국가별 백신 접종률 정보 가져오기

정보 수집 사이트

- bloomberg.com
  - https://www.bloomberg.com/graphics/covid-vaccine-tracker-global-distribution/ (https://www.bloomberg.com/graphics/covid-vaccine-tracker-global-distribution/)

기타 참고 통계 사이트

- our world in data
  - https://ourworldindata.org/covid-vaccinations (https://ourworldindata.org/covid-vaccinations)

In [1]:

```
from IPython.display import display, Image
import os, warnings
import re
warnings.filterwarnings(action='ignore')
```

01 웹 브라우저 띄우기

- 만약 chrome 브라우저와 chromedirver의 버전이 안 맞을 경우, 버전을 맞는 것으로 변경해야 함.(가끔 이 부분에서 에러 발생)
  - 'chrome driver download'로 검색 후, 사이트에 접근 후, 다운로드 가능(window, linux, mac 버전 있음)

In [2]:

```
from selenium import webdriver
from bs4 import BeautifulSoup

driver = webdriver.Chrome('./chromedriver_91')

url = 'https://www.bloomberg.com/graphics/covid-vaccine-tracker-global-distribution/'
driver.get(url)
```

In [3]:

```
import time
time.sleep(3)  # 홈페이지 로딩 시간 3초
```

전체 데이터 보기

- 나라가 여러나라가 있어, 더 보기 버튼을 2번 정도 눌러준다.

In [4]:

```
# //*[@id="dvz-table-global-vaccination"]/div[2]/div[2]/button
# //*[@id="dvz-table-global-vaccination"]/div[2]/div[2]/button
sel_more1 = driver.find_element_by_xpath('//*[@id="dvz-table-global-vaccination"]/div[2]/div[2]/butt
sel_more1.click()
time.sleep(1)
```

In [5]:

```
# //*[@id="dvz-table-usa-vaccination"]/div[2]/div[2]/button
sel_more2 = driver.find_element_by_xpath('//*[@id="dvz-table-global-vaccination"]/div[2]/div[2]/butt
sel_more2.click()
```

TABLE 선택 후, 데이터 가져오기

Countries and regions

- //*[@id="dvz-table-global-vaccination"]/div[2]/div[1]/table/tbody/tr[1]/td[1]
- ..
- //*[@id="dvz-table-global-vaccination"]/div[2]/div[1]/table/tbody/tr[3]/td[1]

Doses administered

- //*[@id="dvz-table-global-vaccination"]/div[2]/div[1]/table/tbody/tr[1]/td[2]

Enough for % of people

- //*[@id="dvz-table-global-vaccination"]/div[2]/div[1]/table/tbody/tr[1]/td[3]

given 1+ dose

- //*[@id="dvz-table-global-vaccination"]/div[2]/div[1]/table/tbody/tr[1]/td[4]

fully vaccinated

- //*[@id="dvz-table-global-vaccination"]/div[2]/div[1]/table/tbody/tr[1]/td[5]

Daily rate of doses administered

- //*[@id="dvz-table-global-vaccination"]/div[2]/div[1]/table/tbody/tr[1]/td[6]

In [6]:

```
all_data = []

for i in range(1, 7, 1):
    data_col = []
    xpath = '//*[@id="dvz-table-global-vaccination"]/div[2]/div[1]/table/tbody/tr/td[%s]' % str(i)
    sel_data = driver.find_elements_by_xpath(xpath)

    for dat in sel_data:
        #print(dat)
        data_col.append(dat.text)
    print(data_col)
    all_data.append(data_col)
```

```
['Global Total', 'Mainland China', 'India', 'EU', 'U.S.', 'Brazil', 'Japan', 'Indone
sia', 'Turkey', 'Germany', 'Mexico', 'France', 'U.K.', '', '', '', '', '', 'Russia',
'Italy', 'Pakistan', 'Spain', 'South Korea', 'Canada', '', '', '', '', '', '', '',
'', '', '', '', '', '', 'Argentina', 'Thailand', 'Philippines', 'Iran', 'Malaysia',
'Saudi Arabia', 'Bangladesh', 'Morocco', 'Colombia', 'Vietnam', 'Poland', 'Chile',
'Australia', 'Sri Lanka', 'Peru', 'Cambodia', 'Netherlands', 'Ecuador', 'Cuba', 'UA
E', 'Uzbekistan', 'Ukraine', 'South Africa', 'Belgium', 'Portugal', 'Egypt', 'Israe
l', 'Taiwan', 'Algeria', 'Venezuela', 'Kazakhstan', 'Sweden', 'Nepal', 'Greece', 'Do
minican Republic', 'Czech Republic', 'Hungary', 'Austria', 'Switzerland', '', 'Roman
ia', 'Singapore', 'Myanmar', 'Denmark', 'Hong Kong', 'Azerbaijan', 'Norway', 'Finlan
d', 'Iraq', 'Tunisia', 'El Salvador', 'Ireland', 'Jordan', 'Guatemala', 'Bolivia',
'Nigeria', 'Serbia', 'Uruguay', 'Panama', 'Costa Rica', 'Zimbabwe', 'Honduras', 'New
Zealand', 'Slovakia', 'Qatar', 'Oman', 'Mongolia', 'Paraguay', 'Laos', 'Tajikistan',
'Mozambique', 'Kenya', 'Ethiopia', 'Rwanda', 'Croatia', 'Belarus', 'Lithuania', 'Ang
ola', 'Afghanistan', 'Lebanon', 'Bahrain', 'Bulgaria', 'Kuwait', 'Slovenia', 'Ugand
a', 'Ivory Coast', 'Senegal', 'Georgia', 'Guinea', 'Albania', 'Sudan', 'Ghana', 'Mau
ritius', 'Latvia', 'Libya', 'North Macedonia', 'Kyrgyzstan', 'Moldova', 'Estonia',
'Kosovo', 'Cyprus', 'Malawi', 'Bhutan', 'Bosnia and Herzegovina', 'Trinidad and Toba
go', 'Togo', 'Fiji', 'Mauritania', 'Malta', 'Nicaragua', 'Luxembourg', 'Maldives',
```

In [7]:

```
import pandas as pd
```

데이터 확인

- Countries and regions : 나라 및 지역 / country 컬럼
- Doses administered : 접종 수 / Doses_administered 컬럼
- Enough for % of people : 접종률 / percent_of_people 컬럼
- given 1+ dose : 1차 접종 / 1_percent 컬럼
- fully vaccinated : 2차 접종 / 2_percent 컬럼
- Daily_rate_of_doses_administered : 일일 투여 용량 / Daily_rate_of_doses 컬럼

In [55]:

```
pd.set_option("display.max_rows", 40)

dat_dict = {'국가':all_data[0],
            '백신접종수':all_data[1],
            'Enough_for_percent_of_people:':all_data[2],
            '1차접종':all_data[3],
            '2차접종':all_data[4],
            '일별접종수':all_data[5]
           }

dat_df = pd.DataFrame(dat_dict)
dat_df
```

Out[55]:

	국가	백신접종수	Enough_for_percent_of_people:	1차접종	2차접종	일별접종수
0	Global Total	6,172,363,261	–	–	–	31,740,869
1	Mainland China	2,200,202,000	78.6	78.6	73.0	3,223,429
2	India	870,566,939	31.8	46.2	16.4	7,488,509
3	EU	562,942,153	63.4	67.2	64.6	999,304
4	U.S.	390,114,328	60.9	64.3	55.3	716,762
...	...	...	...	...	...	...
221	Nauru	14,863	57.2	58.6	55.8	4
222	St. Helena	7,892	65.8	72.7	58.9	8
223	Falkland Islands	4,407	73.5	87.7	59.2	202
224	Montserrat	2,871	28.7	29.7	27.8	2
225	Eritrea	–	–	–	–	–

226 rows × 6 columns

In [56]:

```
dat_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 226 entries, 0 to 225
Data columns (total 6 columns):
 #   Column                          Non-Null Count  Dtype 
---  ------                          --------------  ----- 
 0   국가                              226 non-null    object
 1   백신접종수                           226 non-null    object
 2   Enough_for_percent_of_people:  226 non-null    object
 3   1차접종                            226 non-null    object
 4   2차접종                            226 non-null    object
 5   일별접종수                           226 non-null    object
dtypes: object(6)
memory usage: 10.7+ KB
```

데이터 전처리

- 데이터가 없거나 제대로 얻어지지 못한 부분. 그리고 추가 컬럼 등을 생성
- [].str.len() : 데이터의 길이

In [57]:

```
### 공백행을 삭제
dat_df['국가'].str.len()
```

Out[57]:

```
0      12
1      14
2       5
3       2
4       4
       ..
221     5
222    10
223    16
224    10
225     7
Name: 국가, Length: 226, dtype: int64
```

In [58]:

```
dat_df['국가'].str.len().unique()
```

Out[58]:

```
array([12, 14,  5,  2,  4,  6,  9,  7,  0,  8, 11, 10,  3, 18, 15, 22, 19,
       17, 21, 24, 16, 13, 30], dtype=int64)
```

한나라의 중복 행의 존재로 이 부분은 정보 취득 못함

In [59]:

```
dat_df.loc[ dat_df['국가'].str.len() < 1, : ]
```

Out[59]:

	국가	백신접종수	Enough_for_percent_of_people:	1차접종	2차접종	일별접종수
13						
14						
15						
16						
17						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
76						

데이터가 없는 인덱스를 얻어서 해당 행을 삭제한다.

In [60]:

```
sel_index = dat_df[ dat_df['국가'].str.len() <1 ].index
print(sel_index)
print(dat_df.shape)
dat_df.drop (sel_index, axis=0, inplace=True  )
print(dat_df.shape)
```

```
Int64Index([13, 14, 15, 16, 17, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
            36, 76],
           dtype='int64')
(226, 6)
(207, 6)
```

In [61]:

```
dat_df.head(15)
```

Out[61]:

	국가	백신접종수	Enough_for_percent_of_people:	1차접종	2차접종	일별접종수
0	Global Total	6,172,363,261	–	–	–	31,740,869
1	Mainland China	2,200,202,000	78.6	78.6	73.0	3,223,429
2	India	870,566,939	31.8	46.2	16.4	7,488,509
3	EU	562,942,153	63.4	67.2	64.6	999,304
4	U.S.	390,114,328	60.9	64.3	55.3	716,762
5	Brazil	232,250,878	56.3	71.1	41.4	3,757,702
6	Japan	159,494,782	63.2	68.9	57.4	1,026,446
7	Indonesia	136,941,018	25.7	32.2	18.1	1,755,924
8	Turkey	108,344,725	65.1	64.4	52.7	386,308
9	Germany	107,030,469	64.4	67.8	64.1	194,734
10	Mexico	99,366,403	38.9	49.7	34.9	532,540
11	France	93,817,818	72.4	77.4	74.4	435,139
12	U.K.	93,500,858	70.0	73.0	67.0	63,103
18	Russia	89,682,021	30.6	32.4	28.4	220,487
19	Italy	84,158,581	69.7	74.3	74.4	222,470

인덱스가 일정하지 않아, 인덱스 값 초기화

In [62]:

```
dat_df = dat_df.reindex()
dat_df.shape
```

Out[62]:

```
(207, 6)
```

이상치 '-' 값 확인

In [63]:

```
dat_df.loc[dat_df['백신접종수'] == '-']
```

Out[63]:

	국가	백신접종수	Enough_for_percent_of_people:	1차접종	2차접종	일별접종수
225	Eritrea	–	–	–	–	–

In [65]:

```
dat_df.loc[dat_df['일별접종수'] == '-']
```

Out[65]:

	국가	백신접종수	Enough_for_percent_of_people:	1차접종	2차접종	일별접종수
218	Turkmenistan	41,993	0.4	0.6	0.2	–
225	Eritrea	–	–	–	–	–

In [66]:

```
dat_df.columns
```

Out[66]:

```
Index(['국가', '백신접종수', 'Enough_for_percent_of_people:', '1차접종', '2차접종',
       '일별접종수'],
      dtype='object')
```

In [67]:

```
col_all = dat_df.columns
for one in col_all:
    print("col name : ", one)
    print( dat_df.loc[dat_df[one] == '-', one].count() )
    print("₩n")
```

```
col name :  국가
0


col name :  백신접종수
1


col name :  Enough_for_percent_of_people:
5


col name :  1차접종
6


col name :  2차접종
7


col name :  일별접종수
2
```

결측치(비어 있는 값)를 -999로 처리한다.

In [68]:

```
col_all = dat_df.columns
for one in col_all:
    print("col name : ", one)
    print( dat_df.loc[dat_df[one] == '-', one].count() )
    dat_df.loc[dat_df[one] == '-', one] = "-999"      # -은 이상치 -999로 치환
    dat_df.loc[dat_df[one] == '<0.1', one] = "0.05"   # <0.1은 0.05로 치환

    print("₩n")
```

```
col name :  국가
0


col name :  백신접종수
1


col name :  Enough_for_percent_of_people:
5


col name :  1차접종
6


col name :  2차접종
7


col name :  일별접종수
2
```

결측치 처리 후, 확인

In [70]:

```
col_all = dat_df.columns
for one in col_all:
    print("col name : ", one)
    print( dat_df.loc[dat_df[one] == '-', one].count() )
```

```
col name :  국가
0
col name :  백신접종수
0
col name :  Enough_for_percent_of_people:
0
col name :  1차접종
0
col name :  2차접종
0
col name :  일별접종수
0
```

수치의 ','을 처리

In [72]:

```
dat_df['백신접종수'] = dat_df['백신접종수'].str.replace(',', '')
dat_df['일별접종수'] = dat_df['일별접종수'].str.replace(',', '')
```

In [73]:

```
dat_df.head(10)
```

Out[73]:

	국가	백신접종수	Enough_for_percent_of_people:	1차접종	2차접종	일별접종수
0	Global Total	6172363261	-999	-999	-999	31740869
1	Mainland China	2200202000	78.6	78.6	73.0	3223429
2	India	870566939	31.8	46.2	16.4	7488509
3	EU	562942153	63.4	67.2	64.6	999304
4	U.S.	390114328	60.9	64.3	55.3	716762
5	Brazil	232250878	56.3	71.1	41.4	3757702
6	Japan	159494782	63.2	68.9	57.4	1026446
7	Indonesia	136941018	25.7	32.2	18.1	1755924
8	Turkey	108344725	65.1	64.4	52.7	386308
9	Germany	107030469	64.4	67.8	64.1	194734

데이터 전처리 후, 확인

In [74]:

```
dat_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 207 entries, 0 to 225
Data columns (total 6 columns):
 #   Column                        Non-Null Count  Dtype 
---  ------                        --------------  ----- 
 0   국가                            207 non-null    object
 1   백신접종수                         207 non-null    object
 2   Enough_for_percent_of_people:  207 non-null    object
 3   1차접종                          207 non-null    object
 4   2차접종                          207 non-null    object
 5   일별접종수                         207 non-null    object
dtypes: object(6)
memory usage: 11.3+ KB
```

In [78]:

```
dat_df.isnull().sum()
```

Out[78]:

```
국가                                0
백신접종수                             0
Enough_for_percent_of_people:    0
1차접종                              0
2차접종                              0
일별접종수                             0
dtype: int64
```

In [79]:

```
dat_df['백신접종수'].unique()
```

Out[79]:

```
array(['6172363261', '2200202000', '870566939', '562942153', '390114328',
       '232250878', '159494782', '136941018', '108344725', '107030469',
       '99366403', '93817818', '93500858', '89682021', '84158581',
       '76141484', '69867532', '61329870', '55964812', '51153324',
       '46023016', '43933886', '43372270', '42699197', '41770521',
       '40976791', '40273386', '39610550', '38367246', '37147012',
       '31552753', '26806343', '26002445', '25683070', '23830570',
       '23440062', '20571335', '19991460', '19872799', '18526941',
       '17699201', '16999722', '16623734', '15944324', '15057720',
       '14958007', '14430076', '14082920', '14050000', '13932708',
       '13607052', '12645634', '12066901', '11798933', '11779255',
       '11167215', '10792944', '10330401', '10071932', '9286999',
       '9001616', '8811697', '8600780', '8349945', '7747721', '7523495',
       '7448300', '7425277', '7349843', '7195154', '6942596', '6682813',
       '6651655', '6552979', '6409263', '6190557', '5367816', '5278479',
       '5234875', '5196072', '5045901', '4755737', '4696753', '4636486',
       '4545348', '4488153', '4402770', '3872421', '3644423', '3613357',
       '3535329', '3409470', '3408049', '3389520', '3339375', '3134137',
       '3133227', '2929966', '2561756', '2513769', '2380000', '2107230',
       '2058553', '1968171', '1784832', '1749457', '1699898', '1693555',
       '1650828', '1643172', '1633952', '1616636', '1546967', '1495762',
       '1411946', '1388566', '1359926', '1274156', '1146345', '1087426',
       '1068414', '1060145', '1045498', '997632', '975268', '890122',
       '816767', '788502', '783832', '727664', '715860', '702462',
       '675891', '649564', '621853', '606699', '551110', '533949',
       '533843', '464359', '461201', '450451', '443676', '417648',
       '414300', '408535', '407502', '405053', '391985', '389807',
       '387510', '336863', '332954', '329905', '322934', '304218',
       '284246', '254545', '240002', '234124', '232375', '226269',
       '207195', '187160', '186793', '184461', '180320', '164776',
       '155660', '150891', '149431', '148402', '142548', '134945',
       '129302', '109789', '103751', '102357', '101898', '97744', '94715',
       '93430', '86676', '86542', '82679', '79502', '75827', '73959',
       '73603', '72085', '67229', '64799', '55663', '52374', '49980',
       '49310', '47098', '47088', '44624', '41993', '33077', '18584',
       '14863', '7892', '4407', '2871', '-999'], dtype=object)
```

In [80]:

```
dat_df['백신접종수'].head(15)
```

Out[80]:

```
0     6172363261
1     2200202000
2      870566939
3      562942153
4      390114328
5      232250878
6      159494782
7      136941018
8      108344725
9      107030469
10      99366403
11      93817818
12      93500858
18      89682021
19      84158581
Name: 백신접종수, dtype: object
```

In [81]:

```
dat_df.loc[dat_df['백신접종수'].isna() , : ]
```

Out[81]:

국가	백신접종수	Enough_for_percent_of_people:	1차접종	2차접종	일별접종수

In [82]:

```
dat_df.iloc[ 15:25, :]
```

Out[82]:

	국가	백신접종수	Enough_for_percent_of_people:	1차접종	2차접종	일별접종수
20	Pakistan	76141484	18.6	27.5	12.5	686980
21	Spain	69867532	75.1	80.9	78.4	76386
22	South Korea	61329870	59.3	73.7	44.9	374506
23	Canada	55964812	74.6	76.2	70.6	109885
37	Argentina	51153324	56.9	65.6	47.7	260261
38	Thailand	46023016	33.1	42.4	22.8	567300
39	Philippines	43933886	20.5	23.5	18.9	359982
40	Iran	43372270	26.0	35.5	16.8	1368407
41	Malaysia	42699197	65.5	70.1	60.3	290646
42	Saudi Arabia	41770521	61.3	68.4	54.2	104852

In [83]:

```
dat_df_num = dat_df.iloc[:, 1:]
dat_df_num.columns
```

Out[83]:

```
Index(['백신접종수', 'Enough_for_percent_of_people:', '1차접종', '2차접종', '일별접
종수'], dtype='object')
```

In [84]:

```
dat_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 207 entries, 0 to 225
Data columns (total 6 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   국가                              207 non-null    object
 1   백신접종수                           207 non-null    object
 2   Enough_for_percent_of_people:  207 non-null    object
 3   1차접종                            207 non-null    object
 4   2차접종                            207 non-null    object
 5   일별접종수                           207 non-null    object
dtypes: object(6)
memory usage: 11.3+ KB
```

In [88]:

```
sel_col = dat_df_num.columns
for one in sel_col:
    print("col name :", one)
    dat_df[one] = dat_df[one].astype('float32')
```

```
col name : 백신접종수
col name : Enough_for_percent_of_people:
col name : 1차접종
col name : 2차접종
col name : 일별접종수
```

In [89]:

```
dat_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 207 entries, 0 to 225
Data columns (total 6 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   국가                              207 non-null    object
 1   백신접종수                           207 non-null    float32
 2   Enough_for_percent_of_people:  207 non-null    float32
 3   1차접종                            207 non-null    float32
 4   2차접종                            207 non-null    float32
 5   일별접종수                           207 non-null    float32
dtypes: float32(5), object(1)
memory usage: 7.3+ KB
```

파일 만들기

In [90]:

```
from datetime import datetime
import os

now = datetime.now()
file_make_time = "%04d%02d%02d_%02d" % (now.year, now.month, now.day, now.hour)
print(now.day - 1)
now_day = now.day
now_hour = now.hour

print( file_make_time )
```

```
28
20210929_00
```

In [91]:

```
print( os.getcwd() )
path_dir = os.getcwd() + "₩₩data₩₩"
path_file = path_dir + file_make_time
print( path_dir,  path_file, sep="₩n" )
```

```
C:₩Users₩toto₩Documents₩Github₩corona_analysis
C:₩Users₩toto₩Documents₩Github₩corona_analysis₩data₩
C:₩Users₩toto₩Documents₩Github₩corona_analysis₩data₩20210929_00
```

In [92]:

```
dat_df.to_csv( path_file + "_vaccine_bloomberg.csv", index=False)
dat_df.to_excel( path_file + "_vaccine_bloomberg.xlsx", index=False)
os.listdir(path_dir)
```

Out[92]:

```
['2021-08-07_corona.csv',
 '2021-08-07_corona.xlsx',
 '2021-09-19_corona.csv',
 '2021-09-19_corona.xlsx',
 '2021-09-20_corona.csv',
 '2021-09-20_corona.xlsx',
 '2021-09-28_corona.csv',
 '2021-09-28_corona.xlsx',
 '20210808_16datamerge.csv',
 '20210808_16datamerge.xlsx',
 '20210808_16_today_corona.csv',
 '20210808_16_today_corona.xlsx',
 '20210808_16_vaccine_bloomberg.csv',
 '20210808_16_vaccine_bloomberg.xlsx',
 '20210920_00_vaccine_bloomberg.csv',
 '20210920_00_vaccine_bloomberg.xlsx',
 '20210921_00_datamerge.csv',
 '20210921_00_datamerge.xlsx',
 '20210929_00_vaccine_bloomberg.csv',
 '20210929_00_vaccine_bloomberg.xlsx',
 'country.csv',
 'country.xlsx']
```

- history
  - 2021.08.08 version 01
- 출처를 밝히시고 위의 내용에 대해 자유롭게 사용 가능합니다.