WebScraping

December 29, 2021

Creating a class

[ ]:
```
class wikipedia_webscrap:
  def __init__(self):
    ''' The function reads the website, extracts the table from it and store it
    as a pandas DataFrame'''

    import pandas as pd
    # Read data from the specified website
    self.wiki_page=pd.read_html('https://en.wikipedia.org/wiki/
 ↪COVID-19_pandemic_by_country_and_territory')
    # Read data from the specified table
    self.covid_table=self.wiki_page[9]


  def drop_columns(self,remove_list,axis=1):
    ''' Pass the list of columns to be dropped to the function
    and it will drop the same from DataFrame'''
    self.covid_table.drop(remove_list,axis=axis,inplace=True)

  def check_info(self):
    '''Check the datatypes and null values in the dataset'''
    self.covid_table.info()
```

[ ]:
```
wiki=wikipedia_webscrap()
table=wiki.covid_table
wiki.drop_columns(['Country','Unnamed: 5', 'Unnamed: 6', 'Unnamed: 7'])
column_names=['country','deaths_per_million','deaths','recovered']
table.columns=column_names
wiki.check_info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 218 entries, 0 to 217
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   country             218 non-null    object
 1   deaths_per_million  218 non-null    object
 2   deaths              218 non-null    object
```

```
 3   recovered           218 non-null    object
dtypes: object(4)
memory usage: 6.9+ KB
```

Checking for improper value in country column to drop

[ ]:
```
table['country'].unique()
```

[ ]:
```
array(['World[a]', 'Peru', 'Bulgaria', 'Bosnia and Herzegovina',
       'Hungary', 'Montenegro', 'North Macedonia', 'Georgia',
       'Czech Republic', 'Romania', 'Croatia', 'Slovakia', 'Gibraltar',
       'Brazil', 'San Marino', 'Lithuania', 'Armenia', 'Slovenia',
       'Argentina', 'Colombia', 'Poland', 'United States', 'Belgium',
       'Latvia', 'Moldova', 'Ukraine', 'Paraguay', 'Mexico', 'Italy',
       'French Polynesia', 'United Kingdom', 'Tunisia', 'Russia', 'Chile',
       'Suriname', 'European Union[b]', 'Trinidad and Tobago', 'Greece',
       'Spain', 'Ecuador', 'Portugal', 'Serbia', 'France',
       'Liechtenstein', 'Andorra', 'Bahamas', 'Grenada', 'Uruguay',
       'Bermuda', 'Panama', 'Aruba', 'Kosovo', 'Bolivia', 'Saint Lucia',
       'Iran', 'South Africa', 'Austria', 'Sweden', 'Belize', 'Estonia',
       'Luxembourg', 'Costa Rica', 'Namibia', 'Switzerland', 'Lebanon',
       'Guyana', 'Seychelles', 'Germany', 'British Virgin Islands',
       'Netherlands', 'Jordan', 'Antigua and Barbuda',
       'Republic of Ireland', 'Curaçao', 'Albania', 'Eswatini',
       'Honduras', 'Botswana', 'New Caledonia', 'Monaco', 'Turkey',
       'Kazakhstan', 'Malaysia', 'Palestine', 'Malta', 'Barbados',
       'Israel', 'Guatemala', 'Caribbean Netherlands', 'Jamaica', 'Libya',
       'Azerbaijan', 'Bahrain', 'Canada', 'Oman', 'Isle of Man', 'Fiji',
       'Cuba', 'Saint Vincent and the Grenadines', 'Cyprus', 'Sri Lanka',
       'Turks and Caicos Islands', 'Wallis and Futuna', 'Cabo Verde',
       'Dominica', 'Mongolia', 'Iraq', 'El Salvador', 'Belarus', 'Kuwait',
       'Denmark', 'Saint Kitts and Nevis', 'Indonesia', 'Maldives',
       'Philippines', 'Kyrgyzstan', 'Morocco', 'Nepal',
       'Dominican Republic', 'Myanmar', 'India', 'Anguilla', 'Zimbabwe',
       'Vietnam', 'Thailand', 'Lesotho', 'Finland', 'Faroe Islands',
       'Sao Tome and Principe', 'Saudi Arabia', 'Norway', 'Brunei',
       'United Arab Emirates', 'Qatar', 'Egypt', 'Montserrat', 'Zambia',
       'Djibouti', 'Mauritius', 'Venezuela', 'Afghanistan', 'Mauritania',
       'Cambodia', 'Comoros', 'Bangladesh', 'Cayman Islands', 'Syria',
       'Singapore', 'Japan', 'Algeria', 'Gambia', 'Pakistan', 'Gabon',
       'Equatorial Guinea', 'Malawi', 'Senegal', 'Iceland', 'South Korea',
       'Rwanda', 'Kenya', 'Timor-Leste', 'Australia', 'Somalia',
       'Guinea-Bissau', 'Sudan', 'Uganda', 'Cameroon', 'Haiti', 'Yemen',
       'Republic of the Congo', 'Papua New Guinea', 'Mozambique',
       'Ethiopia', 'Liberia', 'Angola', 'Laos', 'Uzbekistan', 'Ghana',
       'Madagascar', 'Taiwan', 'Nicaragua', 'Mali', 'Togo', 'Guinea',
       'Hong Kong', 'Ivory Coast', 'Central African Republic', 'Eritrea',
       'Greenland', 'Sierra Leone', 'Burkina Faso', 'Nigeria',
```

```
        'Democratic Republic of the Congo', 'Benin', 'Tajikistan',
        'Tanzania', 'South Sudan', 'Niger', 'Chad', 'New Zealand',
        'Bhutan', 'China[c]', 'Vanuatu', 'Burundi', 'Falkland Islands',
        'Solomon Islands', 'Samoa', 'Cook Islands', 'Marshall Islands',
        'Saint Pierre and Miquelon', 'Palau',
        'Federated States of Micronesia', 'Vatican City',
        'Saint Helena, Ascension and Tristan da Cunha', 'Macau',
        'Kiribati', 'Tonga',
        ".mw-parser-output .reflist{font-size:90%;margin-bottom:0.5em;list-style-
type:decimal}.mw-parser-output .reflist .references{font-size:100%;margin-
bottom:0;list-style-type:inherit}.mw-parser-output .reflist-columns-2{column-
width:30em}.mw-parser-output .reflist-columns-3{column-width:25em}.mw-parser-
output .reflist-columns{margin-top:0.3em}.mw-parser-output .reflist-columns
ol{margin-top:0}.mw-parser-output .reflist-columns li{page-break-
inside:avoid;break-inside:avoid-column}.mw-parser-output .reflist-upper-
alpha{list-style-type:upper-alpha}.mw-parser-output .reflist-upper-roman{list-
style-type:upper-roman}.mw-parser-output .reflist-lower-alpha{list-style-
type:lower-alpha}.mw-parser-output .reflist-lower-greek{list-style-type:lower-
greek}.mw-parser-output .reflist-lower-roman{list-style-type:lower-roman} ^
Countries which do not report data for a column are not included in that
column's world total. ^ Data on member states of the European Union are
individually listed, but are also summed here for convenience. They are not
double-counted in world totals. ^ Does not include special administrative
regions (Hong Kong and Macau) or Taiwan."],
      dtype=object)
```

The last two rows are to be removed as they do not represent any country or region

[ ]:
```
wiki.drop_columns([table.shape[0]-1,table.shape[0]-2],0)
```

[ ]:
```
table['country'].unique()
```

[ ]:
```
array(['World[a]', 'Peru', 'Bulgaria', 'Bosnia and Herzegovina',
       'Hungary', 'Montenegro', 'North Macedonia', 'Georgia',
       'Czech Republic', 'Romania', 'Croatia', 'Slovakia', 'Gibraltar',
       'Brazil', 'San Marino', 'Lithuania', 'Armenia', 'Slovenia',
       'Argentina', 'Colombia', 'Poland', 'United States', 'Belgium',
       'Latvia', 'Moldova', 'Ukraine', 'Paraguay', 'Mexico', 'Italy',
       'French Polynesia', 'United Kingdom', 'Tunisia', 'Russia', 'Chile',
       'Suriname', 'European Union[b]', 'Trinidad and Tobago', 'Greece',
       'Spain', 'Ecuador', 'Portugal', 'Serbia', 'France',
       'Liechtenstein', 'Andorra', 'Bahamas', 'Grenada', 'Uruguay',
       'Bermuda', 'Panama', 'Aruba', 'Kosovo', 'Bolivia', 'Saint Lucia',
       'Iran', 'South Africa', 'Austria', 'Sweden', 'Belize', 'Estonia',
       'Luxembourg', 'Costa Rica', 'Namibia', 'Switzerland', 'Lebanon',
       'Guyana', 'Seychelles', 'Germany', 'British Virgin Islands',
       'Netherlands', 'Jordan', 'Antigua and Barbuda',
       'Republic of Ireland', 'Curaçao', 'Albania', 'Eswatini',
```

```
'Honduras', 'Botswana', 'New Caledonia', 'Monaco', 'Turkey',
'Kazakhstan', 'Malaysia', 'Palestine', 'Malta', 'Barbados',
'Israel', 'Guatemala', 'Caribbean Netherlands', 'Jamaica', 'Libya',
'Azerbaijan', 'Bahrain', 'Canada', 'Oman', 'Isle of Man', 'Fiji',
'Cuba', 'Saint Vincent and the Grenadines', 'Cyprus', 'Sri Lanka',
'Turks and Caicos Islands', 'Wallis and Futuna', 'Cabo Verde',
'Dominica', 'Mongolia', 'Iraq', 'El Salvador', 'Belarus', 'Kuwait',
'Denmark', 'Saint Kitts and Nevis', 'Indonesia', 'Maldives',
'Philippines', 'Kyrgyzstan', 'Morocco', 'Nepal',
'Dominican Republic', 'Myanmar', 'India', 'Anguilla', 'Zimbabwe',
'Vietnam', 'Thailand', 'Lesotho', 'Finland', 'Faroe Islands',
'Sao Tome and Principe', 'Saudi Arabia', 'Norway', 'Brunei',
'United Arab Emirates', 'Qatar', 'Egypt', 'Montserrat', 'Zambia',
'Djibouti', 'Mauritius', 'Venezuela', 'Afghanistan', 'Mauritania',
'Cambodia', 'Comoros', 'Bangladesh', 'Cayman Islands', 'Syria',
'Singapore', 'Japan', 'Algeria', 'Gambia', 'Pakistan', 'Gabon',
'Equatorial Guinea', 'Malawi', 'Senegal', 'Iceland', 'South Korea',
'Rwanda', 'Kenya', 'Timor-Leste', 'Australia', 'Somalia',
'Guinea-Bissau', 'Sudan', 'Uganda', 'Cameroon', 'Haiti', 'Yemen',
'Republic of the Congo', 'Papua New Guinea', 'Mozambique',
'Ethiopia', 'Liberia', 'Angola', 'Laos', 'Uzbekistan', 'Ghana',
'Madagascar', 'Taiwan', 'Nicaragua', 'Mali', 'Togo', 'Guinea',
'Hong Kong', 'Ivory Coast', 'Central African Republic', 'Eritrea',
'Greenland', 'Sierra Leone', 'Burkina Faso', 'Nigeria',
'Democratic Republic of the Congo', 'Benin', 'Tajikistan',
'Tanzania', 'South Sudan', 'Niger', 'Chad', 'New Zealand',
'Bhutan', 'China[c]', 'Vanuatu', 'Burundi', 'Falkland Islands',
'Solomon Islands', 'Samoa', 'Cook Islands', 'Marshall Islands',
'Saint Pierre and Miquelon', 'Palau',
'Federated States of Micronesia', 'Vatican City',
'Saint Helena, Ascension and Tristan da Cunha', 'Macau',
'Kiribati'], dtype=object)
```

Replacing the '[charachter]' in country

[ ]:
```
countries=[]
import re
for country in table['country']:
  if len(re.findall('\[\w\]',country))>0:
    countries.append(re.findall('(\w*)\[',country)[0])
  else:
    countries.append(country)
table['country']=countries
```

[ ]:
```
table.head()
```

[ ]:

	country	deaths_per_million	deaths	recovered
0	World	686	5406818	281400646

1	Peru	6070	202524	2279299
2	Bulgaria	4440	30623	735998
3	Bosnia and Herzegovina	4083	13325	288128
4	Hungary	4021	38743	1245319

Assigning country name as index

[ ]:
```
table.index=table['country']
table.drop(table.columns[0],axis=1,inplace=True)
table.head()
```

[ ]:

country	deaths_per_million	deaths	recovered
World	686	5406818	281400646
Peru	6070	202524	2279299
Bulgaria	4440	30623	735998
Bosnia and Herzegovina	4083	13325	288128
Hungary	4021	38743	1245319

Checking for undefined values in columns

[ ]:
```
drop_index_list=[]

for i,row in table.iterrows():
  try:
    _=int(row['deaths_per_million'])
    _=int(row['deaths'])
    _=int(row['recovered'])
    if int(row['deaths'])==0:
      print(i,' has 0 recorded deaths')
  except:
    drop_index_list.append(i)

print('The below list of countries has missing data')
drop_index_list
```

```
The below list of countries has missing data
```

[ ]:
```
['Falkland Islands',
 'Solomon Islands',
 'Samoa',
 'Cook Islands',
 'Marshall Islands',
 'Saint Pierre and Miquelon',
 'Palau',
 'Federated States of Micronesia',
 'Vatican City',
 'Saint Helena, Ascension and Tristan da Cunha',
 'Macau',
```

```
 'Kiribati']
```

We can see that the above Countries have entires with non numeric values. Hence dropping them

```
[ ]: table.drop(drop_index_list,axis=0,inplace=True)
```

Converting the columns to integer type

```
[ ]: for column in table.columns:
       table[column]=table[column].astype('int')
     table.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 204 entries, World to Burundi
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   deaths_per_million  204 non-null    int64
 1   deaths              204 non-null    int64
 2   recovered           204 non-null    int64
dtypes: int64(3)
memory usage: 6.4+ KB
```

Creating new column recovered_per_deaths

```
[ ]: table['recovered_per_deaths']=table['recovered']/table['deaths']
     table.head()
```

```
[ ]:                        deaths_per_million  …  recovered_per_deaths
     country                                    …
     World                                 686  …             52.045518
     Peru                                 6070  …             11.254464
     Bulgaria                             4440  …             24.034157
     Bosnia and Herzegovina               4083  …             21.623114
     Hungary                              4021  …             32.143071

     [5 rows x 4 columns]
```

Sorting the table by recovered per deaths

```
[ ]: table=table.sort_values(by='recovered_per_deaths',ascending=False)
```

```
[ ]: table.head()
```

```
[ ]:                deaths_per_million  deaths  recovered  recovered_per_deaths
     country
     Greenland                      17       1       2306           2306.000000
     Bhutan                          3       3       2660            886.666667
     Cayman Islands                165      11       8386            762.363636
     Burundi                         3      38      26224            690.105263
     Iceland                       107      37      24340            657.837838
```

- Higher value of the ratio indicates that more people have recovered in the country and less people had passed away due to covid.
- Lower the ratio of recovered_per_death indicates that less people have died in the country due to Covid.
- This calculation cannot be completely relivd upon. Because there are countries like Vanuatu, where the number of cases are very less, ie 7. But still they have very low value of recovered_per_death because they have total of 7 cases and 1 deaths. Hence this data needs to be further normalised

[ ]:
```
table.sort_values(by='recovered_per_deaths',ascending=True).head(30)
```

[ ]:

country	deaths_per_million	…	recovered_per_deaths
Yemen	65	…	5.099798
Vanuatu	3	…	7.000000
Peru	6070	…	11.254464
Mexico	2293	…	13.223920
Sudan	73	…	13.998181
Ecuador	1881	…	16.117596
Syria	157	…	17.411254
Somalia	81	…	17.653413
Egypt	207	…	17.662276
Taiwan	35	…	19.918824
Afghanistan	184	…	21.480419
Bosnia and Herzegovina	4083	…	21.623114
Liberia	55	…	21.700348
China	3	…	21.890854
Bulgaria	4440	…	24.034157
Niger	10	…	26.602190
Myanmar	351	…	27.512749
Paraguay	2300	…	28.034202
North Macedonia	3796	…	28.244593
Tunisia	2139	…	28.327277
Bolivia	1652	…	29.549351
Indonesia	521	…	29.583439
Gambia	137	…	29.637427
Grenada	1769	…	29.805000
Mali	31	…	30.577508
Malawi	118	…	30.616602
Romania	3062	…	30.786359
Chad	10	…	31.508287
El Salvador	586	…	31.869372
Hungary	4021	…	32.143071

[30 rows x 4 columns]

[ ]: