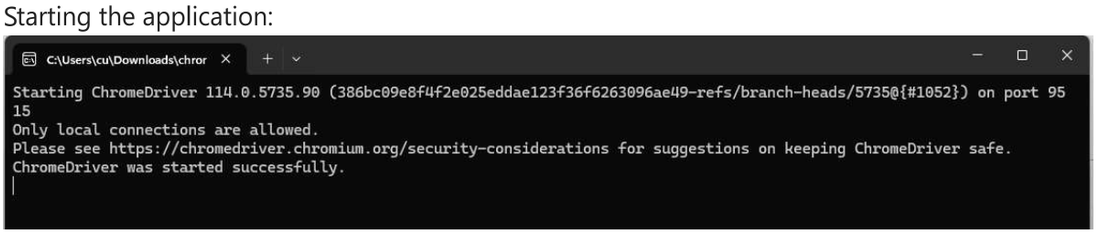
```
!pip3 install selenium
```

```
Requirement already satisfied: selenium in c:\users\cu\anaconda3\lib\site-packages
(4.18.1)
Requirement already satisfied: urllib3<3,>=1.26 in c:\users\cu\anaconda3\lib\site-pac
kages (from urllib3[socks]<3,>=1.26->selenium) (1.26.16)
Requirement already satisfied: trio~=0.17 in c:\users\cu\anaconda3\lib\site-packages
(from selenium) (0.25.0)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\cu\anaconda3\lib\site-
packages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\cu\anaconda3\lib\site-p
ackages (from selenium) (2023.11.17)
Requirement already satisfied: typing_extensions>=4.9.0 in c:\users\cu\anaconda3\lib
\site-packages (from selenium) (4.10.0)
Requirement already satisfied: attrs>=23.2.0 in c:\users\cu\anaconda3\lib\site-packag
es (from trio~=0.17->selenium) (23.2.0)
Requirement already satisfied: sortedcontainers in c:\users\cu\anaconda3\lib\site-pac
kages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in c:\users\cu\anaconda3\lib\site-packages (from
trio~=0.17->selenium) (3.4)
Requirement already satisfied: outcome in c:\users\cu\anaconda3\lib\site-packages (fr
om trio~=0.17->selenium) (1.3.0.post0)
Requirement already satisfied: sniffio>=1.3.0 in c:\users\cu\anaconda3\lib\site-packa
ges (from trio~=0.17->selenium) (1.3.1)
Requirement already satisfied: cffi>=1.14 in c:\users\cu\anaconda3\lib\site-packages
(from trio~=0.17->selenium) (1.15.1)
Requirement already satisfied: wsproto>=0.14 in c:\users\cu\anaconda3\lib\site-packag
es (from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\users\cu\anaconda3\l
ib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\cu\anaconda3\lib\site-packages
(from cffi>=1.14->trio~=0.17->selenium) (2.21)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\cu\anaconda3\lib\site-packag
es (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)
```

I removed the !apt install chromium-chromedriver and instead installed it directly from this site:

https://chromedriver.chromium.org/downloads

Starting the application:



In [27]:

```python
from selenium import webdriver
import time
import requests
import shutil
```

```
import os
import getpass
import urllib.request
import io
import time
from PIL import Image
user = getpass.getuser()
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome(options=chrome_options) # removed 'chromedriver' parameter
search_url = "https://www.google.com/search?q={q}&tbm=isch&tbs=sur%3Afc&hl=en&ved=0CAI
driver.get(search_url.format(q='Car'))

def scroll_to_end(driver):
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
    time.sleep(5) #sleep_between_interactions

def getImageUrls(name,totalImgs,driver):
    search_url = "https://www.google.com/search?q={q}&tbm=isch&tbs=sur%3Afc&hl=en&ved=
    driver.get(search_url.format(q=name))
    img_urls = set()
    img_count = 0
    results_start = 0

    while(img_count<totalImgs): #Extract actual images now
        scroll_to_end(driver)
        thumbnail_results = driver.find_elements_by_xpath("//img[contains(@class,'Q4Lu
        totalResults=len(thumbnail_results)
        print(f"Found: {totalResults} search results. Extracting links from{results_st

        for img in thumbnail_results[results_start:totalResults]:
            img.click()
            time.sleep(2)
            actual_images = driver.find_elements_by_css_selector('img.n3VNCb')
            for actual_image in actual_images:
                if actual_image.get_attribute('src') and 'https' in actual_image.get_a
                    img_urls.add(actual_image.get_attribute('src'))
            img_count=len(img_urls)
            if img_count >= totalImgs:
                print(f"Found: {img_count} image links")
                break
            else:
                print("Found:", img_count, "looking for more image links ...")
                load_more_button = driver.find_element_by_css_selector(".mye4qd")
                driver.execute_script("document.querySelector('.mye4qd').click();")
                results_start = len(thumbnail_results)
    return img_urls

def downloadImages(folder_path,file_name,url):
    try:
        image_content = requests.get(url).content
    except Exception as e:
        print(f"ERROR - COULD NOT DOWNLOAD {url} - {e}")
    try:
        image_file = io.BytesIO(image_content)
        image = Image.open(image_file).convert('RGB')
        file_path = os.path.join(folder_path, file_name)
        with open(file_path, 'wb') as f:
            image.save(f, "JPEG", quality=85)
        print(f"SAVED - {url} - AT: {file_path}")
    except Exception as e:
        print(f"ERROR - COULD NOT SAVE {url} - {e}")

def saveInDestFolder(searchNames,destDir,totalImgs,driver):
    for name in list(searchNames):
        path=os.path.join(destDir,name)
        if not os.path.isdir(path):
            os.mkdir(path)
        print('Current Path',path)
        totalLinks=getImageUrls(name,totalImgs,driver)
        print('totalLinks',totalLinks)
    if totalLinks is None:
        print('images not found for :',name)
    else:
        for i, link in enumerate(totalLinks):
            file_name = f"{i:150}.jpg"
```

```
            downloadImages(path,file_name,link)

searchNames=['cat']
destDir='C:\\result'
totalImgs=5
saveInDestFolder(searchNames,destDir,totalImgs,driver)
```

```
Current Path C:\result\cat
```

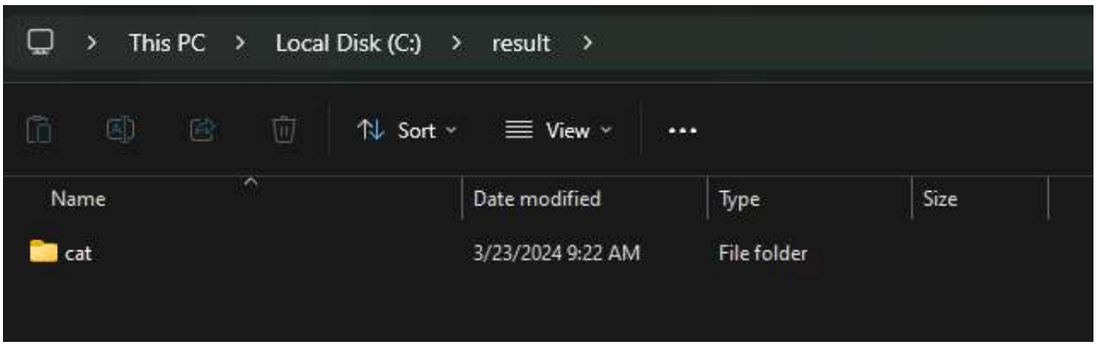
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[27], line 88
     86 destDir='C:\\result'
     87 totalImgs=5
---> 88 saveInDestFolder(searchNames,destDir,totalImgs,driver)

Cell In[27], line 76, in saveInDestFolder(searchNames, destDir, totalImgs, driver)
     74         os.mkdir(path)
     75     print('Current Path',path)
---> 76     totalLinks=getImageUrls(name,totalImgs,driver)
     77     print('totalLinks',totalLinks)
     78 if totalLinks is None:

Cell In[27], line 33, in getImageUrls(name, totalImgs, driver)
     31 while(img_count<totalImgs): #Extract actual images now
     32     scroll_to_end(driver)
---> 33     thumbnail_results = driver.find_elements_by_xpath("//img[contains(@clas
s,'Q4LuWd')]")
     34     totalResults=len(thumbnail_results)
     35     print(f"Found: {totalResults} search results. Extracting links from{resul
ts_start}:{totalResults}")

AttributeError: 'WebDriver' object has no attribute 'find_elements_by_xpath'
```

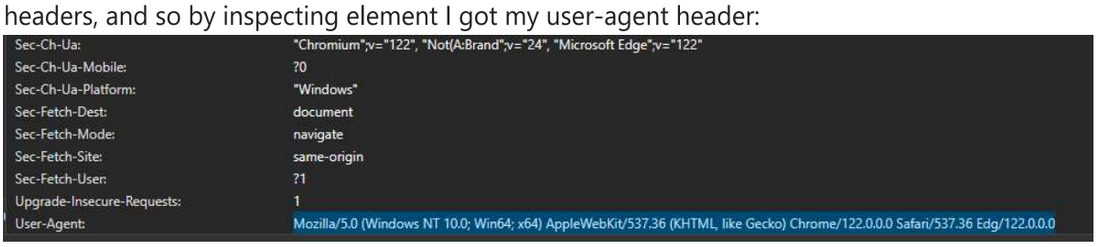
The problem seems to be with the webdriver package by selenium, but the folder for cat was made:



In [38]:
```
from requests import get
url = 'https://www.imdb.com/search/title/?release_date=2017-01-01,2017-12-31&sort=num_
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
response = get(url, headers=headers)
print(response.text[:500])
```

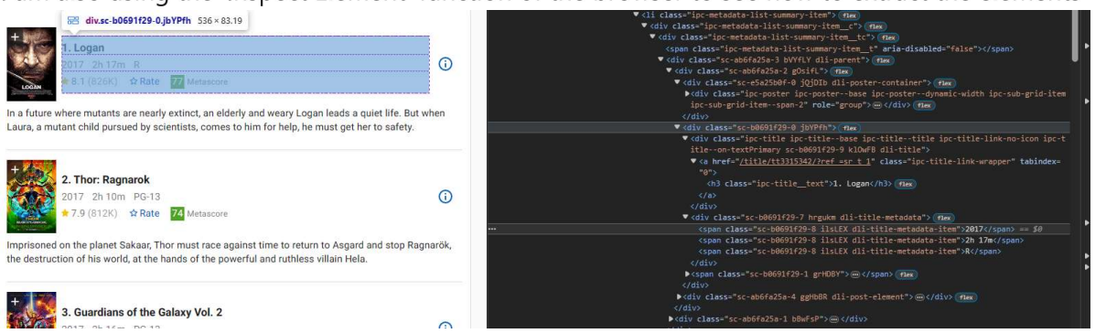
```
<!DOCTYPE html><html lang="en-US" xmlns:og="http://opengraphprotocol.org/schema/" xml
ns:fb="http://www.facebook.com/2008/fbml"><head><meta charSet="utf-8"/><meta name="vi
ewport" content="width=device-width"/><script>if(typeof uet === 'function'){ uet('b
b', 'LoadTitle', {wb: 1}); }</script><script>window.addEventListener('load', (event)
=> {
        if (typeof window.csa !== 'undefined' && typeof window.csa === 'function') {
            var csaLatencyPlugin = window.csa('Content', {
```

It seems that imdb is now not allowing the access of their website via requests without the headers, and so by inspecting element I got my user-agent header:



It should also be noted that the website's layout and html structure had changed since the hands-on activity. I decided to comment the changes for clarity

I am also using the 'Inspect Element' function of the browser to see how to extract the elements



In [39]:
```
from bs4 import BeautifulSoup
html_soup = BeautifulSoup(response.text, 'html.parser')
headers = {'Accept-Language': 'en-US,en;q=0.8'}
type(html_soup)
```

Out[39]:
```
bs4.BeautifulSoup
```

In [42]:
```
movie_containers = html_soup.find_all('div', class_ = 'ipc-metadata-list-summary-item_
print(type(movie_containers))
print(len(movie_containers))
```

```
<class 'bs4.element.ResultSet'>
50
```

In [43]:
```
first_movie = movie_containers[0]
first_movie
```

Out[43]:

```
<div class="ipc-metadata-list-summary-item__tc"><span aria-disabled="false" class="ip
c-metadata-list-summary-item__t"></span><div class="sc-ab6fa25a-3 bVYfLY dli-parent">
<div class="sc-ab6fa25a-2 gOsifL"><div class="sc-e5a25b0f-0 jQjDIb dli-poster-contain
er"><div class="ipc-poster ipc-poster--base ipc-poster--dynamic-width ipc-sub-grid-it
em ipc-sub-grid-item--span-2" role="group"><div aria-label="add to watchlist" class
="ipc-watchlist-ribbon ipc-focusable ipc-watchlist-ribbon--s ipc-watchlist-ribbon--ba
se ipc-watchlist-ribbon--loading ipc-watchlist-ribbon--onImage ipc-poster__watchlist-
ribbon" role="button" tabindex="0"><svg class="ipc-watchlist-ribbon__bg" height="34p
x" role="presentation" viewbox="0 0 24 34" width="24px" xmlns="http://www.w3.org/200
0/svg"><polygon class="ipc-watchlist-ribbon__bg-ribbon" fill="#000000" points="24 0 0
0 0 32 12.2436611 26.2926049 24 31.7728343"></polygon><polygon class="ipc-watchlist-r
ibbon__bg-hover" points="24 0 0 0 0 32 12.2436611 26.2926049 24 31.7728343"></polygon
><polygon class="ipc-watchlist-ribbon__bg-shadow" points="24 31.7728343 24 33.7728343
12.2436611 28.2926049 0 34 0 32 12.2436611 26.2926049"></polygon></svg><div class="ip
c-watchlist-ribbon__icon" role="presentation"><svg class="ipc-loader ipc-loader--circ
le ipc-watchlist-ribbon__loader" data-testid="watchlist-ribbon-loader" height="48px"
role="presentation" version="1.1" viewbox="0 0 48 48" width="48px" xmlns="http://www.
w3.org/2000/svg"><g class="ipc-loader__container" fill="currentColor"><circle class
="ipc-loader__circle ipc-loader__circle--one" cx="24" cy="9" r="4"></circle><circle c
lass="ipc-loader__circle ipc-loader__circle--two" cx="35" cy="14" r="4"></circle><cir
cle class="ipc-loader__circle ipc-loader__circle--three" cx="39" cy="24" r="4"></circ
le><circle class="ipc-loader__circle ipc-loader__circle--four" cx="35" cy="34" r="4">
</circle><circle class="ipc-loader__circle ipc-loader__circle--five" cx="24" cy="39"
r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--six" cx="13" cy
="34" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--seven" cx
="9" cy="24" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circle--eig
ht" cx="13" cy="14" r="4"></circle></g></svg></div></div><div class="ipc-media ipc-me
dia--poster-27x40 ipc-image-media-ratio--poster-27x40 ipc-media--base ipc-media--post
er-m ipc-poster__poster-image ipc-media__img" style="width:100%"><img alt="Hugh Jackm
an in Logan (2017)" class="ipc-image" loading="lazy" sizes="50vw, (min-width: 480px)
34vw, (min-width: 600px) 26vw, (min-width: 1024px) 16vw, (min-width: 1280px) 16vw" sr
c="https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMz
JhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX140_CR0,1,140,207_.jpg" srcset="http
s://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzX
kEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX140_CR0,1,140,207_.jpg 140w, https://m.media-a
mazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQX
VyNjc1NTYyMjg@._V1_QL75_UX210_CR0,2,210,311_.jpg 210w, https://m.media-amazon.com/ima
ges/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg
@._V1_QL75_UX280_CR0,3,280,414_.jpg 280w" width="140"/></div><a aria-label="View titl
e page for Logan" class="ipc-lockup-overlay ipc-focusable" href="/title/tt3315342/?re
f_=sr_i_1"><div class="ipc-lockup-overlay__screen"></div></a></div></div><div class
="sc-b0691f29-0 jbYPfh"><div class="ipc-title ipc-title--base ipc-title--title ipc-ti
tle-link-no-icon ipc-title--on-textPrimary sc-b0691f29-9 klOwFB dli-title"><a class
="ipc-title-link-wrapper" href="/title/tt3315342/?ref_=sr_t_1" tabindex="0"><h3 class
="ipc-title__text">1. Logan</h3></a></div><div class="sc-b0691f29-7 hrgukm dli-title-
metadata"><span class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">2017</span><span
class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">2h 17m</span><span class="sc-b06
91f29-8 ilsLEX dli-title-metadata-item">R-16</span></div><span class="sc-b0691f29-1 g
rHDBY"><div class="sc-e2dbc1a3-0 ajrIH sc-b0691f29-2 bhhtyj dli-ratings-container" da
ta-testid="ratingGroup--container"><span aria-label="IMDb rating: 8.1" class="ipc-rat
ing-star ipc-rating-star--base ipc-rating-star--imdb ratingGroup--imdb-rating" data-t
estid="ratingGroup--imdb-rating"><svg class="ipc-icon ipc-icon--star-inline" fill="cu
rrentColor" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="htt
p://www.w3.org/2000/svg"><path d="M12 20.1l5.82 3.682c1.066.675 2.37-.322 2.09-1.584l
-1.543-6.926 5.146-4.667c.94-.85.435-2.465-.799-2.567l-6.773-.602L13.29.89a1.38 1.38
0 0 0-2.581 0l-2.65 6.53-6.774.602C.052 8.126-.453 9.74.486 10.59l5.147 4.666-1.542
6.926c-.28 1.262 1.023 2.26 2.09 1.585L12 20.099z"></path></svg>8.1<span class="ipc-r
ating-star--voteCount"> (<!-- -->826K<!-- -->)</span></span><button aria-label="Rate
Logan" class="ipc-rate-button sc-e2dbc1a3-1 jboOQc ratingGroup--user-rating ipc-rate-
button--unrated ipc-rate-button--base" data-testid="rate-button"><span class="ipc-rat
ing-star ipc-rating-star--base ipc-rating-star--rate"><svg class="ipc-icon ipc-icon--
star-border-inline" fill="currentColor" height="24" role="presentation" viewbox="0 0
24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M22.724 8.217l-6.786-.5
87-2.65-6.22c-.477-1.133-2.103-1.133-2.58 0l-2.65 6.234-6.772.573c-1.234.098-1.739 1.
636-.8 2.446l5.146 4.446-1.542 6.598c-.28 1.202 1.023 2.153 2.09 1.51l5.818-3.495 5.8
19 3.509c1.065.643 2.37-.308 2.089-1.51l-1.542-6.612 5.145-4.446c.94-.81.45-2.348-.78
5-2.446zm-10.726 8.89l-5.272 3.174 1.402-5.983-4.655-4.026 6.141-.531 2.384-5.634 2.3
98 5.648 6.14.531-4.654 4.026 1.402 5.983-5.286-3.187z"></path></svg><span class="ipc
-rating-star--rate">Rate</span></span></button></div><span class="sc-b0691f29-11 TmkK
M"><span class="sc-b0901df4-0 bcQdDJ metacritic-score-box" style="background-color:#5
4A72A">77</span><span class="metacritic-score-label">Metascore</span></span></span></
div><div class="sc-ab6fa25a-4 ggHbBR dli-post-element"><button aria-disabled="false"
aria-label="See more information about Logan" class="ipc-icon-button dli-info-icon ip
c-icon-button--base ipc-icon-button--onAccent2" role="button" tabindex="0" title="See
more information about Logan"><svg class="ipc-icon ipc-icon--info" fill="currentColo
r" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w
3.org/2000/svg"><path d="M0 0h24v24H0V0z" fill="none"></path><path d="M11 7h2v2h-2zm0
4h2v6h-2zm1-9C6.48 2 2 6.48 2 12s4.48 10 10 10 10-4.48 10-10S17.52 2 12 2zm0 18c-4.41
```

0-8-3.59-8-8s3.59-8 8-8 8 3.59 8 8-3.59 8-8 8z"></path></svg></button> In a future where mutants are nearly extinct, an elderly and weary Logan leads a quiet life. But when Laura, a mutant child pursued by scientists, comes to him for help, he must get her to safety.

In [44]:
```
first_movie.div
```

Out[44]:

```
<div class="sc-ab6fa25a-3 bVYfLY dli-parent"><div class="sc-ab6fa25a-2 gOsifL"><div c
lass="sc-e5a25b0f-0 jQjDIb dli-poster-container"><div class="ipc-poster ipc-poster--b
ase ipc-poster--dynamic-width ipc-sub-grid-item ipc-sub-grid-item--span-2" role="grou
p"><div aria-label="add to watchlist" class="ipc-watchlist-ribbon ipc-focusable ipc-w
atchlist-ribbon--s ipc-watchlist-ribbon--base ipc-watchlist-ribbon--loading ipc-watch
list-ribbon--onImage ipc-poster__watchlist-ribbon" role="button" tabindex="0"><svg cl
ass="ipc-watchlist-ribbon__bg" height="34px" role="presentation" viewbox="0 0 24 34"
width="24px" xmlns="http://www.w3.org/2000/svg"><polygon class="ipc-watchlist-ribbon_
_bg-ribbon" fill="#000000" points="24 0 0 0 0 32 12.2436611 26.2926049 24 31.772834
3"></polygon><polygon class="ipc-watchlist-ribbon__bg-hover" points="24 0 0 0 0 32 1
2.2436611 26.2926049 24 31.7728343"></polygon><polygon class="ipc-watchlist-ribbon__b
g-shadow" points="24 31.7728343 24 33.7728343 12.2436611 28.2926049 0 34 0 32 12.2436
611 26.2926049"></polygon></svg><div class="ipc-watchlist-ribbon__icon" role="present
ation"><svg class="ipc-loader ipc-loader--circle ipc-watchlist-ribbon__loader" data-t
estid="watchlist-ribbon-loader" height="48px" role="presentation" version="1.1" viewb
ox="0 0 48 48" width="48px" xmlns="http://www.w3.org/2000/svg"><g class="ipc-loader__
container" fill="currentColor"><circle class="ipc-loader__circle ipc-loader__circle--
one" cx="24" cy="9" r="4"></circle><circle class="ipc-loader__circle ipc-loader__circ
le--two" cx="35" cy="14" r="4"></circle><circle class="ipc-loader__circle ipc-loader_
_circle--three" cx="39" cy="24" r="4"></circle><circle class="ipc-loader__circle ipc-
loader__circle--four" cx="35" cy="34" r="4"></circle><circle class="ipc-loader__circl
e ipc-loader__circle--five" cx="24" cy="39" r="4"></circle><circle class="ipc-loader_
_circle ipc-loader__circle--six" cx="13" cy="34" r="4"></circle><circle class="ipc-lo
ader__circle ipc-loader__circle--seven" cx="9" cy="24" r="4"></circle><circle class
="ipc-loader__circle ipc-loader__circle--eight" cx="13" cy="14" r="4"></circle></g></
svg></div></div><div class="ipc-media ipc-media--poster-27x40 ipc-image-media-ratio--
poster-27x40 ipc-media--base ipc-media--poster-m ipc-poster__poster-image ipc-media__
img" style="width:100%"><img alt="Hugh Jackman in Logan (2017)" class="ipc-image" loa
ding="lazy" sizes="50vw, (min-width: 480px) 34vw, (min-width: 600px) 26vw, (min-widt
h: 1024px) 16vw, (min-width: 1280px) 16vw" src="https://m.media-amazon.com/images/M/M
V5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_Q
L75_UX140_CR0,1,140,207_.jpg" srcset="https://m.media-amazon.com/images/M/MV5BYzc5MTU
4N2EtYTkyMi00NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX140_
CR0,1,140,207_.jpg 140w, https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00
NjdhLTg3NWEtMTY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX210_CR0,2,210,311
_.jpg 210w, https://m.media-amazon.com/images/M/MV5BYzc5MTU4N2EtYTkyMi00NjdhLTg3NWEtM
TY4OTEyMzJhZTAzXkEyXkFqcGdeQXVyNjc1NTYyMjg@._V1_QL75_UX280_CR0,3,280,414_.jpg 280w" w
idth="140"/></div><a aria-label="View title page for Logan" class="ipc-lockup-overlay
ipc-focusable" href="/title/tt3315342/?ref_=sr_i_1"><div class="ipc-lockup-overlay__s
creen"></div></a></div></div><div class="sc-b0691f29-0 jbYPfh"><div class="ipc-title
ipc-title--base ipc-title--title ipc-title-link-no-icon ipc-title--on-textPrimary sc-
b0691f29-9 klOwFB dli-title"><a class="ipc-title-link-wrapper" href="/title/tt331534
2/?ref_=sr_t_1" tabindex="0"><h3 class="ipc-title__text">1. Logan</h3></a></div><div
class="sc-b0691f29-7 hrgukm dli-title-metadata"><span class="sc-b0691f29-8 ilsLEX dli
-title-metadata-item">2017</span><span class="sc-b0691f29-8 ilsLEX dli-title-metadata
-item">2h 17m</span><span class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">R-16</
span></div><span class="sc-b0691f29-1 grHDBY"><div class="sc-e2dbc1a3-0 ajrIH sc-b069
1f29-2 bhhtyj dli-ratings-container" data-testid="ratingGroup--container"><span aria-
label="IMDb rating: 8.1" class="ipc-rating-star ipc-rating-star--base ipc-rating-star
--imdb ratingGroup--imdb-rating" data-testid="ratingGroup--imdb-rating"><svg class="i
pc-icon ipc-icon--star-inline" fill="currentColor" height="24" role="presentation" vi
ewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M12 20.1l5.8
2 3.682c1.066.675 2.37-.322 2.09-1.584l-1.543-6.926 5.146-4.667c.94-.85.435-2.465-.79
9-2.567l-6.773-.602L13.29.89a1.38 1.38 0 0 0-2.581 0l-2.65 6.53-6.774.602C.052 8.126
-.453 9.74.486 10.59l5.147 4.666-1.542 6.926c-.28 1.262 1.023 2.26 2.09 1.585L12 20.0
99z"></path></svg>8.1<span class="ipc-rating-star--voteCount"> (<!-- -->826K<!-- -->)
</span></span><button aria-label="Rate Logan" class="ipc-rate-button sc-e2dbc1a3-1 jb
oOQc ratingGroup--user-rating ipc-rate-button--unrated ipc-rate-button--base" data-te
stid="rate-button"><span class="ipc-rating-star ipc-rating-star--base ipc-rating-star
--rate"><svg class="ipc-icon ipc-icon--star-border-inline" fill="currentColor" height
="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://www.w3.org/200
0/svg"><path d="M22.724 8.217l-6.786-.587-2.65-6.22c-.477-1.133-2.103-1.133-2.58 0l-
2.65 6.234-6.772.573c-1.234.098-1.739 1.636-.8 2.446l5.146 4.446-1.542 6.598c-.28 1.2
02 1.023 2.153 2.09 1.51l5.818-3.495 5.819 3.509c1.065.643 2.37-.308 2.089-1.51l-1.54
2-6.612 5.145-4.446c.94-.81.45-2.348-.785-2.446zm-10.726 8.89l-5.272 3.174 1.402-5.98
3-4.655-4.026 6.141-.531 2.384-5.634 2.398 5.648 6.14.531-4.654 4.026 1.402 5.983-5.2
86-3.187z"></path></svg><span class="ipc-rating-star--rate">Rate</span></span></butto
n></div><span class="sc-b0691f29-11 TmkKM"><span class="sc-b0901df4-0 bcQdDJ metacrit
ic-score-box" style="background-color:#54A72A">77</span><span class="metacritic-score
-label">Metascore</span></span></span></div><div class="sc-ab6fa25a-4 ggHbBR dli-post
-element"><button aria-disabled="false" aria-label="See more information about Logan"
class="ipc-icon-button dli-info-icon ipc-icon-button--base ipc-icon-button--onAccent
2" role="button" tabindex="0" title="See more information about Logan"><svg class="ip
c-icon ipc-icon--info" fill="currentColor" height="24" role="presentation" viewbox="0
0 24 24" width="24" xmlns="http://www.w3.org/2000/svg"><path d="M0 0h24v24H0V0z" fill
="none"></path><path d="M11 7h2v2h-2zm0 4h2v6h-2zm1-9C6.48 2 2 6.48 2 12s4.48 10 10 1
0 10-4.48 10-10S17.52 2 12 2zm0 18c-4.41 0-8-3.59-8-8s3.59-8 8-8 8 3.59 8 8-3.59 8-8
8z"></path></svg></button></div></div><div class="sc-ab6fa25a-1 bBwFsP"><div class="i
```

```
pc-html-content ipc-html-content--base sc-ab6fa25a-0 bhexuD dli-plot-container" role
="presentation"><div class="ipc-html-content-inner-div">In a future where mutants are
nearly extinct, an elderly and weary Logan leads a quiet life. But when Laura, a muta
nt child pursued by scientists, comes to him for help, he must get her to safety.</di
v></div></div></div>
```

Getting the title

In [45]:
```
first_movie.a
```

Out[45]:
```
<a aria-label="View title page for Logan" class="ipc-lockup-overlay ipc-focusable" hr
ef="/title/tt3315342/?ref_=sr_i_1"><div class="ipc-lockup-overlay__screen"></div></a>
```

In [46]:
```
first_movie.h3
```

Out[46]:
```
<h3 class="ipc-title__text">1. Logan</h3>
```

In [48]:
```
first_movie.h3.a
```

In [49]:
```
first_name = first_movie.h3.a.text
first_name
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[49], line 1
----> 1 first_name = first_movie.h3.a.text
      2 first_name

AttributeError: 'NoneType' object has no attribute 'text'
```

In [53]:
```
# alternate way to get the title
first_movie.h3.text[3:]
```

Out[53]:
```
'Logan'
```

Getting the year of release

In [54]:
```
first_year = first_movie.h3.find('span', class_ = 'lister-item-year text-muted unbold'
first_year
```

In [55]:
```
first_year = first_year.text
first_year
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[55], line 1
----> 1 first_year = first_year.text
      2 first_year

AttributeError: 'NoneType' object has no attribute 'text'
```

In [81]:
```
# alternate way to get the year released
first_movie_metadata1 = html_soup.find('div', class_ = 'sc-b0691f29-7 hrgukm dli-title
print(first_movie_metadata)
```

```
<div class="sc-b0691f29-7 hrgukm dli-title-metadata"><span class="sc-b0691f29-8 ilsLE
X dli-title-metadata-item">2017</span><span class="sc-b0691f29-8 ilsLEX dli-title-met
adata-item">2h 17m</span><span class="sc-b0691f29-8 ilsLEX dli-title-metadata-item">R
-16</span></div>
```

In [82]:
```
first_year = first_movie_metadata1.contents[0].text
first_year
```

Out[82]:
```
'2017'
```

Getting the imdb rating

In [75]:
```
first_movie.strong
```

In [76]:
```
first_imdb = float(first_movie.strong.text)
first_imdb
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[76], line 1
----> 1 first_imdb = float(first_movie.strong.text)
      2 first_imdb

AttributeError: 'NoneType' object has no attribute 'text'
```

In [100…
```
# alternate way to get imdb rating
first_movie_metadata2 = html_soup.find('div', class_ = 'sc-e2dbc1a3-0 ajrIH sc-b0691f2
print(first_movie_metadata2)
```

```
<div class="sc-e2dbc1a3-0 ajrIH sc-b0691f29-2 bhhtyj dli-ratings-container" data-test
id="ratingGroup--container"><span aria-label="IMDb rating: 8.1" class="ipc-rating-sta
r ipc-rating-star--base ipc-rating-star--imdb ratingGroup--imdb-rating" data-testid
="ratingGroup--imdb-rating"><svg class="ipc-icon ipc-icon--star-inline" fill="current
Color" height="24" role="presentation" viewbox="0 0 24 24" width="24" xmlns="http://w
ww.w3.org/2000/svg"><path d="M12 20.1l5.82 3.682c1.066.675 2.37-.322 2.09-1.584l-1.54
3-6.926 5.146-4.667c.94-.85.435-2.465-.799-2.567l-6.773-.602L13.29.89a1.38 1.38 0 0 0
-2.581 0l-2.65 6.53-6.774.602C.052 8.126-.453 9.74.486 10.59l5.147 4.666-1.542 6.926c
-.28 1.262 1.023 2.26 2.09 1.585L12 20.099z"></path></svg>8.1<span class="ipc-rating-
star--voteCount"> (<!-- -->826K<!-- -->)</span></span><button aria-label="Rate Logan"
class="ipc-rate-button sc-e2dbc1a3-1 jboOQc ratingGroup--user-rating ipc-rate-button-
-unrated ipc-rate-button--base" data-testid="rate-button"><span class="ipc-rating-sta
r ipc-rating-star--base ipc-rating-star--rate"><svg class="ipc-icon ipc-icon--star-bo
rder-inline" fill="currentColor" height="24" role="presentation" viewbox="0 0 24 24"
width="24" xmlns="http://www.w3.org/2000/svg"><path d="M22.724 8.217l-6.786-.587-2.65
-6.22c-.477-1.133-2.103-1.133-2.58 0l-2.65 6.234-6.772.573c-1.234.098-1.739 1.636-.8
2.446l5.146 4.446-1.542 6.598c-.28 1.202 1.023 2.153 2.09 1.51l5.818-3.495 5.819 3.50
9c1.065.643 2.37-.308 2.089-1.51l-1.542-6.612 5.145-4.446c.94-.81.45-2.348-.785-2.446
zm-10.726 8.89l-5.272 3.174 1.402-5.983-4.655-4.026 6.141-.531 2.384-5.634 2.398 5.64
8 6.14.531-4.654 4.026 1.402 5.983-5.286-3.187z"></path></svg><span class="ipc-rating
-star--rate">Rate</span></span></button></div>
```

In [106…
```
first_imdb = first_movie_metadata2.contents[0].text[:3]
print(first_imdb)
```

```
8.1
```

Getting the metascore

In [77]:
```
first_mscore = first_movie.find('span', class_ = 'metascore favorable')
first_mscore = int(first_mscore.text)
print(first_mscore)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[77], line 2
      1 first_mscore = first_movie.find('span', class_ = 'metascore favorable')
----> 2 first_mscore = int(first_mscore.text)
      3 print(first_mscore)

AttributeError: 'NoneType' object has no attribute 'text'
```

In [107…
```
# alternate way to get metascore
first_movie_metadata3 = html_soup.find('span', class_ = 'sc-b0901df4-0 bcQdDJ metacrit
print(first_movie_metadata3)
```

```
<span class="sc-b0901df4-0 bcQdDJ metacritic-score-box" style="background-color:#54A7
2A">77</span>
```

In [108…
```
first_mscore = first_movie_metadata3.text
print(first_mscore)
```

```
77
```

Getting the number of votes

In [78]:
```
first_votes = first_movie.find('span', attrs = {'name':'nv'})
first_votes
```

In [79]:
```
first_votes['data-value']
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[79], line 1
----> 1 first_votes['data-value']

TypeError: 'NoneType' object is not subscriptable
```

In [80]:
```
first_votes = int(first_votes['data-value'])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[80], line 1
----> 1 first_votes = int(first_votes['data-value'])

TypeError: 'NoneType' object is not subscriptable
```

In [112…
```
# alternate way to get votes
first_votes = first_movie_metadata2.contents[0].text[5:-1]
print(first_votes)
```

826K

In [113…
```
# Lists to store the scraped data in
names = []
years = []
imdb_ratings = []
metascores = []
votes = []
# Extract data from individual movie container
for container in movie_containers:
# If the movie has Metascore, then extract:
    if container.find('div', class_ = 'ratings-metascore') is not None:
# The name
        name = container.h3.a.text
        names.append(name)
# The year
        year = container.h3.find('span', class_ = 'lister-item-year').text
        years.append(year)
# The IMDB rating
        imdb = float(container.strong.text)
        imdb_ratings.append(imdb)
# The Metascore
        m_score = container.find('span', class_ = 'metascore').text
        metascores.append(int(m_score))
# The number of votes
        vote = container.find('span', attrs = {'name':'nv'})['data-value']
        votes.append(int(vote))
```

In [126…
```
# alternate function
names = []
years = []
imdb_ratings = []
metascores = []
votes = []

for container in movie_containers:
    if container.find('span', class_ = 'sc-b0901df4-0 bcQdDJ metacritic-score-box') is
        name = container.h3.text[3:]
        names.append(name)

        temp1 = container.find('div', class_ = 'sc-b0691f29-7 hrgukm dli-title-metadat
        year = temp1.contents[0].text
        years.append(year)

        temp2 = container.find('div', class_ = 'sc-e2dbc1a3-0 ajrIH sc-b0691f29-2 bhht
        imdb = float(temp2.contents[0].text[:3])
        imdb_ratings.append(imdb)

        temp3 = container.find('span', class_ = 'sc-b0901df4-0 bcQdDJ metacritic-score
        m_score = temp3.text
        metascores.append(int(m_score))

        temp4 = temp2.contents[0].text[5:-1] # assuming all entries have 000K format
        vote = int(temp4[:-1]) * 1000
        votes.append(vote)
```

In [127…

```
import pandas as pd
test_df = pd.DataFrame({'movie': names,
'year': years,
'imdb': imdb_ratings,
'metascore': metascores,
'votes': votes
})
print(test_df.info())
test_df
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41 entries, 0 to 40
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   movie      41 non-null     object
 1   year       41 non-null     object
 2   imdb       41 non-null     float64
 3   metascore  41 non-null     int64
 4   votes      41 non-null     int64
dtypes: float64(1), int64(2), object(2)
memory usage: 1.7+ KB
None
```

Out[127]:

	movie	year	imdb	metascore	votes
0	Logan	2017	8.1	77	826000
1	Thor: Ragnarok	2017	7.9	74	812000
2	Guardians of the Galaxy Vol. 2	2017	7.6	67	756000
3	Dunkirk	2017	7.8	94	736000
4	Spider-Man: Homecoming	2017	7.4	73	716000
5	Wonder Woman	2017	7.3	76	698000
6	Get Out	2017	7.8	85	691000
7	Star Wars: Episode VIII - The Last Jedi	2017	6.9	84	669000
8	Blade Runner 2049	2017	8.0	81	658000
9	Baby Driver	2017	7.5	86	605000
10	It	2017	7.3	69	603000
11	Coco	2017	8.4	81	586000
12	Three Billboards Outside Ebbing, Missouri	2017	8.1	88	552000
13	John Wick: Chapter 2	2017	7.4	75	509000
14	Justice League	2017	6.1	45	477000
15	The Shape of Water	2017	7.3	87	446000
16	Jumanji: Welcome to the Jungle	2017	6.9	58	435000
17	Kingsman: The Golden Circle	2017	6.7	44	361000
18	Kong: Skull Island	2017	6.7	62	345000
19	Pirates of the Caribbean: Salazar's Revenge	2017	6.5	39	343000
20	Beauty and the Beast	2017	7.1	65	333000
21	Lady Bird	2017	7.4	93	326000
22	Call Me by Your Name	2017	7.8	94	312000
23	The Greatest Showman	2017	7.5	48	310000
24	Alien: Covenant	2017	6.4	65	302000
25	Murder on the Orient Express	2017	6.5	52	295000
26	War for the Planet of the Apes	2017	7.4	82	280000
27	Wind River	2017	7.7	73	279000
28	Fast & Furious 8	2017	6.6	56	253000
29	Life	2017	6.6	54	252000
30	Mother!	2017	6.6	76	249000
31	The Hitman's Bodyguard	2017	6.9	47	246000
32	I, Tonya	2017	7.5	77	241000
33	King Arthur: Legend of the Sword	2017	6.7	41	231000
34	Ghost in the Shell	2017	6.3	52	227000
35	Darkest Hour	2017	7.4	75	220000
36	American Made	2017	7.1	65	207000
37	Atomic Blonde	2017	6.7	63	206000
38	The Mummy	2017	5.4	34	205000
39	Baywatch	2017	5.5	37	201000
40	Bright	2017	6.3	29	201000

In [124…

```
from time import time
from time import sleep
from random import randint
```

```
from IPython.core.display import clear_output
pages = [ '1','2','3','4','5']
years_url = [ '2017', '2018', '2019', '2020']
# Redeclaring the lists to store data in
names = []
years = []
imdb_ratings = []
metascores = []
votes = []
# Preparing the monitoring of the loop
start_time = time()
requests = 0
# For every year in the interval 2000-2017
for year_url in years_url:
    # For every page in the interval 1-4
    for page in pages:
        # Make a get request
        response = get('https://www.imdb.com/search/title?release_date=' + year_url +
        '&sort=num_votes,desc&page=' + page, headers = headers)
        # Pause the loop
        sleep(randint(8,15))
        # Monitor the requests
        requests += 1
        elapsed_time = time() - start_time
        print('Request:{}; Frequency: {} requests/s'.format(requests, requests/elapsed
        clear_output(wait = True)
        # Throw a warning for non-200 status codes
        if response.status_code != 200:
            warn('Request: {}; Status code: {}'.format(requests, response.status_code)
        # Break the loop if the number of requests is greater than expected
        if requests > 72:
            warn('Number of requests was greater than expected.')
            break
        # Parse the content of the request with BeautifulSoup
        page_html = BeautifulSoup(response.text, 'html.parser')
        # Select all the 50 movie containers from a single page
        mv_containers = page_html.find_all('div', class_ = 'lister-item mode-advanced'
        # For every movie of these 50
        for container in mv_containers:
            # If the movie has a Metascore, then:
            if container.find('div', class_ = 'ratings-metascore') is not None:
                # Scrape the name
                name = container.h3.a.text
                names.append(name)
                # Scrape the year
                year = container.h3.find('span', class_ = 'lister-item-year').text
                years.append(year)
                # Scrape the IMDB rating
                imdb = float(container.strong.text)
                imdb_ratings.append(imdb)
                # Scrape the Metascore
                m_score = container.find('span', class_ = 'metascore').text
                metascores.append(int(m_score))
                # Scrape the number of votes
                vote = container.find('span', attrs = {'name':'nv'})['data-value']
                votes.append(int(vote))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[124], line 32
     30 # Throw a warning for non-200 status codes
     31 if response.status_code != 200:
---> 32     warn('Request: {}; Status code: {}'.format(requests, response.status_cod
e))
     33 # Break the loop if the number of requests is greater than expected
     34 if requests > 72:

NameError: name 'warn' is not defined
```

The code above did not work as there was an error earlier that states that IPython.core.display is deprecated

In [125…

```
movie_ratings = pd.DataFrame({'movie': names,
'year': years,
'imdb': imdb_ratings,
'metascore': metascores,
'votes': votes
```

```
})
print(movie_ratings.info())
movie_ratings.head(10)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 0 entries
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   movie       0 non-null      float64
 1   year        0 non-null      float64
 2   imdb        0 non-null      float64
 3   metascore   0 non-null      float64
 4   votes       0 non-null      float64
dtypes: float64(5)
memory usage: 132.0 bytes
None
```

Out[125]:

movie	year	imdb	metascore	votes

I used the earlier dataframe instead

In [129…
```
test_df.tail(10)
```

Out[129]:

	movie	year	imdb	metascore	votes
31	The Hitman's Bodyguard	2017	6.9	47	246000
32	I, Tonya	2017	7.5	77	241000
33	King Arthur: Legend of the Sword	2017	6.7	41	231000
34	Ghost in the Shell	2017	6.3	52	227000
35	Darkest Hour	2017	7.4	75	220000
36	American Made	2017	7.1	65	207000
37	Atomic Blonde	2017	6.7	63	206000
38	The Mummy	2017	5.4	34	205000
39	Baywatch	2017	5.5	37	201000
40	Bright	2017	6.3	29	201000

In [130…
```
test_df.to_csv('movie_ratings.csv')
```

In [131…
```
test_df['year'].unique()
```

Out[131]:
```
array(['2017'], dtype=object)
```

In [133…
```
test_df.dtypes
```

Out[133]:
```
movie         object
year          object
imdb         float64
metascore      int64
votes          int64
dtype: object
```

In [134…
```
test_df['year'] = (test_df.year.apply(lambda x:x.replace('(I)','')))
```

In [135…
```
test_df['year'].unique()
```

Out[135]:
```
array(['2017'], dtype=object)
```

In [136…
```
test_df['year'] = (test_df.year.apply(lambda x:x.replace('(II)','')))
```

In [137…
```
test_df['year'].unique()
```

Out[137]:
```
array(['2017'], dtype=object)
```

In [138…
```
test_df['year'] = (test_df.year.apply(lambda x:x.replace('(','')))
```

In [139…
```
test_df['year'].unique()
```

Out[139]:
```
array(['2017'], dtype=object)
```

In [140…
```
test_df['year'] = (test_df.year.apply(lambda x:x.replace(')','')))
```

In [141…
```
test_df['year'].unique()
```

Out[141]:
```
array(['2017'], dtype=object)
```

In [142…
```
test_df['year'] = test_df['year'].astype(int)
```

In [143…
```
test_df['year'].unique()
```

Out[143]:
```
array([2017])
```

In [144…
```
test_df.dtypes
```

Out[144]:
```
movie         object
year           int32
imdb         float64
metascore      int64
votes          int64
dtype: object
```

In [145…
```
test_df.head(10)
```

Out[145]:

	movie	year	imdb	metascore	votes
0	Logan	2017	8.1	77	826000
1	Thor: Ragnarok	2017	7.9	74	812000
2	Guardians of the Galaxy Vol. 2	2017	7.6	67	756000
3	Dunkirk	2017	7.8	94	736000
4	Spider-Man: Homecoming	2017	7.4	73	716000
5	Wonder Woman	2017	7.3	76	698000
6	Get Out	2017	7.8	85	691000
7	Star Wars: Episode VIII - The Last Jedi	2017	6.9	84	669000
8	Blade Runner 2049	2017	8.0	81	658000
9	Baby Driver	2017	7.5	86	605000

In [146…
```
test_df.tail(10)
```

Out[146]:

	movie	year	imdb	metascore	votes
31	The Hitman's Bodyguard	2017	6.9	47	246000
32	I, Tonya	2017	7.5	77	241000
33	King Arthur: Legend of the Sword	2017	6.7	41	231000
34	Ghost in the Shell	2017	6.3	52	227000
35	Darkest Hour	2017	7.4	75	220000
36	American Made	2017	7.1	65	207000
37	Atomic Blonde	2017	6.7	63	206000
38	The Mummy	2017	5.4	34	205000
39	Baywatch	2017	5.5	37	201000
40	Bright	2017	6.3	29	201000

In [147…
```
test_df
```

Out[147]:

	movie	year	imdb	metascore	votes
0	Logan	2017	8.1	77	826000
1	Thor: Ragnarok	2017	7.9	74	812000
2	Guardians of the Galaxy Vol. 2	2017	7.6	67	756000
3	Dunkirk	2017	7.8	94	736000
4	Spider-Man: Homecoming	2017	7.4	73	716000
5	Wonder Woman	2017	7.3	76	698000
6	Get Out	2017	7.8	85	691000
7	Star Wars: Episode VIII - The Last Jedi	2017	6.9	84	669000
8	Blade Runner 2049	2017	8.0	81	658000
9	Baby Driver	2017	7.5	86	605000
10	It	2017	7.3	69	603000
11	Coco	2017	8.4	81	586000
12	Three Billboards Outside Ebbing, Missouri	2017	8.1	88	552000
13	John Wick: Chapter 2	2017	7.4	75	509000
14	Justice League	2017	6.1	45	477000
15	The Shape of Water	2017	7.3	87	446000
16	Jumanji: Welcome to the Jungle	2017	6.9	58	435000
17	Kingsman: The Golden Circle	2017	6.7	44	361000
18	Kong: Skull Island	2017	6.7	62	345000
19	Pirates of the Caribbean: Salazar's Revenge	2017	6.5	39	343000
20	Beauty and the Beast	2017	7.1	65	333000
21	Lady Bird	2017	7.4	93	326000
22	Call Me by Your Name	2017	7.8	94	312000
23	The Greatest Showman	2017	7.5	48	310000
24	Alien: Covenant	2017	6.4	65	302000
25	Murder on the Orient Express	2017	6.5	52	295000
26	War for the Planet of the Apes	2017	7.4	82	280000
27	Wind River	2017	7.7	73	279000
28	Fast & Furious 8	2017	6.6	56	253000
29	Life	2017	6.6	54	252000
30	Mother!	2017	6.6	76	249000
31	The Hitman's Bodyguard	2017	6.9	47	246000
32	I, Tonya	2017	7.5	77	241000
33	King Arthur: Legend of the Sword	2017	6.7	41	231000
34	Ghost in the Shell	2017	6.3	52	227000
35	Darkest Hour	2017	7.4	75	220000
36	American Made	2017	7.1	65	207000
37	Atomic Blonde	2017	6.7	63	206000
38	The Mummy	2017	5.4	34	205000
39	Baywatch	2017	5.5	37	201000
40	Bright	2017	6.3	29	201000

Performing Webscraping to Website of choice

website = https://en.wikipedia.org/wiki/List_of_accidents_and_disasters_by_death_toll

In [1]:
```
from requests import get
url = 'https://en.wikipedia.org/wiki/List_of_accidents_and_disasters_by_death_toll'
response = get(url)
print(response.text[:500])
```

```
<!DOCTYPE html>
<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-lan
guage-in-main-page-header-disabled vector-feature-sticky-header-disabled vector-featu
re-page-tools-pinned-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-m
ain-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-feature-lim
ited-width-content-enabled vector-feature-custom-font-size-clientpref-0 vector-featur
e-client-preferences-disabled vector-feature-client-prefs-p
```

In [2]:
```
from bs4 import BeautifulSoup
html_soup = BeautifulSoup(response.text, 'html.parser')
headers = {'Accept-Language': 'en-US,en;q=0.8'}
type(html_soup)
```

Out[2]:
```
bs4.BeautifulSoup
```

In [8]:
```
tables = html_soup.find_all("table", class_ = "wikitable sortable") # gets all tables
print(tables)
```

[<table class="wikitable sortable">
<tbody><tr>
<th data-sort-type="number">Deaths
</th>
<th>Date
</th>
<th>Incident
</th></tr>
<tr>
<td>20,000</td>
<td>30 May
1626
</td>
<td>Wanggongc
hang Explosion in Beijing, China in the Wanggongchang Gunpowder Factory destroys
part of the city and kills 20,000 people<sup class="reference" id="cite_ref-1"><a hre
f="#cite_note-1">[1]</sup>
</td></tr>
<tr>
<td>3,000</td>
<td>18 Aug
ust 1769
</td>
<td>A lightning bolt caused the <a class="mw-redirect" href="/wiki/Brescia_Explosion"
title="Brescia Explosion">Brescia Explosion of a gunpowder depot in <a href="/wik
i/Brescia" title="Brescia">Brescia (Italy), destroying one-sixth of the city<sup
class="reference" id="cite_ref-2">[2]</sup><sup class="ref
erence" id="cite_ref-3">[3]</sup>
</td></tr>
<tr>
<td data-sort-value="3,000">3,000?</td>
<td>1 Nove
mber 1948
</td>
<td>Boiler and ammunition explosion aboard an unidentified merchant ship evacuating t
roops of the Re
public of China Army from Yingkou, Ch
ina for Taiwan in early November 1948.<sup
class="reference" id="cite_ref-4">[4]</sup><sup class="nop
rint Inline-Template" style="white-space:nowrap;">[<a class="mw-redirect" href="/w
iki/Wikipedia:Disputed_statement" title="Wikipedia:Disputed statement"><span title="T
his claim has reliable sources with contradicting facts (August 2018)">disputed – <a href="/wiki/Talk:List_of_accidents_and_disasters_b
y_death_toll" title="Talk:List of accidents and disasters by death toll">discuss
]</sup> Other sources suggest this figure is inaccurate.<sup class="refere
nce" id="cite_ref-Williams2012_5-0">[5]</sup>
</td></tr>
<tr>
<td>1,400–2,280</td>
<td>6 Marc
h 1862
</td>
<td>Ammunition warehouse explodes and kills almost all of Oaxaca brigade, in <a href
="/wiki/Ciudad_Serd%C3%A1n" title="Ciudad Serdán">San Andrés Chalchicomula, <a hr
ef="/wiki/Mexico" title="Mexico">Mexico, during the first days of <a href="/wiki/
Second_French_intervention_in_Mexico" title="Second French intervention in Mexico">Se
cond French intervention in Mexico<sup class="reference" id="cite_ref-Galindo_6-
0">[6]</sup>
</td></tr>
<tr>
<td>1,950</td>
<td>6 Dece
mber 1917
</td>
<td>Halifax Explosion
in Nova Scotia, Canada<sup class="reference" id="cite_ref-7">
[7]</sup>
</td></tr>
<tr>
<td>1,500</td>
<td>5 June
1941
</td>
<td><a href="/wiki/Smederevo_Fortress_explosion" title="Smederevo Fortress explosio
n">Smederevo Fortress explosion of stockpiled ammunition at <a href="/wiki/Smeder
evo_Fortress" title="Smederevo Fortress">Smederevo Fortress near <a href="/wiki/B
elgrade" title="Belgrade">Belgrade in the <a href="/wiki/Territory_of_the_Militar
y_Commander_in_Serbia" title="Territory of the Military Commander in Serbia">Territor

```
<td>1948 London smog<sup class="reference" id="cite_ref-Smog_report_153-1"><a href="#
cite_note-Smog_report-153">[153]</a></sup>
</td></tr>
<tr>
<td>300–405</td>
<td data-sort-value="1 January 1963">January–February 1963
</td>
<td>1963 <a href="/wiki/New_York_City" title="New York City">New York City</a> smog
</td></tr>
<tr>
<td>300–400</td>
<td data-sort-value="1 December 1962">December 1962
</td>
<td><a href="/wiki/1962_London_smog" title="1962 London smog">1962 London smog</a><su
p class="reference" id="cite_ref-Smog_notes_151-1"><a href="#cite_note-Smog_notes-15
1">[151]</a></sup>
</td></tr>
<tr>
<td>220–240</td>
<td data-sort-value="1 November 1953">November 1953
</td>
<td>1953 New York City smog
</td></tr>
<tr>
<td>168</td>
<td data-sort-value="12 December 1991">23–25 November 1966
</td>
<td><a href="/wiki/1966_New_York_City_smog" title="1966 New York City smog">1966 New
York City smog</a>
</td></tr>
<tr>
<td>160</td>
<td data-sort-value="12 December 1991">12–15 December 1991
</td>
<td>Smog (London, 12–15 December 1991)<sup class="reference" id="cite_ref-154"><a hre
f="#cite_note-154">[154]</a></sup>
</td></tr></tbody></table>]
```

In [30]:

```
headers = ['Deaths', 'Date', 'Incident'] # I decided to only get the date, deaths, and
temp_data = [] # stores every information taken in rows

for table in tables:
    rows = table.find_all('tr')
    for row in rows[1:]:
        data = [data.text for data in row.find_all('td')[:3]] # gets the data per row
        temp_data.append(data)
print(temp_data)
```

[['20,000', '30 May 1626\n', 'Wanggongchang Explosion in Beijing, China in the Wanggo
ngchang Gunpowder Factory destroys part of the city and kills 20,000 people[1]\n'],
['3,000', '18 August 1769\n', 'A lightning bolt caused the Brescia Explosion of a gun
powder depot in Brescia (Italy), destroying one-sixth of the city[2][3]\n'], ['3,00
0?', '1 November 1948\n', 'Boiler and ammunition explosion aboard an unidentified mer
chant ship evacuating troops of the Republic of China Army from Yingkou, China for Ta
iwan in early November 1948.[4][disputed – discuss] Other sources suggest this figur
e is inaccurate.[5]\n'], ['1,400–2,280', '6 March 1862\n', 'Ammunition warehouse expl
odes and kills almost all of Oaxaca brigade, in San Andrés Chalchicomula, Mexico, dur
ing the first days of Second French intervention in Mexico[6]\n'], ['1,950', '6 Decem
ber 1917\n', 'Halifax Explosion in Nova Scotia, Canada[7]\n'], ['1,500', '5 June 1941
\n', 'Smederevo Fortress explosion of stockpiled ammunition at Smederevo Fortress nea
r Belgrade in the Territory of the Military Commander in Serbia[8]\n'], ['256–1,500
+', '16 September 1732\n', 'Military warehouse explodes and kills up to two-thirds of
the population of Campo Maior district of Portalegre, Portugal[9][10]\n'], ['1,300+',
'7 August 1956\n', 'Ammunition trucks explode near a railway station in Cali, Colombi
a[11]\n'], ['1,200', '16 October 1926\n', 'Explosion of ammunition on the Chinese tro
opship Kuang Yuang, near Kiukiang, China[12]\n'], ['1,121', '8 June 1943\n', 'Japanes
e battleship Mutsu, at Hashirajima harbor, 1943 due to magazine explosion; ship sank
on 9 June\n'], ['1,100', '27 January 2002\n', 'Lagos Armoury Explosion, in Lagos, Nig
eria; many deaths were from drowning during the resulting panic\n'], ['1,082', '18 Oc
tober 1998\n', 'Jesse pipeline explosion, near Lagos, Nigeria\n'], ['1,007', '18 Nove
mber 1918\n', 'Ammunition transporter explosion in Hamont, Belgium\n'], ['1,000+', '3
1 August 1794\n', 'Explosion of a gunpowder factory in Grenelle, France\n'], ['843',
'9 July 1917\n', 'Propellant explosion of the British dreadnought battleship HMS\xa0V
anguard\n'], ['800', '14 April 1944\n', 'Bombay Docks Explosion in Bombay, Bombay Pre
sidency, British India\n'], ['738', '26 November 1914\n', 'British pre-dreadnought ba
ttleship HMS\xa0Bulwark destroyed due to a magazine explosion\n'], ['700', '17 August
1989\n', 'Iraqi military plant with facilities explode at Al Hillah, Babil, Iraq\n'],
['621', '27 February 1925\n', 'Explosion of a dynamite depot in Brazil.[13]\n'], ['60
0+', '28 March 1943\n', 'Explosion of the Caterina Costa, at port of Naples; over 300
0 were also injured\n'], ['590', '3 November 1893\n', 'Explosion of dynamite cargo on
the steamship Cabo Machichaco, in at the port of Santander, Spain, with more than 2,0
00 injured.[14]\n'], ['581', '16 April 1947\n', 'Texas City disaster in the Port of T
exas City; over 5,000 were also injured.\n'], ['575', '4 June 1989\n', 'Ufa train dis
aster in Ufa, Soviet Union.\n'], ['565', '21 September 1921\n', 'Oppau explosion at a
BASF plant in Germany; possibly as many as 1,500 were killed\n'], ['542', '9 April 19
45\n', 'SS Charles Henderson ammunition explosion at Bari, Italy; 1,800 injured\n'],
['508', '25 February 1984\n', 'A gasoline pipeline exploded in the favela of Vila Sao
Jose, Cubatão, Brazil[15][16]\n'], ['500+', '19 November 1984\n', 'San Juanico disast
er, in Mexico City, Mexico\n'], ['372', '10 November 1944\n', 'Ammunition ship USS\xa
0Mount Hood exploded at Seeadler Harbor, killing 372 and injuring 371\n'], ['370+',
'27 July 1816\n', 'Powder magazine in Negro Fort in Spanish Florida exploded during b
attle with United States forces.\n'], ['339', '11 September 1905\n', 'Japanese battle
ship Mikasa explosion of magazine (artillery) while at port\n'], ['322', '17 July 194
4\n', 'Port Chicago disaster at Port Chicago, California, United States\n'], ['300 (e
stimate)', '25 September 1911\n', 'An explosion occurred on the French battleship Lib
erté\n'], ['300+', '4 March 2012\n', 'Brazzaville arms dump blasts at Brazzaville, Re
public of the Congo\n'], ['296+', '18 March 1937\n', 'New London School explosion at
New London, Texas, United States\n'], ['256', '18 July 1806\n', 'Gunpowder magazine e
xplosion, Birgu, Malta\n'], ['256', '4 June 2015\n', 'Gas station explosion in Accra,
Ghana\n'], ['235+\n', '29 December 2001\n', 'Fireworks stand explosion in Lima, Peru.
235 bodies recovered, 144+ taken to hospital due to burns.[17]\n'], ['234', '23 Decem
ber 2003\n', 'PetroChina Chuandongbei natural gas field explosion, Guoqiao, Chongqin
g, China\n'], ['233', '20 October 1916\n', 'Russian battleship Imperatritsa Mariya ma
gazine explosion\n'], ['230+', '2 July 2010\n', '2010 South Kivu fuel tank explosion
in the Democratic Republic of the Congo\n'], ['220', '11 December 1944\n', 'Gunpowder
explosion on the train doing Japanese military and civilian transportation, Itoman, O
kinawa, Japan.[18][better\xa0source\xa0needed]\n'], ['220', '25 September 2023\n', 'F
illing station explosion in Berkadzor, Nagorno-Karabakh[19]\n'], ['219', '25 June 201
7\n', 'Bahawalpur explosion, an oil truck exploded in Bahawalpur, Pakistan.\n'], ['21
8', '4 August 2020\n', '2020 Beirut explosion, Beirut, Lebanon\n'], ['217', '11 July
1978\n', 'Los Alfaques disaster, in Sant Carles de la Ràpita, Spain\n'], ['207', '28
July 1948\n', '1948 BASF tank car explosion, Ludwigshafen am Rhein, Germany; 3,818 we
re also injured[20]\n'], ['206', '22 April 1992\n', 'Gas explosions in Guadalajara, J
alisco, Mexico\n'], ['206', '17 September 2015\n', '2015 tanker explosion in Juba, So
uth Sudan[21]\n'], ['200+', '6 June 1991\n', 'Gotera ammunition dump explosion in Add
is Ababa, Ethiopia\n'], ['3,787–16,000', '2–3 December 1984\n', 'Bhopal disaster\n'],
['1,549', '26 April 1942\n', 'Benxihu Colliery explosion\n'], ['1,134', '24 April 201
3\n', 'Rana Plaza collapse[22]\n'], ['1,099', '10 March 1906\n', 'Courrières mine dis
aster\n'], ['687', '15 December 1914\n', 'Mitsubishi Hōjō coal mine disaster\n'], ['6
82', '9 May 1960\n', 'Laobaidong colliery coal dust explosion\n'], ['581', '16 April
1947\n', 'Texas City disaster\n'], ['512', '28 August 1899\n', 'Sumitomo Besshi bronz
e mine area, landslide with debris flow disaster\n'], ['500+', '19 November 1984\n',
'San Juanico Disaster[23]\n'], ['476–1,000', '1931\n', 'Hawks Nest Tunnel Disaster
\n'], ['458', '9 November 1963\n', 'Mitsui Miike Coal Mine disaster[24]\n'], ['439',
'14 October 1913\n', 'Senghenydd Colliery Disaster\n'], ['437', '21 January 1960\n',
'Coalbrook mining disaster\n'], ['426', '6 June 1972\n', 'Wankie coal mine disaster

amia ferry incident (Meghna River, Bangladesh)\n'], ['500\n', '29 January 1986', 'Atl as Star (Dhaleswar River, Munshiganj, Bangladesh)[123][124]\n'], ['900–1,700+\n', '26 December 2004\n', '2004 Sri Lanka tsunami train wreck\n'], ['600–1,000\n', '13 Januar y 1917\n', 'Ciurea rail disaster\n'], ['675–800\n', '12 December 1917\n', 'Saint-Mich el-de-Maurienne derailment\n'], ['235–800\n', '6 June 1981\n', 'Bihar train derailmen t[125][126]\n'], ['600–700\n', '24 January 1944\n', 'Vereshchyovka train disaster \n'], ['600+\n', '22 January 1915\n', 'Guadalajara train disaster[127]\n'], ['575\n', '4 June 1989\n', 'Ufa train disaster\n'], ['521–600+\n', '3 March 1944\n', 'Balvano t rain disaster\n'], ['428\n', '13 January 1985\n', 'Awash rail disaster\n'], ['383+ \n', '20 February 2002\n', 'Al Ayatt train disaster\n'], ['370\n', '27 December 1944 \n', 'Stará Kremnička derailment[128]\n'], ['360\n', '9 July 1981\n', '1981 Chengdu-K unming rail crash\n'], ['358\n', '20 August 1995\n', 'Firozabad rail disaster\n'], ['338\n', '29 April 1997\n', 'Rongjiawan train disaster\xa0[zh]\n'], ['320\n', '18 Fe bruary 2004\n', 'Nishapur train disaster\n'], ['307\n', '4 January 1990\n', 'Sukkur r ail disaster\n'], ['300\n', '7 January 1918\n', 'Changsha rail disaster\xa0[de]\n'], ['300~\n', '21 June 1915\n', 'Monte Morelas[129]\n'], ['300\n', '22 September 1994 \n', 'Tolunda rail disaster\xa0[de][130]\n'], ['300\n', '29 September 1957\n', 'Gamba r train crash[131]\n'], ['296\n', '2 June 2023\n', 'Odisha train collision[132]\n'], ['289\n', '28 October 1995\n', 'Baku Metro fire\n'], ['285\n', '2 August 1999\n', 'Ga isal train disaster\n'], ['281\n', '24 June 2002\n', 'Igandu train disaster\n'], ['27 8\n', '22 December 1939\n', 'Genthin rail disaster\n'], ['248\n', '14 March 1926\n', 'El Virilla train accident[133]\n'], ['230\n', '23 December 1933\n', 'Lagny-Pomponne Railroad Disaster\n'], ['226\n', '22 May 1915\n', 'Quintinshill rail crash\n'], ['212 \n', '26 November 1998\n', 'Khanna rail disaster\n'], ['208\n', '4 October 1972\n', 'Saltillo, Coahuila\n'], ['207\n', '28 July 1948\n', 'BASF tank car explosion\xa0[de] \n'], ['160–200\n', '17 March 1982\n', 'Bau Ca train wreck\xa0[vi]\n'], ['230\n', '2 July 2010\n', 'Sange road tanker explosion[134]\n'], ['219\n', '25 June 2017\n', 'Bah awalpur explosion[135]\n'], ['217\n', '11 July 1978\n', 'Los Alfaques disaster[136] \n'], ['207\n', '15 February 1991\n', 'Thung Maphrao truck explosion[137]\n'], ['206 \n', '17 September 2015\n', '2015 Juba tanker explosion [21]\n'], ['168–3,000\n', '3 November 1982\n', 'Salang Tunnel fire[136][138]\n'], ['154\n', '5 November 2021\n', 'Freetown fuel tanker explosion\n'], ['125\n', '6 December 1965\n', 'Sotouba truck co llision[136]\n'], ['121\n', '12 July 2012\n', 'Okobie road tanker explosion[139]\n'], ['115\n', '17 November 2016\n', 'Caphiridzange tanker explosion [140]\n'], ['113\n', '31 January 2009\n', 'Molo road tanker fire\n'], ['110\n', '12 March 1995\n', 'Tamil Nadu truck collision[141]\n'], ['104\n', '18 August 1968\n', 'Gero bus crash[136] \n'], ['100–200\n', '5 November 2000\n', 'Ibadan road tanker explosion[142]\n'], ['10 0+\n', '10 August 2019\n', 'Morogoro fuel tanker explosion[143]\n'], ['98\n', '28 Mar ch 2007\n', 'Kagarko truck explosion\n'], ['96\n', '12 July 1992\n', 'Machakos bus cr ash[144]\n'], ['94\n', '8 June 1999\n', 'Karnataka bus crash[145]\n'], ['90\n', '15 N ovember 1976\n', 'Mafeteng bus crash\n'], ['90\n', '29 June 1980\n', 'Kashmir bus cra sh[136]\n'], ['90\n', '24 June 2004\n', 'Zahedan bus crash[146]\n'], ['90+\n', '14 De cember 2021\n', 'Cap-Haïtien tanker explosion[147]\n'], ['81–115\n', '6 January 1967 \n', 'Cavite bus crash[148]\n'], ['54-300\n', '24 October 1960\n', 'Nedelin catastrop he, Baikonur Cosmodrome, Kazakh SSR, Soviet Union - occurred on ground before launch \n'], ['48\n', '18 March 1980\n', '1980 Plesetsk launch pad disaster, Plesetsk Cosmod rome, Mirny, Arkhangelsk Oblast, Soviet Union - occurred on ground before launch\n'], ['21\n', '22 August 2003\n', '2003 Alcântara VLS accident, Alcântara, Brazil\xa0– oc curred on ground before launch\n'], ['7', '28 January 1986\n', 'Space Shuttle Challen ger disaster, crashed over Florida, U.S.\n'], ['7', '1 February 2003\n', 'Space Shutt le Columbia disaster, crashed over Texas and Louisiana, U.S.\n'], ['6+\n', '15 Februa ry 1996\n', 'Intelsat 708, Xichang Satellite Launch Center, Sichuan, China, Launch ve hicle veered off course immediately after liftoff and struck a nearby village, offici ally killing at least six people (although other estimates put the death count highe r).\n'], ['3', '27 January 1967\n', 'Apollo 1, Cape Canaveral Air Force Station, Cape Canaveral, FL, U.S.\xa0– occurred on ground during a "plugs out" test.\n'], ['3', '3 0 June 1971\n', 'Soyuz 11, depressurized in space; only deaths in space as of 2024 \n'], ['1', '24 April 1967\n', 'Soyuz 1, southeast of Orenburg, Russia\n'], ['1', '15 November 1967\n', 'X-15 Flight 191, near Edwards AFB, California, U.S.\n'], ['1', '31 October 2014\n', 'VSS Enterprise crash, suborbital, Mojave Desert, California, U. S.\n'], ['12,000', '5–9 December 1952\n', 'The Great Smog of London (5–9 December 195 2). Delayed fatalities rise from 10,000 to 12,000.[149]\n'], ['2,200', '1880\n', 'Coa l smog (London, 1880).[150]\n'], ['1,000', 'December 1956\n', '1956 London smog.[151] \n'], ['780', 'December 1873\n', '1873 London smog[152]\n'], ['779', 'December 1892 \n', '1892 London smog[153]\n'], ['700–800', 'December 1948\n', '1948 London smog[15 3]\n'], ['300–405', 'January–February 1963\n', '1963 New York City smog\n'], ['300–40 0', 'December 1962\n', '1962 London smog[151]\n'], ['220–240', 'November 1953\n', '19 53 New York City smog\n'], ['168', '23–25 November 1966\n', '1966 New York City smog \n'], ['160', '12–15 December 1991\n', 'Smog (London, 12–15 December 1991)[154]\n']]

In [23]:
```
import pandas as pd
deaths_data = pd.DataFrame(temp_data, columns=headers)
deaths_data
```

Out[23]:

	Deaths	Date	Incident
0	20,000	30 May 1626\n	Wanggongchang Explosion in Beijing, China in t...
1	3,000	18 August 1769\n	A lightning bolt caused the Brescia Explosion ...
2	3,000?	1 November 1948\n	Boiler and ammunition explosion aboard an unid...
3	1,400–2,280	6 March 1862\n	Ammunition warehouse explodes and kills almost...
4	1,950	6 December 1917\n	Halifax Explosion in Nova Scotia, Canada[7]\n
...	...	...	...
441	300–405	January–February 1963\n	1963 New York City smog\n
442	300–400	December 1962\n	1962 London smog[151]\n
443	220–240	November 1953\n	1953 New York City smog\n
444	168	23–25 November 1966\n	1966 New York City smog\n
445	160	12–15 December 1991\n	Smog (London, 12–15 December 1991)[154]\n

446 rows × 3 columns

In [24]:
```
deaths_data["Deaths"].unique()
```

Out[24]:
```
array(['20,000', '3,000', '3,000?', '1,400–2,280', '1,950', '1,500',
       '256–1,500+', '1,300+', '1,200', '1,121', '1,100', '1,082',
       '1,007', '1,000+', '843', '800', '738', '700', '621', '600+',
       '590', '581', '575', '565', '542', '508', '500+', '372', '370+',
       '339', '322', '300 (estimate)', '300+', '296+', '256', '235+\n',
       '234', '233', '230+', '220', '219', '218', '217', '207', '206',
       '200+', '3,787–16,000', '1,549', '1,134', '1,099', '687', '682',
       '512', '476–1,000', '458', '439', '437', '426', '422', '405',
       '388', '376', '375', '365', '362', '344', '319', '301', '299[26]',
       '290', '289', '277', '270\n', '268', '266', '263', '262', '259',
       '254', '243', '239', '236', '235', '216', '214', '210', '204',
       '202', '200–6,000\n', '100–240\n', '95–4,000+[31][32]\n', '17\n',
       '13\n', '11\n', '10\n', '26,000–240,000\n', '20,000+\n',
       '11,300\n', '1,800–25,000\n', '2,208\n', '1,134\n', '1,000+\n',
       '1,000\n', '941\n', '500–1500\n', '608\n', '502\n', '400–600+\n',
       '423\n', '356\n', '250–434\n', '270+\n', '268\n', '244\n', '238\n',
       '107–500\n', '226\n', '200+\n', '~200\n', '169\n', '~160\n',
       '120–180\n', '145\n', '144\n', '141\n', '140\n', '139\n', '135\n',
       '128\n', '125\n', '120\n', '117\n', '115\n', '114\n', '111\n',
       '108\n', '100+\n', '100\n', '2,000–3,000\n', '1,700–2845+[48]\n',
       '1,670\n', '900\n', '800\n', '694\n', '658\n', '614\n', '602\n',
       '600\n', '538\n', '530\n', '500\n', '492\n', '448\n', '430\n',
       '400\n', '396\n', '377-470+\n', '360\n', '326+\n', '324\n',
       '322\n', '312\n', '309\n', '291\n', '289\n', '278+\n', '250\n',
       '242\n', '234\n', '225\n', '209\n', '208\n', '200\n', '28\n',
       '15\n', '8\n', '7\n', '6\n', '5\n', '4\n', '3\n', '6,000\n',
       '4,000\n', '2,400+\n', '1,426\n', '1,389\n', '953\n', '363\n',
       '357\n', '354\n', '328\n', '258\n', '251\n', '245\n', '224\n',
       '183\n', '173\n', '162\n', '159\n', '130\n', '124\n', '123\n',
       '118\n', '110\n', '105\n', '102\n', '13,000', '1,112', '614',
       '328', '~200', '135', '127', '97', '93', '84', '83', '79', '71',
       '66', '61', '56', '43', '42', '39', '32', '33', '28', '26',
       '22–24', '22', '21', '19\n', '18\n', '18', '17', '16', '15', '13',
       '12', '11', '583\n', '520\n', '349\n', '346\n', '301\n', '298\n',
       '290\n', '275\n', '273\n', '155\n', '43\n', '20\n', '14\n', '12\n',
       '104\n', '62\n', '31\n', '16\n', '4,385\n', '1,863\n', '1,573\n',
       '1,514\n', 'c.\u20091,500\n', '1,500\n', '1,168\n', '1,159\n',
       '1,024\n', '1,021\n', '1,012\n', 'c.\u20091,000\n', '894\n',
       '852\n', '464–850\n', '844\n', '832\n', '829\n', '826\n', '800+\n',
       '746\n', '741\n', '737\n', '736\n', '731\n', '702\n', '673\n',
       'c.\u2009640\n', '644\n', '627\n', '625\n', '600+\n', '580\n',
       '575\n', '570\n', '565\n', '564\n', '558\n', '550\n', '546\n',
       '533\n', '528\n', '500–1,500\n', '500–600\n', '900–1,700+\n',
       '600–1,000\n', '675–800\n', '235–800\n', '600–700\n', '521–600+\n',
       '428\n', '383+\n', '370\n', '358\n', '338\n', '320\n', '307\n',
       '300\n', '300~\n', '296\n', '285\n', '281\n', '278\n', '248\n',
       '230\n', '212\n', '207\n', '160–200\n', '219\n', '217\n', '206\n',
       '168–3,000\n', '154\n', '121\n', '113\n', '100–200\n', '98\n',
       '96\n', '94\n', '90\n', '90+\n', '81–115\n', '54-300\n', '48\n',
       '21\n', '7', '6+\n', '3', '1', '12,000', '2,200', '1,000', '780',
       '779', '700–800', '300–405', '300–400', '220–240', '168', '160'],
      dtype=object)
```

In [25]:
```
deletions_data['Deaths'] = (deaths_data.Deaths.apply(lambda x: int(''.join(filter(str.isd
# cleans the deaths column by removing every non-digit character
# this has limitations such as not being able to handle ranges (600-700 for example)
deaths_data
```

Out[25]:

	Deaths	Date	Incident
0	20000	30 May 1626\n	Wanggongchang Explosion in Beijing, China in t...
1	3000	18 August 1769\n	A lightning bolt caused the Brescia Explosion ...
2	3000	1 November 1948\n	Boiler and ammunition explosion aboard an unid...
3	14002280	6 March 1862\n	Ammunition warehouse explodes and kills almost...
4	1950	6 December 1917\n	Halifax Explosion in Nova Scotia, Canada[7]\n
...	...	...	...
441	300405	January–February 1963\n	1963 New York City smog\n
442	300400	December 1962\n	1962 London smog[151]\n
443	220240	November 1953\n	1953 New York City smog\n
444	168	23–25 November 1966\n	1966 New York City smog\n
445	160	12–15 December 1991\n	Smog (London, 12–15 December 1991)[154]\n

446 rows × 3 columns

In [31]:
```
deaths_data['Date'] = (deaths_data.Date.apply(lambda x: x.strip())) # removes the \n c
deaths_data
```

Out[31]:

	Deaths	Date	Incident
0	20000	30 May 1626	Wanggongchang Explosion in Beijing, China in t...
1	3000	18 August 1769	A lightning bolt caused the Brescia Explosion ...
2	3000	1 November 1948	Boiler and ammunition explosion aboard an unid...
3	14002280	6 March 1862	Ammunition warehouse explodes and kills almost...
4	1950	6 December 1917	Halifax Explosion in Nova Scotia, Canada[7]\n
...	...	...	...
441	300405	January–February 1963	1963 New York City smog\n
442	300400	December 1962	1962 London smog[151]\n
443	220240	November 1953	1953 New York City smog\n
444	168	23–25 November 1966	1966 New York City smog\n
445	160	12–15 December 1991	Smog (London, 12–15 December 1991)[154]\n

446 rows × 3 columns

In [33]:
```
deaths_data['Incident'] = (deaths_data.Incident.apply(lambda x : x.strip())) # removes
deaths_data
```

Out[33]:

	Deaths	Date	Incident
0	20000	30 May 1626	Wanggongchang Explosion in Beijing, China in t...
1	3000	18 August 1769	A lightning bolt caused the Brescia Explosion ...
2	3000	1 November 1948	Boiler and ammunition explosion aboard an unid...
3	14002280	6 March 1862	Ammunition warehouse explodes and kills almost...
4	1950	6 December 1917	Halifax Explosion in Nova Scotia, Canada[7]
...	...	...	...
441	300405	January–February 1963	1963 New York City smog
442	300400	December 1962	1962 London smog[151]
443	220240	November 1953	1953 New York City smog
444	168	23–25 November 1966	1966 New York City smog
445	160	12–15 December 1991	Smog (London, 12–15 December 1991)[154]

446 rows × 3 columns

In [34]:
```
deaths_data.dtypes
```

Out[34]:
```
Deaths       int64
Date        object
Incident    object
dtype: object
```

I am unable to convert the date to the pandas datetime format as I do not know how to handle ranges such as January-February 1963 or 23-25 December 1966

In [35]:
```
deaths_data.to_csv('deaths_data.csv')
```