In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns #visualisation
import matplotlib.pyplot as plt #visualisation
from sklearn.preprocessing import LabelEncoder #To Encoding
import missingno as msno #Visualisation Matrix NaNs
```

STEPS

Dataframe about Coronavirus Group D

In [2]:

```
from utils.folders_tb import downloader
world = downloader(url='https://covid.ourworldindata.org/data/owid-covid-data.csv')
world
```

Out[2]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	nev
0	AFG	Asia	Afghanistan	2020-02-24	1.0	1.0	NaN	NaN	NaN	
1	AFG	Asia	Afghanistan	2020-02-25	1.0	0.0	NaN	NaN	NaN	
2	AFG	Asia	Afghanistan	2020-02-26	1.0	0.0	NaN	NaN	NaN	
3	AFG	Asia	Afghanistan	2020-02-27	1.0	0.0	NaN	NaN	NaN	
4	AFG	Asia	Afghanistan	2020-02-28	1.0	0.0	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	
62060	ZWE	Africa	Zimbabwe	2021-01-20	29408.0	733.0	736.000	879.0	54.0	
62061	ZWE	Africa	Zimbabwe	2021-01-21	30047.0	639.0	668.429	917.0	38.0	
62062	ZWE	Africa	Zimbabwe	2021-01-22	30523.0	476.0	630.571	962.0	45.0	
62063	ZWE	Africa	Zimbabwe	2021-01-23	31007.0	484.0	589.429	974.0	12.0	
62064	ZWE	Africa	Zimbabwe	2021-01-24	31320.0	313.0	588.143	1005.0	31.0	

62065 rows × 55 columns

In [5]:

```
#This correlation Matrix of NaNs is showing how many columns have a lot of non-values
msno.matrix(world)
```

Out[5]:

<AxesSubplot:>

In [4]:

```
msno.heatmap(world)
```

Out[4]:

```
<AxesSubplot:>
```

In [6]:

```
world.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62047 entries, 0 to 62046
Data columns (total 55 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   iso_code                               61694 non-null  object 
 1   continent                              61325 non-null  object 
 2   location                               62047 non-null  object 
 3   date                                   62047 non-null  object 
 4   total_cases                            61450 non-null  float64
 5   new_cases                              61443 non-null  float64
 6   new_cases_smoothed                     60484 non-null  float64
 7   total_deaths                           52913 non-null  float64
 8   new_deaths                             52912 non-null  float64
 9   new_deaths_smoothed                    60484 non-null  float64
 10  total_cases_per_million                61097 non-null  float64
 11  new_cases_per_million                  61090 non-null  float64
 12  new_cases_smoothed_per_million         60136 non-null  float64
 13  total_deaths_per_million               52573 non-null  float64
 14  new_deaths_per_million                 52572 non-null  float64
 15  new_deaths_smoothed_per_million        60136 non-null  float64
 16  reproduction_rate                      50035 non-null  float64
 17  icu_patients                           6856 non-null   float64
 18  icu_patients_per_million               6856 non-null   float64
 19  hosp_patients                          8114 non-null   float64
 20  hosp_patients_per_million              8114 non-null   float64
 21  weekly_icu_admissions                  672 non-null    float64
 22  weekly_icu_admissions_per_million      672 non-null    float64
 23  weekly_hosp_admissions                 991 non-null    float64
 24  weekly_hosp_admissions_per_million     991 non-null    float64
 25  total_tests                            28702 non-null  float64
 26  new_tests                              28885 non-null  float64
 27  total_tests_per_thousand               28702 non-null  float64
 28  new_tests_per_thousand                 28885 non-null  float64
 29  new_tests_smoothed                     32573 non-null  float64
 30  new_tests_smoothed_per_thousand        32573 non-null  float64
 31  positive_rate                          30771 non-null  float64
 32  tests_per_case                         30282 non-null  float64
 33  tests_units                            33772 non-null  object 
 34  total_vaccinations                     922 non-null    float64
 35  new_vaccinations                       725 non-null    float64
 36  new_vaccinations_smoothed              1196 non-null   float64
 37  total_vaccinations_per_hundred         922 non-null    float64
 38  new_vaccinations_smoothed_per_million  1196 non-null   float64
 39  stringency_index                       55360 non-null  float64
 40  population                             61694 non-null  float64
 41  population_density                     60390 non-null  float64
 42  median_age                             59028 non-null  float64
 43  aged_65_older                          58344 non-null  float64
 44  aged_70_older                          58694 non-null  float64
 45  gdp_per_capita                         59064 non-null  float64
 46  extreme_poverty                        40417 non-null  float64
 47  cardiovasc_death_rate                  59665 non-null  float64
 48  diabetes_prevalence                    60368 non-null  float64
 49  female_smokers                         47084 non-null  float64
 50  male_smokers                           46439 non-null  float64
 51  handwashing_facilities                 29886 non-null  float64
 52  hospital_beds_per_thousand             54798 non-null  float64
 53  life_expectancy                        61377 non-null  float64
 54  human_development_index                59329 non-null  float64
```

```
 54  human_development_index                59329 non-null  float64
dtypes: float64(50), object(5)
memory usage: 26.0+ MB
```

In [8]:

```
#In this section only it is showing how "date" columns has been changed from object to da
tetime to can work in future functions.
world_date = downloader(url='https://covid.ourworldindata.org/data/owid-covid-data.csv')
world_date["date"] = pd.to_datetime(world_date["date"])
world_date.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62047 entries, 0 to 62046
Data columns (total 55 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   iso_code                              61694 non-null  object
 1   continent                             61325 non-null  object
 2   location                              62047 non-null  object
 3   date                                  62047 non-null  datetime64[ns]
 4   total_cases                           61450 non-null  float64
 5   new_cases                             61443 non-null  float64
 6   new_cases_smoothed                    60484 non-null  float64
 7   total_deaths                          52913 non-null  float64
 8   new_deaths                            52912 non-null  float64
 9   new_deaths_smoothed                   60484 non-null  float64
 10  total_cases_per_million               61097 non-null  float64
 11  new_cases_per_million                 61090 non-null  float64
 12  new_cases_smoothed_per_million        60136 non-null  float64
 13  total_deaths_per_million              52573 non-null  float64
 14  new_deaths_per_million                52572 non-null  float64
 15  new_deaths_smoothed_per_million       60136 non-null  float64
 16  reproduction_rate                     50035 non-null  float64
 17  icu_patients                          6856 non-null   float64
 18  icu_patients_per_million              6856 non-null   float64
 19  hosp_patients                         8114 non-null   float64
 20  hosp_patients_per_million             8114 non-null   float64
 21  weekly_icu_admissions                 672 non-null    float64
 22  weekly_icu_admissions_per_million     672 non-null    float64
 23  weekly_hosp_admissions                991 non-null    float64
 24  weekly_hosp_admissions_per_million    991 non-null    float64
 25  total_tests                           28702 non-null  float64
 26  new_tests                             28885 non-null  float64
 27  total_tests_per_thousand              28702 non-null  float64
 28  new_tests_per_thousand                28885 non-null  float64
 29  new_tests_smoothed                    32573 non-null  float64
 30  new_tests_smoothed_per_thousand       32573 non-null  float64
 31  positive_rate                         30771 non-null  float64
 32  tests_per_case                        30282 non-null  float64
 33  tests_units                           33772 non-null  object
 34  total_vaccinations                    922 non-null    float64
 35  new_vaccinations                      725 non-null    float64
 36  new_vaccinations_smoothed             1196 non-null   float64
 37  total_vaccinations_per_hundred        922 non-null    float64
 38  new_vaccinations_smoothed_per_million 1196 non-null   float64
 39  stringency_index                      55360 non-null  float64
 40  population                            61694 non-null  float64
 41  population_density                    60390 non-null  float64
 42  median_age                            59028 non-null  float64
 43  aged_65_older                         58344 non-null  float64
 44  aged_70_older                         58694 non-null  float64
 45  gdp_per_capita                        59064 non-null  float64
 46  extreme_poverty                       40417 non-null  float64
 47  cardiovasc_death_rate                 59665 non-null  float64
 48  diabetes_prevalence                   60368 non-null  float64
 49  female_smokers                        47084 non-null  float64
 50  male_smokers                          46439 non-null  float64
 51  handwashing_facilities                29886 non-null  float64
 52  hospital_beds_per_thousand            54798 non-null  float64
 53  life_expectancy                       61377 non-null  float64
 54  human_development_index               59329 non-null  float64
dtypes: datetime64[ns](1), float64(50), object(4)
```

```
dtypes: datetime64[ns](1), float64(50), object(4)
memory usage: 26.0+ MB
```

In [4]:

```python
from utils.mining_data_tb import countries
gbr = countries(df=world, code= "GBR")
prt = countries(df=world, code= "PRT")
ven = countries(df=world, code= "VEN")
tur = countries(df=world, code= "TUR")
esp = countries(df=world, code= "ESP")
gbr
```

Out[4]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_d
58315	GBR	Europe	United Kingdom	2020-01-31	2.0	2.0	NaN	NaN	NaN	
58316	GBR	Europe	United Kingdom	2020-02-01	2.0	0.0	NaN	NaN	NaN	
58317	GBR	Europe	United Kingdom	2020-02-02	2.0	0.0	NaN	NaN	NaN	
58318	GBR	Europe	United Kingdom	2020-02-03	8.0	6.0	NaN	NaN	NaN	
58319	GBR	Europe	United Kingdom	2020-02-04	8.0	0.0	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	
58670	GBR	Europe	United Kingdom	2021-01-20	3515796.0	38992.0	42120.429	93469.0	1826.0	
58671	GBR	Europe	United Kingdom	2021-01-21	3553773.0	37977.0	40573.714	94765.0	1296.0	
58672	GBR	Europe	United Kingdom	2021-01-22	3594094.0	40321.0	38350.286	96166.0	1401.0	
58673	GBR	Europe	United Kingdom	2021-01-23	3627746.0	33652.0	37239.429	97518.0	1352.0	
58674	GBR	Europe	United Kingdom	2021-01-24	3657857.0	30111.0	36016.714	98129.0	611.0	

360 rows × 55 columns

In [5]:

```python
from utils.mining_data_tb import column_erraser

gbr_dd = column_erraser(df=gbr, col1="date", col2='new_cases', col3="new_deaths")
prt_dd = column_erraser(df=prt, col1="date", col2='new_cases', col3="new_deaths")
ven_dd = column_erraser(df=ven, col1="date", col2='new_cases', col3="new_deaths")
tur_dd = column_erraser(df=tur, col1="date", col2='new_cases', col3="new_deaths")
esp_dd = column_erraser(df=esp, col1="date", col2='new_cases', col3="new_deaths")
```

In [5]:

```python
gbr_dd
```

Out[5]:

	date	new_cases	new_deaths
58315	2020-01-31	2.0	NaN
58316	2020-02-01	0.0	NaN
58317	2020-02-02	0.0	NaN
58318	2020-02-03	6.0	NaN

	date	new_cases	new_deaths
58319	2020-02-04	0.0	NaN
...	...	...	...
58670	2021-01-20	38992.0	1826.0
58671	2021-01-21	37977.0	1296.0
58672	2021-01-22	40321.0	1401.0
58673	2021-01-23	33652.0	1352.0
58674	2021-01-24	30111.0	611.0

360 rows × 3 columns

In [6]:

```
gbr_dd.dropna(inplace= True)
prt_dd.dropna( inplace= True)
ven_dd.dropna( inplace= True)
tur_dd.dropna( inplace= True)
esp_dd.dropna( inplace= True)
```

In [7]:

```
gbr_dd
```

Out[7]:

	date	new_cases	new_deaths
58350	2020-03-06	79.0	1.0
58351	2020-03-07	55.0	1.0
58352	2020-03-08	54.0	0.0
58353	2020-03-09	147.0	1.0
58354	2020-03-10	259.0	4.0
...	...	...	...
58670	2021-01-20	38992.0	1826.0
58671	2021-01-21	37977.0	1296.0
58672	2021-01-22	40321.0	1401.0
58673	2021-01-23	33652.0	1352.0
58674	2021-01-24	30111.0	611.0

325 rows × 3 columns

In [18]:

```
#Pregunta C8
gbr.corr()
prt.corr()
ven.corr()
tur.corr()
esp.corr()
```

Out[18]:

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_
total_cases	1.000000e+00	5.306517e-01	8.070279e-01	8.793744e-01	5.678626e-02	
new_cases	5.306517e-01	1.000000e+00	6.480925e-01	3.881492e-01	3.449407e-01	
new_cases_smoothed	8.070279e-01	6.480925e-01	1.000000e+00	6.128911e-01	2.063171e-01	
total_deaths	8.793744e-01	3.881492e-01	6.128911e-01	1.000000e+00	-1.186537e-01	
[illegible]	[illegible]	[illegible]	[illegible]	-1.186537e-	[illegible]	

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_
new_deaths	5.678626e-02	3.449407e-01	2.063171e-01	01	1.000000e+00	
new_deaths_smoothed	1.435459e-01	1.676450e-01	2.996779e-01	-1.084045e-01	7.224396e-01	
total_cases_per_million	1.000000e+00	5.306517e-01	8.070279e-01	8.793744e-01	5.678626e-02	
new_cases_per_million	5.306517e-01	1.000000e+00	6.480924e-01	3.881493e-01	3.449407e-01	
new_cases_smoothed_per_million	8.070280e-01	6.480924e-01	1.000000e+00	6.128912e-01	2.063171e-01	
total_deaths_per_million	8.793744e-01	3.881493e-01	6.128911e-01	1.000000e+00	-1.186538e-01	
new_deaths_per_million	5.678975e-02	3.449424e-01	2.063185e-01	-1.186484e-01	1.000000e+00	
new_deaths_smoothed_per_million	1.435458e-01	1.676428e-01	2.996795e-01	-1.084037e-01	7.224380e-01	
reproduction_rate	-2.704262e-01	-4.507082e-02	-1.512950e-01	-5.722246e-01	-4.038914e-02	
icu_patients	7.542774e-01	3.425082e-01	6.763201e-01	6.440188e-01	6.224790e-01	
icu_patients_per_million	7.542792e-01	3.425099e-01	6.763184e-01	6.440210e-01	6.224782e-01	
hosp_patients	5.838531e-01	4.455477e-01	8.511182e-01	4.489541e-01	6.179149e-01	
hosp_patients_per_million	5.838531e-01	4.455474e-01	8.511181e-01	4.489542e-01	6.179150e-01	
weekly_icu_admissions	-1.034623e-02	3.329079e-01	1.568608e-01	-4.810924e-01	3.140879e-01	
weekly_icu_admissions_per_million	-1.033943e-02	3.329056e-01	1.568674e-01	-4.810873e-01	3.140857e-01	
weekly_hosp_admissions	1.560007e-02	4.971079e-01	2.239458e-01	-4.724676e-01	5.053663e-01	
weekly_hosp_admissions_per_million	1.559985e-02	4.971079e-01	2.239456e-01	-4.724678e-01	5.053663e-01	
total_tests	9.935334e-01	8.079092e-01	7.706481e-01	9.766989e-01	3.465857e-01	
new_tests	NaN	NaN	NaN	NaN	NaN	
total_tests_per_thousand	9.935333e-01	8.079097e-01	7.706486e-01	9.766988e-01	3.465859e-01	
new_tests_per_thousand	NaN	NaN	NaN	NaN	NaN	
new_tests_smoothed	8.506193e-01	5.695179e-01	9.376576e-01	7.644713e-01	2.675102e-01	
new_tests_smoothed_per_thousand	8.506225e-01	5.694967e-01	9.376329e-01	7.644788e-01	2.675182e-01	
positive_rate	3.259592e-01	3.529368e-01	5.685625e-01	2.010074e-01	2.973048e-01	
tests_per_case	-3.352373e-01	-2.581484e-01	-4.299592e-01	-2.933662e-01	-1.069640e-01	
total_vaccinations	9.962167e-01	3.741305e-01	9.846909e-01	9.900327e-01	5.333185e-01	
new_vaccinations	-1.315952e-01	1.886223e-01	7.621172e-02	-1.466593e-01	-6.665631e-01	
new_vaccinations_smoothed	6.672829e-01	1.935738e-01	7.559267e-01	6.278346e-01	1.513558e-01	
total_vaccinations_per_hundred	9.961878e-01	3.744837e-01	9.846397e-01	9.898782e-01	5.339127e-01	
new_vaccinations_smoothed_per_million	6.673870e-01	1.940945e-01	7.559833e-01	6.279747e-01	1.518692e-01	
stringency_index	3.689012e-01	2.216712e-01	3.386774e-01	2.378178e-01	3.981147e-01	
population	NaN	NaN	NaN	NaN	NaN	
population_density	1.457344e-15	2.342494e-15	1.116635e-15	-3.433818e-15	-4.037768e-16	
median_age	NaN	NaN	NaN	NaN	NaN	
aged_65_older	3.290689e-15	2.735455e-18	1.116635e-15	-3.005899e-16	5.467811e-16	
aged_70_older	4.853011e-15	-1.243446e-15	-1.772436e-16	8.803784e-16	3.148134e-16	
gdp_per_capita	3.290689e-15	2.735455e-18	-1.772436e-16	-2.557054e-	4.343872e-16	

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_
extreme_poverty	NaN	NaN	NaN	NaN	NaN	
cardiovasc_death_rate	4.271298e-16	1.818611e-15	4.070039e-16	2.007449e-15	6.529491e-16	
diabetes_prevalence	-3.145630e-15	-1.593129e-15	-4.741268e-16	1.404248e-15	6.806264e-16	
female_smokers	3.290689e-15	2.735455e-18	6.255658e-17	1.516406e-15	-3.594241e-16	
male_smokers	3.604038e-15	-1.850639e-15	-1.909256e-15	-3.425526e-15	3.386764e-16	
handwashing_facilities	NaN	NaN	NaN	NaN	NaN	
hospital_beds_per_thousand	2.769411e-15	1.616520e-15	-7.877495e-18	-8.803784e-16	-3.148134e-16	
life_expectancy	3.700407e-15	7.547041e-16	-5.812549e-16	2.667679e-15	4.374249e-16	
human_development_index	-4.853011e-15	1.243446e-15	-8.271370e-16	-3.378236e-15	-2.226180e-16	

50 rows × 50 columns

In [8]:

```
#Pregunta A2
worlda_2 = world.drop(['iso_code','continent','location','date', 'aged_65_older', 'tests
_units'],1)
from utils.mining_data_tb import get_top_abs_correlations
get_top_abs_correlations(df=worlda_2, n=7)
```

Out[8]:

```
new_cases           new_cases_smoothed     0.990950
total_cases         total_deaths           0.977766
new_cases_smoothed  total_deaths           0.976873
icu_patients        hosp_patients          0.975158
new_tests           new_tests_smoothed     0.975068
new_deaths          new_deaths_smoothed    0.973488
total_cases         new_cases_smoothed     0.970599
dtype: float64
```

In [9]:

```
worlda_2cor = column_erraser(df= worlda_2, col1='new_cases', col2='new_cases_smoothed',
col3='total_deaths', col4='total_cases', col5='new_tests', col6='new_tests_smoothed', col
7= "icu_patients", col8='hosp_patients', col9='new_deaths', col10='new_deaths_smoothed')
worlda_2cor.corr()
```

Out[9]:

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	icu_
total_cases	1.000000	0.958567	0.970599	0.977766	0.883569	0.906525	0.
new_cases	0.958567	1.000000	0.990950	0.967305	0.933366	0.934160	0.
new_cases_smoothed	0.970599	0.990950	1.000000	0.976873	0.922549	0.944571	0.
total_deaths	0.977766	0.967305	0.976873	1.000000	0.919862	0.944834	0.
new_deaths	0.883569	0.933366	0.922549	0.919862	1.000000	0.973488	0.
new_deaths_smoothed	0.906525	0.934160	0.944571	0.944834	0.973488	1.000000	0.
icu_patients	0.882391	0.890198	0.909069	0.881514	0.915093	0.962369	1.
hosp_patients	0.884858	0.897988	0.915676	0.900303	0.913121	0.956632	0.
new_tests	0.883138	0.857692	0.857943	0.846004	0.701788	0.718675	0.
new_tests_smoothed	0.902680	0.853262	0.870133	0.847494	0.664222	0.713038	0.

In [7]:

```
#Pregunta A.4.b - Ejecutado debajo de este código en la sección Visualization Javi.
from utils.mining_data_tb import A_4_b
gbr1M = A_4_b(df= gbr_dd, column='date', key= "date", freq= '1M')
```

In [8]:

```
gbr10D = A_4_b(df= gbr_dd, column='date', key= "date", freq= '10D')
```

In [25]:

```
prt1M = A_4_b(df= prt_dd, column='date', key= "date", freq= '1M')
```

In [26]:

```
prt10D = A_4_b(df= prt_dd, column='date', key= "date", freq= '10D')
```

In [27]:

```
ven1M = A_4_b(df= ven_dd, column='date', key= "date", freq= '1M')
```

In [28]:

```
ven10D = A_4_b(df= ven_dd, column='date', key= "date", freq= '10D')
```

In [29]:

```
tur1M = A_4_b(df= tur_dd, column='date', key= "date", freq= '1M')
```

In [30]:

```
tur10D = A_4_b(df= tur_dd, column='date', key= "date", freq= '10D')
```

In [31]:

```
esp1M = A_4_b(df= esp_dd, column='date', key= "date", freq= '1M')
```

In [32]:

```
esp10D = A_4_b(df= esp_dd, column='date', key= "date", freq= '10D')
```

Visualization Javi

In [19]:

```
from utils.visualization_tb import b_2_line
gbrb_2_line = b_2_line(df= gbr_dd, x= gbr_dd['date'], y=gbr_dd.columns, title= 'GBR covi
d deaths')
```

In [20]:

```
from utils.visualization_tb import b_2_bar
gbrb_2_bar = b_2_bar(df= gbr_dd, x= gbr_dd['date'], y=gbr_dd.columns, title= 'GBR covid
deaths')
```

In [21]:

```
from utils.visualization_tb import b_2_dots
gbrb_2_dots = b_2_dots(df= gbr_dd, x= gbr_dd['date'], y=gbr_dd.columns, title= 'GBR covi
d deaths')
```

In [33]:

```
prtb_2_line = b_2_line(df= prt_dd, x= prt_dd['date'], y=prt_dd.columns, title= 'PRT covi
d deaths')
```

In [37]:

```
prtb_2_bar = b_2_bar(df= prt_dd, x= prt_dd['date'], y=prt_dd.columns, title= 'PRT covid
deaths')
```

In [38]:

```
prtb_2_dots = b_2_dots(df= prt_dd, x= prt_dd['date'], y=prt_dd.columns, title= 'PRT covi
d deaths')
```

In [39]:

```
venb_2_line = b_2_line(df= ven_dd, x= ven_dd['date'], y=ven_dd.columns, title= 'VEN covi
d deaths')
```

In [40]:

```
venb_2_bar = b_2_bar(df= ven_dd, x= ven_dd['date'], y=ven_dd.columns, title= 'VEN covid
deaths')
```

In [42]:

```
venb_2_dots = b_2_dots(df= ven_dd, x= ven_dd['date'], y=ven_dd.columns, title= 'VEN covi
d deaths')
```

In [43]:

```
turb_2_line = b_2_line(df= tur_dd, x= tur_dd['date'], y=tur_dd.columns, title= 'TUR covi
d deaths')
```

In [44]:

```
turb_2_bar = b_2_bar(df= tur_dd, x= tur_dd['date'], y=tur_dd.columns, title= 'TUR covid
deaths')
```

In [45]:

```
turb_2_dots = b_2_dots(df= tur_dd, x= tur_dd['date'], y=tur_dd.columns, title= 'TUR covi
d deaths')
```

In [46]:

```
espb_2_line = b_2_line(df= esp_dd, x= esp_dd['date'], y=esp_dd.columns, title= 'ESP covi
d deaths')
```

In [47]:

```
espb_2_bar = b_2_bar(df= esp_dd, x= esp_dd['date'], y=esp_dd.columns, title= 'ESP covid
deaths')
```

In [48]:

```
espb_2_dots = b_2_dots(df= esp_dd, x= esp_dd['date'], y=esp_dd.columns, title= 'ESP covi
d deaths')
```

In [13]:

```
sns.heatmap(worlda_2cor.corr())
```

Out[13]:

```
<AxesSubplot:>
```

In [32]:

```
gbrb1M_line = b_2_line(df= gbr1M, x= gbr1M.index, y=gbr1M.columns, title= 'GBR 1M covid
deaths')
```

In [33]:

In [22]:

```
gbr10D_line = b_2_line(df= gbr10D, x= gbr10D.index, y=gbr10D.columns, title= 'GBR 10D co
vid deaths')
```

In [33]:

```
prt1M_line = b_2_line(df= prt1M, x= prt1M.index, y=prt1M.columns, title= 'PRT 1M covid d
eaths')
```

In [34]:

```
prt10D_line = b_2_line(df= prt10D, x= prt10D.index, y=prt10D.columns, title= 'PRT 10D co
vid deaths')
```

In [35]:

```
ven1M_line = b_2_line(df= ven1M, x= ven1M.index, y=ven1M.columns, title= 'VEN 1M covid d
eaths')
```

In [36]:

```
ven10D_line = b_2_line(df= ven10D, x= ven10D.index, y=ven10D.columns, title= 'VEN 10D co
vid deaths')
```

In [37]:

```
tur1M_line = b_2_line(df= tur1M, x= tur1M.index, y=tur1M.columns, title= 'TUR 1M covid d
eaths')
```

In [38]:

```
tur10D_line = b_2_line(df= tur10D, x= tur10D.index, y=tur10D.columns, title= 'TUR 10D co
vid deaths')
```

In [39]:

```
esp1M_line = b_2_line(df= esp1M, x= esp1M.index, y=esp1M.columns, title= 'ESP 1M covid d
eaths')
```

In [40]:

```
esp10D_line = b_2_line(df= esp10D, x= esp10D.index, y=esp10D.columns, title= 'ESP 10D co
vid deaths')
```

Visualization Ariadna

In [3]:

```
#Is the second way: this plot is showing that the 75% of the global Countries about total
deaths are between 250K and more 500K.

from utils.visualization_tb import C10c

C10c(df=world)
```

```
<Figure size 432x288 with 0 Axes>
```

In [3]:

```
#In this graphic is representing total cases and total deaths, so this data it's the tota
l since the covid-19 begun. United Kingdom and Spain are in top 10 Ranking of the worst c
ountries that didn't manage very well.

from utils.visualization_tb import C10a

C10a(df=world)
```

```
<Figure size 432x288 with 0 Axes>
```

In [3]:

```
"""In this graphic has been thought with new cases and new deaths because they are not ac
cumulative"""
#It is showing the position of our countries: Portugal, Venezuela, Turkey, UK and Spain i
n general ranking in over the world. Venezuela is the first country with less new cases a
nd new deaths being 116 position in new cases and 106 in new deaths respect 190 countries
total. Instead of the last country is United Kingdom in 186 position in new cases and dea
ths. Spain is the second last country being 182/190 and 180/190 in new cases and deaths r
espectively.

from utils.visualization_tb import position

position(df=world, col1="iso_code", col2="date", col3="new_cases", col4="new_deaths", co
l5="continent", col6="location", col7="tests_units")
```

```
<Figure size 432x288 with 0 Axes>
```

Visualization Group C's Data - Total Cases

Visualization Ariadna

In [4]:

```
#This graphic from Group C is representing how the cases has gone increased during all Co
ronavirus pandemic.
from utils.visualization_tb import groupC_plot

groupC_plot(df=world)
```

```
<Figure size 432x288 with 0 Axes>
```

In [ ]: