

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

In [5]:



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

## 전체 데이터 가져오기

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', 'Indonesia', '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', 'UAE', 'Uzbekistan', 'Ukraine', 'South Africa', 'Belgium', 'Portugal', 'Egypt', 'Israel', 'Taiwan', 'Algeria', 'Venezuela', 'Kazakhstan', 'Sweden', 'Nepal', 'Greece', 'Dominican Republic', 'Czech Republic', 'Hungary', 'Austria', 'Switzerland', '', 'Romania', 'Singapore', 'Myanmar', 'Denmark', 'Hong Kong', 'Azerbaijan', 'Norway', 'Finland', 'Iraq', 'Tunisia', 'El Salvador', 'Ireland', 'Jordan', 'Guatemala', 'Bolivia', 'Nigeria', 'Serbia', 'Uruguay', 'Panama', 'Costa Rica', 'Zimbabwe', 'Honduras', 'New Zealand', 'Oman', 'Slovakia', 'Qatar', 'Mongolia', 'Paraguay', 'Laos', 'Tajikistan', 'Rwanda', 'Mozambique', 'Kenya', 'Ethiopia', 'Croatia', 'Belarus', 'Lithuania', 'Angola', 'Afghanistan', 'Lebanon', 'Bahrain', 'Bulgaria', 'Kuwait', 'Slovenia', 'Uganda', 'Ivory Coast', 'Senegal', 'Georgia', 'Albania', 'Guinea', 'Ghana', 'Sudan', 'Mauritius', 'Latvia', 'Libya', 'North Macedonia', 'Kyrgyzstan', 'Moldova', 'Estonia', 'Kosovo', 'Cyprus', 'Malawi', 'Bhutan', 'Bosnia and Herzegovina', 'Trinidad and Tobago', 'Togo', 'Fiji', 'Mauritania', 'Malta', 'Nicaragua', 'Luxembourg', 'Jamaica', 'M

In [8]:



```
for i in range(len(all_data)):
    print(len( all_data[i]) )
```

226  
226  
226  
226  
226  
226

In [9]:



```
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 [87]:

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

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

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

Out[87]:

	국가	백신접종수	백신보유율(인구)	1차접종	2차접종	일별접종수
0	Global Total	6,199,443,468	—	—	—	29,152,868
1	Mainland China	2,203,185,000	78.7	78.6	73.0	3,171,286
2	India	876,538,633	32.0	46.7	16.8	7,178,901
3	EU	564,029,956	63.5	67.3	64.7	1,052,903
4	U.S.	391,152,574	61.1	64.4	55.8	701,422
...	...	...	...	...	...	...
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 [88]:

```
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   백신보유율(인구)     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 [89]:



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

Out[89]:

```
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 [90]:



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

Out[90]:

```
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 [91]:



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

Out[91]:

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

## 데이터가 있는 행만 가져오기

In [92]:



```
print(dat_df.shape)
dat_df = dat_df.loc[ dat_df['국가'].str.len() > 1, : ]
print(dat_df.shape)
```

(226, 6)

(207, 6)

In [93]:



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

Out[93]:

	국가	백신접종수	백신보유율(인구)	1차접종	2차접종	일별접종수
0	Global Total	6,199,443,468	—	—	—	29,152,868
1	Mainland China	2,203,185,000	78.7	78.6	73.0	3,171,286
2	India	876,538,633	32.0	46.7	16.8	7,178,901
3	EU	564,029,956	63.5	67.3	64.7	1,052,903
4	U.S.	391,152,574	61.1	64.4	55.8	701,422
5	Brazil	233,261,963	56.5	71.3	41.9	1,317,163
6	Japan	161,057,159	63.8	69.5	58.2	1,134,065
7	Indonesia	139,232,210	26.1	32.9	18.4	1,916,012
8	Turkey	108,761,681	65.4	64.5	53.0	381,129
9	Germany	107,197,993	64.5	67.9	64.2	194,817
10	Mexico	99,366,403	38.9	49.8	35.0	536,693
11	France	94,250,528	72.7	77.4	74.6	229,555
12	U.K.	93,565,474	70.0	73.0	67.1	63,126
18	Russia	89,865,316	30.6	32.6	28.5	214,966
19	Italy	84,351,559	69.9	74.5	74.5	211,345

## 인덱스가 일정하지 않아, 인덱스 값 초기화

In [94]:



```
dat_df = dat_df.reset_index()  
dat_df.shape
```

Out[94]:

(207, 7)

## 이상치 '-' 값 확인

In [95]:



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

Out[95]:

	index	국가	백신접종수	백신보유율(인구)	1차접종	2차접종	일별접종수
206	225	Eritrea	-	-	-	-	-

In [96]:



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

Out[96]:

	index	국가	백신접종수	백신보유율(인구)	1차접종	2차접종	일별접종수
199	218	Turkmenistan	41,993	0.4	0.6	0.2	-
206	225	Eritrea	-	-	-	-	-

In [97]:



```
dat_df.columns
if 'index' in dat_df.columns:
    dat_df.drop(['index'], axis=1, inplace=True )
dat_df.columns
```

Out[97]:

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

## 각 컬럼별 이상치 개수



In [98]:



```
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 : 백신보유율(인구)  
5

col name : 1차접종  
6

col name : 2차접종  
7

col name : 일별접종수  
2

## 이상치 처리

### (1) '-'는 결측치(비어 있는 값)를 문자열 '-999'로 처리

- 문자열로 변경하는 이유 : 현재 컬럼이 문자열이기에

### (2) '<0.1'를 0.05로 임의의 값 고정값으로 변경

In [99]:



```
col_all = dat_df.columns
for one in col_all:
    print("col name : ", one)
    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("col name : {} 이상치 처리 OK".format(one), end="WnWn")
```

col name : 국가  
col name : 국가 이상치 처리 OK

col name : 백신접종수  
col name : 백신접종수 이상치 처리 OK

col name : 백신보유율(인구)  
col name : 백신보유율(인구) 이상치 처리 OK

col name : 1차접종  
col name : 1차접종 이상치 처리 OK

col name : 2차접종  
col name : 2차접종 이상치 처리 OK

col name : 일별접종수  
col name : 일별접종수 이상치 처리 OK

## 결측치 처리 후, 확인

In [100]:



```
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 : 백신보유율(인구)  
0  
col name : 1차접종  
0  
col name : 2차접종  
0  
col name : 일별접종수  
0

## 수치 값의 천단위 구분자 ','을 처리

In [101]:



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

In [102]:



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

Out[102]:

	국가	백신접종수	백신보유율(인구)	1차접종	2차접종	일별접종수
0	Global Total	6199443468	-999	-999	-999	29152868
1	Mainland China	2203185000	78.7	78.6	73.0	3171286
2	India	876538633	32.0	46.7	16.8	7178901
3	EU	564029956	63.5	67.3	64.7	1052903
4	U.S.	391152574	61.1	64.4	55.8	701422
5	Brazil	233261963	56.5	71.3	41.9	1317163
6	Japan	161057159	63.8	69.5	58.2	1134065
7	Indonesia	139232210	26.1	32.9	18.4	1916012
8	Turkey	108761681	65.4	64.5	53.0	381129
9	Germany	107197993	64.5	67.9	64.2	194817

## 데이터 전처리 후, 확인

In [103]:



```
dat_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 207 entries, 0 to 206
Data columns (total 6 columns):
 #   Column                Non-Null Count  Dtype  
---  --
 0   국가                  207 non-null   object 
 1   백신접종수           207 non-null   object 
 2   백신보유율(인구)     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: 9.8+ KB
```

In [104]:



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

Out[104]:

```
국가          0
백신접종수      0
백신보유율(인구)  0
1차접종        0
2차접종        0
일별접종수      0
dtype: int64
```

In [105]:



```
dat_df['국가'].unique()
```

Out[105]:

```
array(['Global Total', 'Mainland China', 'India', 'EU', 'U.S.', 'Brazil',  
      'Japan', 'Indonesia', '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', 'UAE', 'Uzbekistan',  
      'Ukraine', 'South Africa', 'Belgium', 'Portugal', 'Egypt',  
      'Israel', 'Taiwan', 'Algeria', 'Venezuela', 'Kazakhstan', 'Sweden',  
      'Nepal', 'Greece', 'Dominican Republic', 'Czech Republic',  
      'Hungary', 'Austria', 'Switzerland', 'Romania', 'Singapore',  
      'Myanmar', 'Denmark', 'Hong Kong', 'Azerbaijan', 'Norway',  
      'Finland', 'Iraq', 'Tunisia', 'El Salvador', 'Ireland', 'Jordan',  
      'Guatemala', 'Bolivia', 'Nigeria', 'Serbia', 'Uruguay', 'Panama',  
      'Costa Rica', 'Zimbabwe', 'Honduras', 'New Zealand', 'Oman',  
      'Slovakia', 'Qatar', 'Mongolia', 'Paraguay', 'Laos', 'Tajikistan',  
      'Rwanda', 'Mozambique', 'Kenya', 'Ethiopia', 'Croatia', 'Belarus',  
      'Lithuania', 'Angola', 'Afghanistan', 'Lebanon', 'Bahrain',  
      'Bulgaria', 'Kuwait', 'Slovenia', 'Uganda', 'Ivory Coast',  
      'Senegal', 'Georgia', 'Albania', 'Guinea', 'Ghana', 'Sudan',  
      'Mauritius', 'Latvia', 'Libya', 'North Macedonia', 'Kyrgyzstan',  
      'Moldova', 'Estonia', 'Kosovo', 'Cyprus', 'Malawi', 'Bhutan',  
      'Bosnia and Herzegovina', 'Trinidad and Tobago', 'Togo', 'Fiji',  
      'Mauritania', 'Malta', 'Nicaragua', 'Luxembourg', 'Jamaica',  
      'Maldives', 'Zambia', 'Timor-Leste', 'Macau', 'Niger', 'Botswana',  
      'Iceland', 'Guyana', 'Syria', 'Somalia', 'Cameroon', 'Montenegro',  
      'Brunei', 'Madagascar', 'Cape Verde', 'Armenia', 'Namibia', 'Mali',  
      'Suriname', 'Tanzania', 'Equatorial Guinea', 'Comoros', 'Lesotho',  
      'Republic of the Congo', 'Yemen', 'Belize', 'Liberia',  
      'Burkina Faso', 'Eswatini', 'Barbados', 'Sierra Leone', 'Benin',  
      'Gambia', 'Central African Republic', 'Curaçao', 'Bahamas',  
      'Gabon', 'Papua New Guinea', 'Aruba', 'Jersey', 'Samoa',  
      'Seychelles', 'Chad', 'DR Congo', 'Isle of Man', 'Cayman Islands',  
      'South Sudan', 'Greenland', 'Guernsey', 'Solomon Islands',  
      'Andorra', 'Guinea-Bissau', 'Antigua and Barbuda', 'Bermuda',  
      'Sao Tome and Principe', 'Gibraltar', 'St. Lucia', 'Faroe Islands',  
      'Tonga', 'Micronesia', 'Djibouti', 'Haiti', 'Grenada',  
      'Turks & Caicos', 'Monaco', 'Sint Maarten',  
      'Saint Kitts and Nevis', 'San Marino', 'Dominica', 'Turkmenistan',  
      'St. Vincent and the Grenadines', 'Anguilla', 'Nauru',  
      'St. Helena', 'Falkland Islands', 'Montserrat', 'Eritrea'],  
      dtype=object)
```

In [106]:



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

Out[106]:

국가	백신접종수	백신보유율(인구)	1차접종	2차접종	일별접종수
----	-------	-----------	------	------	-------

In [107]:

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

Out[107]:

	국가	백신접종수	백신보유율(인구)	1차접종	2차접종	일별접종수
15	Pakistan	79533208	19.4	28.3	13.4	670870
16	Spain	69936755	75.2	81.0	78.5	72816
17	South Korea	62462921	60.4	74.5	46.3	533067
18	Canada	56036106	74.7	76.2	70.6	105929
19	Argentina	51222138	57.0	65.7	48.3	222650
20	Thailand	46023016	33.1	42.4	22.8	640484
21	Philippines	44361285	20.7	23.5	19.2	369770
22	Iran	43372270	26.0	35.5	16.8	1368407
23	Malaysia	43028919	66.0	70.6	60.7	300427
24	Saudi Arabia	41886224	61.4	68.4	54.4	101318

In [108]:

```
dat_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 207 entries, 0 to 206
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   국가                  207 non-null   object 
1   백신접종수           207 non-null   object 
2   백신보유율(인구)     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: 9.8+ KB
```

## 수치형 변수 값 변환

In [109]:

```
dat_df_num = dat_df.loc[:, "백신접종수":]
dat_df_num.columns
```

Out[109]:

```
Index(['백신접종수', '백신보유율(인구)', '1차접종', '2차접종', '일별접종수'], dtype
='object')
```

In [110]:



```
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 : 백신보유율(인구)
col name : 1차접종
col name : 2차접종
col name : 일별접종수
```

In [111]:



```
dat_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 207 entries, 0 to 206
Data columns (total 6 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   국가                  207 non-null    object
 1   백신접종수           207 non-null    float32
 2   백신보유율(인구)     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: 5.8+ KB
```

## 파일 만들기

In [112]:



```
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 )
```

```
29
20210930_00
```

In [113]:



```
print( os.getcwd() )
path_dir = os.getcwd() + "WWdataWW"
path_file = path_dir + file_make_time
print( path_dir, path_file, sep="Wn" )
```

```
C:\Users\Wtoto\Documents\Github\corona_analysis
C:\Users\Wtoto\Documents\Github\corona_analysis\dataW
C:\Users\Wtoto\Documents\Github\corona_analysis\dataW20210930_00
```

In [114]:



```
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[114]:

```
['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',
 '2021-09-29_corona.csv',
 '2021-09-29_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',
 '20210929_01_datamerge.csv',
 '20210929_01_datamerge.xlsx',
 '20210929_14_vaccine_bloomberg.csv',
 '20210929_14_vaccine_bloomberg.xlsx',
 '20210930_00_vaccine_bloomberg.csv',
 '20210930_00_vaccine_bloomberg.xlsx',
 'country.csv',
 'country.xlsx']
```

- history
  - 2021.09.30 version 01
- 출처를 밝히시고 위의 내용에 대해 자유롭게 사용 가능합니다.

In [ ]:





