

exercice

September 23, 2021

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: sns.set_style("whitegrid")
# sns.set_context("paper")
sns.set(rc={'figure.figsize': (12,8)})
```

0.1 Download the data

First, we download the data from the [John Hopkins Github](#) and make a inspection of the columns:

```
[4]: url = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/
↳csse_covid_19_data/csse_covid_19_time_series/
↳time_series_covid19_confirmed_global.csv"
raw_data = pd.read_csv(url)
```

Now we can inspect the data:

```
[5]: raw_data
```

```
[5]:
```

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

14	Victoria	Australia
15	Western Australia	Australia
16	NaN	Austria
17	NaN	Azerbaijan
18	NaN	Bahamas
19	NaN	Bahrain
20	NaN	Bangladesh
21	NaN	Barbados
22	NaN	Belarus
23	NaN	Belgium
24	NaN	Belize
25	NaN	Benin
26	NaN	Bhutan
27	NaN	Bolivia
28	NaN	Bosnia and Herzegovina
29	NaN	Botswana
..
249	NaN	Timor-Leste
250	NaN	Togo
251	NaN	Trinidad and Tobago
252	NaN	Tunisia
253	NaN	Turkey
254	NaN	US
255	NaN	Uganda
256	NaN	Ukraine
257	NaN	United Arab Emirates
258	Anguilla	United Kingdom
259	Bermuda	United Kingdom
260	British Virgin Islands	United Kingdom
261	Cayman Islands	United Kingdom
262	Channel Islands	United Kingdom
263	Falkland Islands (Malvinas)	United Kingdom
264	Gibraltar	United Kingdom
265	Isle of Man	United Kingdom
266	Montserrat	United Kingdom
267	Saint Helena, Ascension and Tristan da Cunha	United Kingdom
268	Turks and Caicos Islands	United Kingdom
269	NaN	United Kingdom
270	NaN	Uruguay
271	NaN	Uzbekistan
272	NaN	Vanuatu
273	NaN	Venezuela
274	NaN	Vietnam
275	NaN	West Bank and Gaza
276	NaN	Yemen
277	NaN	Zambia
278	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	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	-27.469800	153.025100	0	0	0	0	0	
12	-34.928500	138.600700	0	0	0	0	0	
13	-42.882100	147.327200	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	25.025885	-78.035889	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	4.469936	0	0	0	0	0	
24	17.189900	-88.497600	0	0	0	0	0	
25	9.307700	2.315800	0	0	0	0	0	
26	27.514200	90.433600	0	0	0	0	0	
27	-16.290200	-63.588700	0	0	0	0	0	
28	43.915900	17.679100	0	0	0	0	0	
29	-22.328500	24.684900	0	0	0	0	0	
..	
249	-8.874217	125.727539	0	0	0	0	0	
250	8.619500	0.824800	0	0	0	0	0	
251	10.691800	-61.222500	0	0	0	0	0	
252	33.886917	9.537499	0	0	0	0	0	
253	38.963700	35.243300	0	0	0	0	0	
254	40.000000	-100.000000	1	1	2	2	5	
255	1.373333	32.290275	0	0	0	0	0	
256	48.379400	31.165600	0	0	0	0	0	
257	23.424076	53.847818	0	0	0	0	0	
258	18.220600	-63.068600	0	0	0	0	0	
259	32.307800	-64.750500	0	0	0	0	0	
260	18.420700	-64.640000	0	0	0	0	0	
261	19.313300	-81.254600	0	0	0	0	0	
262	49.372300	-2.364400	0	0	0	0	0	

263	-51.796300	-59.523600	0	0	0	0	0
264	36.140800	-5.353600	0	0	0	0	0
265	54.236100	-4.548100	0	0	0	0	0
266	16.742498	-62.187366	0	0	0	0	0
267	-7.946700	-14.355900	0	0	0	0	0
268	21.694000	-71.797900	0	0	0	0	0
269	55.378100	-3.436000	0	0	0	0	0
270	-32.522800	-55.765800	0	0	0	0	0
271	41.377491	64.585262	0	0	0	0	0
272	-15.376700	166.959200	0	0	0	0	0
273	6.423800	-66.589700	0	0	0	0	0
274	14.058324	108.277199	0	2	2	2	2
275	31.952200	35.233200	0	0	0	0	0
276	15.552727	48.516388	0	0	0	0	0
277	-13.133897	27.849332	0	0	0	0	0
278	-19.015438	29.154857	0	0	0	0	0

	1/27/20	...	9/13/21	9/14/21	9/15/21	9/16/21	9/17/21	\
0	0	...	154094	154180	154283	154361	154487	
1	0	...	157436	158431	159423	160365	161324	
2	0	...	200301	200528	200770	200989	201224	
3	0	...	15096	15099	15108	15113	15124	
4	0	...	50738	51047	51407	51827	52208	
5	0	...	2297	2304	2304	2304	2603	
6	0	...	5226831	5229848	5232358	5234851	5237159	
7	0	...	249146	249803	250559	251323	252082	
8	0	...	652	665	680	710	725	
9	4	...	45782	47007	48341	49611	50919	
10	0	...	203	203	204	204	204	
11	0	...	2009	2010	2013	2014	2015	
12	0	...	895	896	896	898	898	
13	0	...	235	235	235	235	235	
14	1	...	26036	26439	26942	27439	27968	
15	0	...	1089	1089	1089	1089	1092	
16	0	...	711573	713269	715893	718091	720455	
17	0	...	461249	463326	463326	467173	467173	
18	0	...	19601	19601	19795	19795	20030	
19	0	...	273835	273916	273977	274041	274107	
20	0	...	1532366	1534440	1536341	1538203	1540110	
21	0	...	5984	6053	6177	6248	6358	
22	0	...	504961	506591	508514	510481	512460	
23	0	...	1210381	1212106	1215114	1217473	1219814	
24	0	...	17767	17967	18143	18334	18532	
25	0	...	19841	21450	21450	21450	21450	
26	0	...	2596	2596	2596	2596	2596	
27	0	...	495035	495612	496032	496032	496700	
28	0	...	222161	222927	223957	224862	225857	

29	0	...	165644	165644	165644	172252	172252
..
249	0	...	18597	18715	18786	18856	18943
250	0	...	23778	23947	24093	24093	24369
251	0	...	47208	47363	47653	47925	48143
252	0	...	685799	695406	696279	697421	698427
253	0	...	6682834	6710636	6738860	6766978	6794670
254	5	...	41317328	41461662	41627946	41785903	41993789
255	0	...	121587	121617	121687	121784	121984
256	0	...	2420776	2424416	2429354	2435404	2442344
257	0	...	729518	730135	730743	731307	731828
258	0	...	318	318	318	325	331
259	0	...	3465	3877	3877	4027	4218
260	0	...	2642	2642	2642	2642	2642
261	0	...	729	745	745	772	772
262	0	...	11131	11168	11174	11223	11269
263	0	...	67	67	67	67	67
264	0	...	5436	5445	5449	5463	5469
265	0	...	7028	7050	7068	7094	7113
266	0	...	31	31	31	31	32
267	0	...	4	4	4	4	4
268	0	...	2758	2774	2774	2794	2799
269	0	...	7256559	7282810	7312683	7339009	7371301
270	0	...	386873	387028	387156	387299	387449
271	0	...	164890	165421	166025	166644	167268
272	0	...	4	4	4	4	4
273	0	...	347647	348873	349731	350795	353401
274	2	...	624547	635055	645640	656129	667650
275	0	...	372108	374768	377134	379635	381854
276	0	...	8452	8502	8527	8557	8593
277	0	...	207960	208049	208161	208267	208353
278	0	...	126399	126817	127083	127368	127632

	9/18/21	9/19/21	9/20/21	9/21/21	9/22/21
0	154487	154487	154585	154712	154757
1	162173	162953	163404	164276	165096
2	201425	201600	201766	201948	202122
3	15124	15124	15140	15140	15153
4	52307	52307	52644	52968	53387
5	2603	2603	2603	2603	2625
6	5238610	5239232	5241394	5243231	5245265
7	253093	253600	253942	254436	254709
8	742	749	765	782	798
9	51986	52922	53898	54919	55962
10	204	206	206	206	208
11	2015	2017	2018	2019	2021
12	898	898	898	898	899

13	235	235	235	235	235
14	28456	29008	29596	30216	30961
15	1092	1093	1094	1094	1094
16	722357	724035	725434	726674	728696
17	467173	467173	473459	475053	476409
18	20030	20030	20215	20288	20288
19	274179	274264	274383	274452	274524
20	1541300	1542683	1544238	1545800	1547176
21	6358	6527	6631	6736	6904
22	514446	516428	518369	520286	522275
23	1219814	1219814	1224885	1226682	1229236
24	18532	18532	18902	18902	19185
25	21450	21450	21450	21450	21450
26	2597	2597	2597	2599	2599
27	496950	497100	497386	497676	497984
28	225857	225857	227579	228105	229360
29	172252	172252	173788	173788	173788
..
249	18994	19033	19058	19125	19206
250	24519	24599	24655	24791	24902
251	48400	48523	48523	48826	49111
252	699224	699928	700400	700807	702503
253	6820831	6847229	6874917	6904255	6932423
254	42050638	42088171	42289819	42410607	42543510
255	122083	122212	122277	122405	122502
256	2448908	2453240	2455873	2461415	2468567
257	732299	732690	733003	733325	733643
258	331	331	331	340	350
259	4218	4477	4632	4632	4800
260	2642	2642	2642	2642	2642
261	774	774	774	787	791
262	11269	11269	11335	11368	11393
263	67	67	67	67	67
264	5476	5477	5479	5483	5490
265	7141	7166	7184	7218	7265
266	32	33	33	33	33
267	4	4	4	4	4
268	2800	2805	2805	2805	2813
269	7400739	7429746	7465448	7496543	7530103
270	387555	387627	387744	387922	388068
271	167858	168437	168938	169467	169989
272	4	4	4	4	4
273	353401	354239	356262	356262	358462
274	677023	687063	695744	707436	718963
275	382584	384390	386493	388470	390369
276	8630	8667	8718	8752	8789
277	208422	208469	208502	208599	208676

```
278      127739      127938      128186      128186      128804
```

```
[279 rows x 614 columns]
```

0.2 Creating the data for plotting:

We will exclude the latitude and longitude columns since they are not useful:

```
[6]: data = raw_data.drop(columns=["Lat", "Long"])
data
```

```
[6]:
```

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

252	NaN	Tunisia
253	NaN	Turkey
254	NaN	US
255	NaN	Uganda
256	NaN	Ukraine
257	NaN	United Arab Emirates
258	Anguilla	United Kingdom
259	Bermuda	United Kingdom
260	British Virgin Islands	United Kingdom
261	Cayman Islands	United Kingdom
262	Channel Islands	United Kingdom
263	Falkland Islands (Malvinas)	United Kingdom
264	Gibraltar	United Kingdom
265	Isle of Man	United Kingdom
266	Montserrat	United Kingdom
267	Saint Helena, Ascension and Tristan da Cunha	United Kingdom
268	Turks and Caicos Islands	United Kingdom
269	NaN	United Kingdom
270	NaN	Uruguay
271	NaN	Uzbekistan
272	NaN	Vanuatu
273	NaN	Venezuela
274	NaN	Vietnam
275	NaN	West Bank and Gaza
276	NaN	Yemen
277	NaN	Zambia
278	NaN	Zimbabwe

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	\
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	
9	0	0	0	0	3	4	4	4	
10	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	1	
12	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	
14	0	0	0	0	1	1	1	1	
15	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	

18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0
..	
249	0	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0	0
251	0	0	0	0	0	0	0	0	0
252	0	0	0	0	0	0	0	0	0
253	0	0	0	0	0	0	0	0	0
254	1	1	2	2	5	5	5	5	6
255	0	0	0	0	0	0	0	0	0
256	0	0	0	0	0	0	0	0	0
257	0	0	0	0	0	0	0	0	4
258	0	0	0	0	0	0	0	0	0
259	0	0	0	0	0	0	0	0	0
260	0	0	0	0	0	0	0	0	0
261	0	0	0	0	0	0	0	0	0
262	0	0	0	0	0	0	0	0	0
263	0	0	0	0	0	0	0	0	0
264	0	0	0	0	0	0	0	0	0
265	0	0	0	0	0	0	0	0	0
266	0	0	0	0	0	0	0	0	0
267	0	0	0	0	0	0	0	0	0
268	0	0	0	0	0	0	0	0	0
269	0	0	0	0	0	0	0	0	0
270	0	0	0	0	0	0	0	0	0
271	0	0	0	0	0	0	0	0	0
272	0	0	0	0	0	0	0	0	0
273	0	0	0	0	0	0	0	0	0
274	0	2	2	2	2	2	2	2	2
275	0	0	0	0	0	0	0	0	0
276	0	0	0	0	0	0	0	0	0
277	0	0	0	0	0	0	0	0	0
278	0	0	0	0	0	0	0	0	0
	...	9/13/21	9/14/21	9/15/21	9/16/21	9/17/21	9/18/21	\	
0	...	154094	154180	154283	154361	154487	154487		
1	...	157436	158431	159423	160365	161324	162173		

2	...	200301	200528	200770	200989	201224	201425
3	...	15096	15099	15108	15113	15124	15124
4	...	50738	51047	51407	51827	52208	52307
5	...	2297	2304	2304	2304	2603	2603
6	...	5226831	5229848	5232358	5234851	5237159	5238610
7	...	249146	249803	250559	251323	252082	253093
8	...	652	665	680	710	725	742
9	...	45782	47007	48341	49611	50919	51986
10	...	203	203	204	204	204	204
11	...	2009	2010	2013	2014	2015	2015
12	...	895	896	896	898	898	898
13	...	235	235	235	235	235	235
14	...	26036	26439	26942	27439	27968	28456
15	...	1089	1089	1089	1089	1092	1092
16	...	711573	713269	715893	718091	720455	722357
17	...	461249	463326	463326	467173	467173	467173
18	...	19601	19601	19795	19795	20030	20030
19	...	273835	273916	273977	274041	274107	274179
20	...	1532366	1534440	1536341	1538203	1540110	1541300
21	...	5984	6053	6177	6248	6358	6358
22	...	504961	506591	508514	510481	512460	514446
23	...	1210381	1212106	1215114	1217473	1219814	1219814
24	...	17767	17967	18143	18334	18532	18532
25	...	19841	21450	21450	21450	21450	21450
26	...	2596	2596	2596	2596	2596	2597
27	...	495035	495612	496032	496032	496700	496950
28	...	222161	222927	223957	224862	225857	225857
29	...	165644	165644	165644	172252	172252	172252
..
249	...	18597	18715	18786	18856	18943	18994
250	...	23778	23947	24093	24093	24369	24519
251	...	47208	47363	47653	47925	48143	48400
252	...	685799	695406	696279	697421	698427	699224
253	...	6682834	6710636	6738860	6766978	6794670	6820831
254	...	41317328	41461662	41627946	41785903	41993789	42050638
255	...	121587	121617	121687	121784	121984	122083
256	...	2420776	2424416	2429354	2435404	2442344	2448908
257	...	729518	730135	730743	731307	731828	732299
258	...	318	318	318	325	331	331
259	...	3465	3877	3877	4027	4218	4218
260	...	2642	2642	2642	2642	2642	2642
261	...	729	745	745	772	772	774
262	...	11131	11168	11174	11223	11269	11269
263	...	67	67	67	67	67	67
264	...	5436	5445	5449	5463	5469	5476
265	...	7028	7050	7068	7094	7113	7141
266	...	31	31	31	31	32	32

267	...	4	4	4	4	4	4
268	...	2758	2774	2774	2794	2799	2800
269	...	7256559	7282810	7312683	7339009	7371301	7400739
270	...	386873	387028	387156	387299	387449	387555
271	...	164890	165421	166025	166644	167268	167858
272	...	4	4	4	4	4	4
273	...	347647	348873	349731	350795	353401	353401
274	...	624547	635055	645640	656129	667650	677023
275	...	372108	374768	377134	379635	381854	382584
276	...	8452	8502	8527	8557	8593	8630
277	...	207960	208049	208161	208267	208353	208422
278	...	126399	126817	127083	127368	127632	127739

	9/19/21	9/20/21	9/21/21	9/22/21
0	154487	154585	154712	154757
1	162953	163404	164276	165096
2	201600	201766	201948	202122
3	15124	15140	15140	15153
4	52307	52644	52968	53387
5	2603	2603	2603	2625
6	5239232	5241394	5243231	5245265
7	253600	253942	254436	254709
8	749	765	782	798
9	52922	53898	54919	55962
10	206	206	206	208
11	2017	2018	2019	2021
12	898	898	898	899
13	235	235	235	235
14	29008	29596	30216	30961
15	1093	1094	1094	1094
16	724035	725434	726674	728696
17	467173	473459	475053	476409
18	20030	20215	20288	20288
19	274264	274383	274452	274524
20	1542683	1544238	1545800	1547176
21	6527	6631	6736	6904
22	516428	518369	520286	522275
23	1219814	1224885	1226682	1229236
24	18532	18902	18902	19185
25	21450	21450	21450	21450
26	2597	2597	2599	2599
27	497100	497386	497676	497984
28	225857	227579	228105	229360
29	172252	173788	173788	173788
..
249	19033	19058	19125	19206
250	24599	24655	24791	24902

251	48523	48523	48826	49111
252	699928	700400	700807	702503
253	6847229	6874917	6904255	6932423
254	42088171	42289819	42410607	42543510
255	122212	122277	122405	122502
256	2453240	2455873	2461415	2468567
257	732690	733003	733325	733643
258	331	331	340	350
259	4477	4632	4632	4800
260	2642	2642	2642	2642
261	774	774	787	791
262	11269	11335	11368	11393
263	67	67	67	67
264	5477	5479	5483	5490
265	7166	7184	7218	7265
266	33	33	33	33
267	4	4	4	4
268	2805	2805	2805	2813
269	7429746	7465448	7496543	7530103
270	387627	387744	387922	388068
271	168437	168938	169467	169989
272	4	4	4	4
273	354239	356262	356262	358462
274	687063	695744	707436	718963
275	384390	386493	388470	390369
276	8667	8718	8752	8789
277	208469	208502	208599	208676
278	127938	128186	128186	128804

[279 rows x 612 columns]

Since we will exclude Hong-Kong from the China aggregated numbers, we can modify the country in the Hong-Kong province to Hong-Kong,

```
[7]: data.loc[data["Province/State"]=="Hong Kong", "Country/Region"] = "Hong Kong"
data
```

```
[7]:
```

	Province/State	Country/Region \
0	NaN	Afghanistan
1	NaN	Albania
2	NaN	Algeria
3	NaN	Andorra
4	NaN	Angola
5	NaN	Antigua and Barbuda
6	NaN	Argentina
7	NaN	Armenia
8	Australian Capital Territory	Australia

9	New South Wales	Australia
10	Northern Territory	Australia
11	Queensland	Australia
12	South Australia	Australia
13	Tasmania	Australia
14	Victoria	Australia
15	Western Australia	Australia
16	NaN	Austria
17	NaN	Azerbaijan
18	NaN	Bahamas
19	NaN	Bahrain
20	NaN	Bangladesh
21	NaN	Barbados
22	NaN	Belarus
23	NaN	Belgium
24	NaN	Belize
25	NaN	Benin
26	NaN	Bhutan
27	NaN	Bolivia
28	NaN	Bosnia and Herzegovina
29	NaN	Botswana
..
249	NaN	Timor-Leste
250	NaN	Togo
251	NaN	Trinidad and Tobago
252	NaN	Tunisia
253	NaN	Turkey
254	NaN	US
255	NaN	Uganda
256	NaN	Ukraine
257	NaN	United Arab Emirates
258	Anguilla	United Kingdom
259	Bermuda	United Kingdom
260	British Virgin Islands	United Kingdom
261	Cayman Islands	United Kingdom
262	Channel Islands	United Kingdom
263	Falkland Islands (Malvinas)	United Kingdom
264	Gibraltar	United Kingdom
265	Isle of Man	United Kingdom
266	Montserrat	United Kingdom
267	Saint Helena, Ascension and Tristan da Cunha	United Kingdom
268	Turks and Caicos Islands	United Kingdom
269	NaN	United Kingdom
270	NaN	Uruguay
271	NaN	Uzbekistan
272	NaN	Vanuatu
273	NaN	Venezuela

274	NaN	Vietnam
275	NaN	West Bank and Gaza
276	NaN	Yemen
277	NaN	Zambia
278	NaN	Zimbabwe

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	\
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	
9	0	0	0	0	3	4	4	4	
10	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	1	
12	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	
14	0	0	0	0	1	1	1	1	
15	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	
21	0	0	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	
24	0	0	0	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	
26	0	0	0	0	0	0	0	0	
27	0	0	0	0	0	0	0	0	
28	0	0	0	0	0	0	0	0	
29	0	0	0	0	0	0	0	0	
..	
249	0	0	0	0	0	0	0	0	
250	0	0	0	0	0	0	0	0	
251	0	0	0	0	0	0	0	0	
252	0	0	0	0	0	0	0	0	
253	0	0	0	0	0	0	0	0	
254	1	1	2	2	5	5	5	6	
255	0	0	0	0	0	0	0	0	
256	0	0	0	0	0	0	0	0	
257	0	0	0	0	0	0	0	4	

258	0	0	0	0	0	0	0	0
259	0	0	0	0	0	0	0	0
260	0	0	0	0	0	0	0	0
261	0	0	0	0	0	0	0	0
262	0	0	0	0	0	0	0	0
263	0	0	0	0	0	0	0	0
264	0	0	0	0	0	0	0	0
265	0	0	0	0	0	0	0	0
266	0	0	0	0	0	0	0	0
267	0	0	0	0	0	0	0	0
268	0	0	0	0	0	0	0	0
269	0	0	0	0	0	0	0	0
270	0	0	0	0	0	0	0	0
271	0	0	0	0	0	0	0	0
272	0	0	0	0	0	0	0	0
273	0	0	0	0	0	0	0	0
274	0	2	2	2	2	2	2	2
275	0	0	0	0	0	0	0	0
276	0	0	0	0	0	0	0	0
277	0	0	0	0	0	0	0	0
278	0	0	0	0	0	0	0	0

	...	9/13/21	9/14/21	9/15/21	9/16/21	9/17/21	9/18/21	\
0	...	154094	154180	154283	154361	154487	154487	
1	...	157436	158431	159423	160365	161324	162173	
2	...	200301	200528	200770	200989	201224	201425	
3	...	15096	15099	15108	15113	15124	15124	
4	...	50738	51047	51407	51827	52208	52307	
5	...	2297	2304	2304	2304	2603	2603	
6	...	5226831	5229848	5232358	5234851	5237159	5238610	
7	...	249146	249803	250559	251323	252082	253093	
8	...	652	665	680	710	725	742	
9	...	45782	47007	48341	49611	50919	51986	
10	...	203	203	204	204	204	204	
11	...	2009	2010	2013	2014	2015	2015	
12	...	895	896	896	898	898	898	
13	...	235	235	235	235	235	235	
14	...	26036	26439	26942	27439	27968	28456	
15	...	1089	1089	1089	1089	1092	1092	
16	...	711573	713269	715893	718091	720455	722357	
17	...	461249	463326	463326	467173	467173	467173	
18	...	19601	19601	19795	19795	20030	20030	
19	...	273835	273916	273977	274041	274107	274179	
20	...	1532366	1534440	1536341	1538203	1540110	1541300	
21	...	5984	6053	6177	6248	6358	6358	
22	...	504961	506591	508514	510481	512460	514446	
23	...	1210381	1212106	1215114	1217473	1219814	1219814	

24	...	17767	17967	18143	18334	18532	18532
25	...	19841	21450	21450	21450	21450	21450
26	...	2596	2596	2596	2596	2596	2597
27	...	495035	495612	496032	496032	496700	496950
28	...	222161	222927	223957	224862	225857	225857
29	...	165644	165644	165644	172252	172252	172252
..
249	...	18597	18715	18786	18856	18943	18994
250	...	23778	23947	24093	24093	24369	24519
251	...	47208	47363	47653	47925	48143	48400
252	...	685799	695406	696279	697421	698427	699224
253	...	6682834	6710636	6738860	6766978	6794670	6820831
254	...	41317328	41461662	41627946	41785903	41993789	42050638
255	...	121587	121617	121687	121784	121984	122083
256	...	2420776	2424416	2429354	2435404	2442344	2448908
257	...	729518	730135	730743	731307	731828	732299
258	...	318	318	318	325	331	331
259	...	3465	3877	3877	4027	4218	4218
260	...	2642	2642	2642	2642	2642	2642
261	...	729	745	745	772	772	774
262	...	11131	11168	11174	11223	11269	11269
263	...	67	67	67	67	67	67
264	...	5436	5445	5449	5463	5469	5476
265	...	7028	7050	7068	7094	7113	7141
266	...	31	31	31	31	32	32
267	...	4	4	4	4	4	4
268	...	2758	2774	2774	2794	2799	2800
269	...	7256559	7282810	7312683	7339009	7371301	7400739
270	...	386873	387028	387156	387299	387449	387555
271	...	164890	165421	166025	166644	167268	167858
272	...	4	4	4	4	4	4
273	...	347647	348873	349731	350795	353401	353401
274	...	624547	635055	645640	656129	667650	677023
275	...	372108	374768	377134	379635	381854	382584
276	...	8452	8502	8527	8557	8593	8630
277	...	207960	208049	208161	208267	208353	208422
278	...	126399	126817	127083	127368	127632	127739

	9/19/21	9/20/21	9/21/21	9/22/21
0	154487	154585	154712	154757
1	162953	163404	164276	165096
2	201600	201766	201948	202122
3	15124	15140	15140	15153
4	52307	52644	52968	53387
5	2603	2603	2603	2625
6	5239232	5241394	5243231	5245265
7	253600	253942	254436	254709

8	749	765	782	798
9	52922	53898	54919	55962
10	206	206	206	208
11	2017	2018	2019	2021
12	898	898	898	899
13	235	235	235	235
14	29008	29596	30216	30961
15	1093	1094	1094	1094
16	724035	725434	726674	728696
17	467173	473459	475053	476409
18	20030	20215	20288	20288
19	274264	274383	274452	274524
20	1542683	1544238	1545800	1547176
21	6527	6631	6736	6904
22	516428	518369	520286	522275
23	1219814	1224885	1226682	1229236
24	18532	18902	18902	19185
25	21450	21450	21450	21450
26	2597	2597	2599	2599
27	497100	497386	497676	497984
28	225857	227579	228105	229360
29	172252	173788	173788	173788
..
249	19033	19058	19125	19206
250	24599	24655	24791	24902
251	48523	48523	48826	49111
252	699928	700400	700807	702503
253	6847229	6874917	6904255	6932423
254	42088171	42289819	42410607	42543510
255	122212	122277	122405	122502
256	2453240	2455873	2461415	2468567
257	732690	733003	733325	733643
258	331	331	340	350
259	4477	4632	4632	4800
260	2642	2642	2642	2642
261	774	774	787	791
262	11269	11335	11368	11393
263	67	67	67	67
264	5477	5479	5483	5490
265	7166	7184	7218	7265
266	33	33	33	33
267	4	4	4	4
268	2805	2805	2805	2813
269	7429746	7465448	7496543	7530103
270	387627	387744	387922	388068
271	168437	168938	169467	169989
272	4	4	4	4

273	354239	356262	356262	358462
274	687063	695744	707436	718963
275	384390	386493	388470	390369
276	8667	8718	8752	8789
277	208469	208502	208599	208676
278	127938	128186	128186	128804

[279 rows x 612 columns]

We will also exclude all the France regions that are not metropolitane France. First, let's take a look at the possible provinces:

```
[8]: data[data["Country/Region"]=="France"]["Province/State"].values
```

```
[8]: array(['French Guiana', 'French Polynesia', 'Guadeloupe', 'Martinique',
        'Mayotte', 'New Caledonia', 'Reunion', 'Saint Barthelemy',
        'Saint Pierre and Miquelon', 'St Martin', 'Wallis and Futuna', nan],
       dtype=object)
```

Now, we can exclude all the values besides the last one:

```
[10]: list_of_regions = data[data["Country/Region"]=="France"]["Province/State"].
        ↪values[:-1]
data = data.drop(data[(data["Country/Region"] == "France") & (data["Province/
        ↪State"].isin(list_of_regions))].index)
```

We do the same thing with Netherlands and UK:

```
[11]: print(data[data["Country/Region"]=="United Kingdom"]["Province/State"].values)
print(data[data["Country/Region"]=="Netherlands"]["Province/State"].values)
```

```
['Anguilla' 'Bermuda' 'British Virgin Islands' 'Cayman Islands'
 'Channel Islands' 'Falkland Islands (Malvinas)' 'Gibraltar' 'Isle of Man'
 'Montserrat' 'Saint Helena, Ascension and Tristan da Cunha'
 'Turks and Caicos Islands' nan]
['Aruba' 'Bonaire, Sint Eustatius and Saba' 'Curacao' 'Sint Maarten' nan]
```

```
[12]: regions_netherlands = data[data["Country/Region"]=="Netherlands"]["Province/
        ↪State"].values[:-1]
data = data.drop(data[(data["Country/Region"] == "Netherlands") &
        ↪(data["Province/State"].isin(regions_netherlands))].index)

regions_uk = data[data["Country/Region"]=="United Kingdom"]["Province/State"].
        ↪values[:-1]
data = data.drop(data[(data["Country/Region"] == "United Kingdom") &
        ↪(data["Province/State"].isin(regions_uk))].index)
```

0.3 Visualizations:

Now we can make our plots, but first we need to group the data by country:

```
[13]: data_grouped = data.groupby('Country/Region').sum()
data_grouped.head()
```

```
[13]:
```

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	\
Country/Region								
Afghanistan	0	0	0	0	0	0	0	
Albania	0	0	0	0	0	0	0	
Algeria	0	0	0	0	0	0	0	
Andorra	0	0	0	0	0	0	0	
Angola	0	0	0	0	0	0	0	

	1/29/20	1/30/20	1/31/20	...	9/13/21	9/14/21	9/15/21	\
Country/Region				...				
Afghanistan	0	0	0	...	154094	154180	154283	
Albania	0	0	0	...	157436	158431	159423	
Algeria	0	0	0	...	200301	200528	200770	
Andorra	0	0	0	...	15096	15099	15108	
Angola	0	0	0	...	50738	51047	51407	

	9/16/21	9/17/21	9/18/21	9/19/21	9/20/21	9/21/21	9/22/21
Country/Region							
Afghanistan	154361	154487	154487	154487	154585	154712	154757
Albania	160365	161324	162173	162953	163404	164276	165096
Algeria	200989	201224	201425	201600	201766	201948	202122
Andorra	15113	15124	15124	15124	15140	15140	15153
Angola	51827	52208	52307	52307	52644	52968	53387

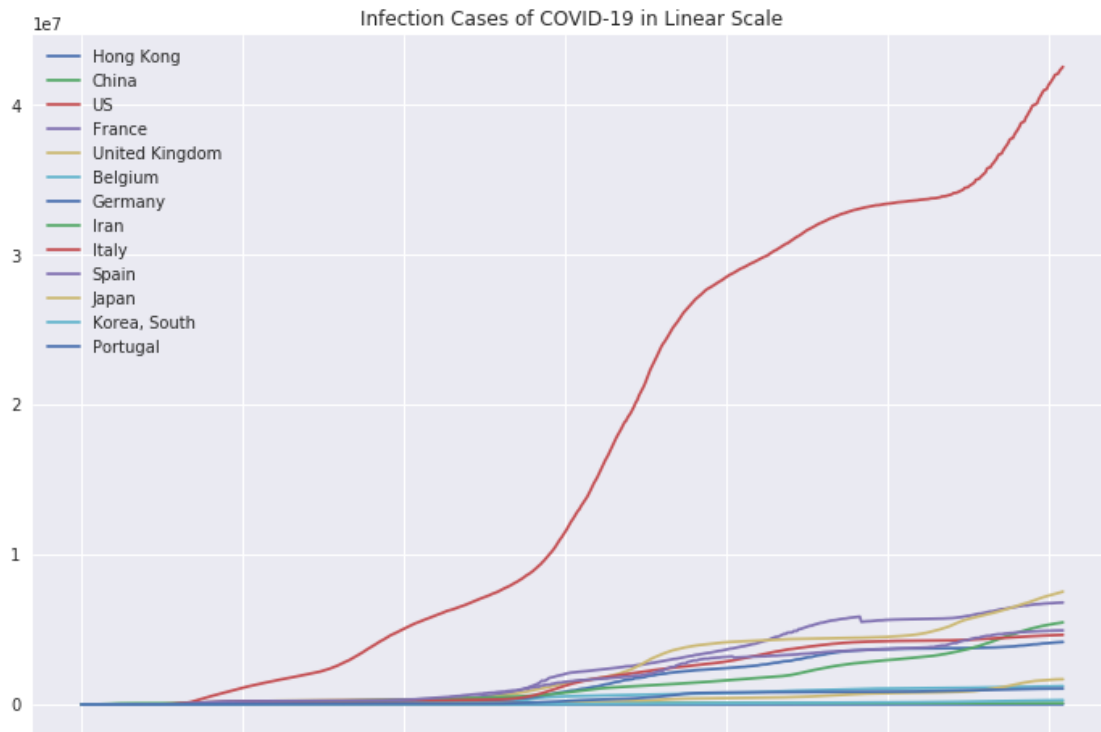
[5 rows x 610 columns]

Now we can plot our selected countries:

```
[15]: data_grouped.loc['Hong Kong'].plot()
data_grouped.loc['China'].plot()
data_grouped.loc['US'].plot()
data_grouped.loc['France'].plot()
data_grouped.loc['United Kingdom'].plot()
data_grouped.loc['Belgium'].plot()
data_grouped.loc['Germany'].plot()
data_grouped.loc['Iran'].plot()
data_grouped.loc['Italy'].plot()
data_grouped.loc['Spain'].plot()
data_grouped.loc['Japan'].plot()
data_grouped.loc['Korea, South'].plot()
data_grouped.loc['Portugal'].plot()
```

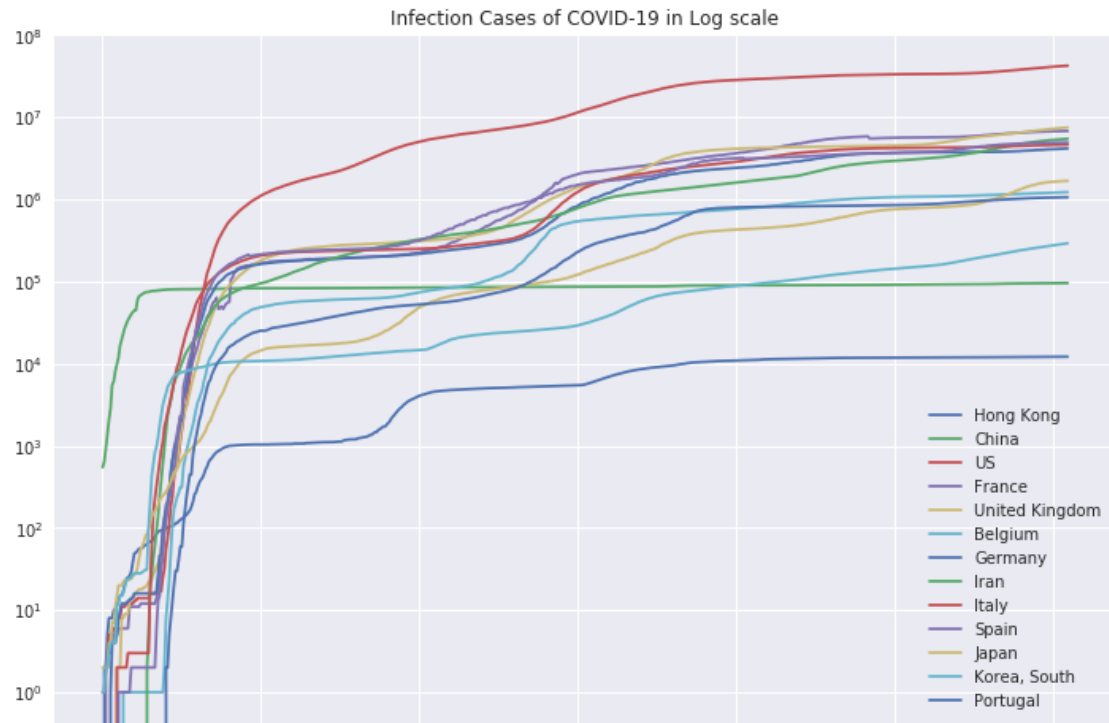
```
plt.legend()
plt.title('Infection Cases of COVID-19 in Linear Scale')
```

```
[15]: Text(0.5,1,'Infection Cases of COVID-19 in Linear Scale')
```



```
[16]: data_grouped.loc['Hong Kong'].plot()
data_grouped.loc['China'].plot()
data_grouped.loc['US'].plot()
data_grouped.loc['France'].plot()
data_grouped.loc['United Kingdom'].plot()
data_grouped.loc['Belgium'].plot()
data_grouped.loc['Germany'].plot()
data_grouped.loc['Iran'].plot()
data_grouped.loc['Italy'].plot()
data_grouped.loc['Spain'].plot()
data_grouped.loc['Japan'].plot()
data_grouped.loc['Korea, South'].plot()
data_grouped.loc['Portugal'].plot()
plt.legend()
plt.yscale("log")
plt.title('Infection Cases of COVID-19 in Log scale')
```

```
[16]: Text(0.5,1,'Infection Cases of COVID-19 in Log scale')
```



[]: