

# CLEAN & VISUALIZE

 [youtube.com/isnaalfi](https://youtube.com/isnaalfi)



## Siapkan Pandas

```
In [194]: import pandas as pd
```

## import data

```
In [195]: dfc=pd.read_csv("time_series_covid19_confirmed_global.csv")
```

```
In [197]: dfc.head(3) #tail --> bawah
```

```
Out[197]:
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...

3 rows × 156 columns

```
In [198]: dfr=pd.read_csv("time_series_covid19_recovered_global.csv")
```

```
In [199]: dfd=pd.read_csv("time_series_covid19_deaths_global.csv")
```

## Bersihkan Data

hapus kolom provinsi

```
In [200]: del dfc["Province/State"]
del dfr["Province/State"]
del dfd["Province/State"]
```

```
In [201]: dfr.head()
```

```
Out[201]:
```

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...	6/12/20
0	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	0	...	3928
1	Albania	41.1533	20.1683	0	0	0	0	0	0	0	...	1034
2	Algeria	28.0339	1.6596	0	0	0	0	0	0	0	...	7322
3	Andorra	42.5063	1.5218	0	0	0	0	0	0	0	...	781
4	Angola	-11.2027	17.8739	0	0	0	0	0	0	0	...	42

5 rows × 155 columns

## Cek shape

```
In [202]: dfc.shape
```

```
Out[202]: (266, 155)
```

```
In [203]: dfr.shape
```

```
Out[203]: (253, 155)
```

```
In [204]: dfd.shape
```

```
Out[204]: (266, 155)
```

Lhoh kok nggak sama? samakan dulu yuk!

## Melihat data lebih detail

### Melihat data negara..

```
In [205]: dfc['Country/Region'].head(3)
```

```
Out[205]: 0    Afghanistan
1         Albania
2         Algeria
Name: Country/Region, dtype: object
```

```
In [207]: compares=set(dfc['Country/Region']).intersection(set(dfr['Country/Region']))
len(compares)
```

```
Out[207]: 188
```

```
In [209]: compares=set(dfc['Country/Region']).difference(set(dfr['Country/Region']))
len(compares)
```

```
Out[209]: 0
```

```
In [211]: len(dfc['Country/Region'].unique())
```

```
Out[211]: 188
```

```
In [60]: len(dfc['Country/Region'].unique())
```

```
Out[60]: 188
```

## Cek data yang aneh

### Index ke berapa ada data yang tidak sama di dfc dan dfr?

```
In [ ]: list_aneh=[] #menyimpan index yang datanya tidak sama antara dfc & dfr
# 0, UK,UK
# 1, UK,Indonesia --> beda! --> list_aneh=[1,2,3,4,5]
def cekdata(a,b): #diisi dengan df yang dibandingkan: dfc, dfr
    for i in range(len(b)): #dfr : 253 baris --> 0,252
        if (a.iloc[i]["Country/Region"]) != (b.iloc[i]["Country/Region"]):
            list_aneh.append(i)
    return list_aneh
cekal=cekdata(dfc,dfr)
cekal
```

### Yang beda apa ya?

```
In [215]: list_cekal=[]
all_cekal=[]
for c in range(len(cekal)): #0--> cecal :180 data
    list_cekal=dfc.iloc[[cekal[c]]["Country/Region"].values.tolist(),dfr.iloc
[[cekal[c]]["Country/Region"].values.tolist()
    all_cekal.append(list_cekal)
all_cekal
```

```

Out[215]: [['Benin'], ['Belize']],
          [['Bhutan'], ['Benin']],
          [['Bolivia'], ['Bhutan']],
          [['Bosnia and Herzegovina'], ['Bolivia']],
          [['Brazil'], ['Bosnia and Herzegovina']],
          [['Brunei'], ['Brazil']],
          [['Bulgaria'], ['Brunei']],
          [['Burkina Faso'], ['Bulgaria']],
          [['Cabo Verde'], ['Burkina Faso']],
          [['Cambodia'], ['Cabo Verde']],
          [['Cameroon'], ['Cambodia']],
          [['Canada'], ['Cameroon']],
          [['Canada'], ['Central African Republic']],
          [['Canada'], ['Chad']],
          [['Canada'], ['Chile']],
          [['Canada'], ['China']],
          [['Canada'], ['China']],
          [['Canada'], ['China']],
          [['Canada'], ['China']],
          [['Canada'], ['China']],
          [['Canada'], ['China']],
          [['Central African Republic'], ['China']],
          [['Chad'], ['China']],
          [['Chile'], ['China']],
          [['China'], ['Colombia']],
          [['China'], ['Congo (Brazzaville)']],
          [['China'], ['Congo (Kinshasa)']],
          [['China'], ['Costa Rica']],
          [['China'], ['Cote d'Ivoire']],
          [['China'], ['Croatia']],
          [['China'], ['Diamond Princess']],
          [['China'], ['Cuba']],
          [['China'], ['Cyprus']],
          [['Colombia'], ['Czechia']],
          [['Congo (Brazzaville)'], ['Denmark']],
          [['Congo (Kinshasa)'], ['Denmark']],
          [['Costa Rica'], ['Denmark']],
          [['Cote d'Ivoire'], ['Djibouti']],
          [['Croatia'], ['Dominica']],
          [['Diamond Princess'], ['Dominican Republic']],
          [['Cuba'], ['Ecuador']],
          [['Cyprus'], ['Egypt']],
          [['Czechia'], ['El Salvador']],
          [['Denmark'], ['Equatorial Guinea']],
          [['Denmark'], ['Eritrea']],
          [['Denmark'], ['Estonia']],
          [['Djibouti'], ['Eswatini']],
          [['Dominican Republic'], ['Ethiopia']],
          [['Ecuador'], ['Fiji']],
          [['Egypt'], ['Finland']],
          [['El Salvador'], ['France']],
          [['Equatorial Guinea'], ['France']],
          [['Eritrea'], ['France']],
          [['Estonia'], ['France']],
          [['Eswatini'], ['France']],
          [['Ethiopia'], ['France']],
          [['Fiji'], ['France']],
          [['Finland'], ['France']],
          [['France'], ['Gabon']],
          [['France'], ['Gambia']],
          [['France'], ['Georgia']],
          [['France'], ['Germany']],
          [['France'], ['Ghana']],
          [['France'], ['Grenada']],
          [['France'], ['Greece']],
          [['France'], ['Guatemala']],
          [['Gabon'], ['Guinea']],
          [['Gambia'], ['Guinea-Bissau']],
          [['Georgia'], ['Guyana']]

```

## Lhoh, kok "Belize" disejajarkan dengan "Benin" ?

Ada nggak sih Belize di daftar dfc?

```
In [222]: listb=dfc["Country/Region"].values  
         "Belize" in listb
```

```
Out[222]: True
```

## Sort dulu deh..

```
In [220]: dfc.sort_values("Country/Region", inplace=True) #sort, ascending=False  
dfc
```

Out[220]:

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...
0	Afghanistan	33.000000	65.000000	0	0	0	0	0	0	0	...
1	Albania	41.153300	20.168300	0	0	0	0	0	0	0	...
2	Algeria	28.033900	1.659600	0	0	0	0	0	0	0	...
3	Andorra	42.506300	1.521800	0	0	0	0	0	0	0	...
4	Angola	-11.202700	17.873900	0	0	0	0	0	0	0	...
5	Antigua and Barbuda	17.060800	-61.796400	0	0	0	0	0	0	0	...
6	Argentina	-38.416100	-63.616700	0	0	0	0	0	0	0	...
7	Armenia	40.069100	45.038200	0	0	0	0	0	0	0	...
15	Australia	-31.950500	115.860500	0	0	0	0	0	0	0	...
14	Australia	-37.813600	144.963100	0	0	0	0	1	1	1	...
13	Australia	-41.454500	145.970700	0	0	0	0	0	0	0	...
12	Australia	-34.928500	138.600700	0	0	0	0	0	0	0	...
10	Australia	-12.463400	130.845600	0	0	0	0	0	0	0	...
9	Australia	-33.868800	151.209300	0	0	0	0	3	4	4	...
8	Australia	-35.473500	149.012400	0	0	0	0	0	0	0	...
11	Australia	-28.016700	153.400000	0	0	0	0	0	0	0	...
16	Austria	47.516200	14.550100	0	0	0	0	0	0	0	...
17	Azerbaijan	40.143100	47.576900	0	0	0	0	0	0	0	...
18	Bahamas	25.034300	-77.396300	0	0	0	0	0	0	0	...
19	Bahrain	26.027500	50.550000	0	0	0	0	0	0	0	...
20	Bangladesh	23.685000	90.356300	0	0	0	0	0	0	0	...
21	Barbados	13.193900	-59.543200	0	0	0	0	0	0	0	...
22	Belarus	53.709800	27.953400	0	0	0	0	0	0	0	...
23	Belgium	50.833300	4.000000	0	0	0	0	0	0	0	...
237	Belize	13.193900	-59.543200	0	0	0	0	0	0	0	...
24	Benin	9.307700	2.315800	0	0	0	0	0	0	0	...
25	Bhutan	27.514200	90.433600	0	0	0	0	0	0	0	...
26	Bolivia	-16.290200	-63.588700	0	0	0	0	0	0	0	...
27	Bosnia and Herzegovina	43.915900	17.679100	0	0	0	0	0	0	0	...
252	Botswana	-22.328500	24.684900	0	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
209	Thailand	15.000000	101.000000	2	3	5	7	8	8	14	...
236	Timor-Leste	-8.874217	125.727539	0	0	0	0	0	0	0	...
210	Togo	8.619500	0.824800	0	0	0	0	0	0	0	...
211	Trinidad and Tobago	10.691800	-61.222500	0	0	0	0	0	0	0	...
212	Tunisia	34.000000	9.000000	0	0	0	0	0	0	0	...
213	Turkey	38.963700	35.243300	0	0	0	0	0	0	0	...
225	US	37.090200	-95.712900	1	1	2	2	5	5	5	...
214	Uganda	1.000000	32.000000	0	0	0	0	0	0	0	...
215	Ukraine	48.379400	31.165600	0	0	0	0	0	0	0	...
216	United Arab Emirates	24.000000	54.000000	0	0	0	0	0	0	0	...



## Jangan lupa reset index

```
In [224]: #ganti/reset index
dfc.index = range(len(dfc.index)) #mengenerate angka list =[0-266]
dfc.tail(3)
```

Out[224]:

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...
263	Yemen	15.552727	48.516388	0	0	0	0	0	0	0	...
264	Zambia	-15.416700	28.283300	0	0	0	0	0	0	0	...
265	Zimbabwe	-20.000000	30.000000	0	0	0	0	0	0	0	...

3 rows × 155 columns

Lakukan pada df lainnya juga ya..

```
In [225]: dfr.sort_values("Country/Region", inplace=True) #sort  
          dfr.index = range(len(dfr.index)) #reset  
          dfr
```

Out[225]:

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...
0	Afghanistan	33.000000	65.000000	0	0	0	0	0	0	0	...
1	Albania	41.153300	20.168300	0	0	0	0	0	0	0	...
2	Algeria	28.033900	1.659600	0	0	0	0	0	0	0	...
3	Andorra	42.506300	1.521800	0	0	0	0	0	0	0	...
4	Angola	-11.202700	17.873900	0	0	0	0	0	0	0	...
5	Antigua and Barbuda	17.060800	-61.796400	0	0	0	0	0	0	0	...
6	Argentina	-38.416100	-63.616700	0	0	0	0	0	0	0	...
7	Armenia	40.069100	45.038200	0	0	0	0	0	0	0	...
8	Australia	-35.473500	149.012400	0	0	0	0	0	0	0	...
9	Australia	-33.868800	151.209300	0	0	0	0	0	0	0	...
10	Australia	-12.463400	130.845600	0	0	0	0	0	0	0	...
11	Australia	-28.016700	153.400000	0	0	0	0	0	0	0	...
12	Australia	-34.928500	138.600700	0	0	0	0	0	0	0	...
13	Australia	-41.454500	145.970700	0	0	0	0	0	0	0	...
14	Australia	-37.813600	144.963100	0	0	0	0	0	0	0	...
15	Australia	-31.950500	115.860500	0	0	0	0	0	0	0	...
16	Austria	47.516200	14.550100	0	0	0	0	0	0	0	...
17	Azerbaijan	40.143100	47.576900	0	0	0	0	0	0	0	...
18	Bahamas	25.034300	-77.396300	0	0	0	0	0	0	0	...
19	Bahrain	26.027500	50.550000	0	0	0	0	0	0	0	...
20	Bangladesh	23.685000	90.356300	0	0	0	0	0	0	0	...
21	Barbados	13.193900	-59.543200	0	0	0	0	0	0	0	...
22	Belarus	53.709800	27.953400	0	0	0	0	0	0	0	...
23	Belgium	50.833300	4.000000	0	0	0	0	0	0	0	...
24	Belize	13.193900	-59.543200	0	0	0	0	0	0	0	...
25	Benin	9.307700	2.315800	0	0	0	0	0	0	0	...
26	Bhutan	27.514200	90.433600	0	0	0	0	0	0	0	...
27	Bolivia	-16.290200	-63.588700	0	0	0	0	0	0	0	...
28	Bosnia and Herzegovina	43.915900	17.679100	0	0	0	0	0	0	0	...
29	Botswana	-22.328500	24.684900	0	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
223	Thailand	15.000000	101.000000	0	0	0	0	2	2	5	...
224	Timor-Leste	-8.874200	125.727500	0	0	0	0	0	0	0	...
225	Togo	8.619500	0.824800	0	0	0	0	0	0	0	...
226	Trinidad and Tobago	10.691800	-61.222500	0	0	0	0	0	0	0	...
227	Tunisia	34.000000	9.000000	0	0	0	0	0	0	0	...
228	Turkey	38.963700	35.243300	0	0	0	0	0	0	0	...
229	US	37.090200	-95.712900	0	0	0	0	0	0	0	...
230	Uganda	1.000000	32.000000	0	0	0	0	0	0	0	...
231	Ukraine	48.379400	31.165600	0	0	0	0	0	0	0	...
232	United Arab Emirates	24.000000	54.000000	0	0	0	0	0	0	0	...

```
In [226]: dfd.sort_values("Country/Region", inplace=True) #sort
          dfd.index = range(len(dfd.index)) #reset
          dfd
```

Out[226]:

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...
0	Afghanistan	33.000000	65.000000	0	0	0	0	0	0	0	...
1	Albania	41.153300	20.168300	0	0	0	0	0	0	0	...
2	Algeria	28.033900	1.659600	0	0	0	0	0	0	0	...
3	Andorra	42.506300	1.521800	0	0	0	0	0	0	0	...
4	Angola	-11.202700	17.873900	0	0	0	0	0	0	0	...
5	Antigua and Barbuda	17.060800	-61.796400	0	0	0	0	0	0	0	...
6	Argentina	-38.416100	-63.616700	0	0	0	0	0	0	0	...
7	Armenia	40.069100	45.038200	0	0	0	0	0	0	0	...
8	Australia	-31.950500	115.860500	0	0	0	0	0	0	0	...
9	Australia	-37.813600	144.963100	0	0	0	0	0	0	0	...
10	Australia	-41.454500	145.970700	0	0	0	0	0	0	0	...
11	Australia	-34.928500	138.600700	0	0	0	0	0	0	0	...
12	Australia	-12.463400	130.845600	0	0	0	0	0	0	0	...
13	Australia	-33.868800	151.209300	0	0	0	0	0	0	0	...
14	Australia	-35.473500	149.012400	0	0	0	0	0	0	0	...
15	Australia	-28.016700	153.400000	0	0	0	0	0	0	0	...
16	Austria	47.516200	14.550100	0	0	0	0	0	0	0	...
17	Azerbaijan	40.143100	47.576900	0	0	0	0	0	0	0	...
18	Bahamas	25.034300	-77.396300	0	0	0	0	0	0	0	...
19	Bahrain	26.027500	50.550000	0	0	0	0	0	0	0	...
20	Bangladesh	23.685000	90.356300	0	0	0	0	0	0	0	...
21	Barbados	13.193900	-59.543200	0	0	0	0	0	0	0	...
22	Belarus	53.709800	27.953400	0	0	0	0	0	0	0	...
23	Belgium	50.833300	4.000000	0	0	0	0	0	0	0	...
24	Belize	13.193900	-59.543200	0	0	0	0	0	0	0	...
25	Benin	9.307700	2.315800	0	0	0	0	0	0	0	...
26	Bhutan	27.514200	90.433600	0	0	0	0	0	0	0	...
27	Bolivia	-16.290200	-63.588700	0	0	0	0	0	0	0	...
28	Bosnia and Herzegovina	43.915900	17.679100	0	0	0	0	0	0	0	...
29	Botswana	-22.328500	24.684900	0	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
236	Thailand	15.000000	101.000000	0	0	0	0	0	0	0	...
237	Timor-Leste	-8.874217	125.727539	0	0	0	0	0	0	0	...
238	Togo	8.619500	0.824800	0	0	0	0	0	0	0	...
239	Trinidad and Tobago	10.691800	-61.222500	0	0	0	0	0	0	0	...
240	Tunisia	34.000000	9.000000	0	0	0	0	0	0	0	...
241	Turkey	38.963700	35.243300	0	0	0	0	0	0	0	...
242	US	37.090200	-95.712900	0	0	0	0	0	0	0	...
243	Uganda	1.000000	32.000000	0	0	0	0	0	0	0	...
244	Ukraine	48.379400	31.165600	0	0	0	0	0	0	0	...
245	United Arab Emirates	24.000000	54.000000	0	0	0	0	0	0	0	...

```
In [227]: dfc.shape
```

```
Out[227]: (266, 155)
```

```
In [228]: dfr.shape
```

```
Out[228]: (253, 155)
```

## Lhah masih sama?

Kita gabungkan saja untuk negara yang sama

```
In [229]: dfc.loc[dfc["Country/Region"] == "Australia"]  
#dfr.loc[dfr["Country/Region"] == "Australia"]  
#dfd.loc[dfd["Country/Region"] == "Australia"]
```

```
Out[229]:
```

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...	6/12
8	Australia	-31.9505	115.8605	0	0	0	0	0	0	0	...	6
9	Australia	-37.8136	144.9631	0	0	0	0	1	1	1	...	17
10	Australia	-41.4545	145.9707	0	0	0	0	0	0	0	...	2
11	Australia	-34.9285	138.6007	0	0	0	0	0	0	0	...	4
12	Australia	-12.4634	130.8456	0	0	0	0	0	0	0	...	
13	Australia	-33.8688	151.2093	0	0	0	0	3	4	4	...	3
14	Australia	-35.4735	149.0124	0	0	0	0	0	0	0	...	
15	Australia	-28.0167	153.4000	0	0	0	0	0	0	0	...	10

8 rows × 155 columns

```
In [80]: 605+1847+228+440+29+3151+108+1066
```

```
Out[80]: 7474
```

```
In [231]: dfc.loc[dfc["Country/Region"] == "Australia"]
```

```
Out[231]:
```

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...	6/12
8	Australia	-31.9505	115.8605	0	0	0	0	0	0	0	...	6
9	Australia	-37.8136	144.9631	0	0	0	0	1	1	1	...	17
10	Australia	-41.4545	145.9707	0	0	0	0	0	0	0	...	2
11	Australia	-34.9285	138.6007	0	0	0	0	0	0	0	...	4
12	Australia	-12.4634	130.8456	0	0	0	0	0	0	0	...	
13	Australia	-33.8688	151.2093	0	0	0	0	3	4	4	...	3
14	Australia	-35.4735	149.0124	0	0	0	0	0	0	0	...	
15	Australia	-28.0167	153.4000	0	0	0	0	0	0	0	...	10

8 rows × 155 columns

```
In [233]: indo=dfc.loc[dfc["Country/Region"] == "Indonesia"].sum()[3:]  
indo
```

```
Out[233]: 1/22/20      0  
1/23/20      0  
1/24/20      0  
1/25/20      0  
1/26/20      0  
1/27/20      0  
1/28/20      0  
1/29/20      0  
1/30/20      0  
1/31/20      0  
2/1/20       0  
2/2/20       0  
2/3/20       0  
2/4/20       0  
2/5/20       0  
2/6/20       0  
2/7/20       0  
2/8/20       0  
2/9/20       0  
2/10/20      0  
2/11/20      0  
2/12/20      0  
2/13/20      0  
2/14/20      0  
2/15/20      0  
2/16/20      0  
2/17/20      0  
2/18/20      0  
2/19/20      0  
2/20/20      0  
  
...  
5/23/20      21745  
5/24/20      22271  
5/25/20      22750  
5/26/20      23165  
5/27/20      23851  
5/28/20      24538  
5/29/20      25216  
5/30/20      25773  
5/31/20      26473  
6/1/20       26940  
6/2/20       27549  
6/3/20       28233  
6/4/20       28818  
6/5/20       29521  
6/6/20       30514  
6/7/20       31186  
6/8/20       32033  
6/9/20       33076  
6/10/20      34316  
6/11/20      35295  
6/12/20      36406  
6/13/20      37420  
6/14/20      38277  
6/15/20      39294  
6/16/20      40400  
6/17/20      41431  
6/18/20      42762  
6/19/20      43803  
6/20/20      45029  
6/21/20      45891  
Length: 152, dtype: object
```

```
In [232]: ausie=dfc.loc[dfc["Country/Region"] == "Australia"].sum()[3:]
```

```
ausie
```

```
Out[232]: 1/22/20      0
          1/23/20      0
          1/24/20      0
          1/25/20      0
          1/26/20      4
          1/27/20      5
          1/28/20      5
          1/29/20      6
          1/30/20      9
          1/31/20      9
          2/1/20      12
          2/2/20      12
          2/3/20      12
          2/4/20      13
          2/5/20      13
          2/6/20      14
          2/7/20      15
          2/8/20      15
          2/9/20      15
          2/10/20     15
          2/11/20     15
          2/12/20     15
          2/13/20     15
          2/14/20     15
          2/15/20     15
          2/16/20     15
          2/17/20     15
          2/18/20     15
          2/19/20     15
          2/20/20     15
          ...
          5/23/20    7114
          5/24/20    7114
          5/25/20    7126
          5/26/20    7139
          5/27/20    7150
          5/28/20    7165
          5/29/20    7184
          5/30/20    7192
          5/31/20    7202
          6/1/20     7221
          6/2/20     7229
          6/3/20     7240
          6/4/20     7247
          6/5/20     7252
          6/6/20     7259
          6/7/20     7265
          6/8/20     7267
          6/9/20     7274
          6/10/20    7285
          6/11/20    7289
          6/12/20    7294
          6/13/20    7320
          6/14/20    7335
          6/15/20    7347
          6/16/20    7370
          6/17/20    7391
          6/18/20    7409
          6/19/20    7411
          6/20/20    7461
          6/21/20    7474
          Length: 152, dtype: object
```



```
In [234]: dfc.head(2)
```

```
Out[234]:
```

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...	6/12/20
0	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	0	...	23546
1	Albania	41.1533	20.1683	0	0	0	0	0	0	0	...	1416

2 rows × 155 columns

```
In [237]: latitude=dfc.loc[dfc["Country/Region"] == "Australia"].mean()
latitude
#longitude=dfx.loc[dfx[idx] == i].mean()[2:3].values.tolist()
```

```
Out[237]: Lat      -31.996188
Long      141.232788
1/22/20    0.000000
1/23/20    0.000000
1/24/20    0.000000
1/25/20    0.000000
1/26/20    0.500000
1/27/20    0.625000
1/28/20    0.625000
1/29/20    0.750000
1/30/20    1.125000
1/31/20    1.125000
2/1/20     1.500000
2/2/20     1.500000
2/3/20     1.500000
2/4/20     1.625000
2/5/20     1.625000
2/6/20     1.750000
2/7/20     1.875000
2/8/20     1.875000
2/9/20     1.875000
2/10/20    1.875000
2/11/20    1.875000
2/12/20    1.875000
2/13/20    1.875000
2/14/20    1.875000
2/15/20    1.875000
2/16/20    1.875000
2/17/20    1.875000
2/18/20    1.875000
...
5/23/20    889.250000
5/24/20    889.250000
5/25/20    890.750000
5/26/20    892.375000
5/27/20    893.750000
5/28/20    895.625000
5/29/20    898.000000
5/30/20    899.000000
5/31/20    900.250000
6/1/20     902.625000
6/2/20     903.625000
6/3/20     905.000000
6/4/20     905.875000
6/5/20     906.500000
6/6/20     907.375000
6/7/20     908.125000
6/8/20     908.375000
6/9/20     909.250000
6/10/20    910.625000
6/11/20    911.125000
6/12/20    911.750000
6/13/20    915.000000
6/14/20    916.875000
6/15/20    918.375000
6/16/20    921.250000
6/17/20    923.875000
6/18/20    926.125000
6/19/20    926.375000
6/20/20    932.625000
6/21/20    934.250000
Length: 154, dtype: float64
```

In [ ]:

```
In [238]: def hitungSum(dfx,idx):
            nama_negara=list(dfx[idx].unique()) #[afganns...]
            latitude=[]
            longitude=[]
            per_negara=[] #data seluruh negara, per hari
            all_n = []

            for i in nama_negara: # i= Afghanistan --> aljazai
                latitude= dfc.loc[dfc["Country/Region"] == i,"Lat"].mean()
                longitude= dfc.loc[dfc["Country/Region"] == i,"Long"].mean()

                per_negara=dfc.loc[dfx[idx] == i].sum()[3:].values.tolist() #100an kolo
m
                per_negara.insert(0,i) #nama negara
                per_negara.insert(1,latitude)
                per_negara.insert(2,longitude) #3+100an kolom

                all_n.append(per_negara)

            return all_n

headerdf=dfc.columns.values.tolist() #ambil nama kolom dari dfc

dfc_bersih=pd.DataFrame(hitungSum(dfc,"Country/Region"), columns=headerdf)
dfc_bersih
```

Out[238]:

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...
0	Afghanistan	33.000000	65.000000	0	0	0	0	0	0	0	...
1	Albania	41.153300	20.168300	0	0	0	0	0	0	0	...
2	Algeria	28.033900	1.659600	0	0	0	0	0	0	0	...
3	Andorra	42.506300	1.521800	0	0	0	0	0	0	0	...
4	Angola	-11.202700	17.873900	0	0	0	0	0	0	0	...
5	Antigua and Barbuda	17.060800	-61.796400	0	0	0	0	0	0	0	...
6	Argentina	-38.416100	-63.616700	0	0	0	0	0	0	0	...
7	Armenia	40.069100	45.038200	0	0	0	0	0	0	0	...
8	Australia	-31.996188	141.232788	0	0	0	0	4	5	5	...
9	Austria	47.516200	14.550100	0	0	0	0	0	0	0	...
10	Azerbaijan	40.143100	47.576900	0	0	0	0	0	0	0	...
11	Bahamas	25.034300	-77.396300	0	0	0	0	0	0	0	...
12	Bahrain	26.027500	50.550000	0	0	0	0	0	0	0	...
13	Bangladesh	23.685000	90.356300	0	0	0	0	0	0	0	...
14	Barbados	13.193900	-59.543200	0	0	0	0	0	0	0	...
15	Belarus	53.709800	27.953400	0	0	0	0	0	0	0	...
16	Belgium	50.833300	4.000000	0	0	0	0	0	0	0	...
17	Belize	13.193900	-59.543200	0	0	0	0	0	0	0	...
18	Benin	9.307700	2.315800	0	0	0	0	0	0	0	...
19	Bhutan	27.514200	90.433600	0	0	0	0	0	0	0	...
20	Bolivia	-16.290200	-63.588700	0	0	0	0	0	0	0	...
21	Bosnia and Herzegovina	43.915900	17.679100	0	0	0	0	0	0	0	...
22	Botswana	-22.328500	24.684900	0	0	0	0	0	0	0	...
23	Brazil	-14.235000	-51.925300	0	0	0	0	0	0	0	...
24	Brunei	4.535300	114.727700	0	0	0	0	0	0	0	...
25	Bulgaria	42.733900	25.485800	0	0	0	0	0	0	0	...
26	Burkina Faso	12.238300	-1.561600	0	0	0	0	0	0	0	...
27	Burma	21.916200	95.956000	0	0	0	0	0	0	0	...
28	Burundi	-3.373100	29.918900	0	0	0	0	0	0	0	...
29	Cabo Verde	16.538800	-23.041800	0	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
158	Spain	40.000000	-4.000000	0	0	0	0	0	0	0	...
159	Sri Lanka	7.000000	81.000000	0	0	0	0	0	1	1	...
160	Sudan	12.862800	30.217600	0	0	0	0	0	0	0	...
161	Suriname	3.919300	-56.027800	0	0	0	0	0	0	0	...
162	Sweden	63.000000	16.000000	0	0	0	0	0	0	0	...
163	Switzerland	46.818200	8.227500	0	0	0	0	0	0	0	...
164	Syria	34.802075	38.996815	0	0	0	0	0	0	0	...
165	Taiwan*	23.700000	121.000000	1	1	3	3	4	5	8	...
166	Tajikistan	38.861034	71.276093	0	0	0	0	0	0	0	...
167	Tanzania	-6.369000	34.888800	0	0	0	0	0	0	0	...
168	Thailand	15.000000	101.000000	2	3	5	7	8	8	14	...
...	...	...	...	...	...	...	...	...	...	...	...

```
In [99]: dfc.head(20)
```

```
Out[99]:
```

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...	6/12/20
0	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	0	...	23%
1	Albania	41.1533	20.1683	0	0	0	0	0	0	0	...	14%
2	Algeria	28.0339	1.6596	0	0	0	0	0	0	0	...	10%
3	Andorra	42.5063	1.5218	0	0	0	0	0	0	0	...	8%
4	Angola	-11.2027	17.8739	0	0	0	0	0	0	0	...	7%
5	Antigua and Barbuda	17.0608	-61.7964	0	0	0	0	0	0	0	...	6%
6	Argentina	-38.4161	-63.6167	0	0	0	0	0	0	0	...	28%
7	Armenia	40.0691	45.0382	0	0	0	0	0	0	0	...	15%
8	Australia	-31.9505	115.8605	0	0	0	0	0	0	0	...	6%
9	Australia	-37.8136	144.9631	0	0	0	0	1	1	1	...	17%
10	Australia	-41.4545	145.9707	0	0	0	0	0	0	0	...	2%
11	Australia	-34.9285	138.6007	0	0	0	0	0	0	0	...	4%
12	Australia	-12.4634	130.8456	0	0	0	0	0	0	0	...	3%
13	Australia	-33.8688	151.2093	0	0	0	0	3	4	4	...	3%
14	Australia	-35.4735	149.0124	0	0	0	0	0	0	0	...	7%
15	Australia	-28.0167	153.4000	0	0	0	0	0	0	0	...	10%
16	Austria	47.5162	14.5501	0	0	0	0	0	0	0	...	17%
17	Azerbaijan	40.1431	47.5769	0	0	0	0	0	0	0	...	9%
18	Bahamas	25.0343	-77.3963	0	0	0	0	0	0	0	...	7%
19	Bahrain	26.0275	50.5500	0	0	0	0	0	0	0	...	17%

20 rows × 155 columns

## Lakukan hal yang sama untuk dfr dan dfd

```
In [240]: dfr_bersih=pd.DataFrame(hitungSum(dfr,"Country/Region"), columns=headerdf)
          dfd_bersih=pd.DataFrame(hitungSum(dfd,"Country/Region"), columns=headerdf)
```

```
In [241]: dfc_bersih.shape
```

```
Out[241]: (188, 155)
```

```
In [244]: dfr_bersih.head()
```

```
Out[244]:
```

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...	6/12/20
0	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	0	...	392%
1	Albania	41.1533	20.1683	0	0	0	0	0	0	0	...	1034%
2	Algeria	28.0339	1.6596	0	0	0	0	0	0	0	...	7322%
3	Andorra	42.5063	1.5218	0	0	0	0	0	0	0	...	781%
4	Angola	-11.2027	17.8739	0	0	0	0	0	0	0	...	42%

5 rows × 155 columns

```
In [243]: dfd_bersih.shape
```

```
Out[243]: (188, 155)
```

```
In [245]: dfc.isna().sum()
```

```
Out[245]: Country/Region    0
Lat                        0
Long                      0
1/22/20                   0
1/23/20                   0
1/24/20                   0
1/25/20                   0
1/26/20                   0
1/27/20                   0
1/28/20                   0
1/29/20                   0
1/30/20                   0
1/31/20                   0
2/1/20                    0
2/2/20                    0
2/3/20                    0
2/4/20                    0
2/5/20                    0
2/6/20                    0
2/7/20                    0
2/8/20                    0
2/9/20                    0
2/10/20                   0
2/11/20                   0
2/12/20                   0
2/13/20                   0
2/14/20                   0
2/15/20                   0
2/16/20                   0
2/17/20                   0
..
5/23/20                   0
5/24/20                   0
5/25/20                   0
5/26/20                   0
5/27/20                   0
5/28/20                   0
5/29/20                   0
5/30/20                   0
5/31/20                   0
6/1/20                    0
6/2/20                    0
6/3/20                    0
6/4/20                    0
6/5/20                    0
6/6/20                    0
6/7/20                    0
6/8/20                    0
6/9/20                    0
6/10/20                   0
6/11/20                   0
6/12/20                   0
6/13/20                   0
6/14/20                   0
6/15/20                   0
6/16/20                   0
6/17/20                   0
6/18/20                   0
6/19/20                   0
6/20/20                   0
6/21/20                   0
Length: 155, dtype: int64
```



```
In [246]: dfc_bersih.dropna(inplace=True)
          dfr_bersih.dropna(inplace=True)
          dfd_bersih.dropna(inplace=True)

          dfr_bersih.shape
```

```
Out[246]: (188, 155)
```

**YAY!!**

## Membuat dataframe khusus Indonesia saja

**membuat list untuk menyimpan header kolom**

```
In [118]: list_date=dfc_bersih.columns.values.tolist()[3:]
          len(list_date)
```

```
Out[118]: 152
```

## Menggabungkan data positif/confirm, sembuh/recovered dan meninggal/death

**Membuat list untuk menyimpan masing-masing value dari dfc, dfr, dan dfd**

```
In [247]: conf_id=dfc_bersih[dfc_bersih["Country/Region"]=="Indonesia"].values[0].tolist()
          [3:] #mulai dari 3. longitude kan tidak berubah
          rec_id=dfr_bersih[dfr_bersih["Country/Region"]=="Indonesia"].values[0].tolist()
          [3:]
          death_id=dfd_bersih[dfd_bersih["Country/Region"]=="Indonesia"].values[0].tolist()
          [3:]
```

```
In [117]: len(conf_id)
```

```
Out[117]: 152
```

```
In [120]: len(rec_id)
```

```
Out[120]: 152
```

```
In [119]: len(death_id)
```

```
Out[119]: 152
```

```
In [248]: case_id=list((list_date, conf_id, rec_id, death_id))
          len(case_id)
```

```
Out[248]: 4
```

```
In [249]: case_id
```

```
Out[249]: [['1/22/20',
            '1/23/20',
            '1/24/20',
            '1/25/20',
            '1/26/20',
            '1/27/20',
            '1/28/20',
            '1/29/20',
            '1/30/20',
            '1/31/20',
            '2/1/20',
            '2/2/20',
            '2/3/20',
            '2/4/20',
            '2/5/20',
            '2/6/20',
            '2/7/20',
            '2/8/20',
            '2/9/20',
            '2/10/20',
            '2/11/20',
            '2/12/20',
            '2/13/20',
            '2/14/20',
            '2/15/20',
            '2/16/20',
            '2/17/20',
            '2/18/20',
            '2/19/20',
            '2/20/20',
            '2/21/20',
            '2/22/20',
            '2/23/20',
            '2/24/20',
            '2/25/20',
            '2/26/20',
            '2/27/20',
            '2/28/20',
            '2/29/20',
            '3/1/20',
            '3/2/20',
            '3/3/20',
            '3/4/20',
            '3/5/20',
            '3/6/20',
            '3/7/20',
            '3/8/20',
            '3/9/20',
            '3/10/20',
            '3/11/20',
            '3/12/20',
            '3/13/20',
            '3/14/20',
            '3/15/20',
            '3/16/20',
            '3/17/20',
            '3/18/20',
            '3/19/20',
            '3/20/20',
            '3/21/20',
            '3/22/20',
            '3/23/20',
            '3/24/20',
            '3/25/20',
            '3/26/20',
            '3/27/20',
            '3/28/20',
            '3/29/20',
            '3/30/20']]
```

## Buat DF ID

```
In [251]: df_id=pd.DataFrame(case_id).T
df_id.columns=["tanggal","positif","sembuh","meninggal"]
df_id.set_index("tanggal",inplace=True)
df_id.tail(10)
```

Out[251]:

	positif	sembuh	meninggal
tanggal			
6/12/20	36406	13213	2048
6/13/20	37420	13776	2091
6/14/20	38277	14531	2134
6/15/20	39294	15123	2198
6/16/20	40400	15703	2231
6/17/20	41431	16243	2276
6/18/20	42762	16798	2339
6/19/20	43803	17349	2373
6/20/20	45029	17883	2429
6/21/20	45891	18404	2465

## Convert ke CSV

```
In [252]: df_id.to_csv("data_covid_id.csv")
```

## Membuat DF untuk data "Total" seluruh dunia

```
In [128]: global_conf=dfc_bersih.values.tolist()  
          global_conf
```

[illegible]

```
In [138]: dfc_bersih
```

Out[138]:

	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	...
0	Afghanistan	33.000000	65.000000	0	0	0	0	0	0	0	...
1	Albania	41.153300	20.168300	0	0	0	0	0	0	0	...
2	Algeria	28.033900	1.659600	0	0	0	0	0	0	0	...
3	Andorra	42.506300	1.521800	0	0	0	0	0	0	0	...
4	Angola	-11.202700	17.873900	0	0	0	0	0	0	0	...
5	Antigua and Barbuda	17.060800	-61.796400	0	0	0	0	0	0	0	...
6	Argentina	-38.416100	-63.616700	0	0	0	0	0	0	0	...
7	Armenia	40.069100	45.038200	0	0	0	0	0	0	0	...
8	Australia	-31.996188	141.232788	0	0	0	0	4	5	5	...
9	Austria	47.516200	14.550100	0	0	0	0	0	0	0	...
10	Azerbaijan	40.143100	47.576900	0	0	0	0	0	0	0	...
11	Bahamas	25.034300	-77.396300	0	0	0	0	0	0	0	...
12	Bahrain	26.027500	50.550000	0	0	0	0	0	0	0	...
13	Bangladesh	23.685000	90.356300	0	0	0	0	0	0	0	...
14	Barbados	13.193900	-59.543200	0	0	0	0	0	0	0	...
15	Belarus	53.709800	27.953400	0	0	0	0	0	0	0	...
16	Belgium	50.833300	4.000000	0	0	0	0	0	0	0	...
17	Belize	13.193900	-59.543200	0	0	0	0	0	0	0	...
18	Benin	9.307700	2.315800	0	0	0	0	0	0	0	...
19	Bhutan	27.514200	90.433600	0	0	0	0	0	0	0	...
20	Bolivia	-16.290200	-63.588700	0	0	0	0	0	0	0	...
21	Bosnia and Herzegovina	43.915900	17.679100	0	0	0	0	0	0	0	...
22	Botswana	-22.328500	24.684900	0	0	0	0	0	0	0	...
23	Brazil	-14.235000	-51.925300	0	0	0	0	0	0	0	...
24	Brunei	4.535300	114.727700	0	0	0	0	0	0	0	...
25	Bulgaria	42.733900	25.485800	0	0	0	0	0	0	0	...
26	Burkina Faso	12.238300	-1.561600	0	0	0	0	0	0	0	...
27	Burma	21.916200	95.956000	0	0	0	0	0	0	0	...
28	Burundi	-3.373100	29.918900	0	0	0	0	0	0	0	...
29	Cabo Verde	16.538800	-23.041800	0	0	0	0	0	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...
158	Spain	40.000000	-4.000000	0	0	0	0	0	0	0	...
159	Sri Lanka	7.000000	81.000000	0	0	0	0	0	1	1	...
160	Sudan	12.862800	30.217600	0	0	0	0	0	0	0	...
161	Suriname	3.919300	-56.027800	0	0	0	0	0	0	0	...
162	Sweden	63.000000	16.000000	0	0	0	0	0	0	0	...
163	Switzerland	46.818200	8.227500	0	0	0	0	0	0	0	...
164	Syria	34.802075	38.996815	0	0	0	0	0	0	0	...
165	Taiwan*	23.700000	121.000000	1	1	3	3	4	5	8	...
166	Tajikistan	38.861034	71.276093	0	0	0	0	0	0	0	...
167	Tanzania	-6.369000	34.888800	0	0	0	0	0	0	0	...
168	Thailand	15.000000	101.000000	2	3	5	7	8	8	14	...
...	...	...	...	...	...	...	...	...	...	...	...



**BIKIN dataframe untuk global per tanggal**

```
In [148]: sum_c=dfc_bersih.sum()[3:].values.tolist() # data positif per hari seluruh dunia
sum_r=dfr_bersih.sum()[3:].values.tolist()
sum_d=dfd_bersih.sum()[3:].values.tolist()

type(sum_c)
sum_c
```

```
Out[148]: [555,  
654,  
941,  
1434,  
2118,  
2927,  
5578,  
6166,  
8234,  
9927,  
12038,  
16787,  
19881,  
23892,  
27635,  
30794,  
34391,  
37120,  
40150,  
42762,  
44802,  
45221,  
60368,  
66885,  
69030,  
71224,  
73258,  
75136,  
75639,  
76197,  
76819,  
78572,  
78958,  
79525,  
80372,  
81346,  
82704,  
84070,  
85967,  
88325,  
90262,  
92795,  
95075,  
97844,  
101761,  
105782,  
109754,  
113536,  
118375,  
125704,  
130909,  
145204,  
156283,  
167022,  
181452,  
196917,  
216161,  
243084,  
272698,  
304844,  
337597,  
378381,  
418569,  
468155,  
530138,  
594178,  
661544,  
720695,  
783580.]
```

## Menggabungkan dataframe positif,sembuh dan meninggal jadi 1

```
In [253]: df_total=pd.DataFrame(list((list_date,sum_c,sum_r,sum_d))).T
df_total.columns=["tanggal","positif","sembuh","meninggal"]
df_total.set_index("tanggal",inplace=True)
df_total.head()
```

Out[253]:

	positif	sembuh	meninggal
tanggal			
1/22/20	555	28	17
1/23/20	654	30	18
1/24/20	941	36	26
1/25/20	1434	39	42
1/26/20	2118	52	56

```
In [254]: df_total.to_csv("data_covid_total.csv")
```

## Membuat data global covid per negara

ambil data terakhir saja tanggal 6/21/20

```
In [173]: dfc_bersih["Country/Region"]
```

```
Out[173]: 0      Afghanistan
1      Albania
2      Algeria
3      Andorra
4      Angola
5      Antigua and Barbuda
6      Argentina
7      Armenia
8      Australia
9      Austria
10     Azerbaijan
11     Bahamas
12     Bahrain
13     Bangladesh
14     Barbados
15     Belarus
16     Belgium
17     Belize
18     Benin
19     Bhutan
20     Bolivia
21     Bosnia and Herzegovina
22     Botswana
23     Brazil
24     Brunei
25     Bulgaria
26     Burkina Faso
27     Burma
28     Burundi
29     Cabo Verde
...
158     Spain
159     Sri Lanka
160     Sudan
161     Suriname
162     Sweden
163     Switzerland
164     Syria
165     Taiwan*
166     Tajikistan
167     Tanzania
168     Thailand
169     Timor-Leste
170     Togo
171     Trinidad and Tobago
172     Tunisia
173     Turkey
174     US
175     Uganda
176     Ukraine
177     United Arab Emirates
178     United Kingdom
179     Uruguay
180     Uzbekistan
181     Venezuela
182     Vietnam
183     West Bank and Gaza
184     Western Sahara
185     Yemen
186     Zambia
187     Zimbabwe
Name: Country/Region, Length: 188, dtype: object
```

```
In [190]: lin=dfc_bersih["Country/Region"].values.tolist()
lila=dfc_bersih["Lat"].values.tolist()
lilo=dfc_bersih["Long"].values.tolist()
lipo=dfc_bersih["6/21/20"].values.tolist()
lisem=dfr_bersih["6/21/20"].values.tolist()
limen=dfd_bersih["6/21/20"].values.tolist()

totli=[]
totli.append(lin)
totli.append(lila)
totli.append(lilo)
totli.append(lipo)
totli.append(lisem)
totli.append(limen)

pd_li=pd.DataFrame(totli).T
pd_li.columns=["negara","latitude","longitude","positif","sembuh","meninggal"]
pd_li
```

Out[190]:

	negara	latitude	longitude	positif	sembuh	meninggal
0	Afghanistan	33	65	28833	8764	581
1	Albania	41.1533	20.1683	1962	1134	44
2	Algeria	28.0339	1.6596	11771	8422	845
3	Andorra	42.5063	1.5218	855	792	52
4	Angola	-11.2027	17.8739	183	77	9
5	Antigua and Barbuda	17.0608	-61.7964	26	22	3
6	Argentina	-38.4161	-63.6167	42785	12728	1011
7	Armenia	40.0691	45.0382	20268	9002	350
8	Australia	-31.9962	141.233	7474	6903	102
9	Austria	47.5162	14.5501	17341	16197	690
10	Azerbaijan	40.1431	47.5769	12729	6799	154
11	Bahamas	25.0343	-77.3963	104	77	11
12	Bahrain	26.0275	50.55	21764	16419	63
13	Bangladesh	23.685	90.3563	112306	45077	1464
14	Barbados	13.1939	-59.5432	97	85	7
15	Belarus	53.7098	27.9534	58505	37666	346
16	Belgium	50.8333	4	60550	16771	9696
17	Belize	13.1939	-59.5432	22	17	2
18	Benin	9.3077	2.3158	765	253	13
19	Bhutan	27.5142	90.4336	68	30	0
20	Bolivia	-16.2902	-63.5887	24388	5454	773
21	Bosnia and Herzegovina	43.9159	17.6791	3273	2241	169
22	Botswana	-22.3285	24.6849	89	25	1
23	Brazil	-14.235	-51.9253	1083341	588118	50591
24	Brunei	4.5353	114.728	141	138	3
25	Bulgaria	42.7339	25.4858	3905	2074	199
26	Burkina Faso	12.2383	-1.5616	903	814	53
27	Burma	21.9162	95.956	290	200	6
28	Burundi	-3.3731	29.9189	144	93	1
29	Cabo Verde	16.5388	-23.0418	890	413	8
...	...	...	...	...	...	...
158	Spain	40	-4	246272	150376	28323
159	Sri Lanka	7	81	1950	1498	11
160	Sudan	12.8628	30.2176	8580	3325	521
161	Suriname	3.9193	-56.0278	314	106	8
162	Sweden	63	16	56043	0	5053
163	Switzerland	46.8182	8.2275	31292	29000	1956
164	Syria	34.8021	38.9968	204	83	7
165	Taiwan*	23.7	121	446	434	7
166	Tajikistan	38.861	71.2761	5457	3995	52
167	Tanzania	-6.369	34.8888	509	183	21
168	Thailand	15	101	3148	3018	58
169	Timor-Leste	-8.87422	125.728	24	24	0
170	Togo	6.6185	0.8218	500	275	10

```
In [255]: df_global=dfc_bersih[["Country/Region"]]  
df_global["latitude"]=dfc_bersih[["Lat"]]  
df_global["longitude"]=dfc_bersih[["Long"]]  
df_global["positif"]=dfc_bersih[["6/21/20"]]  
df_global["sembuh"]=dfr_bersih[["6/21/20"]]  
df_global["meninggal"]=dfd_bersih[["6/21/20"]]
```

D:\DataScience\lib\site-packages\ipykernel\_launcher.py:2: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

D:\DataScience\lib\site-packages\ipykernel\_launcher.py:3: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

This is separate from the ipykernel package so we can avoid doing imports until

D:\DataScience\lib\site-packages\ipykernel\_launcher.py:4: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

after removing the cwd from sys.path.

D:\DataScience\lib\site-packages\ipykernel\_launcher.py:5: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

"""

D:\DataScience\lib\site-packages\ipykernel\_launcher.py:6: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

```
In [256]: df_global.columns=['negara','latitude','longitude','positif', 'sembuh', 'meninggal']
```



```
In [257]: df_global
```

Out[257]:

	negara	latitude	longitude	positif	sembuh	meninggal
0	Afghanistan	33.000000	65.000000	28833	8764	581
1	Albania	41.153300	20.168300	1962	1134	44
2	Algeria	28.033900	1.659600	11771	8422	845
3	Andorra	42.506300	1.521800	855	792	52
4	Angola	-11.202700	17.873900	183	77	9
5	Antigua and Barbuda	17.060800	-61.796400	26	22	3
6	Argentina	-38.416100	-63.616700	42785	12728	1011
7	Armenia	40.069100	45.038200	20268	9002	350
8	Australia	-31.996188	141.232788	7474	6903	102
9	Austria	47.516200	14.550100	17341	16197	690
10	Azerbaijan	40.143100	47.576900	12729	6799	154
11	Bahamas	25.034300	-77.396300	104	77	11
12	Bahrain	26.027500	50.550000	21764	16419	63
13	Bangladesh	23.685000	90.356300	112306	45077	1464
14	Barbados	13.193900	-59.543200	97	85	7
15	Belarus	53.709800	27.953400	58505	37666	346
16	Belgium	50.833300	4.000000	60550	16771	9696
17	Belize	13.193900	-59.543200	22	17	2
18	Benin	9.307700	2.315800	765	253	13
19	Bhutan	27.514200	90.433600	68	30	0
20	Bolivia	-16.290200	-63.588700	24388	5454	773
21	Bosnia and Herzegovina	43.915900	17.679100	3273	2241	169
22	Botswana	-22.328500	24.684900	89	25	1
23	Brazil	-14.235000	-51.925300	1083341	588118	50591
24	Brunei	4.535300	114.727700	141	138	3
25	Bulgaria	42.733900	25.485800	3905	2074	199
26	Burkina Faso	12.238300	-1.561600	903	814	53
27	Burma	21.916200	95.956000	290	200	6
28	Burundi	-3.373100	29.918900	144	93	1
29	Cabo Verde	16.538800	-23.041800	890	413	8
...	...	...	...	...	...	...
158	Spain	40.000000	-4.000000	246272	150376	28323
159	Sri Lanka	7.000000	81.000000	1950	1498	11
160	Sudan	12.862800	30.217600	8580	3325	521
161	Suriname	3.919300	-56.027800	314	106	8
162	Sweden	63.000000	16.000000	56043	0	5053
163	Switzerland	46.818200	8.227500	31292	29000	1956
164	Syria	34.802075	38.996815	204	83	7
165	Taiwan*	23.700000	121.000000	446	434	7
166	Tajikistan	38.861034	71.276093	5457	3995	52
167	Tanzania	-6.369000	34.888800	509	183	21
168	Thailand	15.000000	101.000000	3148	3018	58
169	Timor-Leste	-8.874217	125.727539	24	24	0
170	Togo	6.616500	0.824000	500	275	10

```
In [259]: df_global.shape
```

```
Out[259]: (188, 6)
```

```
In [261]: df_global.set_index('negara', inplace=True)
```

```
In [169]: df_global.shape
```

```
Out[169]: (188, 5)
```

```
In [262]: df_global.head()
```

```
Out[262]:
```

	latitude	longitude	positif	sembuh	meninggal
negara					
Afghanistan	33.0000	65.0000	28833	8764	581
Albania	41.1533	20.1683	1962	1134	44
Algeria	28.0339	1.6596	11771	8422	845
Andorra	42.5063	1.5218	855	792	52
Angola	-11.2027	17.8739	183	77	9

```
In [263]: df_global.to_csv("data_global.csv")
```

```
In [ ]:
```