

exercice_fr

July 4, 2024

1 Sujet 7 : Autour du SARS-CoV-2 (Covid-19)

1.1 Analyse rapide des donnees recuperees

Imports des packages necessaires a l'analyse

```
[1]: ## import
      %matplotlib inline
      import matplotlib.pyplot as plt
      from matplotlib.pyplot import cm
      import pandas as pd
      import numpy as np
      import datetime
      import os.path
      from urllib.request import urlretrieve
```

Pour cette analyse, nous utiliserons les données compilées par le Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) mises à disposition sur GitHub, plus particulièrement les données **time_series_covid19_confirmed_global.csv**

Ces donnees sont disponibles aussi à l'adresse : https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv.

Nous commencons par verifier la presence d'une copie des donnees. Si elle n'existe pas, nous allons la recuperer sur le site et creons la copie locale.

```
[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_local = "time_series_covid19_confirmed_global.csv"
      if not os.path.isfile(data_local):
          urlretrieve(data_url, data_local)
```

1.1.1 1ere visualisation des donnees :

Comptage rapide du nombre de donnees presentes :

```
[3]: raw_data = pd.read_csv(data_local)
raw_data.shape
```

```
[3]: (289, 1147)
```

Le fichier contient 289 lignes pour 1147 colonnes

On regarde le formatage des donnees a savoir * chaque pays/province est indiquee en ligne * les donnees des colonnes 1 a 4 correspondent aux infos du pays/region convenue * les donnees des colonnes 5 et suivantes correspondent aux nombres de cas cummule par jour (jour indique en etiquette de la colonne)

```
[4]: raw_data
```

```
[4]:
```

	Province/State	Country/Region \
0	NaN	Afghanistan
1	NaN	Albania
2	NaN	Algeria
3	NaN	Andorra
4	NaN	Angola
5	NaN	Antarctica
6	NaN	Antigua and Barbuda
7	NaN	Argentina
8	NaN	Armenia
9	Australian Capital Territory	Australia
10	New South Wales	Australia
11	Northern Territory	Australia
12	Queensland	Australia
13	South Australia	Australia
14	Tasmania	Australia
15	Victoria	Australia
16	Western Australia	Australia
17	NaN	Austria
18	NaN	Azerbaijan
19	NaN	Bahamas
20	NaN	Bahrain
21	NaN	Bangladesh
22	NaN	Barbados
23	NaN	Belarus
24	NaN	Belgium
25	NaN	Belize
26	NaN	Benin
27	NaN	Bhutan
28	NaN	Bolivia
29	NaN	Bosnia and Herzegovina
..	...	...
259	NaN	Tuvalu
260	NaN	US

261		NaN	Uganda
262		NaN	Ukraine
263		NaN	United Arab Emirates
264		Anguilla	United Kingdom
265		Bermuda	United Kingdom
266		British Virgin Islands	United Kingdom
267		Cayman Islands	United Kingdom
268		Channel Islands	United Kingdom
269		Falkland Islands (Malvinas)	United Kingdom
270		Gibraltar	United Kingdom
271		Guernsey	United Kingdom
272		Isle of Man	United Kingdom
273		Jersey	United Kingdom
274		Montserrat	United Kingdom
275		Pitcairn Islands	United Kingdom
276		Saint Helena, Ascension and Tristan da Cunha	United Kingdom
277		Turks and Caicos Islands	United Kingdom
278		NaN	United Kingdom
279		NaN	Uruguay
280		NaN	Uzbekistan
281		NaN	Vanuatu
282		NaN	Venezuela
283		NaN	Vietnam
284		NaN	West Bank and Gaza
285		NaN	Winter Olympics 2022
286		NaN	Yemen
287		NaN	Zambia
288		NaN	Zimbabwe

	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	\
0	33.939110	67.709953	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	-71.949900	23.347000	0	0	0	0	0	
6	17.060800	-61.796400	0	0	0	0	0	
7	-38.416100	-63.616700	0	0	0	0	0	
8	40.069100	45.038200	0	0	0	0	0	
9	-35.473500	149.012400	0	0	0	0	0	
10	-33.868800	151.209300	0	0	0	0	3	
11	-12.463400	130.845600	0	0	0	0	0	
12	-27.469800	153.025100	0	0	0	0	0	
13	-34.928500	138.600700	0	0	0	0	0	
14	-42.882100	147.327200	0	0	0	0	0	
15	-37.813600	144.963100	0	0	0	0	1	
16	-31.950500	115.860500	0	0	0	0	0	

17	47.516200	14.550100	0	0	0	0	0
18	40.143100	47.576900	0	0	0	0	0
19	25.025885	-78.035889	0	0	0	0	0
20	26.027500	50.550000	0	0	0	0	0
21	23.685000	90.356300	0	0	0	0	0
22	13.193900	-59.543200	0	0	0	0	0
23	53.709800	27.953400	0	0	0	0	0
24	50.833300	4.469936	0	0	0	0	0
25	17.189900	-88.497600	0	0	0	0	0
26	9.307700	2.315800	0	0	0	0	0
27	27.514200	90.433600	0	0	0	0	0
28	-16.290200	-63.588700	0	0	0	0	0
29	43.915900	17.679100	0	0	0	0	0
..	...	...	...	...	...	...	...
259	-7.109500	177.649300	0	0	0	0	0
260	40.000000	-100.000000	1	1	2	2	5
261	1.373333	32.290275	0	0	0	0	0
262	48.379400	31.165600	0	0	0	0	0
263	23.424076	53.847818	0	0	0	0	0
264	18.220600	-63.068600	0	0	0	0	0
265	32.307800	-64.750500	0	0	0	0	0
266	18.420700	-64.640000	0	0	0	0	0
267	19.313300	-81.254600	0	0	0	0	0
268	49.372300	-2.364400	0	0	0	0	0
269	-51.796300	-59.523600	0	0	0	0	0
270	36.140800	-5.353600	0	0	0	0	0
271	49.448196	-2.589490	0	0	0	0	0
272	54.236100	-4.548100	0	0	0	0	0
273	49.213800	-2.135800	0	0	0	0	0
274	16.742498	-62.187366	0	0	0	0	0
275	-24.376800	-128.324200	0	0	0	0	0
276	-7.946700	-14.355900	0	0	0	0	0
277	21.694000	-71.797900	0	0	0	0	0
278	55.378100	-3.436000	0	0	0	0	0
279	-32.522800	-55.765800	0	0	0	0	0
280	41.377491	64.585262	0	0	0	0	0
281	-15.376700	166.959200	0	0	0	0	0
282	6.423800	-66.589700	0	0	0	0	0
283	14.058324	108.277199	0	2	2	2	2
284	31.952200	35.233200	0	0	0	0	0
285	39.904200	116.407400	0	0	0	0	0
286	15.552727	48.516388	0	0	0	0	0
287	-13.133897	27.849332	0	0	0	0	0
288	-19.015438	29.154857	0	0	0	0	0

	1/27/20	...	2/28/23	3/1/23	3/2/23	3/3/23	\
0	0	...	209322	209340	209358	209362	

1	0	...	334391	334408	334408	334427
2	0	...	271441	271448	271463	271469
3	0	...	47866	47875	47875	47875
4	0	...	105255	105277	105277	105277
5	0	...	11	11	11	11
6	0	...	9106	9106	9106	9106
7	0	...	10044125	10044125	10044125	10044125
8	0	...	446819	446819	446819	446819
9	0	...	232018	232018	232619	232619
10	4	...	3900969	3900969	3908129	3908129
11	0	...	104931	104931	105021	105021
12	0	...	1796633	1796633	1800236	1800236
13	0	...	880207	880207	881911	881911
14	0	...	286264	286264	286264	286897
15	1	...	2874262	2874262	2877260	2877260
16	0	...	1291077	1291077	1293461	1293461
17	0	...	5911294	5919616	5926148	5931247
18	0	...	828548	828588	828628	828648
19	0	...	37491	37491	37491	37491
20	0	...	707480	707828	708061	708532
21	0	...	2037773	2037829	2037829	2037829
22	0	...	106645	106645	106645	106645
23	0	...	994037	994037	994037	994037
24	0	...	4717655	4717655	4727795	4727795
25	0	...	70757	70757	70757	70757
26	0	...	27990	27990	27990	27990
27	0	...	62615	62620	62620	62620
28	0	...	1193009	1193256	1193418	1193650
29	0	...	401575	401636	401636	401636
..	...	...	...	...	...	...
259	0	...	2805	2805	2805	2805
260	5	...	103443455	103533872	103589757	103648690
261	0	...	170504	170504	170504	170504
262	0	...	5693846	5701249	5701333	5701474
263	0	...	1051998	1052122	1052247	1052382
264	0	...	3904	3904	3904	3904
265	0	...	18799	18814	18814	18814
266	0	...	7305	7305	7305	7305
267	0	...	31472	31472	31472	31472
268	0	...	0	0	0	0
269	0	...	1930	1930	1930	1930
270	0	...	20423	20423	20423	20433
271	0	...	34867	34929	34929	34929
272	0	...	38008	38008	38008	38008
273	0	...	66391	66391	66391	66391
274	0	...	1403	1403	1403	1403
275	0	...	4	4	4	4

276	0	...	2166	2166	2166	2166
277	0	...	6551	6551	6551	6551
278	0	...	24370150	24370150	24396530	24396530
279	0	...	1034303	1034303	1034303	1034303
280	0	...	250932	251071	251071	251071
281	0	...	12014	12014	12014	12014
282	0	...	551981	551986	551986	552014
283	2	...	11526917	11526926	11526937	11526950
284	0	...	703228	703228	703228	703228
285	0	...	535	535	535	535
286	0	...	11945	11945	11945	11945
287	0	...	343012	343012	343079	343079
288	0	...	263921	264127	264127	264127

	3/4/23	3/5/23	3/6/23	3/7/23	3/8/23	3/9/23
0	209369	209390	209406	209436	209451	209451
1	334427	334427	334427	334427	334443	334457
2	271469	271477	271477	271490	271494	271496
3	47875	47875	47875	47875	47890	47890
4	105277	105277	105277	105277	105288	105288
5	11	11	11	11	11	11
6	9106	9106	9106	9106	9106	9106
7	10044125	10044125	10044957	10044957	10044957	10044957
8	446819	446819	446819	446819	447308	447308
9	232619	232619	232619	232619	232619	232974
10	3908129	3908129	3908129	3908129	3908129	3915992
11	105021	105021	105021	105021	105021	105111
12	1800236	1800236	1800236	1800236	1800236	1800236
13	881911	881911	881911	881911	881911	883620
14	286897	286897	286897	286897	286897	287507
15	2877260	2877260	2877260	2877260	2877260	2880559
16	1293461	1293461	1293461	1293461	1293461	1293461
17	5936666	5940935	5943417	5949418	5955860	5961143
18	828682	828721	828730	828783	828819	828825
19	37491	37491	37491	37491	37491	37491
20	708768	709230	709230	709858	710306	710693
21	2037829	2037829	2037829	2037829	2037871	2037871
22	106645	106645	106645	106645	106645	106798
23	994037	994037	994037	994037	994037	994037
24	4727795	4727795	4727795	4727795	4727795	4739365
25	70757	70757	70757	70757	70757	70757
26	27990	27990	27990	27999	27999	27999
27	62620	62620	62620	62620	62627	62627
28	1193815	1193908	1193970	1194069	1194187	1194277
29	401636	401636	401636	401636	401729	401729
..	...	...	...	...	...	...
259	2805	2805	2805	2805	2805	2805

260	103650837	103646975	103655539	103690910	103755771	103802702
261	170504	170504	170504	170504	170544	170544
262	5701602	5701743	5701855	5701959	5711818	5711929
263	1052519	1052664	1052664	1052926	1053068	1053213
264	3904	3904	3904	3904	3904	3904
265	18814	18814	18814	18814	18828	18828
266	7305	7305	7305	7305	7305	7305
267	31472	31472	31472	31472	31472	31472
268	0	0	0	0	0	0
269	1930	1930	1930	1930	1930	1930
270	20433	20433	20433	20433	20433	20433
271	34929	34929	34929	34929	34991	34991
272	38008	38008	38008	38008	38008	38008
273	66391	66391	66391	66391	66391	66391
274	1403	1403	1403	1403	1403	1403
275	4	4	4	4	4	4
276	2166	2166	2166	2166	2166	2166
277	6551	6551	6551	6557	6557	6561
278	24396530	24396530	24396530	24396530	24396530	24425309
279	1034303	1034303	1034303	1034303	1034303	1034303
280	251071	251071	251071	251071	251247	251247
281	12014	12014	12014	12014	12014	12014
282	552051	552051	552125	552157	552157	552162
283	11526962	11526966	11526966	11526986	11526994	11526994
284	703228	703228	703228	703228	703228	703228
285	535	535	535	535	535	535
286	11945	11945	11945	11945	11945	11945
287	343079	343135	343135	343135	343135	343135
288	264127	264127	264127	264127	264276	264276

[289 rows x 1147 columns]

1.1.2 Verification rapide de l'integrite des donnees

Verification d'absence de donnees ou de donnees negatives On regarde ici si on a une colonne de donnees (colonne 5 et suivante) dont le comptage serait negatif.

```
[5]: columns_to_study = raw_data.iloc[:,4:].columns

for i in raw_data.index :
    for d in range(len(columns_to_study[:-1])):
        if (pd.isna(raw_data.iloc[i,d+4]) or raw_data.iloc[i,d+4]<0):
            print(raw_data.iloc[i,d+4])
```

-> aucun retour donc aucune valeur manquante ou negative

Est-ce qu'il y a des donnees qui sont superieures a celle du jour suivant ? On peut aussi regarder si chaque valeur est inferieure a la valeur suivante.

```
[6]: flag = 0
table_of_errors = []
for i in raw_data.index :
    for d in range(len(columns_to_study[:-1])):
        if (int(raw_data.iloc[i,d+4]) > int(raw_data.iloc[i,d+4 +1])):
            data_problem = (raw_data.iloc[i, 1:2], columns_to_study[d],
↪columns_to_study[d+1])
            table_of_errors.append((i,d+4))
            table_of_errors.append((i,d+5))
            flag = flag +1
print("Il y a %s donnees superieures a celle de la donnee suivante" % str(flag))
```

Il y a 362 donnees superieures a celle de la donnee suivante

→ il y a donc bien des valeurs possiblement incoherentes avec des jours ou le taux cummule de cas decroit par rapport a la veille. Est-ce une rectification due a un mauvais diagnostique initial ? une modification de la methode de comptage ?

Dans le doute, on “supprime” les 2 donnees en les fixant a na grace a la sauvegarde des couple (row, col) dans table_of_errors

```
[7]: clean_data = raw_data.copy()
columns_to_study = clean_data.iloc[:,4:].columns

for coord in table_of_errors :
    clean_data.iloc[coord[0],coord[1]] = np.nan
clean_data
```

```
[7]:
```

	Province/State	Country/Region \
0	NaN	Afghanistan
1	NaN	Albania
2	NaN	Algeria
3	NaN	Andorra
4	NaN	Angola
5	NaN	Antarctica
6	NaN	Antigua and Barbuda
7	NaN	Argentina
8	NaN	Armenia
9	Australian Capital Territory	Australia
10	New South Wales	Australia
11	Northern Territory	Australia
12	Queensland	Australia
13	South Australia	Australia
14	Tasmania	Australia
15	Victoria	Australia
16	Western Australia	Australia

17		NaN	Austria
18		NaN	Azerbaijan
19		NaN	Bahamas
20		NaN	Bahrain
21		NaN	Bangladesh
22		NaN	Barbados
23		NaN	Belarus
24		NaN	Belgium
25		NaN	Belize
26		NaN	Benin
27		NaN	Bhutan
28		NaN	Bolivia
29		NaN	Bosnia and Herzegovina
..		...	...
259		NaN	Tuvalu
260		NaN	US
261		NaN	Uganda
262		NaN	Ukraine
263		NaN	United Arab Emirates
264		Anguilla	United Kingdom
265		Bermuda	United Kingdom
266		British Virgin Islands	United Kingdom
267		Cayman Islands	United Kingdom
268		Channel Islands	United Kingdom
269		Falkland Islands (Malvinas)	United Kingdom
270		Gibraltar	United Kingdom
271		Guernsey	United Kingdom
272		Isle of Man	United Kingdom
273		Jersey	United Kingdom
274		Montserrat	United Kingdom
275		Pitcairn Islands	United Kingdom
276		Saint Helena, Ascension and Tristan da Cunha	United Kingdom
277		Turks and Caicos Islands	United Kingdom
278		NaN	United Kingdom
279		NaN	Uruguay
280		NaN	Uzbekistan
281		NaN	Vanuatu
282		NaN	Venezuela
283		NaN	Vietnam
284		NaN	West Bank and Gaza
285		NaN	Winter Olympics 2022
286		NaN	Yemen
287		NaN	Zambia
288		NaN	Zimbabwe

	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	\
0	33.939110	67.709953	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	-71.949900	23.347000	0	0	0	0	0
6	17.060800	-61.796400	0	0	0	0	0
7	-38.416100	-63.616700	0	0	0	0	0
8	40.069100	45.038200	0	0	0	0	0
9	-35.473500	149.012400	0	0	0	0	0
10	-33.868800	151.209300	0	0	0	0	3
11	-12.463400	130.845600	0	0	0	0	0
12	-27.469800	153.025100	0	0	0	0	0
13	-34.928500	138.600700	0	0	0	0	0
14	-42.882100	147.327200	0	0	0	0	0
15	-37.813600	144.963100	0	0	0	0	1
16	-31.950500	115.860500	0	0	0	0	0
17	47.516200	14.550100	0	0	0	0	0
18	40.143100	47.576900	0	0	0	0	0
19	25.025885	-78.035889	0	0	0	0	0
20	26.027500	50.550000	0	0	0	0	0
21	23.685000	90.356300	0	0	0	0	0
22	13.193900	-59.543200	0	0	0	0	0
23	53.709800	27.953400	0	0	0	0	0
24	50.833300	4.469936	0	0	0	0	0
25	17.189900	-88.497600	0	0	0	0	0
26	9.307700	2.315800	0	0	0	0	0
27	27.514200	90.433600	0	0	0	0	0
28	-16.290200	-63.588700	0	0	0	0	0
29	43.915900	17.679100	0	0	0	0	0
..	...	...	...	...	...	...	...
259	-7.109500	177.649300	0	0	0	0	0
260	40.000000	-100.000000	1	1	2	2	5
261	1.373333	32.290275	0	0	0	0	0
262	48.379400	31.165600	0	0	0	0	0
263	23.424076	53.847818	0	0	0	0	0
264	18.220600	-63.068600	0	0	0	0	0
265	32.307800	-64.750500	0	0	0	0	0
266	18.420700	-64.640000	0	0	0	0	0
267	19.313300	-81.254600	0	0	0	0	0
268	49.372300	-2.364400	0	0	0	0	0
269	-51.796300	-59.523600	0	0	0	0	0
270	36.140800	-5.353600	0	0	0	0	0
271	49.448196	-2.589490	0	0	0	0	0
272	54.236100	-4.548100	0	0	0	0	0
273	49.213800	-2.135800	0	0	0	0	0
274	16.742498	-62.187366	0	0	0	0	0
275	-24.376800	-128.324200	0	0	0	0	0

276	-7.946700	-14.355900	0	0	0	0	0
277	21.694000	-71.797900	0	0	0	0	0
278	55.378100	-3.436000	0	0	0	0	0
279	-32.522800	-55.765800	0	0	0	0	0
280	41.377491	64.585262	0	0	0	0	0
281	-15.376700	166.959200	0	0	0	0	0
282	6.423800	-66.589700	0	0	0	0	0
283	14.058324	108.277199	0	2	2	2	2
284	31.952200	35.233200	0	0	0	0	0
285	39.904200	116.407400	0	0	0	0	0
286	15.552727	48.516388	0	0	0	0	0
287	-13.133897	27.849332	0	0	0	0	0
288	-19.015438	29.154857	0	0	0	0	0

	1/27/20	...	2/28/23	3/1/23	3/2/23	3/3/23	\
0	0	...	209322.0	209340.0	209358.0	209362	
1	0	...	334391.0	334408.0	334408.0	334427	
2	0	...	271441.0	271448.0	271463.0	271469	
3	0	...	47866.0	47875.0	47875.0	47875	
4	0	...	105255.0	105277.0	105277.0	105277	
5	0	...	11.0	11.0	11.0	11	
6	0	...	9106.0	9106.0	9106.0	9106	
7	0	...	10044125.0	10044125.0	10044125.0	10044125	
8	0	...	446819.0	446819.0	446819.0	446819	
9	0	...	232018.0	232018.0	232619.0	232619	
10	4	...	3900969.0	3900969.0	3908129.0	3908129	
11	0	...	104931.0	104931.0	105021.0	105021	
12	0	...	1796633.0	1796633.0	1800236.0	1800236	
13	0	...	880207.0	880207.0	881911.0	881911	
14	0	...	286264.0	286264.0	286264.0	286897	
15	1	...	2874262.0	2874262.0	2877260.0	2877260	
16	0	...	1291077.0	1291077.0	1293461.0	1293461	
17	0	...	5911294.0	5919616.0	5926148.0	5931247	
18	0	...	828548.0	828588.0	828628.0	828648	
19	0	...	37491.0	37491.0	37491.0	37491	
20	0	...	707480.0	707828.0	708061.0	708532	
21	0	...	2037773.0	2037829.0	2037829.0	2037829	
22	0	...	106645.0	106645.0	106645.0	106645	
23	0	...	994037.0	994037.0	994037.0	994037	
24	0	...	4717655.0	4717655.0	4727795.0	4727795	
25	0	...	70757.0	70757.0	70757.0	70757	
26	0	...	27990.0	27990.0	27990.0	27990	
27	0	...	62615.0	62620.0	62620.0	62620	
28	0	...	1193009.0	1193256.0	1193418.0	1193650	
29	0	...	401575.0	401636.0	401636.0	401636	
..	...	...	...	...	...	...	
259	0	...	2805.0	2805.0	2805.0	2805	

260	5	...	103443455.0	103533872.0	103589757.0	103648690
261	0	...	170504.0	170504.0	170504.0	170504
262	0	...	5693846.0	5701249.0	5701333.0	5701474
263	0	...	1051998.0	1052122.0	1052247.0	1052382
264	0	...	3904.0	3904.0	3904.0	3904
265	0	...	18799.0	18814.0	18814.0	18814
266	0	...	7305.0	7305.0	7305.0	7305
267	0	...	31472.0	31472.0	31472.0	31472
268	0	...	0.0	0.0	0.0	0
269	0	...	1930.0	1930.0	1930.0	1930
270	0	...	20423.0	20423.0	20423.0	20433
271	0	...	34867.0	34929.0	34929.0	34929
272	0	...	38008.0	38008.0	38008.0	38008
273	0	...	66391.0	66391.0	66391.0	66391
274	0	...	1403.0	1403.0	1403.0	1403
275	0	...	4.0	4.0	4.0	4
276	0	...	2166.0	2166.0	2166.0	2166
277	0	...	6551.0	6551.0	6551.0	6551
278	0	...	24370150.0	24370150.0	24396530.0	24396530
279	0	...	1034303.0	1034303.0	1034303.0	1034303
280	0	...	250932.0	251071.0	251071.0	251071
281	0	...	12014.0	12014.0	12014.0	12014
282	0	...	551981.0	551986.0	551986.0	552014
283	2	...	11526917.0	11526926.0	11526937.0	11526950
284	0	...	703228.0	703228.0	703228.0	703228
285	0	...	535.0	535.0	535.0	535
286	0	...	11945.0	11945.0	11945.0	11945
287	0	...	343012.0	343012.0	343079.0	343079
288	0	...	263921.0	264127.0	264127.0	264127

	3/4/23	3/5/23	3/6/23	3/7/23	3/8/23	3/9/23
0	209369.0	209390.0	209406.0	209436	209451	209451
1	334427.0	334427.0	334427.0	334427	334443	334457
2	271469.0	271477.0	271477.0	271490	271494	271496
3	47875.0	47875.0	47875.0	47875	47890	47890
4	105277.0	105277.0	105277.0	105277	105288	105288
5	11.0	11.0	11.0	11	11	11
6	9106.0	9106.0	9106.0	9106	9106	9106
7	10044125.0	10044125.0	10044957.0	10044957	10044957	10044957
8	446819.0	446819.0	446819.0	446819	447308	447308
9	232619.0	232619.0	232619.0	232619	232619	232974
10	3908129.0	3908129.0	3908129.0	3908129	3908129	3915992
11	105021.0	105021.0	105021.0	105021	105021	105111
12	1800236.0	1800236.0	1800236.0	1800236	1800236	1800236
13	881911.0	881911.0	881911.0	881911	881911	883620
14	286897.0	286897.0	286897.0	286897	286897	287507
15	2877260.0	2877260.0	2877260.0	2877260	2877260	2880559

16	1293461.0	1293461.0	1293461.0	1293461	1293461	1293461
17	5936666.0	5940935.0	5943417.0	5949418	5955860	5961143
18	828682.0	828721.0	828730.0	828783	828819	828825
19	37491.0	37491.0	37491.0	37491	37491	37491
20	708768.0	709230.0	709230.0	709858	710306	710693
21	2037829.0	2037829.0	2037829.0	2037829	2037871	2037871
22	106645.0	106645.0	106645.0	106645	106645	106798
23	994037.0	994037.0	994037.0	994037	994037	994037
24	4727795.0	4727795.0	4727795.0	4727795	4727795	4739365
25	70757.0	70757.0	70757.0	70757	70757	70757
26	27990.0	27990.0	27990.0	27999	27999	27999
27	62620.0	62620.0	62620.0	62620	62627	62627
28	1193815.0	1193908.0	1193970.0	1194069	1194187	1194277
29	401636.0	401636.0	401636.0	401636	401729	401729
..	...	...	...	...	...	...
259	2805.0	2805.0	2805.0	2805	2805	2805
260	NaN	NaN	103655539.0	103690910	103755771	103802702
261	170504.0	170504.0	170504.0	170504	170544	170544
262	5701602.0	5701743.0	5701855.0	5701959	5711818	5711929
263	1052519.0	1052664.0	1052664.0	1052926	1053068	1053213
264	3904.0	3904.0	3904.0	3904	3904	3904
265	18814.0	18814.0	18814.0	18814	18828	18828
266	7305.0	7305.0	7305.0	7305	7305	7305
267	31472.0	31472.0	31472.0	31472	31472	31472
268	0.0	0.0	0.0	0	0	0
269	1930.0	1930.0	1930.0	1930	1930	1930
270	20433.0	20433.0	20433.0	20433	20433	20433
271	34929.0	34929.0	34929.0	34929	34991	34991
272	38008.0	38008.0	38008.0	38008	38008	38008
273	66391.0	66391.0	66391.0	66391	66391	66391
274	1403.0	1403.0	1403.0	1403	1403	1403
275	4.0	4.0	4.0	4	4	4
276	2166.0	2166.0	2166.0	2166	2166	2166
277	6551.0	6551.0	6551.0	6557	6557	6561
278	24396530.0	24396530.0	24396530.0	24396530	24396530	24425309
279	1034303.0	1034303.0	1034303.0	1034303	1034303	1034303
280	251071.0	251071.0	251071.0	251071	251247	251247
281	12014.0	12014.0	12014.0	12014	12014	12014
282	552051.0	552051.0	552125.0	552157	552157	552162
283	11526962.0	11526966.0	11526966.0	11526986	11526994	11526994
284	703228.0	703228.0	703228.0	703228	703228	703228
285	535.0	535.0	535.0	535	535	535
286	11945.0	11945.0	11945.0	11945	11945	11945
287	343079.0	343135.0	343135.0	343135	343135	343135
288	264127.0	264127.0	264127.0	264127	264276	264276

[289 rows x 1147 columns]

Est-ce une rectification due a un mauvais diagnostique initial ? une modification de la methode de comptage ? Quoi qu'il en soit, ce nombre d'incoherence (362) est tres faible par rapport au nombre de donnees total (288 lignes de donnees * 1142 comptage = 328896 donnees totales), et ne devrait pas impacter particulierement les conclusions qu'elles soient presentes ou non dans la table.

1.2 1eres analyses realisees uniquement sur la France

On commence par ne recuperer que la ligne de donnees correspondant a la France

→ Country/Region = France ET Province/State = Nan

```
[8]: df_france = clean_data.loc[(raw_data['Country/Region'] == "France") &
    ↪ (raw_data['Province/State'].isnull())]
df_france
```

```
[8]: Province/State Country/Region Lat Long 1/22/20 1/23/20 1/24/20 \
131 NaN France 46.2276 2.2137 0 0 2

1/25/20 1/26/20 1/27/20 ... 2/28/23 3/1/23 3/2/23 \
131 3 3 3 ... 38579269.0 38583794.0 38587990.0

3/3/23 3/4/23 3/5/23 3/6/23 3/7/23 3/8/23 \
131 38591184 38591184.0 38591184.0 38599330.0 38606393 38612201

3/9/23
131 38618509

[1 rows x 1147 columns]
```

Pour le plot, on transpose les donnees en ne conservant que les lignes des incidences cummulees - a partir de la colonne 5

```
[9]: df_france_final = df_france.transpose()[5:]
```

Pour plus de clarte on change le nom de la colonne pour le nom du pays France

```
[10]: df_france_final.rename(columns={131: "France"}, inplace=True)
df_france_final.head()
```

```
[10]: France
1/23/20 0
1/24/20 2
1/25/20 3
1/26/20 3
1/27/20 3
```

On change les dates en un format interpretable par pandas

```
[11]: all_dates = pd.to_datetime(df_france_final.index)
all_dates
```

```
[11]: DatetimeIndex(['2020-01-23', '2020-01-24', '2020-01-25', '2020-01-26',
                    '2020-01-27', '2020-01-28', '2020-01-29', '2020-01-30',
                    '2020-01-31', '2020-02-01',
                    ...,
                    '2023-02-28', '2023-03-01', '2023-03-02', '2023-03-03',
                    '2023-03-04', '2023-03-05', '2023-03-06', '2023-03-07',
                    '2023-03-08', '2023-03-09'],
                    dtype='datetime64[ns]', length=1142, freq=None)
```

On reinitialise ces dates formatées comme index de la table de données

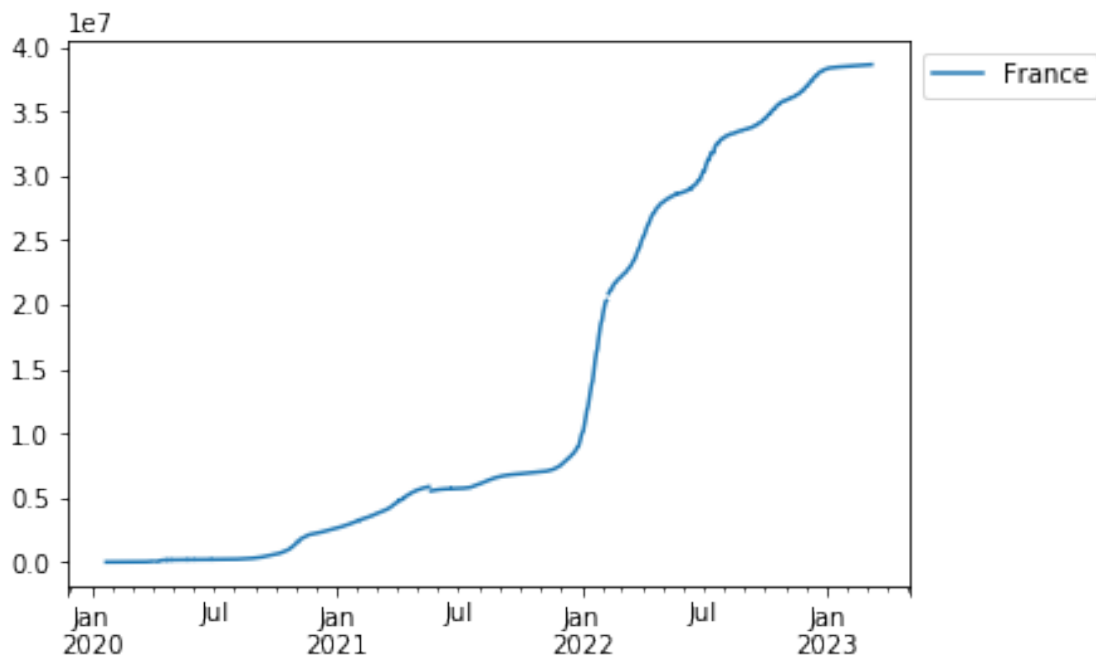
```
[12]: df_france_final.index = all_dates
df_france_final.head()
```

```
[12]:
```

	France
2020-01-23	0
2020-01-24	2
2020-01-25	3
2020-01-26	3
2020-01-27	3

On peut ploter l'incidence en France

```
[13]: fig = plt.figure()
ax = fig.add_subplot(111)
df_france_final.plot(ax=ax)
ax.legend(bbox_to_anchor=(1, 1), loc="upper left")
plt.show()
```



1.3 Generalisation aux pays d'intérêt que sont :

- la Belgique (Belgium)
- la Chine - toutes les provinces sauf Hong-Kong (China)
- Hong Kong (China, Hong-Kong)
- la France métropolitaine (France)
- l'Allemagne (Germany)
- l'Iran (Iran)
- l'Italie (Italy)
- le Japon (Japan)
- la Corée du Sud (Korea, South)
- la Hollande sans les colonies (Netherlands)
- le Portugal (Portugal)
- l'Espagne (Spain)
- le Royaume-Uni sans les colonies (United Kingdom)
- les États-Unis (US).

1.3.1 Création d'un pays "Hong-Kong"

Hong-Kong apparaît comme une province de la Chine. Pour plus de facilité à récupérer les données, nous remplaçons le pays anciennement "China" par Hong Kong pour la province Hong Kong uniquement. .

Je choisis de faire une copie du fichier initial pour pouvoir y revenir le cas échéant.

```
[14]: new_data= clean_data.copy()
new_data.loc[(new_data['Province/State'] == "Hong Kong"), 'Country/Region'] = "Hong Kong"
new_data.loc[(new_data['Province/State'] == "Hong Kong"), 'Province/State'] = np.nan
new_data.loc[(new_data['Country/Region'] == "Hong Kong")]
```

```
[14]: Province/State Country/Region Lat Long 1/22/20 1/23/20 1/24/20 \
71 NaN Hong Kong 22.3 114.2 0 2 2

1/25/20 1/26/20 1/27/20 ... 2/28/23 3/1/23 3/2/23 \
71 5 8 8 ... 2876106.0 2876106.0 2876106.0

3/3/23 3/4/23 3/5/23 3/6/23 3/7/23 3/8/23 3/9/23
71 2876106 2876106.0 2876106.0 2876106.0 2876106 2876106 2876106

[1 rows x 1147 columns]
```

1.3.2 Gestion particuliere de la Chine

La Chine apparait sous de multiples province que nous allons sommer en un unique pays.

On commence par recuperer toutes les donnees de Chine

```
[36]: df_china = new_data.loc[(new_data['Country/Region'] == "China")]
df_china.shape
```

```
[36]: (33, 1147)
```

On somme toutes les donnees pour les 33 provinces de Chine et on reinitialise les province, latitude, longitude a NA, le pays a China.

On travaille sur une Serie pandas, on la reformate en dataframe avec une tranposition.

```
[16]: df_China_combined = df_china.sum()
df_China_combined["Province/State"] = np.nan
df_China_combined["Lat"] = np.nan
df_China_combined["Long"] = np.nan
df_China_combined["Country/Region"] = "China"
df_China_combined = pd.DataFrame(df_China_combined)
df_China_combined = df_China_combined.transpose()
df_China_combined
```

```
[16]: Province/State Country/Region Lat Long 1/22/20 1/23/20 1/24/20 1/25/20 \
0 NaN China NaN NaN 548 641 918 1401

1/26/20 1/27/20 ... 2/28/23 3/1/23 3/2/23 3/3/23 \
0 2067 2869 ... 2.02742e+06 2.02742e+06 2.02742e+06 2027418
```

```

          3/4/23      3/5/23      3/6/23   3/7/23   3/8/23   3/9/23
0  2.02742e+06  2.02742e+06  2.02742e+06  2027418  2027418  2027418

```

[1 rows x 1147 columns]

On ajoute les donnees China “total” dans un nouveau dataframe pandas “newSet”

```

[17]: newSet = pd.concat([new_data,df_China_combined])
      newSet.loc[(newSet['Country/Region'] == "China")]

```

```

[17]: Province/State Country/Region      Lat      Long 1/22/20 1/23/20 1/24/20 \
59      Anhui          China  31.8257  117.2264      1      9      15
60      Beijing        China  40.1824  116.4142     14     22     36
61      Chongqing       China  30.0572  107.8740      6      9     27
62      Fujian          China  26.0789  117.9874      1      5     10
63      Gansu           China  35.7518  104.2861      0      2      2
64      Guangdong       China  23.3417  113.4244     26     32     53
65      Guangxi         China  23.8298  108.7881      2      5     23
66      Guizhou         China  26.8154  106.8748      1      3      3
67      Hainan          China  19.1959  109.7453      4      5      8
68      Hebei           China  39.5490  116.1306      1      1      2
69      Heilongjiang     China  47.8620  127.7615      0      2      4
70      Henan           China  37.8957  114.9042      5      5      9
72      Hubei           China  30.9756  112.2707    444    444    549
73      Hunan           China  27.6104  111.7088      4      9     24
74      Inner Mongolia   China  44.0935  113.9448      0      0      1
75      Jiangsu          China  32.9711  119.4550      1      5      9
76      Jiangxi          China  27.6140  115.7221      2      7     18
77      Jilin           China  43.6661  126.1923      0      1      3
78      Liaoning         China  41.2956  122.6085      2      3      4
79      Macau           China  22.1667  113.5500      1      2      2
80      Ningxia          China  37.2692  106.1655      1      1      2
81      Qinghai          China  35.7452   95.9956      0      0      0
82      Shaanxi          China  35.1917  108.8701      0      3      5
83      Shandong         China  36.3427  118.1498      2      6     15
84      Shanghai         China  31.2020  121.4491      9     16     20
85      Shanxi           China  37.5777  112.2922      1      1      1
86      Sichuan          China  30.6171  102.7103      5      8     15
87      Tianjin          China  39.3054  117.3230      4      4      8
88      Tibet           China  31.6927   88.0924      0      0      0
89      Unknown          China      NaN      NaN      0      0      0
90      Xinjiang         China  41.1129   85.2401      0      2      2
91      Yunnan           China  24.9740  101.4870      1      2      5
92      Zhejiang         China  29.1832  120.0934     10     27     43
0          NaN          China      NaN      NaN    548    641    918

```

	1/25/20	1/26/20	1/27/20	...	2/28/23	3/1/23	3/2/23	\
59	39	60	70	...	2275	2275	2275	
60	41	68	80	...	40774	40774	40774	
61	57	75	110	...	14715	14715	14715	
62	18	35	59	...	17122	17122	17122	
63	4	7	14	...	1742	1742	1742	
64	78	111	151	...	103248	103248	103248	
65	23	36	46	...	13371	13371	13371	
66	4	5	7	...	2534	2534	2534	
67	19	22	33	...	10483	10483	10483	
68	8	13	18	...	3292	3292	3292	
69	9	15	21	...	6603	6603	6603	
70	32	83	128	...	9948	9948	9948	
72	761	1058	1423	...	72131	72131	72131	
73	43	69	100	...	7437	7437	7437	
74	7	7	11	...	8847	8847	8847	
75	18	33	47	...	5075	5075	5075	
76	18	36	72	...	3423	3423	3423	
77	4	4	6	...	40764	40764	40764	
78	17	21	27	...	3547	3547	3547	
79	2	5	6	...	3514	3514	3514	
80	3	4	7	...	1276	1276	1276	
81	1	1	6	...	782	782	782	
82	15	22	35	...	7326	7326	7326	
83	27	46	75	...	5880	5880	5880	
84	33	40	53	...	67040	67040	67040	
85	6	9	13	...	7167	7167	7167	
86	28	44	69	...	14567	14567	14567	
87	10	14	23	...	4392	4392	4392	
88	0	0	0	...	1647	1647	1647	
89	0	0	0	...	1.52182e+06	1.52182e+06	1.52182e+06	
90	3	4	5	...	3089	3089	3089	
91	11	16	26	...	9743	9743	9743	
92	62	104	128	...	11848	11848	11848	
0	1401	2067	2869	...	2.02742e+06	2.02742e+06	2.02742e+06	
	3/3/23	3/4/23	3/5/23		3/6/23	3/7/23	3/8/23	3/9/23
59	2275	2275	2275		2275	2275	2275	2275
60	40774	40774	40774		40774	40774	40774	40774
61	14715	14715	14715		14715	14715	14715	14715
62	17122	17122	17122		17122	17122	17122	17122
63	1742	1742	1742		1742	1742	1742	1742
64	103248	103248	103248		103248	103248	103248	103248
65	13371	13371	13371		13371	13371	13371	13371
66	2534	2534	2534		2534	2534	2534	2534
67	10483	10483	10483		10483	10483	10483	10483
68	3292	3292	3292		3292	3292	3292	3292

69	6603	6603	6603	6603	6603	6603	6603
70	9948	9948	9948	9948	9948	9948	9948
72	72131	72131	72131	72131	72131	72131	72131
73	7437	7437	7437	7437	7437	7437	7437
74	8847	8847	8847	8847	8847	8847	8847
75	5075	5075	5075	5075	5075	5075	5075
76	3423	3423	3423	3423	3423	3423	3423
77	40764	40764	40764	40764	40764	40764	40764
78	3547	3547	3547	3547	3547	3547	3547
79	3514	3514	3514	3514	3514	3514	3514
80	1276	1276	1276	1276	1276	1276	1276
81	782	782	782	782	782	782	782
82	7326	7326	7326	7326	7326	7326	7326
83	5880	5880	5880	5880	5880	5880	5880
84	67040	67040	67040	67040	67040	67040	67040
85	7167	7167	7167	7167	7167	7167	7167
86	14567	14567	14567	14567	14567	14567	14567
87	4392	4392	4392	4392	4392	4392	4392
88	1647	1647	1647	1647	1647	1647	1647
89	1521816	1.52182e+06	1.52182e+06	1.52182e+06	1521816	1521816	1521816
90	3089	3089	3089	3089	3089	3089	3089
91	9743	9743	9743	9743	9743	9743	9743
92	11848	11848	11848	11848	11848	11848	11848
0	2027418	2.02742e+06	2.02742e+06	2.02742e+06	2027418	2027418	2027418

[34 rows x 1147 columns]

1.3.3 Recuperation des donnees pour les pays d'interet listes ci dessus

On cree une liste avec les pays d'interet "interest_countries".

On recupere par la suite un sous jeu de donnees avec uniquement ces pays et "NA" en province.

```
[18]: interest_countries = ["Belgium", "China", "France", "Germany", "Hong Kong",
    ↪ "Iran", "Italy", "Japan", "Korea, South", "Netherlands", "Portugal",
    ↪ "Spain", "United Kingdom", "US"]
df_allCountries = newSet.loc[(newSet['Country/Region'].
    ↪ isin(interest_countries)) & (newSet['Province/State'].isnull()) ,]
df_allCountries
```

```
[18]: Province/State Country/Region Lat Long 1/22/20 1/23/20 \
24 NaN Belgium 50.833300 4.469936 0 0
71 NaN Hong Kong 22.300000 114.200000 0 2
131 NaN France 46.227600 2.213700 0 0
135 NaN Germany 51.165691 10.451526 0 0
150 NaN Iran 32.427908 53.688046 0 0
154 NaN Italy 41.871940 12.567380 0 0
```

156	NaN	Japan	36.204824	138.252924	2	2
162	NaN	Korea, South	35.907757	127.766922	1	1
200	NaN	Netherlands	52.132600	5.291300	0	0
218	NaN	Portugal	39.399900	-8.224500	0	0
241	NaN	Spain	40.463667	-3.749220	0	0
260	NaN	US	40.000000	-100.000000	1	1
278	NaN	United Kingdom	55.378100	-3.436000	0	0
0	NaN	China	NaN	NaN	548	641

	1/24/20	1/25/20	1/26/20	1/27/20	...	2/28/23	3/1/23	\
24	0	0	0	0	...	4.71766e+06	4.71766e+06	
71	2	5	8	8	...	2.87611e+06	2.87611e+06	
131	2	3	3	3	...	3.85793e+07	3.85838e+07	
135	0	0	0	1	...	3.81689e+07	3.819e+07	
150	0	0	0	0	...	7.56791e+06	7.5689e+06	
154	0	0	0	0	...	2.55769e+07	2.55769e+07	
156	2	2	4	4	...	3.32272e+07	3.32412e+07	
162	2	2	3	4	...	3.0526e+07	3.05336e+07	
200	0	0	0	0	...	8.59616e+06	8.59616e+06	
218	0	0	0	0	...	5.56671e+06	5.56808e+06	
241	0	0	0	0	...	1.37633e+07	1.37633e+07	
260	2	2	5	5	...	1.03443e+08	1.03534e+08	
278	0	0	0	0	...	2.43702e+07	2.43702e+07	
0	918	1401	2067	2869	...	2.02742e+06	2.02742e+06	

	3/2/23	3/3/23	3/4/23	3/5/23	3/6/23	3/7/23	\
24	4.7278e+06	4727795	4.7278e+06	4.7278e+06	4.7278e+06	4727795	
71	2.87611e+06	2876106	2.87611e+06	2.87611e+06	2.87611e+06	2876106	
131	3.8588e+07	38591184	3.85912e+07	3.85912e+07	3.85993e+07	38606393	
135	3.82026e+07	38210850	3.82108e+07	3.82109e+07	3.82109e+07	38231610	
150	7.56926e+06	7569483	7.56977e+06	7.57023e+06	7.57074e+06	7571352	
154	2.55769e+07	25603510	2.56035e+07	2.56035e+07	2.56035e+07	25603510	
156	3.32527e+07	33263208	3.32736e+07	3.32824e+07	3.32866e+07	33298799	
162	3.0544e+07	30555102	3.05551e+07	3.05692e+07	3.05815e+07	30594297	
200	8.59616e+06	8598043	8.59804e+06	8.59804e+06	8.59804e+06	8599981	
218	5.56808e+06	5568084	5.56808e+06	5.56808e+06	5.56808e+06	5568084	
241	1.37633e+07	13770429	1.37704e+07	1.37704e+07	1.37704e+07	13770429	
260	1.0359e+08	103648690	NaN	NaN	1.03656e+08	103690910	
278	2.43965e+07	24396530	2.43965e+07	2.43965e+07	2.43965e+07	24396530	
0	2.02742e+06	2027418	2.02742e+06	2.02742e+06	2.02742e+06	2027418	

	3/8/23	3/9/23
24	4727795	4739365
71	2876106	2876106
131	38612201	38618509
135	38241231	38249060
150	7571996	7572311

```

154    25603510    25603510
156    33310604    33320438
162    30605187    30615522
200     8599981     8599981
218     5570473     5570473
241    13770429    13770429
260   103755771   103802702
278    24396530    24425309
0       2027418     2027418

```

```
[14 rows x 1147 columns]
```

1.3.4 Analyse de l'évolution du nombre de cas cumulés au cours du temps

On transforma la table pour être plus compréhensible par matplotlib pour faire le graphique. Globalement on réalise une transposition en supprimant les data latitude/longitude pour le moment et en renommant les colonnes avec le nom du pays correspondant.

```
[19]: df_allCountries_final = df_allCountries.transpose()[5:]
df_allCountries_final.columns = df_allCountries["Country/Region"]
df_allCountries_final.head()
```

```
[19]: Country/Region Belgium Hong Kong France Germany Iran Italy Japan Korea, South \
1/23/20          0          2          0          0          0          0          2          1
1/24/20          0          2          2          0          0          0          2          2
1/25/20          0          5          3          0          0          0          2          2
1/26/20          0          8          3          0          0          0          4          3
1/27/20          0          8          3          1          0          0          4          4
```

```
Country/Region Netherlands Portugal Spain US United Kingdom China
1/23/20          0          0          0  1          0  641
1/24/20          0          0          0  2          0  918
1/25/20          0          0          0  2          0 1401
1/26/20          0          0          0  5          0 2067
1/27/20          0          0          0  5          0 2869
```

On reformate les dates

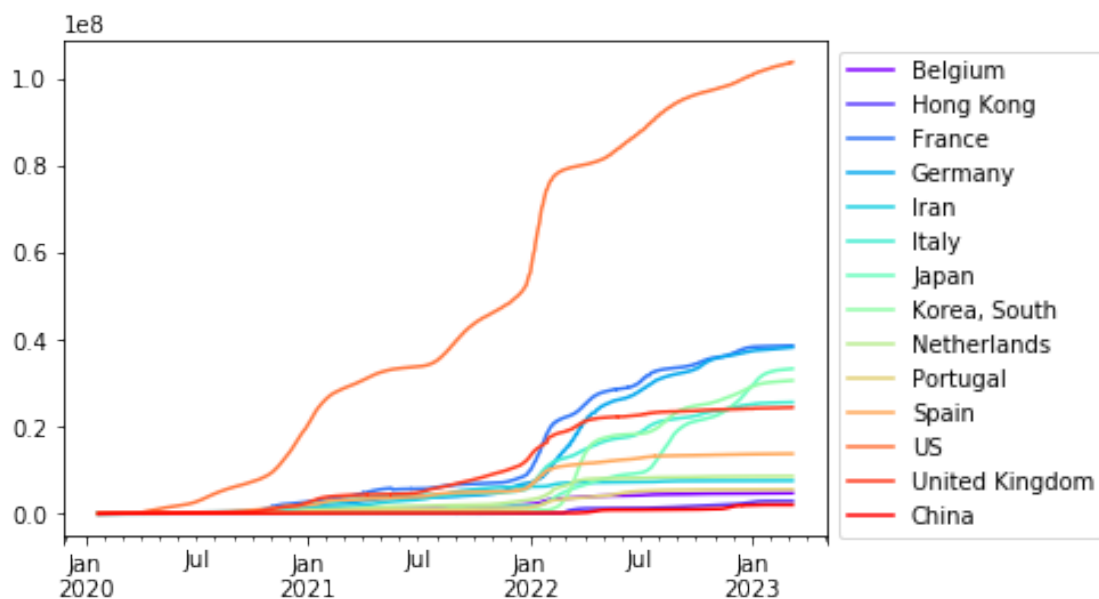
```
[20]: all_dates = pd.to_datetime(df_allCountries_final.index)
df_allCountries_final.index = all_dates
df_allCountries_final.head()
```

```
[20]: Country/Region Belgium Hong Kong France Germany Iran Italy Japan Korea, South \
2020-01-23          0          2          0          0          0          0          2          1
2020-01-24          0          2          2          0          0          0          2          2
2020-01-25          0          5          3          0          0          0          2          2
2020-01-26          0          8          3          0          0          0          4          3
```

2020-01-27	0	8	3	1	0	0	4	4
Country/Region	Netherlands	Portugal	Spain	US	United Kingdom	China		
2020-01-23	0	0	0	1	0	641		
2020-01-24	0	0	0	2	0	918		
2020-01-25	0	0	0	2	0	1401		
2020-01-26	0	0	0	5	0	2067		
2020-01-27	0	0	0	5	0	2869		

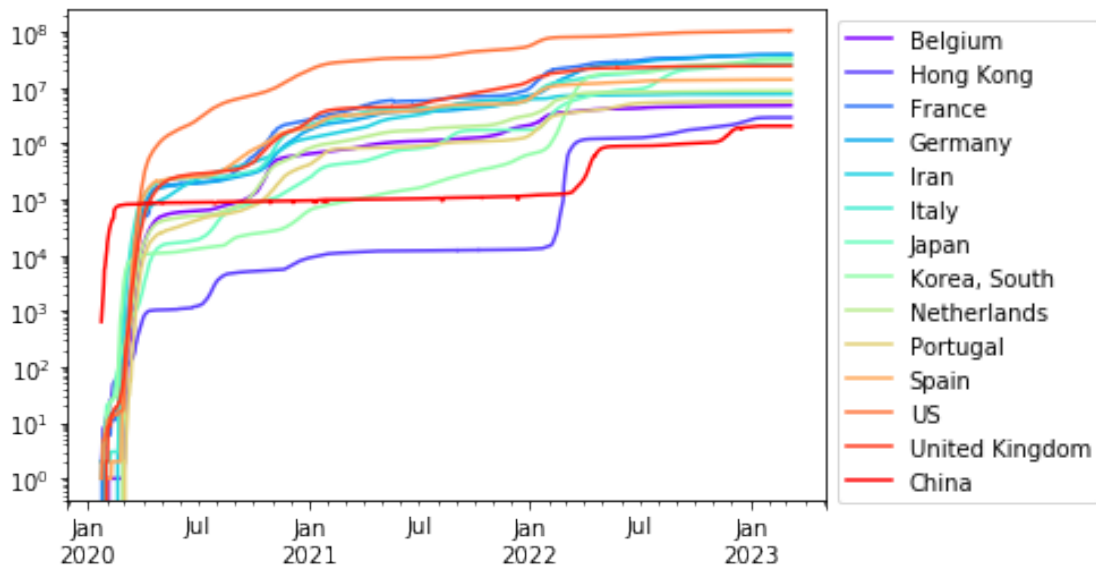
On plot le graph en format classique avec le nombre de cas en fonction des jours, en utilisant un code couleur pour les pays consideres.

```
[21]: color = cm.rainbow(np.linspace(0, 1, len(interest_countries)))
fig = plt.figure()
ax = fig.add_subplot(111)
df_allCountries_final.plot(ax=ax, color=color)
ax.legend(bbox_to_anchor=(1, 1), loc="upper left")
plt.show()
```



On modifie l'échelle des y en log pour mieux voir les distribution.

```
[22]: fig = plt.figure()
ax = fig.add_subplot(111)
plt.yscale("log")
df_allCountries_final.plot(ax=ax, color=color)
ax.legend(bbox_to_anchor=(1, 1), loc="upper left")
plt.show()
```



C'est donc les USA ayant eu le plus de cas recences, mais a normaliser par le nombre d'habitant global de chaque territoire et/ou du nombre de deces.

1.4 Question subsidiaire - utilisqtion des donnees de deces

On recupere les donnees de deces en faisant une copie local au besoin.

```
[23]: death_url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/
↳ csse_covid_19_data/csse_covid_19_time_series/
↳ time_series_covid19_deaths_global.csv"
death_file = "time_series_covid19_deaths_global.csv"

if not os.path.isfile(death_file):
    urlretrieve(death_url, death_file)
```

Comme precedemment, on se doit de regarder les donnees apres chargement.

```
[24]: death_data = pd.read_csv(death_file)
death_data.shape
```

```
[24]: (289, 1147)
```

```
[25]: death_data
```

```
[25]:
```

	Province/State	Country/Region \
0	NaN	Afghanistan
1	NaN	Albania

2		NaN	Algeria
3		NaN	Andorra
4		NaN	Angola
5		NaN	Antarctica
6		NaN	Antigua and Barbuda
7		NaN	Argentina
8		NaN	Armenia
9	Australian Capital Territory		Australia
10	New South Wales		Australia
11	Northern Territory		Australia
12	Queensland		Australia
13	South Australia		Australia
14	Tasmania		Australia
15	Victoria		Australia
16	Western Australia		Australia
17		NaN	Austria
18		NaN	Azerbaijan
19		NaN	Bahamas
20		NaN	Bahrain
21		NaN	Bangladesh
22		NaN	Barbados
23		NaN	Belarus
24		NaN	Belgium
25		NaN	Belize
26		NaN	Benin
27		NaN	Bhutan
28		NaN	Bolivia
29		NaN	Bosnia and Herzegovina
..		...	...
259		NaN	Tuvalu
260		NaN	US
261		NaN	Uganda
262		NaN	Ukraine
263		NaN	United Arab Emirates
264	Anguilla		United Kingdom
265	Bermuda		United Kingdom
266	British Virgin Islands		United Kingdom
267	Cayman Islands		United Kingdom
268	Channel Islands		United Kingdom
269	Falkland Islands (Malvinas)		United Kingdom
270	Gibraltar		United Kingdom
271	Guernsey		United Kingdom
272	Isle of Man		United Kingdom
273	Jersey		United Kingdom
274	Montserrat		United Kingdom
275	Pitcairn Islands		United Kingdom
276	Saint Helena, Ascension and Tristan da Cunha		United Kingdom

277	Turks and Caicos Islands	United Kingdom
278	NaN	United Kingdom
279	NaN	Uruguay
280	NaN	Uzbekistan
281	NaN	Vanuatu
282	NaN	Venezuela
283	NaN	Vietnam
284	NaN	West Bank and Gaza
285	NaN	Winter Olympics 2022
286	NaN	Yemen
287	NaN	Zambia
288	NaN	Zimbabwe

	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	\
0	33.939110	67.709953	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	-71.949900	23.347000	0	0	0	0	0	
6	17.060800	-61.796400	0	0	0	0	0	
7	-38.416100	-63.616700	0	0	0	0	0	
8	40.069100	45.038200	0	0	0	0	0	
9	-35.473500	149.012400	0	0	0	0	0	
10	-33.868800	151.209300	0	0	0	0	0	
11	-12.463400	130.845600	0	0	0	0	0	
12	-27.469800	153.025100	0	0	0	0	0	
13	-34.928500	138.600700	0	0	0	0	0	
14	-42.882100	147.327200	0	0	0	0	0	
15	-37.813600	144.963100	0	0	0	0	0	
16	-31.950500	115.860500	0	0	0	0	0	
17	47.516200	14.550100	0	0	0	0	0	
18	40.143100	47.576900	0	0	0	0	0	
19	25.025885	-78.035889	0	0	0	0	0	
20	26.027500	50.550000	0	0	0	0	0	
21	23.685000	90.356300	0	0	0	0	0	
22	13.193900	-59.543200	0	0	0	0	0	
23	53.709800	27.953400	0	0	0	0	0	
24	50.833300	4.469936	0	0	0	0	0	
25	17.189900	-88.497600	0	0	0	0	0	
26	9.307700	2.315800	0	0	0	0	0	
27	27.514200	90.433600	0	0	0	0	0	
28	-16.290200	-63.588700	0	0	0	0	0	
29	43.915900	17.679100	0	0	0	0	0	
..	...	...	...	...	...	...	...	
259	-7.109500	177.649300	0	0	0	0	0	
260	40.000000	-100.000000	0	0	0	0	0	

261	1.373333	32.290275	0	0	0	0	0
262	48.379400	31.165600	0	0	0	0	0
263	23.424076	53.847818	0	0	0	0	0
264	18.220600	-63.068600	0	0	0	0	0
265	32.307800	-64.750500	0	0	0	0	0
266	18.420700	-64.640000	0	0	0	0	0
267	19.313300	-81.254600	0	0	0	0	0
268	49.372300	-2.364400	0	0	0	0	0
269	-51.796300	-59.523600	0	0	0	0	0
270	36.140800	-5.353600	0	0	0	0	0
271	49.448196	-2.589490	0	0	0	0	0
272	54.236100	-4.548100	0	0	0	0	0
273	49.213800	-2.135800	0	0	0	0	0
274	16.742498	-62.187366	0	0	0	0	0
275	-24.376800	-128.324200	0	0	0	0	0
276	-7.946700	-14.355900	0	0	0	0	0
277	21.694000	-71.797900	0	0	0	0	0
278	55.378100	-3.436000	0	0	0	0	0
279	-32.522800	-55.765800	0	0	0	0	0
280	41.377491	64.585262	0	0	0	0	0
281	-15.376700	166.959200	0	0	0	0	0
282	6.423800	-66.589700	0	0	0	0	0
283	14.058324	108.277199	0	0	0	0	0
284	31.952200	35.233200	0	0	0	0	0
285	39.904200	116.407400	0	0	0	0	0
286	15.552727	48.516388	0	0	0	0	0
287	-13.133897	27.849332	0	0	0	0	0
288	-19.015438	29.154857	0	0	0	0	0

	1/27/20	...	2/28/23	3/1/23	3/2/23	3/3/23	3/4/23	3/5/23	\
0	0	...	7896	7896	7896	7896	7896	7896	
1	0	...	3598	3598	3598	3598	3598	3598	
2	0	...	6881	6881	6881	6881	6881	6881	
3	0	...	165	165	165	165	165	165	
4	0	...	1933	1933	1933	1933	1933	1933	
5	0	...	0	0	0	0	0	0	
6	0	...	146	146	146	146	146	146	
7	0	...	130463	130463	130463	130463	130463	130463	
8	0	...	8721	8721	8721	8721	8721	8721	
9	0	...	224	224	228	228	228	228	
10	0	...	6464	6464	6493	6493	6493	6493	
11	0	...	90	90	90	90	90	90	
12	0	...	2760	2760	2783	2783	2783	2783	
13	0	...	1322	1322	1322	1322	1322	1322	
14	0	...	252	252	252	253	253	253	
15	0	...	7317	7317	7338	7338	7338	7338	
16	0	...	944	944	952	952	952	952	

17	0	...	21887	21891	21899	21907	21921	21922
18	0	...	10119	10119	10122	10126	10127	10129
19	0	...	833	833	833	833	833	833
20	0	...	1548	1549	1550	1552	1552	1552
21	0	...	29445	29445	29445	29445	29445	29445
22	0	...	575	575	575	575	575	575
23	0	...	7118	7118	7118	7118	7118	7118
24	0	...	33717	33717	33775	33775	33775	33775
25	0	...	688	688	688	688	688	688
26	0	...	163	163	163	163	163	163
27	0	...	21	21	21	21	21	21
28	0	...	22365	22365	22365	22365	22365	22365
29	0	...	16278	16279	16279	16279	16279	16279
..	...	...	...	...	...	...	...	...
259	0	...	0	0	0	0	0	0
260	0	...	1119917	1120897	1121658	1122165	1122172	1122134
261	0	...	3630	3630	3630	3630	3630	3630
262	0	...	119149	119209	119210	119211	119212	119213
263	0	...	2349	2349	2349	2349	2349	2349
264	0	...	12	12	12	12	12	12
265	0	...	160	160	160	160	160	160
266	0	...	64	64	64	64	64	64
267	0	...	37	37	37	37	37	37
268	0	...	0	0	0	0	0	0
269	0	...	0	0	0	0	0	0
270	0	...	111	111	111	111	111	111
271	0	...	66	66	66	66	66	66
272	0	...	116	116	116	116	116	116
273	0	...	161	161	161	161	161	161
274	0	...	8	8	8	8	8	8
275	0	...	0	0	0	0	0	0
276	0	...	0	0	0	0	0	0
277	0	...	38	38	38	38	38	38
278	0	...	219948	219948	219948	219948	219948	219948
279	0	...	7617	7617	7617	7617	7617	7617
280	0	...	1637	1637	1637	1637	1637	1637
281	0	...	14	14	14	14	14	14
282	0	...	5853	5854	5854	5854	5854	5854
283	0	...	43186	43186	43186	43186	43186	43186
284	0	...	5708	5708	5708	5708	5708	5708
285	0	...	0	0	0	0	0	0
286	0	...	2159	2159	2159	2159	2159	2159
287	0	...	4057	4057	4057	4057	4057	4057
288	0	...	5663	5668	5668	5668	5668	5668

	3/6/23	3/7/23	3/8/23	3/9/23
0	7896	7896	7896	7896

1	3598	3598	3598	3598
2	6881	6881	6881	6881
3	165	165	165	165
4	1933	1933	1933	1933
5	0	0	0	0
6	146	146	146	146
7	130472	130472	130472	130472
8	8721	8721	8727	8727
9	228	228	228	228
10	6493	6493	6493	6529
11	90	90	90	91
12	2783	2783	2783	2783
13	1322	1322	1322	1365
14	253	253	253	256
15	7338	7338	7338	7370
16	952	952	952	952
17	21923	21941	21949	21970
18	10129	10135	10138	10138
19	833	833	833	833
20	1552	1553	1553	1553
21	29445	29445	29445	29445
22	575	575	575	579
23	7118	7118	7118	7118
24	33775	33775	33775	33814
25	688	688	688	688
26	163	163	163	163
27	21	21	21	21
28	22365	22365	22365	22365
29	16279	16279	16280	16280
..	...	...	...	...
259	0	0	0	0
260	1122181	1122516	1123246	1123836
261	3630	3630	3630	3630
262	119216	119217	119281	119283
263	2349	2349	2349	2349
264	12	12	12	12
265	160	160	160	160
266	64	64	64	64
267	37	37	37	37
268	0	0	0	0
269	0	0	0	0
270	111	111	111	111
271	66	66	66	66
272	116	116	116	116
273	161	161	161	161
274	8	8	8	8
275	0	0	0	0

276	0	0	0	0
277	38	38	38	38
278	219948	219948	219948	219948
279	7617	7617	7617	7617
280	1637	1637	1637	1637
281	14	14	14	14
282	5854	5854	5854	5854
283	43186	43186	43186	43186
284	5708	5708	5708	5708
285	0	0	0	0
286	2159	2159	2159	2159
287	4057	4057	4057	4057
288	5668	5668	5671	5671

[289 rows x 1147 columns]

Le format est donc identique aux données précédentes et peuvent être traitées de la même manière après les mêmes vérifications * présence de données manquantes

```
[26]: columns_to_study = death_data.iloc[:,4:].columns

for i in raw_data.index :
    for d in range(len(columns_to_study[:-1])):
        if (pd.isna(death_data.iloc[i,d+4]) or death_data.iloc[i,d+4]<0):
            print(death_data.iloc[i,d+4])
```

→ pas de données manquantes * incohérence avec un nombre de décès cumulé qui décroît ?

```
[27]: flag = 0
table_of_errors = []
for i in death_data.index :
    for d in range(len(columns_to_study[:-1])):
        if (int(death_data.iloc[i,d+4]) > int(death_data.iloc[i,d+4 +1])):
            data_problem = (death_data.iloc[i, 1:2], columns_to_study[d],
↪columns_to_study[d+1])
            table_of_errors.append((i,d+4))
            table_of_errors.append((i,d+5))
            flag = flag +1
print("Il y a %s données supérieures à celle de la donnée suivante" % str(flag))
```

Il y a 291 données supérieures à celle de la donnée suivante

→ on a le même type d'incohérence que précédemment, minoritaire comparée à la quantité de données, mais on choisit toutefois de supprimer en remettant ces valeurs aberrantes à na

```
[28]: clean_death_data = death_data.copy()
columns_to_study = clean_death_data.iloc[:,4:].columns
```

```

for coord in table_of_errors :
    clean_death_data.iloc[coord[0],coord[1]] = np.nan
clean_death_data

```

```

[28]:

```

	Province/State	Country/Region \
0	NaN	Afghanistan
1	NaN	Albania
2	NaN	Algeria
3	NaN	Andorra
4	NaN	Angola
5	NaN	Antarctica
6	NaN	Antigua and Barbuda
7	NaN	Argentina
8	NaN	Armenia
9	Australian Capital Territory	Australia
10	New South Wales	Australia
11	Northern Territory	Australia
12	Queensland	Australia
13	South Australia	Australia
14	Tasmania	Australia
15	Victoria	Australia
16	Western Australia	Australia
17	NaN	Austria
18	NaN	Azerbaijan
19	NaN	Bahamas
20	NaN	Bahrain
21	NaN	Bangladesh
22	NaN	Barbados
23	NaN	Belarus
24	NaN	Belgium
25	NaN	Belize
26	NaN	Benin
27	NaN	Bhutan
28	NaN	Bolivia
29	NaN	Bosnia and Herzegovina
..	...	...
259	NaN	Tuvalu
260	NaN	US
261	NaN	Uganda
262	NaN	Ukraine
263	NaN	United Arab Emirates
264	Anguilla	United Kingdom
265	Bermuda	United Kingdom
266	British Virgin Islands	United Kingdom
267	Cayman Islands	United Kingdom
268	Channel Islands	United Kingdom
269	Falkland Islands (Malvinas)	United Kingdom

270	Gibraltar	United Kingdom
271	Guernsey	United Kingdom
272	Isle of Man	United Kingdom
273	Jersey	United Kingdom
274	Montserrat	United Kingdom
275	Pitcairn Islands	United Kingdom
276	Saint Helena, Ascension and Tristan da Cunha	United Kingdom
277	Turks and Caicos Islands	United Kingdom
278	NaN	United Kingdom
279	NaN	Uruguay
280	NaN	Uzbekistan
281	NaN	Vanuatu
282	NaN	Venezuela
283	NaN	Vietnam
284	NaN	West Bank and Gaza
285	NaN	Winter Olympics 2022
286	NaN	Yemen
287	NaN	Zambia
288	NaN	Zimbabwe

	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	\
0	33.939110	67.709953	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	-71.949900	23.347000	0	0	0	0	0	
6	17.060800	-61.796400	0	0	0	0	0	
7	-38.416100	-63.616700	0	0	0	0	0	
8	40.069100	45.038200	0	0	0	0	0	
9	-35.473500	149.012400	0	0	0	0	0	
10	-33.868800	151.209300	0	0	0	0	0	
11	-12.463400	130.845600	0	0	0	0	0	
12	-27.469800	153.025100	0	0	0	0	0	
13	-34.928500	138.600700	0	0	0	0	0	
14	-42.882100	147.327200	0	0	0	0	0	
15	-37.813600	144.963100	0	0	0	0	0	
16	-31.950500	115.860500	0	0	0	0	0	
17	47.516200	14.550100	0	0	0	0	0	
18	40.143100	47.576900	0	0	0	0	0	
19	25.025885	-78.035889	0	0	0	0	0	
20	26.027500	50.550000	0	0	0	0	0	
21	23.685000	90.356300	0	0	0	0	0	
22	13.193900	-59.543200	0	0	0	0	0	
23	53.709800	27.953400	0	0	0	0	0	
24	50.833300	4.469936	0	0	0	0	0	
25	17.189900	-88.497600	0	0	0	0	0	

26	9.307700	2.315800	0	0	0	0	0
27	27.514200	90.433600	0	0	0	0	0
28	-16.290200	-63.588700	0	0	0	0	0
29	43.915900	17.679100	0	0	0	0	0
..	...	...	...	...	...	...	...
259	-7.109500	177.649300	0	0	0	0	0
260	40.000000	-100.000000	0	0	0	0	0
261	1.373333	32.290275	0	0	0	0	0
262	48.379400	31.165600	0	0	0	0	0
263	23.424076	53.847818	0	0	0	0	0
264	18.220600	-63.068600	0	0	0	0	0
265	32.307800	-64.750500	0	0	0	0	0
266	18.420700	-64.640000	0	0	0	0	0
267	19.313300	-81.254600	0	0	0	0	0
268	49.372300	-2.364400	0	0	0	0	0
269	-51.796300	-59.523600	0	0	0	0	0
270	36.140800	-5.353600	0	0	0	0	0
271	49.448196	-2.589490	0	0	0	0	0
272	54.236100	-4.548100	0	0	0	0	0
273	49.213800	-2.135800	0	0	0	0	0
274	16.742498	-62.187366	0	0	0	0	0
275	-24.376800	-128.324200	0	0	0	0	0
276	-7.946700	-14.355900	0	0	0	0	0
277	21.694000	-71.797900	0	0	0	0	0
278	55.378100	-3.436000	0	0	0	0	0
279	-32.522800	-55.765800	0	0	0	0	0
280	41.377491	64.585262	0	0	0	0	0
281	-15.376700	166.959200	0	0	0	0	0
282	6.423800	-66.589700	0	0	0	0	0
283	14.058324	108.277199	0	0	0	0	0
284	31.952200	35.233200	0	0	0	0	0
285	39.904200	116.407400	0	0	0	0	0
286	15.552727	48.516388	0	0	0	0	0
287	-13.133897	27.849332	0	0	0	0	0
288	-19.015438	29.154857	0	0	0	0	0

	1/27/20	...	2/28/23	3/1/23	3/2/23	3/3/23	3/4/23	\
0	0	...	7896.0	7896	7896	7896	7896.0	
1	0	...	3598.0	3598	3598	3598	3598.0	
2	0	...	6881.0	6881	6881	6881	6881.0	
3	0	...	165.0	165	165	165	165.0	
4	0	...	1933.0	1933	1933	1933	1933.0	
5	0	...	0.0	0	0	0	0.0	
6	0	...	146.0	146	146	146	146.0	
7	0	...	130463.0	130463	130463	130463	130463.0	
8	0	...	8721.0	8721	8721	8721	8721.0	
9	0	...	224.0	224	228	228	228.0	

10	0	...	6464.0	6464	6493	6493	6493.0
11	0	...	90.0	90	90	90	90.0
12	0	...	2760.0	2760	2783	2783	2783.0
13	0	...	1322.0	1322	1322	1322	1322.0
14	0	...	252.0	252	252	253	253.0
15	0	...	7317.0	7317	7338	7338	7338.0
16	0	...	944.0	944	952	952	952.0
17	0	...	21887.0	21891	21899	21907	21921.0
18	0	...	10119.0	10119	10122	10126	10127.0
19	0	...	833.0	833	833	833	833.0
20	0	...	1548.0	1549	1550	1552	1552.0
21	0	...	29445.0	29445	29445	29445	29445.0
22	0	...	575.0	575	575	575	575.0
23	0	...	7118.0	7118	7118	7118	7118.0
24	0	...	33717.0	33717	33775	33775	33775.0
25	0	...	688.0	688	688	688	688.0
26	0	...	163.0	163	163	163	163.0
27	0	...	21.0	21	21	21	21.0
28	0	...	22365.0	22365	22365	22365	22365.0
29	0	...	16278.0	16279	16279	16279	16279.0
..	...	...	...	...	...	...	...
259	0	...	0.0	0	0	0	0.0
260	0	...	1119917.0	1120897	1121658	1122165	NaN
261	0	...	3630.0	3630	3630	3630	3630.0
262	0	...	119149.0	119209	119210	119211	119212.0
263	0	...	2349.0	2349	2349	2349	2349.0
264	0	...	12.0	12	12	12	12.0
265	0	...	160.0	160	160	160	160.0
266	0	...	64.0	64	64	64	64.0
267	0	...	37.0	37	37	37	37.0
268	0	...	0.0	0	0	0	0.0
269	0	...	0.0	0	0	0	0.0
270	0	...	111.0	111	111	111	111.0
271	0	...	66.0	66	66	66	66.0
272	0	...	116.0	116	116	116	116.0
273	0	...	161.0	161	161	161	161.0
274	0	...	8.0	8	8	8	8.0
275	0	...	0.0	0	0	0	0.0
276	0	...	0.0	0	0	0	0.0
277	0	...	38.0	38	38	38	38.0
278	0	...	219948.0	219948	219948	219948	219948.0
279	0	...	7617.0	7617	7617	7617	7617.0
280	0	...	1637.0	1637	1637	1637	1637.0
281	0	...	14.0	14	14	14	14.0
282	0	...	5853.0	5854	5854	5854	5854.0
283	0	...	43186.0	43186	43186	43186	43186.0
284	0	...	5708.0	5708	5708	5708	5708.0

285	0	...	0.0	0	0	0	0.0
286	0	...	2159.0	2159	2159	2159	2159.0
287	0	...	4057.0	4057	4057	4057	4057.0
288	0	...	5663.0	5668	5668	5668	5668.0

	3/5/23	3/6/23	3/7/23	3/8/23	3/9/23
0	7896.0	7896	7896	7896	7896
1	3598.0	3598	3598	3598	3598
2	6881.0	6881	6881	6881	6881
3	165.0	165	165	165	165
4	1933.0	1933	1933	1933	1933
5	0.0	0	0	0	0
6	146.0	146	146	146	146
7	130463.0	130472	130472	130472	130472
8	8721.0	8721	8721	8727	8727
9	228.0	228	228	228	228
10	6493.0	6493	6493	6493	6529
11	90.0	90	90	90	91
12	2783.0	2783	2783	2783	2783
13	1322.0	1322	1322	1322	1365
14	253.0	253	253	253	256
15	7338.0	7338	7338	7338	7370
16	952.0	952	952	952	952
17	21922.0	21923	21941	21949	21970
18	10129.0	10129	10135	10138	10138
19	833.0	833	833	833	833
20	1552.0	1552	1553	1553	1553
21	29445.0	29445	29445	29445	29445
22	575.0	575	575	575	579
23	7118.0	7118	7118	7118	7118
24	33775.0	33775	33775	33775	33814
25	688.0	688	688	688	688
26	163.0	163	163	163	163
27	21.0	21	21	21	21
28	22365.0	22365	22365	22365	22365
29	16279.0	16279	16279	16280	16280
..	...	...	...	...	...
259	0.0	0	0	0	0
260	NaN	1122181	1122516	1123246	1123836
261	3630.0	3630	3630	3630	3630
262	119213.0	119216	119217	119281	119283
263	2349.0	2349	2349	2349	2349
264	12.0	12	12	12	12
265	160.0	160	160	160	160
266	64.0	64	64	64	64
267	37.0	37	37	37	37
268	0.0	0	0	0	0

269	0.0	0	0	0	0
270	111.0	111	111	111	111
271	66.0	66	66	66	66
272	116.0	116	116	116	116
273	161.0	161	161	161	161
274	8.0	8	8	8	8
275	0.0	0	0	0	0
276	0.0	0	0	0	0
277	38.0	38	38	38	38
278	219948.0	219948	219948	219948	219948
279	7617.0	7617	7617	7617	7617
280	1637.0	1637	1637	1637	1637
281	14.0	14	14	14	14
282	5854.0	5854	5854	5854	5854
283	43186.0	43186	43186	43186	43186
284	5708.0	5708	5708	5708	5708
285	0.0	0	0	0	0
286	2159.0	2159	2159	2159	2159
287	4057.0	4057	4057	4057	4057
288	5668.0	5668	5668	5671	5671

[289 rows x 1147 columns]

On fait les memes manipulations de gestion de pays pour Hong Kong et la Chine aue precedemment

```
[29]: new_data_death= clean_death_data.copy()
# Hong Kong
new_data_death.loc[(new_data_death['Province/State'] == "Hong Kong"),'Country/
↳Region'] = "Hong Kong"
new_data_death.loc[(new_data_death['Province/State'] == "Hong Kong"),'Province/
↳State'] = np.nan
new_data_death.loc[(new_data_death['Country/Region'] == "Hong Kong")]
# China
df_china_death = new_data_death.loc[(new_data['Country/Region'] == "China")]
df_China_death_combined = df_china_death.sum()
df_China_death_combined["Province/State"] = np.nan
df_China_death_combined["Lat"] = np.nan
df_China_death_combined["Long"] = np.nan
df_China_death_combined["Country/Region"] = "China"
df_China_death_combined = pd.DataFrame(df_China_death_combined)
df_China_death_combined = df_China_death_combined.transpose()
# compilation of data
newSet_death = pd.concat([new_data_death,df_China_death_combined])
newSet_death.loc[(newSet['Country/Region'] == "China")]
```

```
[29]: Province/State Country/Region Lat Long 1/22/20 1/23/20 1/24/20 \
59 Anhui China 31.8257 117.2264 0 0 0
```

60	Beijing	China	40.1824	116.4142	0	0	0
61	Chongqing	China	30.0572	107.8740	0	0	0
62	Fujian	China	26.0789	117.9874	0	0	0
63	Gansu	China	35.7518	104.2861	0	0	0
64	Guangdong	China	23.3417	113.4244	0	0	0
65	Guangxi	China	23.8298	108.7881	0	0	0
66	Guizhou	China	26.8154	106.8748	0	0	0
67	Hainan	China	19.1959	109.7453	0	0	0
68	Hebei	China	39.5490	116.1306	0	1	1
69	Heilongjiang	China	47.8620	127.7615	0	0	1
70	Henan	China	37.8957	114.9042	0	0	0
72	Hubei	China	30.9756	112.2707	17	17	24
73	Hunan	China	27.6104	111.7088	0	0	0
74	Inner Mongolia	China	44.0935	113.9448	0	0	0
75	Jiangsu	China	32.9711	119.4550	0	0	0
76	Jiangxi	China	27.6140	115.7221	0	0	0
77	Jilin	China	43.6661	126.1923	0	0	0
78	Liaoning	China	41.2956	122.6085	0	0	0
79	Macau	China	22.1667	113.5500	0	0	0
80	Ningxia	China	37.2692	106.1655	0	0	0
81	Qinghai	China	35.7452	95.9956	0	0	0
82	Shaanxi	China	35.1917	108.8701	0	0	0
83	Shandong	China	36.3427	118.1498	0	0	0
84	Shanghai	China	31.2020	121.4491	0	0	0
85	Shanxi	China	37.5777	112.2922	0	0	0
86	Sichuan	China	30.6171	102.7103	0	0	0
87	Tianjin	China	39.3054	117.3230	0	0	0
88	Tibet	China	31.6927	88.0924	0	0	0
89	Unknown	China	NaN	NaN	0	0	0
90	Xinjiang	China	41.1129	85.2401	0	0	0
91	Yunnan	China	24.9740	101.4870	0	0	0
92	Zhejiang	China	29.1832	120.0934	0	0	0
0	NaN	China	NaN	NaN	17	18	26

	1/25/20	1/26/20	1/27/20	...	2/28/23	3/1/23	3/2/23	3/3/23	3/4/23	3/5/23	\
59	0	0	0	...	7	7	7	7	7	7	
60	0	0	1	...	20	20	20	20	20	20	
61	0	0	0	...	11	11	11	11	11	11	
62	0	0	0	...	2	2	2	2	2	2	
63	0	0	0	...	2	2	2	2	2	2	
64	0	0	0	...	10	10	10	10	10	10	
65	0	0	0	...	2	2	2	2	2	2	
66	0	0	0	...	2	2	2	2	2	2	
67	0	0	1	...	6	6	6	6	6	6	
68	1	1	1	...	7	7	7	7	7	7	
69	1	1	1	...	18	18	18	18	18	18	
70	0	1	1	...	23	23	23	23	23	23	

72	40	52	76	...	4515	4515	4515	4515	4515	4515
73	0	0	0	...	4	4	4	4	4	4
74	0	0	0	...	1	1	1	1	1	1
75	0	0	0	...	0	0	0	0	0	0
76	0	0	0	...	2	2	2	2	2	2
77	0	0	0	...	5	5	5	5	5	5
78	0	0	0	...	2	2	2	2	2	2
79	0	0	0	...	121	121	121	121	121	121
80	0	0	0	...	0	0	0	0	0	0
81	0	0	0	...	0	0	0	0	0	0
82	0	0	0	...	5	5	5	5	5	5
83	0	0	0	...	10	10	10	10	10	10
84	0	1	1	...	595	595	595	595	595	595
85	0	0	0	...	1	1	1	1	1	1
86	0	0	0	...	12	12	12	12	12	12
87	0	0	0	...	3	3	3	3	3	3
88	0	0	0	...	0	0	0	0	0	0
89	0	0	0	...	82195	82195	82195	82195	82195	82195
90	0	0	0	...	3	3	3	3	3	3
91	0	0	0	...	4	4	4	4	4	4
92	0	0	0	...	1	1	1	1	1	1
0	42	56	82	...	87589	87589	87589	87589	87589	87589

	3/6/23	3/7/23	3/8/23	3/9/23
59	7	7	7	7
60	20	20	20	20
61	11	11	11	11
62	2	2	2	2
63	2	2	2	2
64	10	10	10	10
65	2	2	2	2
66	2	2	2	2
67	6	6	6	6
68	7	7	7	7
69	18	18	18	18
70	23	23	23	23
72	4515	4515	4515	4515
73	4	4	4	4
74	1	1	1	1
75	0	0	0	0
76	2	2	2	2
77	5	5	5	5
78	2	2	2	2
79	121	121	121	121
80	0	0	0	0
81	0	0	0	0
82	5	5	5	5

```

83      10      10      10      10
84     595     595     595     595
85       1       1       1       1
86      12      12      12      12
87       3       3       3       3
88       0       0       0       0
89    82195    82195    82195    82195
90       3       3       3       3
91       4       4       4       4
92       1       1       1       1
0     87589    87589    87589    87589

```

[34 rows x 1147 columns]

On selectionne alors uniquement les pays d'interet comme precedemment

```

[30]: interest_countries = ["Belgium", "China", "France", "Germany", "Hong Kong",
    ↪ "Iran", "Italy", "Japan", "Korea, South", "Netherlands", "Portugal",
    ↪ "Spain", "United Kingdom", "US"]
df_allCountries_death = newSet_death.loc[(newSet_death['Country/Region'].
    ↪ isin(interest_countries)) & (newSet_death['Province/State'].isnull()) ,]
df_allCountries_death

```

```

[30]: Province/State Country/Region Lat Long 1/22/20 1/23/20 \
24      NaN Belgium 50.833300 4.469936 0 0
71      NaN Hong Kong 22.300000 114.200000 0 0
131     NaN France 46.227600 2.213700 0 0
135     NaN Germany 51.165691 10.451526 0 0
150     NaN Iran 32.427908 53.688046 0 0
154     NaN Italy 41.871940 12.567380 0 0
156     NaN Japan 36.204824 138.252924 0 0
162     NaN Korea, South 35.907757 127.766922 0 0
200     NaN Netherlands 52.132600 5.291300 0 0
218     NaN Portugal 39.399900 -8.224500 0 0
241     NaN Spain 40.463667 -3.749220 0 0
260     NaN US 40.000000 -100.000000 0 0
278     NaN United Kingdom 55.378100 -3.436000 0 0
0      NaN China NaN NaN 17 18

1/24/20 1/25/20 1/26/20 1/27/20 ... 2/28/23 3/1/23 3/2/23 \
24 0 0 0 0 ... 33717 33717 33775
71 0 0 0 0 ... 13459 13462 13462
131 0 0 0 0 ... 161340 161365 161386
135 0 0 0 0 ... 168086 168175 168296
150 0 0 0 0 ... 144845 144858 144864
154 0 0 0 0 ... 188094 188094 188094
156 0 0 0 0 ... 72395 72494 72581

```

162	0	0	0	0	...	33988	34003	34014
200	0	0	0	0	...	22990	22990	22990
218	0	0	0	0	...	26117	26180	26180
241	0	0	0	0	...	119380	119380	119380
260	0	0	0	0	...	1.11992e+06	1120897	1121658
278	0	0	0	0	...	219948	219948	219948
0	26	42	56	82	...	87589	87589	87589

	3/3/23	3/4/23	3/5/23	3/6/23	3/7/23	3/8/23	3/9/23
24	33775	33775	33775	33775	33775	33775	33814
71	13463	13464	13465	13466	13466	13466	13467
131	161407	161407	161407	161450	161474	161501	161512
135	168397	168397	168397	168397	168709	168808	168935
150	144867	144878	144893	144902	144907	144922	144933
154	188322	188322	188322	188322	188322	188322	188322
156	72648	72729	72779	72813	72848	72917	72997
162	34020	34020	34034	34049	34061	34081	34093
200	22990	22990	22990	22990	22990	22990	22990
218	26180	26180	26180	26180	26180	26266	26266
241	119479	119479	119479	119479	119479	119479	119479
260	1122165	NaN	NaN	1122181	1122516	1123246	1123836
278	219948	219948	219948	219948	219948	219948	219948
0	87589	87589	87589	87589	87589	87589	87589

[14 rows x 1147 columns]

On supprime les informations de latitude/longitude et on reformatte les dates

```
[31]: df_allCountries_death_final = df_allCountries_death.transpose()[5:]
df_allCountries_death_final.columns = df_allCountries_death["Country/Region"]

all_dates = pd.to_datetime(df_allCountries_death_final.index)
df_allCountries_death_final.index = all_dates
df_allCountries_death_final
```

```
[31]: Country/Region Belgium Hong Kong France Germany Iran Italy Japan \
2020-01-23 0 0 0 0 0 0 0
2020-01-24 0 0 0 0 0 0 0
2020-01-25 0 0 0 0 0 0 0
2020-01-26 0 0 0 0 0 0 0
2020-01-27 0 0 0 0 0 0 0
2020-01-28 0 0 0 0 0 0 0
2020-01-29 0 0 0 0 0 0 0
2020-01-30 0 0 0 0 0 0 0
2020-01-31 0 0 0 0 0 0 0
2020-02-01 0 0 0 0 0 0 0
2020-02-02 0 0 0 0 0 0 0
```

2020-02-03	0	0	0	0	0	0	0
2020-02-04	0	1	0	0	0	0	0
2020-02-05	0	1	0	0	0	0	0
2020-02-06	0	1	0	0	0	0	0
2020-02-07	0	1	0	0	0	0	0
2020-02-08	0	1	0	0	0	0	0
2020-02-09	0	1	0	0	0	0	0
2020-02-10	0	1	0	0	0	0	0
2020-02-11	0	1	0	0	0	0	0
2020-02-12	0	1	0	0	0	0	0
2020-02-13	0	1	0	0	0	0	1
2020-02-14	0	1	0	0	0	0	1
2020-02-15	0	1	1	0	0	0	1
2020-02-16	0	1	1	0	0	0	1
2020-02-17	0	1	1	0	0	0	1
2020-02-18	0	1	1	0	0	0	2
2020-02-19	0	2	1	0	2	0	2
2020-02-20	0	2	1	0	2	0	2
2020-02-21	0	2	1	0	4	1	2
...	...	...	...	...	...	...	...
2023-02-08	33582	13403	160880	166526	144771	187272	69970
2023-02-09	33616	13409	160897	166660	144775	187272	70193
2023-02-10	33616	13414	160917	166660	144779	187551	70385
2023-02-11	33616	13418	160917	166763	144779	187551	70566
2023-02-12	33616	13420	160917	166763	144781	187551	70703
2023-02-13	33616	13424	160965	166875	144783	187551	70796
2023-02-14	33616	13427	161000	166999	144788	187551	70931
2023-02-15	33616	13431	161035	167124	144789	187551	71144
2023-02-16	33616	13433	161062	167214	144793	187551	71316
2023-02-17	33663	13435	161090	167301	144793	187850	71457
2023-02-18	33663	13440	161090	167301	144793	187850	71587
2023-02-19	33663	13442	161090	167301	144804	187850	71694
2023-02-20	33663	13445	161137	167387	144812	187850	71745
2023-02-21	33663	13447	161169	167491	144817	187850	71817
2023-02-22	33663	13447	161206	167604	144824	187850	71931
2023-02-23	33717	13447	161225	167723	144828	187850	72059
2023-02-24	33717	13449	161254	167812	144832	188094	72142
2023-02-25	33717	13451	161254	167812	144835	188094	72214
2023-02-26	33717	13453	161254	167812	144842	188094	72276
2023-02-27	33717	13453	161306	167951	144845	188094	72328
2023-02-28	33717	13459	161340	168086	144845	188094	72395
2023-03-01	33717	13462	161365	168175	144858	188094	72494
2023-03-02	33775	13462	161386	168296	144864	188094	72581
2023-03-03	33775	13463	161407	168397	144867	188322	72648
2023-03-04	33775	13464	161407	168397	144878	188322	72729
2023-03-05	33775	13465	161407	168397	144893	188322	72779
2023-03-06	33775	13466	161450	168397	144902	188322	72813

2023-03-07	33775	13466	161474	168709	144907	188322	72848
2023-03-08	33775	13466	161501	168808	144922	188322	72917
2023-03-09	33814	13467	161512	168935	144933	188322	72997

Country/Region	Korea, South	Netherlands	Portugal	Spain	US	\
2020-01-23	0	0	0	0	0	
2020-01-24	0	0	0	0	0	
2020-01-25	0	0	0	0	0	
2020-01-26	0	0	0	0	0	
2020-01-27	0	0	0	0	0	
2020-01-28	0	0	0	0	0	
2020-01-29	0	0	0	0	0	
2020-01-30	0	0	0	0	0	
2020-01-31	0	0	0	0	0	
2020-02-01	0	0	0	0	0	
2020-02-02	0	0	0	0	0	
2020-02-03	0	0	0	0	0	
2020-02-04	0	0	0	0	0	
2020-02-05	0	0	0	0	0	
2020-02-06	0	0	0	0	0	
2020-02-07	0	0	0	0	0	
2020-02-08	0	0	0	0	0	
2020-02-09	0	0	0	0	0	
2020-02-10	0	0	0	0	0	
2020-02-11	0	0	0	0	0	
2020-02-12	0	0	0	0	0	
2020-02-13	0	0	0	0	0	
2020-02-14	0	0	0	0	0	
2020-02-15	0	0	0	0	0	
2020-02-16	0	0	0	0	0	
2020-02-17	0	0	0	0	0	
2020-02-18	0	0	0	0	0	
2020-02-19	0	0	0	0	0	
2020-02-20	1	0	0	0	0	
2020-02-21	2	0	0	0	0	

...	...	...	...	...	...
2023-02-08	33680	22990	26052	118712	1.11343e+06
2023-02-09	33697	22990	26052	118712	1.11438e+06
2023-02-10	33697	22990	26059	118976	1.11449e+06
2023-02-11	33736	22990	26059	118976	1114529
2023-02-12	33747	22990	26059	118976	1.11454e+06
2023-02-13	33758	22990	26059	118976	1.11471e+06
2023-02-14	33782	22990	26103	118976	1.11516e+06
2023-02-15	33804	22990	26103	118976	1.11574e+06
2023-02-16	33832	22990	26117	118976	1.11685e+06
2023-02-17	33844	22990	26117	119186	1.11757e+06
2023-02-18	33856	22990	26117	119186	1117589

2023-02-19	33865	22990	26117	119186	1117590
2023-02-20	33873	22990	26117	119186	1117663
2023-02-21	33887	22990	26117	119186	1.11802e+06
2023-02-22	33909	22990	26117	119186	1.11889e+06
2023-02-23	33929	22990	26117	119186	1119521
2023-02-24	33940	22990	26117	119380	1119573
2023-02-25	33940	22990	26117	119380	1119587
2023-02-26	33961	22990	26117	119380	NaN
2023-02-27	33977	22990	26117	119380	NaN
2023-02-28	33988	22990	26117	119380	1.11992e+06
2023-03-01	34003	22990	26180	119380	1120897
2023-03-02	34014	22990	26180	119380	1121658
2023-03-03	34020	22990	26180	119479	1122165
2023-03-04	34020	22990	26180	119479	NaN
2023-03-05	34034	22990	26180	119479	NaN
2023-03-06	34049	22990	26180	119479	1122181
2023-03-07	34061	22990	26180	119479	1122516
2023-03-08	34081	22990	26266	119479	1123246
2023-03-09	34093	22990	26266	119479	1123836

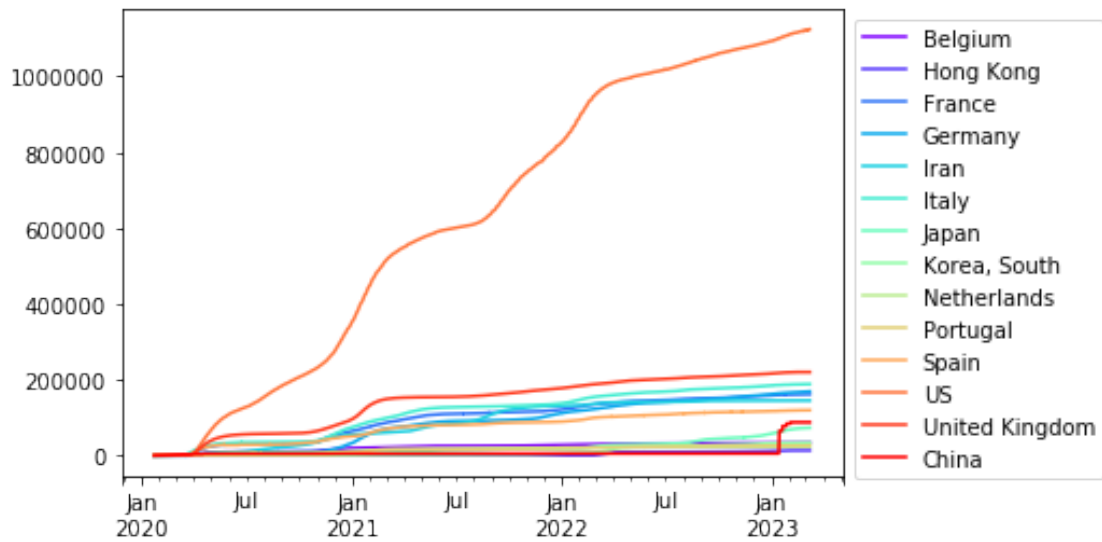
Country/Region	United Kingdom	China
2020-01-23	0	18
2020-01-24	0	26
2020-01-25	0	42
2020-01-26	0	56
2020-01-27	0	82
2020-01-28	0	131
2020-01-29	0	133
2020-01-30	1	171
2020-01-31	1	213
2020-02-01	1	259
2020-02-02	2	361
2020-02-03	2	425
2020-02-04	2	490
2020-02-05	2	562
2020-02-06	2	632
2020-02-07	2	717
2020-02-08	2	804
2020-02-09	2	904
2020-02-10	2	1011
2020-02-11	2	1111
2020-02-12	2	1116
2020-02-13	2	1368
2020-02-14	2	1520
2020-02-15	2	1662
2020-02-16	2	1765
2020-02-17	2	1863

2020-02-18	2	2002
2020-02-19	2	2114
2020-02-20	2	2236
2020-02-21	2	2236
...	...	...
2023-02-08	219819	87589
2023-02-09	219880	87589
2023-02-10	219948	87589
2023-02-11	219948	87589
2023-02-12	219948	87589
2023-02-13	219948	87589
2023-02-14	219948	87589
2023-02-15	219948	87589
2023-02-16	219948	87589
2023-02-17	219948	87589
2023-02-18	219948	87589
2023-02-19	219948	87589
2023-02-20	219948	87589
2023-02-21	219948	87589
2023-02-22	219948	87589
2023-02-23	219948	87589
2023-02-24	219948	87589
2023-02-25	219948	87589
2023-02-26	219948	87589
2023-02-27	219948	87589
2023-02-28	219948	87589
2023-03-01	219948	87589
2023-03-02	219948	87589
2023-03-03	219948	87589
2023-03-04	219948	87589
2023-03-05	219948	87589
2023-03-06	219948	87589
2023-03-07	219948	87589
2023-03-08	219948	87589
2023-03-09	219948	87589

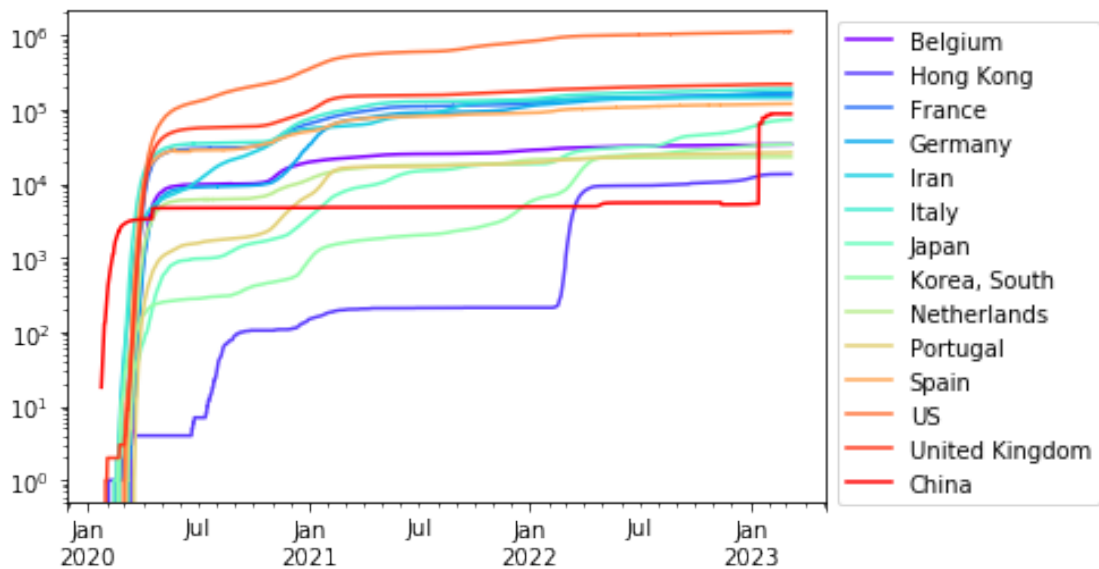
[1142 rows x 14 columns]

On regarde alors la distribution des deces dans les pays concernes. Attention ici on est en effectif cummule et ces donnees ne sont pas normalisees par le taux de deces classiquement observe hors epidemie de covid ni par la population de ces territoires.

```
[34]: fig = plt.figure()
ax = fig.add_subplot(111)
df_allCountries_death_final.plot(ax=ax, color=color)
ax.legend(bbox_to_anchor=(1, 1), loc="upper left")
plt.show()
```



```
[35]: fig = plt.figure()
ax = fig.add_subplot(111)
plt.yscale("log")
df_allCountries_death_final.plot(ax=ax, color=color)
ax.legend(bbox_to_anchor=(1, 1), loc="upper left")
plt.show()
```



On voit une fois encore que ce sont les USA avec le plus grand effectif de décès, ce qui, compte tenu de sa population de environ 333 millions d'habitants n'est pas étonnant.

Les epidemies se succedant (aujourd'hui 4 juillet 2024) malgre l'apparition des vaccins qui ont diminue le risque de mortalite, je vais m'arreter ici dans les analyses malgre une normalisation qui s'imposerait.

[]: