Welcome to my Portfolio Project on WebScraping!

In this project, I am going to analyze the Top 100 Covered Albums of All-Time. This project is definitely more personal to me as opposed to my other exercises. As an Amateur Musician, not only do I have a passion for music, but I most definitely love discovering covers of Songs/Albums where the artist provides his/her own twist to the original.

[55]
```
#Make the necessary imports immediately to avoid errors.

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import requests
from bs4 import BeautifulSoup
```

[56]
```
#Here we are making an HTTP request to pull the HTML code from
the URL we desire.

URL = "https://secondhandsongs.com/statistics?
sort=covers&list=stats_release_covers"
r = requests.get(URL)
```

[57]
```
soup = BeautifulSoup(r.content, "html5lib")
soup.prettify()
#We would like to extract the <tr> tags belonging to the table,
whose id = "vw"
rows = soup.select('#vw tr')
```

[74]
```
#We would like to extract the <tr> tags belonging to the table,
whose id = "vw"
rows = soup.select('#vw tr')
print (rows[:5]) #Prints out top 5 TR values to show the format.
```

```
[<tr><th class="field-index "></th>
 <th class="field-release ">Release</th>
 <th class="field-performer ">Performer</th>
 <th class="field-covers text-right">Covers <i class="fa fa-caret-down
fa- "></i></th>
 </tr>, <tr>
   <td class="field-index ">1</td>
   <td class="field-release "><a class="link-release"
href="/release/712">The Beatles [White Album]</a></td>
   <td class="field-performer "><a class="link-performer"
href="/artist/41">The Beatles</a></td>
   <td class="field-covers text-right">1646</td>
  </tr>, <tr>
   <td class="field-index ">2</td>
   <td class="field-release "><a class="link-release"
href="/release/156">Rubber Soul</a></td>
   <td class="field-performer "><a class="link-performer"
href="/artist/41">The Beatles</a></td>
   <td class="field-covers text-right">1516</td>
  </tr>, <tr>
   <td class="field-index ">3</td>
   <td class="field-release "><a class="link-release"
href="/release/1095">Revolver</a></td>
   <td class="field-performer "><a class="link-performer"
href="/artist/41">The Beatles</a></td>
   <td class="field-covers text-right">1509</td>
  </tr>, <tr>
   <td class="field-index ">4</td>
   <td class="field-release "><a class="link-release"
href="/release/243">Abbey Road</a></td>
   <td class="field-performer "><a class="link-performer"
href="/artist/41">The Beatles</a></td>
   <td class="field-covers text-right">1480</td>
  </tr>]
```

[59]
```
frame = []
for row in rows:
    frame.append([td.text for td in row.select('td')])
    print([td.text for td in row.select('td')])
```

```
[]
['1', 'The Beatles [White Album]', 'The Beatles', '1646']
['2', 'Rubber Soul', 'The Beatles', '1516']
['3', 'Revolver', 'The Beatles', '1509']
```

['4', 'Abbey Road', 'The Beatles', '1480']
['5', 'Meet Me in St. Louis', 'Judy Garland with Georgie Stoll and His Orchestra', '1445']
['6', 'Silent Night, Hallowed Night', 'Haydn Quartet', '1411']
['7', 'The Christmas Song (Merry Christmas to You)', 'The King Cole Trio with String Choir', '1226']
['8', "Sgt. Pepper's Lonely Hearts Club Band", 'The Beatles', '1114']
['9', 'Help!', 'The Beatles', '1083']
['10', 'Were You Fooling', 'Richard Himber & His Orchestra', '1025']
['11', 'Jingle Bells', 'Edison Male Quartette', '1006']
['12', 'Body and Soul', 'Ambrose and His Orchestra', '959']
['13', 'God Rest Ye Merry, Gentlemen', 'Meister Glee Singers', '923']
['14', "A Hard Day's Night", 'The Beatles', '901']
['15', "I'll Be Home for Christmas (If Only in My Dreams)", 'Bing Crosby with John Scott Trotter and His Orchestra', '858']
['16', 'The First Nowell', 'Tally-Ho!', '856']
['17', 'Christmas with The Trapp Family Singers', 'The Trapp Family Singers', '810']
['18', "The Freewheelin' Bob Dylan", 'Bob Dylan', '804']
['19', 'O Amor o Sorriso e a Flor', 'João Gilberto', '750']
['20', "Somebody's Gotta Go", 'Cootie Williams and His Orchestra', '740']
['21', 'Let It Snow! Let It Snow! Let It Snow!', 'Vaughn Monroe and His Orchestra', '717']
['22', 'Tapestry', 'Carole King', '679']
['23', 'One Night in Havana', 'Hoagy Carmichael & His Pals', '630']
['24', 'Caravan', 'Barney Bigard and His Jazzopators', '628']
['25', 'Moon River', 'Henry Mancini, His Orchestra and Chorus', '625']
['26', 'Bridge over Troubled Water', 'Simon and Garfunkel', '617']
['27', 'Hesitating Blues', "Prince's Band", '602']
['28', 'St. Louis Woman - Original Broadway Cast', '', '597']
['29', 'Kind of Blue', 'Miles Davis', '581']
['30', 'Orfeu Negro - Bande Originale du Film de Marcel Camus', '', '581']
['31', 'Georgia (On My Mind)', 'Hoagy Carmichael and His Orchestra', '565']
['32', 'Sleigh Ride', 'Boston Pops Orchestra', '546']
['33', 'Songs in the Key of Life', 'Stevie Wonder', '545']
['34', 'Lover Man (Oh, Where Can You Be?)', 'Billie Holiday', '541']
['35', 'Giant Steps', 'John Coltrane', '537']
['36', 'Love Letters', 'Victor Young and His Concert Orchestra', '536']
['37', 'Rudolph, the Red-Nosed Reindeer', 'Gene Autry and The Pinafores with Orchestral Accompaniment', '535']
['38', 'Imagine', 'John Lennon', '532']
['39', 'Willow Weep for Me', 'Ted Fiorito & His Orchestra', '528']
['40', 'Harlem on Parade', 'Gene Krupa & His Orchestra', '515']
['41', 'Fools Rush In', 'Chick Bullock & His Orchestra', '512']
['42', 'Blue', 'Joni Mitchell', '506']
['43', 'A Fine Romance', 'Guy Lombardo and His Royal Canadians', '502']
['44', 'Yearning Just for You', 'Ben Bernie and His Hotel Roosevelt Orchestra', '501']
['45', 'A Charlie Brown Christmas', 'Vince Guaraldi', '496']

['46', 'In a Sentimental Mood', 'Duke Ellington and His Orchestra', '491']
['47', 'Love Is Here to Stay', 'Ella Logan', '489']
['48', 'A Shine on Your Shoes', 'Leo Reisman and His Orchestra', '486']
['49', 'Equinox', "Sergio Mendes & Brasil '66", '484']
['50', "Don't Do Something to Someone Else (That You Wouldn't Want Done to You)", 'Gordon Jenkins and His Orchestra', '484']
['51', 'What a Wonderful World', 'Louis Armstrong', '482']
['52', 'The Sandpiper - Original Motion Picture Soundtrack', 'Johnny Mandel', '481']
['53', 'Thriller', 'Michael Jackson', '480']
['54', 'Talking Book', 'Stevie Wonder', '480']
['55', 'Blue Christmas', "Doye O'Dell", '480']
['56', 'Poor Hawthorne', 'Ukrainian National Chorus', '477']
['57', 'Nature Boy', 'King Cole', '465']
['58', "I'll Follow You", 'Paul Whiteman and His Orchestra with vocal refrain by Red McKenzie', '465']
['59', 'Thelonious', 'Thelonious Monk Trio', '464']
['60', 'Misty', 'Erroll Garner Trio', '461']
['61', 'West Side Story [OBC]', 'Leonard Bernstein with Irwin Kostal and Sid Ramin - Original Broadway Cast', '459']
['62', 'Cry Me a River', 'Julie London', '454']
['63', 'Nevermind', 'Nirvana [US]', '447']
['64', 'Hey Jude', 'The Beatles', '447']
['65', 'Blue Hawaii', 'Elvis Presley', '444']
['66', 'Turn on the Heat', 'Bert Stock and His Orchestra', '443']
['67', 'Fever', 'Little Willie John', '438']
['68', '"Tryout" - A Series of Private Rehearsal Recordings', 'Kurt Weill and Ira Gershwin', '437']
['69', "They Can't Take That Away from Me", 'Fred Astaire with Johnny Green and His Orchestra', '437']
['70', 'True', 'Al Bowlly', '434']
['71', 'Angel Eyes', 'Herb Jeffries', '431']
['72', 'Merry-Go-Round', 'Duke Ellington and His Orchestra', '429']
['73', 'The Sound of Music', '', '426']
['74', 'Never No Lament', 'Duke Ellington and His Famous Orchestra', '423']
['75', 'Various Positions', 'Leonard Cohen', '421']
['76', 'Oh Lonesome Me', 'Don Gibson', '418']
['77', 'Take the "A" Train', 'Duke Ellington and His Famous Orchestra', '418']
['78', 'Wildflowers', 'Judy Collins', '416']
['79', 'You Go to My Head', 'Larry Clinton & His Orchestra', '411']
['80', 'Music from Beyond the Moon', 'Vic Damone and Music by Camarata', '409']
['81', "When You Wish Upon a Star - I've Got No Strings", 'Cliff Edwards with Victor Young and His Orchestra and The Ken Darby Singers', '402']
['82', 'The Wall', 'Pink Floyd', '400']
['83', "You'd Be So Nice to Come Home To", 'Dick Jurgens and His Orchestra', '398']
['84', 'What the World Needs Now - Stan Getz Plays Bacharach and David', 'Stan Getz', '396']

```
['85', 'New Britain', 'The Original Sacred Harp Choir', '395']
['86', 'God Bless the Child', 'Billie Holiday', '387']
['87', 'The Bootleg Series Volumes 1-3', 'Bob Dylan', '387']
['88', 'The Joshua Tree', 'U2', '385']
['89', "Don't Blame Me", "Sarah Vaughan with George Treadwell's
Orchestra", '383']
['90', "Comme d'habitude", 'Claude François', '382']
['91', 'Secret Love', 'Doris Day', '381']
['92', 'Carnegie Hall, November 13, 1948', 'Duke Ellington and His
Orchestra', '380']
['93', 'Innervisions', 'Stevie Wonder', '379']
['94', 'Bye Bye Blackbird', "Sam Lanin's Dance Orchestra", '378']
['95', "After You've Gone / When We Meet in the Sweet Bye and Bye",
'Henry Burr & Albert Campbell - Sterling Trio', '378']
['96', "I've Got You Under My Skin", 'Frances Langford', '378']
['97', 'Bringing It All Back Home', 'Bob Dylan', '377']
['98', 'Let It Be', 'The Beatles', '371']
['99', 'Dreamy Blues', 'The Jungle Band', '371']
['100', 'Saturday Night Fever', '', '370']
```

Another way we could've approached this scenario would be to use Pandas, which may have been easier. However, I'd prefer to use BeautifulSoup for purposes of this example.

[60]
```
#table = pd.read_html('https://secondhandsongs.com/statistics?
sort=covers&list=stats_release_covers')[0]
#print(table)
```

[61]
```
df = pd.DataFrame(frame)
df.columns = ['Rank', 'Album', 'Artist', 'Count']
df.reset_index()
df.drop(0, inplace=True)
#Remove the zero index. Now everything is ranked 1-100
```

[62]
```
df.tail(5)
```

	Rank	Album	Artist	Count
96	96	I've Got You Under My Skin	Frances Langford	378

	Rank	Album	Artist	Count

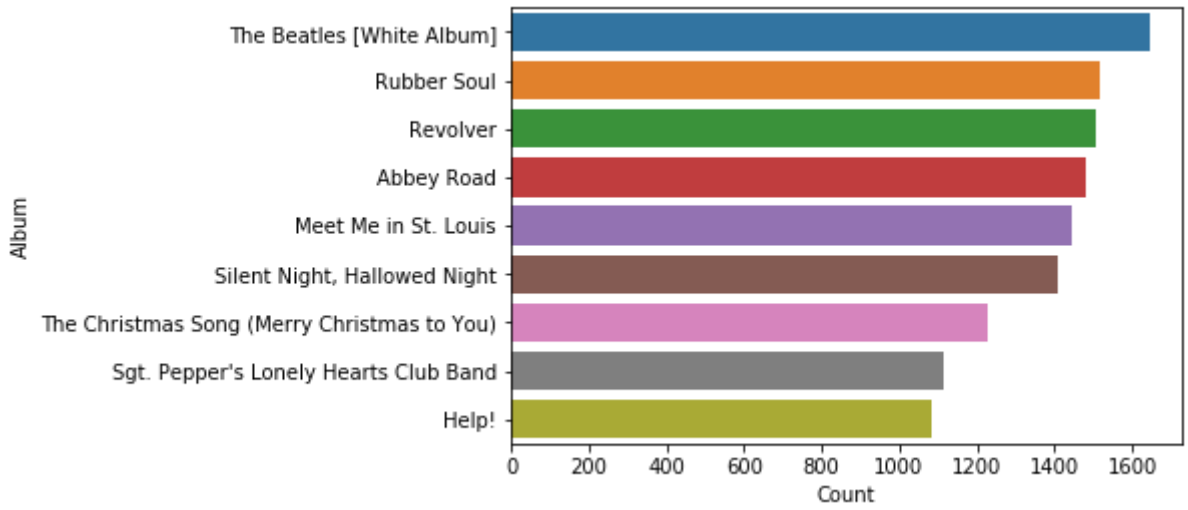
97	97	Bringing It All Back Home	Bob Dylan	377
98	98	Let It Be	The Beatles	371
99	99	Dreamy Blues	The Jungle Band	371
100	100	Saturday Night Fever		370

[63]
```
#Create a copy of our DataFrame for some Data Visualization.
df1 = df[0:9]
df1 = df1.astype({'Count':int,'Rank':int}) #Converting the
columns to be numeric, in order to be used for Plotting.
```

We will use the Seaborn library (my favorite for Data Visualization), to create a graph to analyze the data more easily. This graph represents the top 10 covered Albums and their respective counts. We can clearly tell that for some reason, 6 of the 10 most covered records belong to The Beatles!

[64]
```
sns.barplot(x= "Count",y="Album",data=df1)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c4ea799198>
```



[65]
```
df['Count'] = df['Count'].astype(int)
```

[66]
```
df['Count'].sum()
```

```
59705
```

[67]
```
#The total Count of covers belonging to a Beatles Album:
Beatle = df.loc[df['Artist'] == 'The Beatles', 'Count'].sum()
Beatle
```

```
10067
```

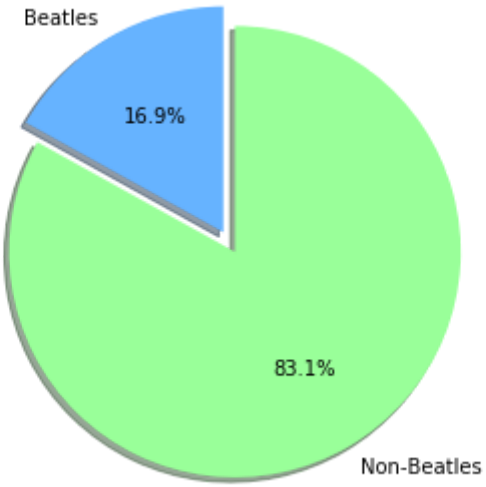
[68]
```
#The total Count of covers not belonging to a Beatles Album:
Not_beatle = Beatle = df.loc[df['Artist'] != 'The Beatles',
'Count'].sum()
Not_beatle
```

```
49638
```

[69]
```
labels = ['Beatles', 'Non-Beatles']
sizes = [9977,48948]
colors = ['gold', 'yellowgreen']
explode = [0.1, 0]
```

Using a simple Pie chart, we can see that the Beatles hold roughly 1/6 of the total number of Covers, in addition to holding the top 4 ranks. People clearly love this band!

[70]
```
plt.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
colors = ['#66b3ff', '#99ff99'],
        shadow=True, startangle=90)
plt.tight_layout()
plt.axis('equal')
plt.show()
```



This is the end of this project. We've conducted WebScraping & Data Manipulation/Visualization using various Python modules (BeautifulSoup, Numpy, Pandas, and Seaborn). Thanks for stopping by and please feel free to visit my Data Science blog: helloworldofdata.webnode.com