Part 1 (1-6)

What difficulties or limitations are there when using regular expressions?
I found that the capabilities are the same when using either BeautifulSoup or re, but the implementation gets significantly more manageable when using BeautifulSoup.

In [ ]:
```
##### -- Imports -- #####
import requests
import re
from bs4 import BeautifulSoup
import pandas as pd

##### -- Variables -- #####
newsFront = 'https://www.bbc.com/news'

##### -- Functions -- #####
def getData(data):
    response = requests.get(data)
    contents = response.text
    return contents

def matches(data):
    regexHeder = re.compile(r'<h\d(?:.*?)>(.*?)<\/h\d>')
    matches = regexHeder.findall(getData(data))
    return matches

def headerList(matchLst):
    lst = []
    for elements in matchLst:
        elements = elements.replace("&#x27;", '')
        elements = re.sub(r'<span(?:.*?)>', '', elements)
        elements = re.sub(r'<\/span>', ' ', elements)
        lst.append(elements)
    return lst

def cleaner(input):
    lst = []
    for elements in input:
        elements = str(elements)
        elements = re.sub(r'<h\d(?:.*?)>', '', elements)
        elements = re.sub(r'<\/h\d>', '', elements)
        elements = re.sub(r'<span(?:.*?)>', '', elements)
        elements = re.sub(r'<\/span>', ' ', elements)
        lst.append(elements)
    return lst

def soupHeders(data):
    soup = BeautifulSoup(getData(data), 'html.parser')
    soupList =  soup.find_all('h1') + soup.find_all('h2') + soup.find_all('h3') +  sou
    return cleaner(soupList)

def topStories(data):
    soup = BeautifulSoup(getData(data), 'html.parser')
```

```
    def find_all(tag):
        return soup.find_all(tag, class_='gs-c-promo-heading__title gel-pica-bold nw-c
    soupList = find_all('h1') + find_all('h2') + find_all('h3') + find_all('h4')
    return cleaner(soupList)


##### -- Calls -- #####
print("")
print("RequestList")
print(headerList(matches(newsFront)))

print("")
print("SoupList")
print(soupHeders(newsFront))

print("")
print("soupTopStories")
print(topStories(newsFront))
```

```
RequestList
['Accessibility links', 'News Navigation', 'BBC News Home', 'Breaking Breaking news
', 'Top Stories', 'US and Russia trade blows over Ukraine at G20', 'US and Russia tra
de blows over Ukraine at G20', 'Related content', 'Angry protests erupt over Greek ra
il disaster', 'Messages reveal battle over UK Covid policy', 'Isabel Oakeshott: Why I
leaked ministers messages', 'Harry and Meghan told to vacate UK Cottage', '1894 shipw
reck confirms tale of treacherous lifeboat', 'Whiskey fungus forces Jack Daniels to s
top construction', 'Argentinas power largely restored after fire', 'Explosive found i
n check-in luggage at US airport', 'Starbucks illegally fired workers over union - ju
dge', 'Aboriginal spears taken in 1770 to return to Sydney', 'Starbucks illegally fir
ed workers over union - judge', 'Aboriginal spears taken in 1770 to return to Sydne
y', 'West Africans to leave Tunisia after race row', 'How fake copyright complaints a
re muzzling journalists', 'Murdaugh jurors visit murder scene as trial closes', 'Gree
ce train crash', 'Human error to blame for train crash - Greek PM', 'Survivors descri
be nightmarish seconds as trains crashed', 'Greece train crash: What we know so far',
'Rescuers search wreckage of deadly Greece train crash', 'Pictures show devastation a
fter Greece train disaster', 'Must see', 'The devastated towns near Ukraines front li
ne', 'BBC World News TV', 'BBC World Service Radio', 'Watch: Dog found alive after 23
days under rubble in Turkey', 'Rare Jurassic-era bug found at Arkansas Walmart', 'Bla
ckpink lead top stars back on the road in Asia', 'Record numbers of guide dog volunte
ers after BBC story', 'Extremely fragile coronation chair being restored', 'Most watc
hed', 'Full story', 'Why did you torture me?', 'How 10% of Nigerian registered voters
delivered victory', 'Sake brewers toast big rise in global sales', 'The Indian-Americ
an CEO who wants to be US president', 'Why Covid lab-leak theory is so disputed', 'Ex
ploring the rigging claims in Nigerias elections', 'Bola Tinubu - the godfather set t
o lead Nigeria', 'Most read', 'Around the BBC', 'The forests where trees go missing',
'The problem confronting women of colour', 'The surprising truth about Champagne', 'T
he greatest monster film ever made', 'The effect of TikTok’s beauty filters', 'A new
way to navigate work and life', 'The return of the US lost language', 'Sport', 'Grims
by shock Southampton &amp;amp; greatest FA Cup moments', 'Yorkshire racism hearing - day
two set to begin', 'Premier League must re-examine Newcastles Saudi deal', 'Fifas app
ointment of supermodel tone deaf', 'In title race to stay - Arsenal show mentality of
champions', 'Welcome to the future - LIV mocks PGA shake-up', 'India v Australia - th
ird Test scorecard', 'Find us here', 'News daily newsletter', 'Mobile app', 'Get in t
ouch', 'News Navigation', 'Explore the BBC']

SoupList
['BBC News Home', 'Accessibility links', 'News Navigation', 'Breaking Breaking news
', 'Top Stories', 'Greece train crash', 'Must see', 'Most watched', 'Full story', 'Mo
st read', 'Around the BBC', 'Sport', 'Find us here', 'News Navigation', 'Explore the
BBC', 'US and Russia trade blows over Ukraine at G20', 'US and Russia trade blows ove
r Ukraine at G20', 'Angry protests erupt over Greek rail disaster', 'Messages reveal
battle over UK Covid policy', "Isabel Oakeshott: Why I leaked minister's messages",
"Harry and Meghan told to 'vacate' UK Cottage", '1894 shipwreck confirms tale of trea
cherous lifeboat', 'Whiskey fungus forces Jack Daniels to stop construction', "Argent
ina's power largely restored after fire", 'Explosive found in check-in luggage at US
airport', 'Starbucks illegally fired workers over union - judge', 'Aboriginal spears
taken in 1770 to return to Sydney', 'Starbucks illegally fired workers over union - j
udge', 'Aboriginal spears taken in 1770 to return to Sydney', 'West Africans to leave
Tunisia after race row', 'How fake copyright complaints are muzzling journalists', 'M
urdaugh jurors visit murder scene as trial closes', 'Human error to blame for train c
rash - Greek PM', "Survivors describe 'nightmarish seconds' as trains crashed", 'Gree
ce train crash: What we know so far', 'Rescuers search wreckage of deadly Greece trai
n crash', 'Pictures show devastation after Greece train disaster', "The devastated to
wns near Ukraine's front line", 'BBC World News TV', 'BBC World Service Radio', 'Watc
h: Dog found alive after 23 days under rubble in Turkey', 'Rare Jurassic-era bug foun
d at Arkansas Walmart', 'Blackpink lead top stars back on the road in Asia', 'Record
numbers of guide dog volunteers after BBC story', 'Extremely fragile coronation chair
being restored', "'Why did you torture me?'", 'How 10% of Nigerian registered voters
delivered victory', 'Sake brewers toast big rise in global sales', 'The Indian-Americ
```

```
an CEO who wants to be US president', 'Why Covid lab-leak theory is so disputed', "Ex
ploring the rigging claims in Nigeria's elections", "Bola Tinubu - the 'godfather' se
t to lead Nigeria", 'The forests where trees go missing', 'The problem confronting wo
men of colour', 'The surprising truth about Champagne', 'The greatest monster film ev
er made', 'The effect of TikTok’s beauty filters', 'A new way to navigate work and li
fe', "The return of the US' lost language", 'Grimsby shock Southampton &amp;amp; greatest
FA Cup moments', 'Yorkshire racism hearing - day two set to begin', "Premier League
'must re-examine' Newcastle's Saudi deal", "Fifa's appointment of supermodel 'tone de
af'", "'In title race to stay - Arsenal show mentality of champions'", "'Welcome to t
he future' - LIV mocks PGA shake-up", 'India v Australia - third Test scorecard', 'Ne
ws daily newsletter', 'Mobile app', 'Get in touch', 'Related content']

soupTopStories
['Angry protests erupt over Greek rail disaster', 'Messages reveal battle over UK Cov
id policy', "Isabel Oakeshott: Why I leaked minister's messages", "Harry and Meghan t
old to 'vacate' UK Cottage", '1894 shipwreck confirms tale of treacherous lifeboat',
'Whiskey fungus forces Jack Daniels to stop construction', "Argentina's power largely
restored after fire", 'Explosive found in check-in luggage at US airport', 'Starbucks
illegally fired workers over union - judge', 'Aboriginal spears taken in 1770 to retu
rn to Sydney', 'Starbucks illegally fired workers over union - judge', 'Aboriginal sp
ears taken in 1770 to return to Sydney', 'West Africans to leave Tunisia after race r
ow', 'How fake copyright complaints are muzzling journalists', 'Murdaugh jurors visit
murder scene as trial closes', 'Human error to blame for train crash - Greek PM', "Su
rvivors describe 'nightmarish seconds' as trains crashed", 'Greece train crash: What
we know so far', 'Rescuers search wreckage of deadly Greece train crash', 'Pictures s
how devastation after Greece train disaster', 'BBC World News TV', 'BBC World Service
Radio', 'Watch: Dog found alive after 23 days under rubble in Turkey', 'Rare Jurassic
-era bug found at Arkansas Walmart', 'Blackpink lead top stars back on the road in As
ia', 'Record numbers of guide dog volunteers after BBC story', 'Extremely fragile cor
onation chair being restored', 'How 10% of Nigerian registered voters delivered victo
ry', 'Sake brewers toast big rise in global sales', 'The Indian-American CEO who want
s to be US president', 'Why Covid lab-leak theory is so disputed', "Exploring the rig
ging claims in Nigeria's elections", "Bola Tinubu - the 'godfather' set to lead Niger
ia", 'The problem confronting women of colour', 'The surprising truth about Champagn
e', 'The greatest monster film ever made', 'The effect of TikTok’s beauty filters',
'A new way to navigate work and life', "The return of the US' lost language", 'Yorksh
ire racism hearing - day two set to begin', "Premier League 'must re-examine' Newcast
le's Saudi deal", "Fifa's appointment of supermodel 'tone deaf'", "'In title race to
stay - Arsenal show mentality of champions'", "'Welcome to the future' - LIV mocks PG
A shake-up", 'India v Australia - third Test scorecard']
```

Part 1 (7-8)

In [ ]:

```python
##### -- Functions -- #####
def cleaner2(input):
    lst = []
    for elements in input:
        elements = str(elements)
        elements = re.sub(r'<p(?:.*?)>', '', elements)
        elements = re.sub(r'<\/p>', '', elements)
        elements = re.sub(r'</polygon>', '', elements)
        elements = re.sub(r'</svg>', '', elements)
        elements = re.sub(r'</span>', '', elements)
        elements = re.sub(r'<h3(?:.*?)>', '', elements)
        elements = re.sub(r'<\/h3>', '', elements)
        lst.append(elements)
    return lst
```

```
def soupSummary(data):
    soup = BeautifulSoup(getData(data), 'html.parser')
    soupList = soup.find_all('p', class_='gs-c-promo-summary gel-long-primer gs-u-mt r
    out = cleaner2(soupList)
    return out

def divider(input):
    def getHeader(string):
        string = re.search(r'<h\d(?:.*?)>(.*?)<\/h\d>', string)
        string = string.group(0)
        string = cleaner([string])
        return string[0]
    def getSummary(string):
        string = re.search(r'<p(?:.*?)>.*<\/p>', string)
        if string == None: return "NoSummery"
        string = string.group(0)
        string = cleaner2([string])
        return string[0]
    def getCategory(string):
        string = re.search(r'<span aria-hidden="true">(.*?)<\/span>', string)
        if string == None: return "NoCategory"
        string = string.group(0)
        string = re.sub(r'<span aria-hidden="true">', '', string)
        string = re.sub(r'<\/span>', '', string)
        string = re.sub(r'&amp;', 'and', string)
        return string
    lst = [getHeader(input), getSummary(input), getCategory(input)]
    return lst

def Div(data):
    soup = BeautifulSoup(getData(data), 'html.parser')
    soupList = soup.find_all('div', class_='gs-c-promo-body gel-1/2@xs gel-1/1@m gs-u-
    lst = []
    for elements in soupList:
        elements = str(elements)
        elements = divider(elements)
        lst.append(elements)
    return lst

def lstToPandasCsv(input):
    df = pd.DataFrame(input)
    df.columns = ['header', 'summary', 'category']
    df.to_csv('csvOut')
    return df

##### -- Calls -- #####
lstToPandasCsv(Div(newsFront))
```

Out[ ]:

	header	summary	category
0	Angry protests erupt over Greek rail disaster	Many protesters see the crash which claimed 43...	Europe
1	Messages reveal battle over UK Covid policy	Leaked WhatsApp messages show ministers clashi...	UK Politics
2	Isabel Oakeshott: Why I leaked minister's mess...	Isabel Oakeshott: Why I leaked minister's mess...	UK
3	Harry and Meghan told to 'vacate' UK Cottage	It was earlier reported that the home, in Wind...	UK
4	1894 shipwreck confirms tale of treacherous li...	1894 shipwreck confirms tale of treacherous li...	US and Canada
5	Watch: Dog found alive after 23 days under rub...	NoSummery	NoCategory
6	Rare Jurassic-era bug found at Arkansas Walmart	NoSummery	NoCategory
7	Blackpink lead top stars back on the road in Asia	NoSummery	NoCategory
8	Record numbers of guide dog volunteers after B...	NoSummery	NoCategory
9	Extremely fragile coronation chair being restored	NoSummery	NoCategory
10	How 10% of Nigerian registered voters delivere...	NoSummery	NoCategory
11	Sake brewers toast big rise in global sales	NoSummery	NoCategory
12	The Indian-American CEO who wants to be US pre...	NoSummery	NoCategory
13	Why Covid lab-leak theory is so disputed	NoSummery	NoCategory
14	Exploring the rigging claims in Nigeria's elec...	NoSummery	NoCategory
15	Bola Tinubu - the 'godfather' set to lead Nigeria	NoSummery	NoCategory
16	The problem confronting women of colour	NoSummery	BBC Worklife
17	The surprising truth about Champagne	NoSummery	BBC Travel
18	The greatest monster film ever made	NoSummery	BBC Culture
19	The effect of TikTok’s beauty filters	NoSummery	BBC Future
20	A new way to navigate work and life	NoSummery	BBC Worklife
21	The return of the US' lost language	NoSummery	BBC Travel
22	Yorkshire racism hearing - day two set to begin	NoSummery	Cricket
23	Premier League 'must re-examine' Newcastle's S...	NoSummery	Football

	header	summary	category
24	Fifa's appointment of supermodel 'tone deaf'	NoSummery	Football
25	'In title race to stay - Arsenal show mentalit...	NoSummery	Football
26	'Welcome to the future' - LIV mocks PGA shake-up	NoSummery	Golf
27	India v Australia - third Test scorecard	NoSummery	NoCategory

Part 2

Describe the tools used and the challenges faced when creating the dataframe.
When creating the articleList function, which returns a list of links referring to all the articles, the temp function uses the beautifulSoup html.parser and the find_all property to get a string with the content division element that contains all the needed links. Temp takes a letter as input and concatenates this letter to the Wikinews link. The following letters: K, L, M, N, O, P, R, S, T, and U are run on temp inside the articleList function by splitting the list from the assignment description and running it in a for-loop.

The Pandas dataframe is created in the createTable function. A for-loop goes through all the links and readies the text for cleaning using request and bs4; the string is then fed to a tuple of functions that extract the relevant data. This data is then appended to a list and pasted into a Pandas dataframe.

Of the three cleaning functions, getContent unraveled the most challenges. The first challenge was that the content consisted of multiple chunks of phrases. To deal with this defContent creates a list of all \<p>'s and converts every list element into strings. The join function then unpacks the list to prepare a string for cleaning. This leads to the next challenge: cleaning the data in a helper function called cleanAll. cleanAll consists of regular expressions that remove the date and unwanted tags. When looking at the HTML code, I noticed that the content is always appearing before the tags \
, \<img>, and \<b>. Therefore everything coming after one of these tags is removed.

Assess whether it is a reasonable choice to trust the sources when they aren't labeled.
If it's assumed that whoever created the list of articles only included articles they believed were legit (or at least wanted the reader to think were legit), such labels would come from the same source and therefore have the same credibility. Because of this, the existence of labels doesn't matter.

When using the list to practice programming skills, it doesn't matter either. The code would be identical if the articles were written in lorem ipsum.

In [ ]:
```
def cleanAll(input):
    tag = ['p', 'span', 'a', 'i']
```

```
    text = input
    text = re.sub(r'(<strong(?:.*?)>).*(<\/strong>)', '', text)
    text = re.sub(r'\s+', ' ', text)
    for element in tag:
        x = '<' + element + '(?:.*?)>'
        y = '<\/' + element + '>'
        text = re.sub(x, '', text)
        text = re.sub(y, ' ', text)
    text = re.sub(r'\s+', ' ', text)
    text = re.sub(r'\s,', '', text)
    text = re.sub('\<br(.*)', '', text)
    text = re.sub('\<img(.*)', '', text)
    text = re.sub('\<b(.*)', '', text)
    return text

def getHeader(input):
    text = input.find('span', class_='mw-page-title-main')
    text = str(text)
    text = cleanAll(text)
    return text

def getDate(input):
    text = input.find('strong', class_='published')
    text = str(text)
    text = re.sub(r'<strong(?:.*?)>', '', text)
    text = re.sub(r'<\/strong>', ' ', text)
    text = re.sub(r'<span(?:.*?)>', '', text)
    text = re.sub(r'<\/span>', ' ', text)
    return text

def getContent(input):
    text = input.find_all('p')
    lst = []
    for elm in text:
        elm = str(elm)
        lst.append(elm)
    string = ' '.join(lst)
    string = cleanAll(string)
    return string

def articleList():
    def temp(input):
        page = 'https://en.wikinews.org/w/index.php?title=Category:Politics_and_confli
        divGroup = BeautifulSoup(getData(page), 'html.parser')
        divGroup = divGroup.find_all('div', id='mw-pages')
        divGroup = divGroup[0].find_all('div', class_='mw-category-group')
        divGroup = divGroup[0].find_all('a')
        lst = []
        for element in divGroup:
            element = str(element)
            href_regex = r'href="([^"]+)"'
            element = re.search(href_regex, element)
            element = element.group(1)
            element = 'https://en.wikinews.org/' + element
            lst.append(element)
        return lst
    letters = "ABCDEFGHIJKLMNOPRSTUVWZABCDEFGHIJKLMNOPRSTUVWZ"[10%23:10%23+10]
    letters = [*letters]
    lst = []
    for elemement in letters:
```

```
        elememement = temp(elememement)
        lst = lst + elememement
    return lst

def createTable():
    links = (articleList())
    lst = []
    for elm in links:
        response = requests.get(elm)
        contents = response.text
        x = BeautifulSoup(contents, 'html.parser')
        x = [getHeader(x), getDate(x), getContent(x)]
        lst.append(x)

    df = pd.DataFrame(lst)
    df.columns = ['header', 'date', 'content']
    return df
print(createTable())
```

```
                                                 header  \
0     K'nesset Member Natan Sharansky resigns from c...
1     Kaczynski elected as the new president of Pola...
2       Kaczyński takes the office of Polish president
3     Kansas Professor assaulted by angry intelligen...
4                 Karachi, Pakistan shut down by strike
...                                                 ...
1881  UK defers junk food deals, advertisement restr...
1882  UK denies pressuring Scotland into Lockerbie r...
1883  UK drugs policy petition reaches 100,000 signa...
1884  UK economy shrinks by 0.3% in fourth quarter o...
1885  UK elections: David Cameron becomes Prime Mini...

                               date  \
0              Tuesday, May 3, 2005
1           Sunday, October 23, 2005
2          Friday, December 23, 2005
3          Tuesday, December 6, 2005
4               Monday, May 14, 2007
...                              ...
1881          Tuesday, May 17, 2022
1882   Wednesday, September 2, 2009
1883      Friday, February 14, 2014
1884       Friday, January 25, 2013
1885          Tuesday, May 11, 2010

                                              content
0     Sharansky’s resignation as Minister of Diaspo...
1     Lech Kaczyński has been elected as the new pr...
2
3     Professor Paul Mirecki the chairman of the Re...
4     After two days of violence in Karachi Pakista...
...                                                 ...
1881  The United Kingdom Department of Health and S...
1882  Since the August 20 release of Abdelbaset Ali...
1883  The Backbench Business Committee of the House...
1884  The United Kingdom economy shrank by 0.3% in ...
1885  David Cameron was today appointed the new Bri...

[1886 rows x 3 columns]
```

The Pandas dateframe may look odd compared to part 1. This happens only in Jupyter and works correctly when the script is run in the command prompt. Note that I only refer to the visualization, not the actual structure.