Data Wrangling Assignment

Nick Sun

1/29/2020

Process

Blink-182 was one of my favorite bands growing up and certainly falls into the category of bands that have changed their sound over the years. Their new style is noticeably more pop and less juvenile than their earlier work, but some elements, such as the use of repetitive melodic recitatives (example: "na na na na na na") remain the same.

I thought it might be interesting to gather a corpus of Blink-182 lyrics to analyze. While there are APIs that have lyrical content for certain artists available, I decided to use web scraping instead since many fan sourced sites have transcriptions of live songs that are not found on commercial APIs. The site I used is azlyrics.com since it contains not only the song lyrics but album information and year of release. I used Python for this assignment, specifically the `BeautifulSoup` and `requests` libraries.

The first step to getting a corpus of lyrics is getting a list of all the songs available for that artist on azlyrics. This can be done by getting the html for Blink-182 artists page and then using a basic CSS selector to grab both the names of all the songs and the corresponding href attribute for the URL to the individual lyrics page and storing this information as dictionaries. Overall there are 176 songs on azlyrics.com for Blink-182, so 176 dictionaries will be created and stored together in a list.

Afterwards, I converted this list of dictionaries into a pandas dataframe to iterate through the list of song URLs. These URLs led to individual song pages that contained both the lyrics and other information such as the album and year of release. The lyrics were contained in a div that did not have any identifying attributes, so the only way to get to this text in the DOM is to use a CSS selector. Thankfully, the individual song pages all have about the same format with only some variation. The text did need to be cleaned slightly since it contained newline and carriage return whitespace characters.

Scraping the other information off the page was slightly more complicated since the div which contains the album name and release date needs to be cleaned using regular expressions. However, some trial and error eventually yielded the correct expression to extract both the album name and the year. There are likely some edge cases that are being missed with this particular regex, such as Unicode characters, but for the Blink-182 discography this approach seems to work.

Some other considerations that I had were using an inner join to connect the dataframe containing the song information with the original dataframe containing all the song titles and URLs. Finally, I had to make sure to put a sleep in between scrapes of the individual song pages, since otherwise azlyrics.com would think I am malicious traffic and block me from their server. This took a few tries to get right, since it was difficult for me to estimate how quick I could make the sleeps without being blocked. I did implement a print statement to stdout so I could tell if and where the script stopped working. The final corpus of lyrics was exported using pandas as both a csv and json file.

Some fun ideas I have for this dataset include analyzing the sentiment of Blink-182 songs over time, charting the length of songs between different albums, or possibly training a Markov Chain to write my own Blink-182 songs!

Initial Data

The initial data of this project were HTML webpages.

From the scale provided on this assignment, I estimate that the data is suitable for this assignment. Using the assignment point system as a checklist, here is how the web scraped lyrics corpus matches up:

- Was mostly in a standardized form, but some songs' pages were structured differently (for example, to accomodate featured artists) so the script had to be adjusted accordingly for those songs
- Data was split across multiple webpages, and hence, multple HTML files
- Data is in a format other than CSV, JSON, or in a database
- Data does contain punctuations, but for the purposes of this assignment I opted not to remove them
- Dataset is relatively small at only 212KB total
- The way I scraped the data involved a two dataframes, one listing the songs, and one with the song information and between the two of them there are a few types of related data (ex: year vs album name)
- Data was accessed using `requests` library, not downloading a file or connecting to a database

Some of the points will be subjective, but I estimate the difficulty of this dataset to be at least a 4.

Below is an excerpt of an example song lyrics webpage from azlyrics.com, with only the relevant sections which I scrape data from. These individual sections are separated with an ellipsis.

```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<meta content="Blink-182 &quot;Brohemian Rhapsody&quot;: There's something about you That I can't quite
<meta content="Brohemian Rhapsody lyrics, Blink-182 Brohemian Rhapsody lyrics, Blink-182 lyrics" name="
<meta content="noarchive" name="robots"/>
<meta content="//www.azlyrics.com/az_logo_tr.png" property="og:image"/>
<title>Blink-182 - Brohemian Rhapsody Lyrics | AZLyrics.com</title>
...
<div class="div-share"><h1>"Brohemian Rhapsody" lyrics</h1></div>
<div class="lyricsh">
<h2><b>Blink-182 Lyrics</b></h2>
</div>
<div class="ringtone">
<span id="cf_text_top"></span>
</div>
<b>"Brohemian Rhapsody"</b><br/>
<br/>
<div>
<!-- Usage of azlyrics.com content by any third-party lyrics provider is prohibited by our licensing ag
There's something about you <br/>
That I can't quite put my finger in
</div>
<br/><br/>
...
<script type="text/javascript">
ArtistName = "Blink-182";
```

```
SongName = "Brohemian Rhapsody";
function submitCorrections(){
    document.getElementById('corlyr').submit();
    return false;
}
</script>
```

Example Data: CSV

Below are the first few rows of the resulting csv. Note that a column of this data is song lyrics, held as a string, which takes a large amount of space and cannot be accurately displayed on the document.

Additionally, I can provide the actual csv files as proof of the extraction working correctly.

```
title,url,lyrics,album_name,year
Reebok Commercial,https://www.azlyrics.com/lyrics/blink182/reebokcommercial.html,"You are better than me
Time,https://www.azlyrics.com/lyrics/blink182/time.html,When the clock strikes two There's so much to d
Red Skies,https://www.azlyrics.com/lyrics/blink182/redskies.html,"Why can't people just understand  mone
Alone,https://www.azlyrics.com/lyrics/blink182/alone.html,"what were doing here, now  no one knows  the
```

If the above is not sufficient, here is a screenshot of the csv in Excel:

	A	B	C	D	E
1	title	url	lyrics	album_name	year
2	Reebok Cc	https://www.a	You are better than me,girls,money,and everything I try to compete v	Flyswatter	1992
3	Time	https://www.a	When the clock strikes two There's so much to do And I cant explain	Flyswatter	1992
4	Red Skies	https://www.a	Why can't people just understand money's something in the nature	Flyswatter	1992
5	Alone	https://www.a	what were doing here, now no one knows the thoughts, the things t	Flyswatter	1992
6	Point Of V	https://www.a	Two different people, two different places Through a one way windc	Flyswatter	1992

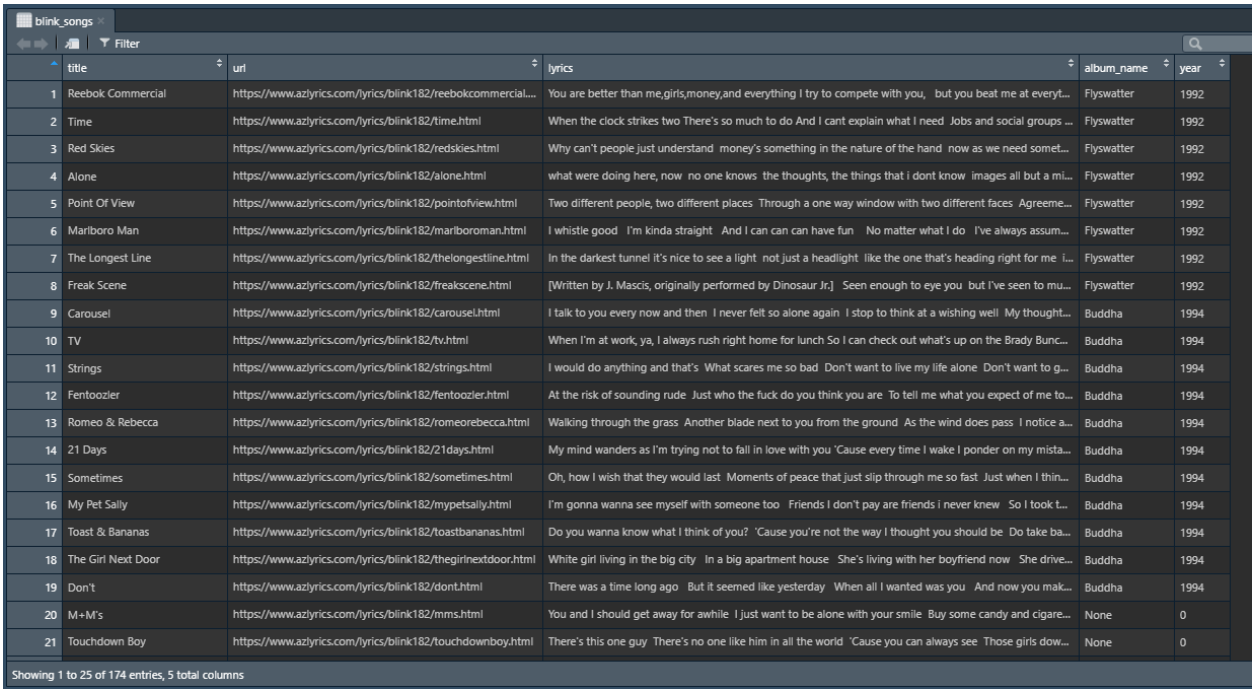
Example Data: JSON

This data is output using the `to_json` method in pandas DataFrames. By default, it outputs the data as `orient='index'`, but can instead be output in other formats. In the example below, I output it with `orient='records'`.

```
[
  {
    "title": "Reebok Commercial",
    "url": "https://www.azlyrics.com/lyrics/blink182/reebokcommercial.html",
    "lyrics": "You are better than me,girls,money,and everything I try to compete with you,   but you b
    "album_name": "Flyswatter",
    "year": "1992"
  },
  {
    "title": "Time",
    "url": "https://www.azlyrics.com/lyrics/blink182/time.html",
    "lyrics": "When the clock strikes two There's so much to do And I cant explain what I need  Jobs an
    "album_name": "Flyswatter",
    "year": "1992"
  }
]
```

Example of the Loaded Data

Below is the head of the csv after it is loaded into RStudio using `readr`.



blink_songs

Filter

	title	url	lyrics	album_name	year
1	Reebok Commercial	https://www.azlyrics.com/lyrics/blink182/reebokcommercial....	You are better than me,girls,money,and everything I try to compete with you, but you beat me at everyt...	Flyswatter	1992
2	Time	https://www.azlyrics.com/lyrics/blink182/time.html	When the clock strikes two There's so much to do And I cant explain what I need Jobs and social groups ...	Flyswatter	1992
3	Red Skies	https://www.azlyrics.com/lyrics/blink182/redskies.html	Why can't people just understand money's something in the nature of the hand now as we need somet...	Flyswatter	1992
4	Alone	https://www.azlyrics.com/lyrics/blink182/alone.html	what were doing here, now no one knows the thoughts, the things that i dont know images all but a mi...	Flyswatter	1992
5	Point Of View	https://www.azlyrics.com/lyrics/blink182/pointofview.html	Two different people, two different places Through a one way window with two different faces Agreeme...	Flyswatter	1992
6	Marlboro Man	https://www.azlyrics.com/lyrics/blink182/marlboroman.html	I whistle good I'm kinda straight And I can can can have fun No matter what I do I've always assum...	Flyswatter	1992
7	The Longest Line	https://www.azlyrics.com/lyrics/blink182/thelongestline.html	In the darkest tunnel it's nice to see a light not just a headlight like the one that's heading right for me i...	Flyswatter	1992
8	Freak Scene	https://www.azlyrics.com/lyrics/blink182/freakscene.html	[Written by J. Mascis, originally performed by Dinosaur Jr.] Seen enough to eye you but I've seen to mu...	Flyswatter	1992
9	Carousel	https://www.azlyrics.com/lyrics/blink182/carousel.html	I talk to you every now and then I never felt so alone again I stop to think at a wishing well My thought...	Buddha	1994
10	TV	https://www.azlyrics.com/lyrics/blink182/tv.html	When I'm at work, ya, I always rush right home for lunch So I can check out what's up on the Brady Bunc...	Buddha	1994
11	Strings	https://www.azlyrics.com/lyrics/blink182/strings.html	I would do anything and that's What scares me so bad Don't want to live my life alone Don't want to g...	Buddha	1994
12	Fentoozler	https://www.azlyrics.com/lyrics/blink182/fentoozler.html	At the risk of sounding rude Just who the fuck do you think you are To tell me what you expect of me to...	Buddha	1994
13	Romeo & Rebecca	https://www.azlyrics.com/lyrics/blink182/romeorebecca.html	Walking through the grass Another blade next to you from the ground As the wind does pass I notice a...	Buddha	1994
14	21 Days	https://www.azlyrics.com/lyrics/blink182/21days.html	My mind wanders as I'm trying not to fall in love with you 'Cause every time I wake I ponder on my mista...	Buddha	1994
15	Sometimes	https://www.azlyrics.com/lyrics/blink182/sometimes.html	Oh, how I wish that they would last Moments of peace that just slip through me so fast Just when I thin...	Buddha	1994
16	My Pet Sally	https://www.azlyrics.com/lyrics/blink182/mypetsally.html	I'm gonna wanna see myself with someone too Friends I don't pay are friends i never knew So I took t...	Buddha	1994
17	Toast & Bananas	https://www.azlyrics.com/lyrics/blink182/toastbananas.html	Do you wanna know what I think of you? 'Cause you're not the way I thought you should be Do take ba...	Buddha	1994
18	The Girl Next Door	https://www.azlyrics.com/lyrics/blink182/thegirlnextdoor.html	White girl living in the big city In a big apartment house She's living with her boyfriend now She drive...	Buddha	1994
19	Don't	https://www.azlyrics.com/lyrics/blink182/dont.html	There was a time long ago But it seemed like yesterday When all I wanted was you And now you mak...	Buddha	1994
20	M+M's	https://www.azlyrics.com/lyrics/blink182/mms.html	You and I should get away for awhile I just want to be alone with your smile Buy some candy and cigare...	None	0
21	Touchdown Boy	https://www.azlyrics.com/lyrics/blink182/touchdownboy.html	There's this one guy There's no one like him in all the world 'Cause you can always see Those girls dow...	None	0

Showing 1 to 25 of 174 entries, 5 total columns

And here is how it actually appears in `Rstudio` using the `knitr` package.

title	url	lyrics
Reebok Commercial	https://www.azlyrics.com/lyrics/blink182/reebokcommercial.html	You are better than me,girls,m
Time	https://www.azlyrics.com/lyrics/blink182/time.html	When the clock strikes two Th

Scripts

The Python module I wrote to scrape the data is provided below. Additionally, I have a utility function written using the pandas library to convert json files to csv.

```
#!/usr/bin/env python

import pandas as pd
import requests
import time
import re
from bs4 import BeautifulSoup

def get_songs(url):
    """
    Function to get list of songs and corresponding URLs under an artist page
    Input: url to an artist's page on azlyrics.com
    Output: Returns a list of dicts ({'title' : [TITLE], 'url' : [URL]})
    """
    songlist = requests.get(url)
    soup = BeautifulSoup(songlist.text)
```

```
    songinfo = []

    for row in soup.findAll('div', attrs = {'class' : 'listalbum-item'}):
        song = {}
        song['title'] = row.a.text
        song['url'] = row.a['href'].replace("..", "https://www.azlyrics.com")
        songinfo.append(song)

    return(songinfo)

def get_song_info(url):
    """
    Function to get lyrics, album, and year information for individual song
    Input: url to a song's page on azlyrics.com
    Output: Dict ({'lyrics' : [string LYRICS],
                   'url' : [string URL],
                   'album_name' : [string ALBUM_NAME],
                   'year': [int YEAR]})
    """
    tmp = requests.get(url)
    tmpsoup = BeautifulSoup(tmp.text)

    try:
        tmpalbum = tmpsoup.find('div', attrs = {'class' : 'songinalbum_title'}).text.replace("album: ",
        albumpattern = r'"([A-Za-z0-9]*)"'
        album_name = re.search(albumpattern, tmpalbum).group(0).replace('\"', '')

        yearpattern = r'\([0-9]*\)'
        album_year = re.search(yearpattern, tmpalbum).group(0).replace('(', '').replace(')', '')
    except:
        album_name = "None"
        album_year = 0

    try:
        tmplyrics = tmpsoup.select('body > div.container.main-page > div > div.col-xs-12.col-lg-8.text-
        lyrics = tmplyrics.replace('\n', ' ').replace('\r', ' ').strip()
    except:
        tmplyrics = tmpsoup.select('body > div.container.main-page > div > div.col-xs-12.col-lg-8.text-
        lyrics = tmplyrics.replace('\n', ' ').replace('\r', ' ').strip()

    output = {}
    output['lyrics'] = lyrics
    output['album_name'] = album_name
    output['year'] = album_year
    output['url'] = url

    return(output)

def json_to_csv(json_file):
    '''
    Converts json to csv files.

    Args:
```

```
        json_file (str) : PATH to json file

    Returns:
        .csv file in current working directory
    '''

    try:
        df = pd.read_json(json_file)
        output = df.to_csv(encoding='utf-8',
                           index=False)
        return(output)
    except:
        print("Error in converting to json")
```

The actual `main.py` file to run the script is provided here:

```
from lyrics_scraper import get_song_info, get_songs

import pandas as pd
import requests
import time
import re
from bs4 import BeautifulSoup

url = 'https://www.azlyrics.com/b/blink.html'

songinfo = get_songs(url)

blink = []

for song in songinfo:
    print("Working on: {}".format(song['title']))
    song_output = get_song_info(song['url'])
    blink.append(song_output)
    time.sleep(10)

song_info = pd.DataFrame(songinfo)
song_lyrics = pd.DataFrame(blink)

blink_songs = pd.merge(left = song_info,
                       right = song_lyrics,
                       left_on = 'url',
                       right_on = 'url')

blink_songs = blink_songs.drop_duplicates()

blink_songs.to_csv(r'blink_songs.csv',
                   index = None,
                   header = True)

blink_songs.to_json(r'blink_songs.json')
```