In [ ]:
```
# import library
import pandas as pd
import numpy as np
df = pd.read_csv('./data/BL-Flickr-Images-Book.csv')
df.head()
```

Out[ ]:

	Identifier	Edition Statement	Place of Publication	Date of Publication	Publisher	Title	Author	Con
0	206	NaN	London	1879 [1878]	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	
1	216	NaN	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A., A. A.	BU Pau -
2	218	NaN	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of “All for Gr...	A., A. A.	BU Pau -
3	472	NaN	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	A
4	480	A new edition, revised, etc.	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	A., E. S.	Jo

In [ ]:
```
to_drop = ['Edition Statement','Corporate Author','Corporate Contributors','Form
df.drop(to_drop, inplace=True, axis=1)
```

In [ ]:
```
df.head()
```

Out[ ]:

	Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr
0	206	London	1879 [1878]	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	http://www.flickr./photos/britishlib
1	216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A., A. A.	http://www.flickr./photos/britishlib
2	218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of "All for Gr...	A., A. A.	http://www.flickr./photos/britishlib
3	472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	http://www.flickr./photos/britishlib
4	480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	A., E. S.	http://www.flickr./photos/britishlib

Dari script di atas, apa yang akan terjadi? Berikan penjelasan!

Jawab:

Jadi pada fungsi di atas akan menghapus column

In [ ]:
```
df['Identifier'].is_unique
```

Out[ ]:
```
True
```

In [ ]:
```
df = df.set_index('Identifier')
df.head()
```

Out[ ]:

Identifier	Place of Publication	Date of Publication	Publisher	Title	Author	Flickr URL
206	London	1879 [1878]	S. Tinsley & Co.	Walter Forbes. [A novel.] By A. A	A. A.	http://www.flickr.com/photos/britishlibrary/ta..
216	London; Virtue & Yorston	1868	Virtue & Co.	All for Greed. [A novel. The dedication signed...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta..
218	London	1869	Bradbury, Evans & Co.	Love the Avenger. By the author of "All for Gr...	A., A. A.	http://www.flickr.com/photos/britishlibrary/ta..
472	London	1851	James Darling	Welsh Sketches, chiefly ecclesiastical, to the...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta..
480	London	1857	Wertheim & Macintosh	[The World in which I live, and my place in it...	A., E. S.	http://www.flickr.com/photos/britishlibrary/ta..

Silahkan coba script di atas dan apakah identifier berhasil menjadi index?

Jawab:

ya kolum indentifier berhasil menjadi index

In [ ]:
```
df.loc[206]
```

Out[ ]:
```
Place of Publication                                          London
Date of Publication                                     1879 [1878]
Publisher                                          S. Tinsley & Co.
Title                                  Walter Forbes. [A novel.] By A. A
Author                                                        A. A.
Flickr URL              http://www.flickr.com/photos/britishlibrary/ta...
Name: 206, dtype: object
```

In [ ]:
```
df.loc[1905:, 'Date of Publication'].head(10)
```

Out[ ]:
```
Identifier
1905           1888
1929    1839, 38-54
2836           1897
2854           1865
2956        1860-63
2957           1873
3017           1866
3131           1899
4598           1814
4884           1820
Name: Date of Publication, dtype: object
```

In [ ]:
```
extr = df['Date of Publication'].str.extract(r'^(\d{4})', expand=False)
extr.head()
```

Out[ ]:
```
Identifier
206    1879
216    1868
218    1869
472    1851
480    1857
Name: Date of Publication, dtype: object
```

Silahkan ikuti script di atas dan amati serta berikan penjelasan mengenani hasil yang didapat.

Jawab:

berfungsi untuk mengambil data tahun pada kolom Date of Publication

In [ ]:
```
df['Date of Publication'] = pd.to_numeric(extr)
df['Date of Publication'].dtype
```

Out[ ]:
```
dtype('float64')
```

In [ ]:
```
df['Place of Publication'].head(10)
```

Out[ ]:
```
Identifier
206                                    London
216                 London; Virtue & Yorston
218                                    London
472                                    London
480                                    London
481                                    London
519                                    London
667     pp. 40. G. Bryan & Co: Oxford, 1898
874                                   London]
1143                                   London
Name: Place of Publication, dtype: object
```

In [ ]:
```
pub = df['Place of Publication']
london = pub.str.contains('London')
oxford = pub.str.contains('Oxford')
```

In [ ]:
```
df['Place of Publication'] = np.where(london, 'London', np.where(oxford, 'Oxford
df['Place of Publication'].head()
```

Out[ ]:
```
Identifier
206    London
216    London
218    London
472    London
480    London
Name: Place of Publication, dtype: object
```

Silahkan ikuti script di atas, dan berikan penjelasan apa yang terjadi!

Jawab:

berfungsi untuk mengambil data tahun pada kolom Date of Publication berdasarkan regex yang di tentukan np.where

berfungsi untuk mengganti '-' dengan ' ' pada kolom Date of Publication

In [ ]:
```
university_towns = []
with open('./data/university_towns.txt') as file:
    for line in file:
        if '[edit]' in line:
            state = line
        else:
            university_towns.append((state, line))
university_towns[:5]
```

Out[ ]:
```
[('Alabama[edit]\n', 'Auburn (Auburn University)[1]\n'),
 ('Alabama[edit]\n', 'Florence (University of North Alabama)\n'),
 ('Alabama[edit]\n', 'Jacksonville (Jacksonville State University)[2]\n'),
 ('Alabama[edit]\n', 'Livingston (University of West Alabama)[2]\n'),
 ('Alabama[edit]\n', 'Montevallo (University of Montevallo)[2]\n')]
```

In [ ]:
```
towns_df = pd.DataFrame(university_towns, columns=['State', 'RegionName'])
towns_df.head()
```

Out[ ]:

	State	RegionName
0	Alabama[edit]\n	Auburn (Auburn University)[1]\n
1	Alabama[edit]\n	Florence (University of North Alabama)\n
2	Alabama[edit]\n	Jacksonville (Jacksonville State University)[2]\n
3	Alabama[edit]\n	Livingston (University of West Alabama)[2]\n
4	Alabama[edit]\n	Montevallo (University of Montevallo)[2]\n

Silahkan ikuti script di atas dan jelaskan apa hasil dari script tersebut.

Jawab:

membaca file "university_towns.txt" dan membentuk daftar university_towns. Dalam loop, kode memeriksa apakah baris berisi "[edit]". Jika ya, itu dianggap sebagai negara bagian. Jika tidak, baris ditambahkan ke daftar sebagai kota perguruan tinggi dengan negara bagian terakhir yang terdeteksi. Hasilnya adalah daftar tupel yang berisi pasangan negara bagian dan nama kota perguruan tinggi. Baris terakhir menampilkan lima entri pertama dalam daftar sebagai pemantauan.

towns_df membuat DataFrame dari university_towns dengan kolom 'State' dan 'RegionName'. head() menampilkan lima baris pertama, memudahkan pemeriksaan struktur dan isi DataFrame.

In [ ]:
```
def get_citystate(item):
    if ' (' in item:
        return item[:item.find(' (')]
    elif '[' in item:
        return item[:item.find('[')]
    else:
        return item

towns_df = towns_df.applymap(get_citystate)
```

In [ ]:
```
towns_df.head()
```

Out[ ]:

	State	RegionName
0	Alabama	Auburn
1	Alabama	Florence
2	Alabama	Jacksonville
3	Alabama	Livingston
4	Alabama	Montevallo

Silahkan ikuti script di atas dan berikan penjelasan mengenai hasilnya!

Jawab:

Fungsi get_citystate memproses setiap elemen DataFrame towns_df. Jika terdapat ' (' atau '[', hanya bagian sebelumnya yang diambil. Metode applymap memanggil fungsi ini pada setiap sel, membersihkan data dari informasi tambahan.

In [ ]:
```
olympics_df = pd.read_csv('./data/olympics.csv')
olympics_df.head()
```

Out[ ]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	1
0	NaN	? Summer	01 !	02 !	03 !	Total	? Winter	01 !	02 !	03 !	Total	? Games	01 !	02 !	0
1	Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	0	
2	Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	2	
3	Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	24	2
4	Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	2	

In [ ]:
```
olympics_df = pd.read_csv('./data/olympics.csv', header=1)
olympics_df.head()
```

Out[ ]:

	Unnamed: 0	? Summer	01 !	02 !	03 !	Total	? Winter	01 !.1	02 !.1	03 !.1	Total.1	? Games	01 !.2	0 !
0	Afghanistan (AFG)	13	0	0	2	2	0	0	0	0	0	13	0	
1	Algeria (ALG)	12	5	2	8	15	3	0	0	0	0	15	5	
2	Argentina (ARG)	23	18	24	28	70	18	0	0	0	0	41	18	
3	Armenia (ARM)	5	1	2	9	12	6	0	0	0	0	11	1	
4	Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	0	0	2	3	

Silahkan amati dan jelaskan perbedaan dari gambar sebelumnya!

Jawab:

Pada tabel sebelumnya header column nya berupa index, sedangkan pada tabel ini header column nya berupa nama column

In [ ]:
```
new_names =  {'Unnamed: 0': 'Country',
              '? Summer': 'Summer Olympics',
              '01 !': 'Gold',
              '02 !': 'Silver',
              '03 !': 'Bronze',
              '? Winter': 'Winter Olympics',
              '01 !.1': 'Gold.1',
              '02 !.1': 'Silver.1',
              '03 !.1': 'Bronze.1',
```

```
                '? Games': '# Games',
                '01 !.2': 'Gold.2',
                '02 !.2': 'Silver.2',
                '03 !.2': 'Bronze.2'}
olympics_df.rename(columns=new_names, inplace=True)
```

In [ ]:
```
olympics_df.head()
```

Out[ ]:

	Country	Summer Olympics	Gold	Silver	Bronze	Total	Winter Olympics	Gold.1	Silver.1	Bronz
0	Afghanistan (AFG)	13	0	0	2	2	0	0	0	
1	Algeria (ALG)	12	5	2	8	15	3	0	0	
2	Argentina (ARG)	23	18	24	28	70	18	0	0	
3	Armenia (ARM)	5	1	2	9	12	6	0	0	
4	Australasia (ANZ) [ANZ]	2	3	4	5	12	0	0	0	

Ikuti script di atas dan berikan penjelasan mengenai hasilnya!

Jawab:

mengganti nama kolom pada DataFrame olympics_df menggunakan variable new_names. Operasi ini memberi label kolom-kolom dengan nama yang lebih deskriptif untuk analisis Olimpiade, dan inplace=True mengubah DataFrame secara langsung.