exercice

June 9, 2020

1 The SARS-CoV-2 (Covid-19) epidemic analysis

[1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd
import isoweek
import os.path
from os import path
```

The data on the Covid-19 incidence are available here. We download them as a file in CSV format.

[2]:
```
data_url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/
 ↪csse_covid_19_data/csse_covid_19_time_series/
 ↪time_series_covid19_confirmed_global.csv"
```

Data downloaded on 09.06.2020

This is the documentation of the data from the download site:

Column name	Description
`Province/State`	Province/State
`Country/Region`	Country/Region
`Lat`	Latitude
`Long`	Longitude
`1/22/20`	Dates

[3]:
```
exists = os.path.isfile('statistics.csv')
if exists:
    raw_data = pd.read_csv('statistics.csv')
    df = pd.DataFrame(raw_data)
else:
    raw_data = pd.read_csv(data_url)
    df = pd.DataFrame(raw_data)
    df.to_csv('statistics.csv')

raw_data
```

[3]:

	Unnamed: 0	Province/State	Country/Region
0	0	NaN	Afghanistan
1	1	NaN	Albania
2	2	NaN	Algeria
3	3	NaN	Andorra
4	4	NaN	Angola
5	5	NaN	Antigua and Barbuda
6	6	NaN	Argentina
7	7	NaN	Armenia
8	8	Australian Capital Territory	Australia
9	9	New South Wales	Australia
10	10	Northern Territory	Australia
11	11	Queensland	Australia
12	12	South Australia	Australia
13	13	Tasmania	Australia
14	14	Victoria	Australia
15	15	Western Australia	Australia
16	16	NaN	Austria
17	17	NaN	Azerbaijan
18	18	NaN	Bahamas
19	19	NaN	Bahrain
20	20	NaN	Bangladesh
21	21	NaN	Barbados
22	22	NaN	Belarus
23	23	NaN	Belgium
24	24	NaN	Benin
25	25	NaN	Bhutan
26	26	NaN	Bolivia
27	27	NaN	Bosnia and Herzegovina
28	28	NaN	Brazil
29	29	NaN	Brunei
..	…	…	…
236	236	NaN	Timor-Leste
237	237	NaN	Belize
238	238	NaN	Laos
239	239	NaN	Libya
240	240	NaN	West Bank and Gaza
241	241	NaN	Guinea-Bissau
242	242	NaN	Mali
243	243	NaN	Saint Kitts and Nevis
244	244	Northwest Territories	Canada
245	245	Yukon	Canada
246	246	NaN	Kosovo
247	247	NaN	Burma
248	248	Anguilla	United Kingdom
249	249	British Virgin Islands	United Kingdom
250	250	Turks and Caicos Islands	United Kingdom

251	251		NaN	MS Zaandam
252	252		NaN	Botswana
253	253		NaN	Burundi
254	254		NaN	Sierra Leone
255	255	Bonaire, Sint Eustatius and Saba		Netherlands
256	256		NaN	Malawi
257	257	Falkland Islands (Malvinas)		United Kingdom
258	258	Saint Pierre and Miquelon		France
259	259		NaN	South Sudan
260	260		NaN	Western Sahara
261	261		NaN	Sao Tome and Principe
262	262		NaN	Yemen
263	263		NaN	Comoros
264	264		NaN	Tajikistan
265	265		NaN	Lesotho

	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	\
0	33.000000	65.000000	0	0	0	0	0	
1	41.153300	20.168300	0	0	0	0	0	
2	28.033900	1.659600	0	0	0	0	0	
3	42.506300	1.521800	0	0	0	0	0	
4	-11.202700	17.873900	0	0	0	0	0	
5	17.060800	-61.796400	0	0	0	0	0	
6	-38.416100	-63.616700	0	0	0	0	0	
7	40.069100	45.038200	0	0	0	0	0	
8	-35.473500	149.012400	0	0	0	0	0	
9	-33.868800	151.209300	0	0	0	0	3	
10	-12.463400	130.845600	0	0	0	0	0	
11	-28.016700	153.400000	0	0	0	0	0	
12	-34.928500	138.600700	0	0	0	0	0	
13	-41.454500	145.970700	0	0	0	0	0	
14	-37.813600	144.963100	0	0	0	0	1	
15	-31.950500	115.860500	0	0	0	0	0	
16	47.516200	14.550100	0	0	0	0	0	
17	40.143100	47.576900	0	0	0	0	0	
18	25.034300	-77.396300	0	0	0	0	0	
19	26.027500	50.550000	0	0	0	0	0	
20	23.685000	90.356300	0	0	0	0	0	
21	13.193900	-59.543200	0	0	0	0	0	
22	53.709800	27.953400	0	0	0	0	0	
23	50.833300	4.000000	0	0	0	0	0	
24	9.307700	2.315800	0	0	0	0	0	
25	27.514200	90.433600	0	0	0	0	0	
26	-16.290200	-63.588700	0	0	0	0	0	
27	43.915900	17.679100	0	0	0	0	0	
28	-14.235000	-51.925300	0	0	0	0	0	
29	4.535300	114.727700	0	0	0	0	0	

..	…	…	…	…	…	…	…	
236	-8.874217	125.727539	0	0	0	0	0	
237	13.193900	-59.543200	0	0	0	0	0	
238	19.856270	102.495496	0	0	0	0	0	
239	26.335100	17.228331	0	0	0	0	0	
240	31.952200	35.233200	0	0	0	0	0	
241	11.803700	-15.180400	0	0	0	0	0	
242	17.570692	-3.996166	0	0	0	0	0	
243	17.357822	-62.782998	0	0	0	0	0	
244	64.825500	-124.845700	0	0	0	0	0	
245	64.282300	-135.000000	0	0	0	0	0	
246	42.602636	20.902977	0	0	0	0	0	
247	21.916200	95.956000	0	0	0	0	0	
248	18.220600	-63.068600	0	0	0	0	0	
249	18.420700	-64.640000	0	0	0	0	0	
250	21.694000	-71.797900	0	0	0	0	0	
251	0.000000	0.000000	0	0	0	0	0	
252	-22.328500	24.684900	0	0	0	0	0	
253	-3.373100	29.918900	0	0	0	0	0	
254	8.460555	-11.779889	0	0	0	0	0	
255	12.178400	-68.238500	0	0	0	0	0	
256	-13.254308	34.301525	0	0	0	0	0	
257	-51.796300	-59.523600	0	0	0	0	0	
258	46.885200	-56.315900	0	0	0	0	0	
259	6.877000	31.307000	0	0	0	0	0	
260	24.215500	-12.885800	0	0	0	0	0	
261	0.186360	6.613081	0	0	0	0	0	
262	15.552727	48.516388	0	0	0	0	0	
263	-11.645500	43.333300	0	0	0	0	0	
264	38.861034	71.276093	0	0	0	0	0	
265	-29.609988	28.233608	0	0	0	0	0	

	…	5/30/20	5/31/20	6/1/20	6/2/20	6/3/20	6/4/20	6/5/20	6/6/20	\
0	…	14525	15205	15750	16509	17267	18054	18969	19551	
1	…	1122	1137	1143	1164	1184	1197	1212	1232	
2	…	9267	9394	9513	9626	9733	9831	9935	10050	
3	…	764	764	765	844	851	852	852	852	
4	…	84	86	86	86	86	86	86	88	
5	…	25	26	26	26	26	26	26	26	
6	…	16214	16851	17415	18319	19268	20197	21037	22020	
7	…	8927	9282	9492	10009	10524	11221	11817	12364	
8	…	107	107	107	107	107	107	107	108	
9	…	3095	3098	3104	3104	3106	3110	3110	3109	
10	…	29	29	29	29	29	29	29	29	
11	…	1058	1058	1059	1059	1060	1060	1061	1061	
12	…	440	440	440	440	440	440	440	440	
13	…	228	228	228	228	228	228	228	228	

14	…	1649	1653	1663	1670	1678	1681	1681	1685
15	…	586	589	591	592	592	592	596	599
16	…	16685	16731	16733	16759	16771	16805	16843	16898
17	…	5246	5494	5662	5935	6260	6522	6860	7239
18	…	102	102	102	102	102	102	102	103
19	…	10793	11398	11871	12311	12815	13296	13835	14383
20	…	44608	47153	49534	52445	55140	57563	60391	63026
21	…	92	92	92	92	92	92	92	92
22	…	41658	42556	43403	44255	45116	45981	46868	47751
23	…	58186	58381	58517	58615	58685	58767	58907	59072
24	…	224	232	243	244	244	261	261	261
25	…	33	43	43	47	47	47	48	48
26	…	9592	9982	10531	10991	11638	12245	12728	13358
27	…	2494	2510	2524	2535	2551	2594	2606	2606
28	…	498440	514849	526447	555383	584016	614941	645771	672846
29	…	141	141	141	141	141	141	141	141
..	…	…	…	…	…	…	…	…	…
236	…	24	24	24	24	24	24	24	24
237	…	18	18	18	18	18	18	19	19
238	…	19	19	19	19	19	19	19	19
239	…	130	156	168	182	196	209	239	256
240	…	447	448	449	451	457	464	464	464
241	…	1256	1256	1339	1339	1339	1339	1368	1368
242	…	1250	1265	1315	1351	1386	1461	1485	1523
243	…	15	15	15	15	15	15	15	15
244	…	5	5	5	5	5	5	5	5
245	…	11	11	11	11	11	11	11	11
246	…	1064	1064	1064	1064	1142	1142	1142	1142
247	…	224	224	228	232	233	236	236	240
248	…	3	3	3	3	3	3	3	3
249	…	8	8	8	8	8	8	8	8
250	…	12	12	12	12	12	12	12	12
251	…	9	9	9	9	9	9	9	9
252	…	35	35	38	40	40	40	40	40
253	…	63	63	63	63	63	63	63	83
254	…	852	861	865	896	909	914	929	946
255	…	6	6	7	7	7	7	7	7
256	…	279	284	336	358	369	393	409	409
257	…	13	13	13	13	13	13	13	13
258	…	1	1	1	1	1	1	1	1
259	…	994	994	994	994	994	994	994	994
260	…	9	9	9	9	9	9	9	9
261	…	479	483	484	484	484	485	499	499
262	…	310	323	354	399	419	453	469	482
263	…	106	106	106	132	132	132	132	141
264	…	3807	3930	4013	4100	4191	4289	4370	4453
265	…	2	2	2	2	4	4	4	4

	6/7/20	6/8/20
0	20342	20917
1	1246	1263
2	10154	10265
3	852	852
4	91	92
5	26	26
6	22794	23620
7	13130	13325
8	108	108
9	3112	3114
10	29	29
11	1062	1062
12	440	440
13	228	228
14	1687	1687
15	599	599
16	16902	16968
17	7553	7876
18	103	103
19	14763	15417
20	65769	68504
21	92	92
22	48630	49453
23	59226	59348
24	261	288
25	59	59
26	13643	13949
27	2606	2704
28	691758	707412
29	141	141
..	…	…
236	24	24
237	19	19
238	19	19
239	256	332
240	472	473
241	1368	1389
242	1533	1547
243	15	15
244	5	5
245	11	11
246	1142	1263
247	242	244
248	3	3
249	8	8

```
250      12      12
251       9       9
252      40      42
253      83      83
254     969    1001
255       7       7
256     438     443
257      13      13
258       1       1
259    1317    1604
260       9       9
261     513     513
262     484     496
263     141     141
264    4529    4609
265       4       4

[266 rows x 144 columns]
```

Remove Long and Lat columns (just for convenience) and make a spared copy in df_total for the "world" graph

[4]:
```
df_total=df.drop(columns=['Lat', 'Long'])
df=df_total
```

Remove "not interesting" countries

[5]:
```
df=df.drop(df[(df['Country/Region'] != 'Belgium') & (df['Country/Region'] !=␣
↪'China') & (df['Country/Region'] != 'France') & (df['Country/Region'] !=␣
↪'Germany') & (df['Country/Region'] != 'Iran') & (df['Country/Region'] !=␣
↪'Italy') & (df['Country/Region'] != 'Japan') & (df['Country/Region'] !=␣
↪'Korea South') & (df['Country/Region'] != 'Netherlands') & (df['Country/
↪Region'] != 'Portugal') & (df['Country/Region'] != 'Spain') & (df['Country/
↪Region'] != 'United Kingdom') & (df['Country/Region'] != 'US')].index)
```

For convenience change China to Hong Kong in the Hong Kong line

[6]:
```
df=df.set_value(df[(df['Province/State'] == 'Hong Kong')].index, 'Country/
↪Region', 'Hong Kong')
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:1: FutureWarning:
set_value is deprecated and will be removed in a future release. Please use
.at[] or .iat[] accessors instead
  """Entry point for launching an IPython kernel.
```

Remove colonies of France, Netherlands and UK

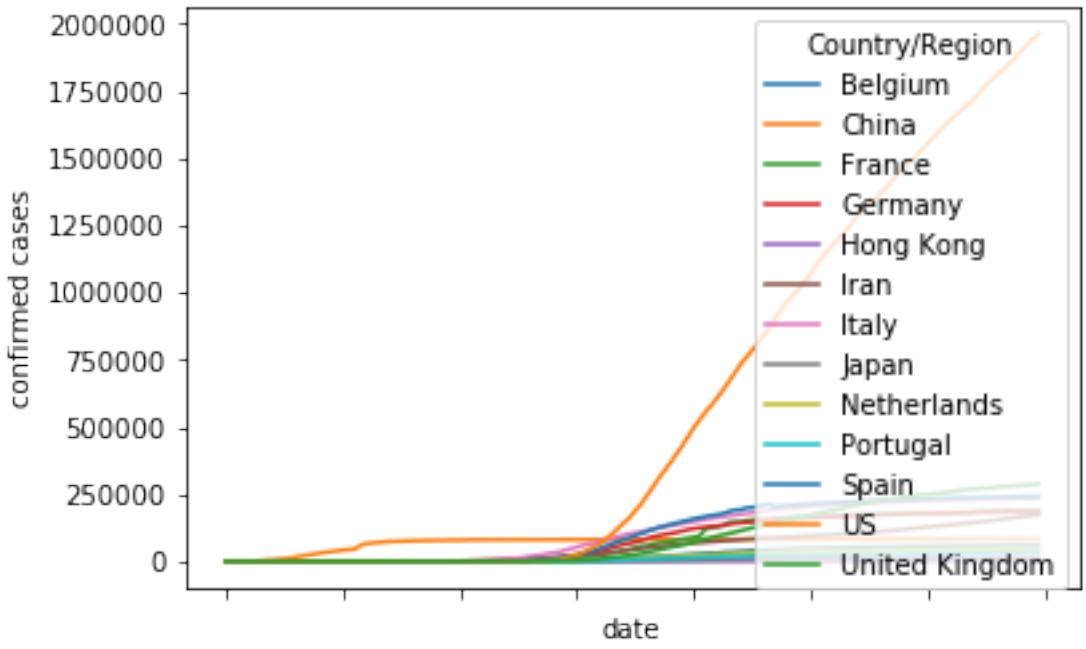
[7]:
```
fr=df[(df['Country/Region']=='France')]
fr=fr['Province/State']
fr=fr.dropna()

ne=df[(df['Country/Region']=='Netherlands')]
ne=ne['Province/State']
ne=ne.dropna()

uk=df[(df['Country/Region']=='United Kingdom')]
uk=uk['Province/State']
uk=uk.dropna()

df=df.drop(fr.index)
df=df.drop(ne.index)
df=df.drop(uk.index)
```

Remove Province/State column and compute total daily sum for China

[8]:
```
df.drop('Province/State', axis = 1, inplace = True)
grouped=df.groupby('Country/Region')
df=grouped.sum()
```

Construct graphs for the countries above

[9]:
```
ax=df.transpose().plot()
ax.set_xlabel("date")
ax.set_ylabel("confirmed cases")
plt.show()
```



Next we make the analogous graph for the Covid-19 incidence in the world

[10]:
```
df=df_total
df.drop('Province/State', axis = 1, inplace = True)
df.drop('Country/Region', axis = 1, inplace = True)
df=df.sum(axis=0)

ax=df.transpose().plot()
ax.set_xlabel("date")
ax.set_ylabel("confirmed cases")
plt.show()
```