data understanding

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
% matplotlib inline
sns.set()
```

```
UsageError: Line magic function `%` not found.
```

In [ ]:
```
df=pd.read_csv('https://covid.ourworldindata.org/data/owid-covid-data.csv')
df
```

In [39]:
```
df.head()
```

Out[39]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_dea
0	AFG	Asia	Afghanistan	2020-02-24	5.0	5.0	NaN	N
1	AFG	Asia	Afghanistan	2020-02-25	5.0	0.0	NaN	N
2	AFG	Asia	Afghanistan	2020-02-26	5.0	0.0	NaN	N
3	AFG	Asia	Afghanistan	2020-02-27	5.0	0.0	NaN	N
4	AFG	Asia	Afghanistan	2020-02-28	5.0	0.0	NaN	N

5 rows × 67 columns

In [4]:
```
df.tail()
```

Out[4]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total
189950	ZWE	Africa	Zimbabwe	2022-05-24	250929.0	227.0	168.714	
189951	ZWE	Africa	Zimbabwe	2022-05-25	251228.0	299.0	174.429	
189952	ZWE	Africa	Zimbabwe	2022-05-26	251646.0	418.0	205.714	
189953	ZWE	Africa	Zimbabwe	2022-05-27	251959.0	313.0	212.857	
189954	ZWE	Africa	Zimbabwe	2022-05-28	251959.0	NaN	NaN	

5 rows × 67 columns

In [40]:
```
df.columns
```

Out[40]:
```
Index(['iso_code', 'continent', 'location', 'date', 'total_cases', 'new_cases',
       'new_cases_smoothed', 'total_deaths', 'new_deaths',
       'new_deaths_smoothed', 'total_cases_per_million',
       'new_cases_per_million', 'new_cases_smoothed_per_million',
       'total_deaths_per_million', 'new_deaths_per_million',
       'new_deaths_smoothed_per_million', 'reproduction_rate', 'icu_patients',
       'icu_patients_per_million', 'hosp_patients',
       'hosp_patients_per_million', 'weekly_icu_admissions',
       'weekly_icu_admissions_per_million', 'weekly_hosp_admissions',
       'weekly_hosp_admissions_per_million', 'total_tests', 'new_tests',
       'total_tests_per_thousand', 'new_tests_per_thousand',
       'new_tests_smoothed', 'new_tests_smoothed_per_thousand',
       'positive_rate', 'tests_per_case', 'tests_units', 'total_vaccinations',
       'people_vaccinated', 'people_fully_vaccinated', 'total_boosters',
       'new_vaccinations', 'new_vaccinations_smoothed',
       'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred',
       'people_fully_vaccinated_per_hundred', 'total_boosters_per_hundred',
       'new_vaccinations_smoothed_per_million',
       'new_people_vaccinated_smoothed',
       'new_people_vaccinated_smoothed_per_hundred', 'stringency_index',
       'population', 'population_density', 'median_age', 'aged_65_older',
       'aged_70_older', 'gdp_per_capita', 'extreme_poverty',
       'cardiovasc_death_rate', 'diabetes_prevalence', 'female_smokers',
       'male_smokers', 'handwashing_facilities', 'hospital_beds_per_thousand',
       'life_expectancy', 'human_development_index',
       'excess_mortality_cumulative_absolute', 'excess_mortality_cumulative',
       'excess_mortality', 'excess_mortality_cumulative_per_million'],
      dtype='object')
```

In [7]:
```
df.shape
```

Out[7]:
```
(188799, 67)
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188799 entries, 0 to 188798
Data columns (total 67 columns):
 #   Column                                    Non-Null Count   Dtype  
---  ------                                    --------------   -----  
 0   iso_code                                  188799 non-null  object 
 1   continent                                 177816 non-null  object 
 2   location                                  188799 non-null  object 
 3   date                                      188799 non-null  object 
 4   total_cases                               181315 non-null  float64
 5   new_cases                                 181041 non-null  float64
 6   new_cases_smoothed                        179867 non-null  float64
 7   total_deaths                              162915 non-null  float64
 8   new_deaths                                162903 non-null  float64
 9   new_deaths_smoothed                       161741 non-null  float64
 10  total_cases_per_million                   180478 non-null  float64
 11  new_cases_per_million                     180204 non-null  float64
 12  new_cases_smoothed_per_million            179035 non-null  float64
 13  total_deaths_per_million                  162091 non-null  float64
 14  new_deaths_per_million                    162079 non-null  float64
 15  new_deaths_smoothed_per_million           160922 non-null  float64
 16  reproduction_rate                         140710 non-null  float64
 17  icu_patients                              25192 non-null   float64
 18  icu_patients_per_million                  25192 non-null   float64
 19  hosp_patients                             26442 non-null   float64
 20  hosp_patients_per_million                 26442 non-null   float64
 21  weekly_icu_admissions                     6123 non-null    float64
 22  weekly_icu_admissions_per_million         6123 non-null    float64
 23  weekly_hosp_admissions                    12217 non-null   float64
 24  weekly_hosp_admissions_per_million        12217 non-null   float64
 25  total_tests                               76851 non-null   float64
 26  new_tests                                 73800 non-null   float64
 27  total_tests_per_thousand                  76851 non-null   float64
 28  new_tests_per_thousand                    73800 non-null   float64
 29  new_tests_smoothed                        99393 non-null   float64
 30  new_tests_smoothed_per_thousand           99393 non-null   float64
 31  positive_rate                             91663 non-null   float64
 32  tests_per_case                            90135 non-null   float64
 33  tests_units                               102220 non-null  object 
 34  total_vaccinations                        51326 non-null   float64
 35  people_vaccinated                         48884 non-null   float64
 36  people_fully_vaccinated                   46331 non-null   float64
 37  total_boosters                            23498 non-null   float64
 38  new_vaccinations                          41986 non-null   float64
 39  new_vaccinations_smoothed                 101111 non-null  float64
 40  total_vaccinations_per_hundred            51326 non-null   float64
 41  people_vaccinated_per_hundred             48884 non-null   float64
 42  people_fully_vaccinated_per_hundred       46331 non-null   float64
 43  total_boosters_per_hundred                23498 non-null   float64
 44  new_vaccinations_smoothed_per_million     101111 non-null  float64
 45  new_people_vaccinated_smoothed            100109 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred  100109 non-null  float64
 47  stringency_index                          147306 non-null  float64
 48  population                                187645 non-null  float64
 49  population_density                        168240 non-null  float64
```

```
 50  median_age                                  155929 non-null  float64
 51  aged_65_older                               154276 non-null  float64
 52  aged_70_older                               155110 non-null  float64
 53  gdp_per_capita                              155097 non-null  float64
 54  extreme_poverty                             101258 non-null  float64
 55  cardiovasc_death_rate                       155580 non-null  float64
 56  diabetes_prevalence                         163179 non-null  float64
 57  female_smokers                              117673 non-null  float64
 58  male_smokers                                116060 non-null  float64
 59  handwashing_facilities                      76433 non-null   float64
 60  hospital_beds_per_thousand                  138044 non-null  float64
 61  life_expectancy                             176559 non-null  float64
 62  human_development_index                     151553 non-null  float64
 63  excess_mortality_cumulative_absolute        6525 non-null    float64
 64  excess_mortality_cumulative                 6525 non-null    float64
 65  excess_mortality                            6525 non-null    float64
 66  excess_mortality_cumulative_per_million     6525 non-null    float64
dtypes: float64(62), object(5)
memory usage: 96.5+ MB
```

In [41]:
```
df.describe()
```

Out[41]:

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_death
count	1.824600e+05	1.822020e+05	1.810280e+05	1.640300e+05	164004.000000	16
mean	3.247384e+06	1.236697e+04	1.240822e+04	6.433518e+04	158.998159	
std	2.038470e+07	8.748121e+04	8.594522e+04	3.353773e+05	793.254371	
min	1.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00	0.000000	
25%	2.642000e+03	0.000000e+00	6.714000e+00	9.300000e+01	0.000000	
50%	3.491950e+04	7.100000e+01	1.012860e+02	9.100000e+02	1.000000	
75%	3.747608e+05	1.034000e+03	1.141714e+03	8.383000e+03	18.000000	
max	5.287209e+08	4.079827e+06	3.437214e+06	6.287246e+06	18151.000000	1

8 rows × 62 columns

exploring world data

listing all countries in our data

In [44]:
```
df['location'].unique()
```

Out[44]:
```
array(['Afghanistan', 'Africa', 'Albania', 'Algeria', 'Andorra', 'Angola',
       'Anguilla', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Aruba',
       'Asia', 'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
       'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
       'Bermuda', 'Bhutan', 'Bolivia', 'Bonaire Sint Eustatius and Saba',
       'Bosnia and Herzegovina', 'Botswana', 'Brazil',
       'British Virgin Islands', 'Brunei', 'Bulgaria', 'Burkina Faso',
       'Burundi', 'Cambodia', 'Cameroon', 'Canada', 'Cape Verde',
       'Cayman Islands', 'Central African Republic', 'Chad', 'Chile',
       'China', 'Colombia', 'Comoros', 'Congo', 'Cook Islands',
       'Costa Rica', "Cote d'Ivoire", 'Croatia', 'Cuba', 'Curacao',
       'Cyprus', 'Czechia', 'Democratic Republic of Congo', 'Denmark',
       'Djibouti', 'Dominica', 'Dominican Republic', 'Ecuador', 'Egypt',
       'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia',
       'Eswatini', 'Ethiopia', 'Europe', 'European Union',
       'Faeroe Islands', 'Falkland Islands', 'Fiji', 'Finland', 'France',
       'French Polynesia', 'Gabon', 'Gambia', 'Georgia', 'Germany',
       'Ghana', 'Gibraltar', 'Greece', 'Greenland', 'Grenada', 'Guam',
       'Guatemala', 'Guernsey', 'Guinea', 'Guinea-Bissau', 'Guyana',
       'Haiti', 'High income', 'Honduras', 'Hong Kong', 'Hungary',
       'Iceland', 'India', 'Indonesia', 'International', 'Iran', 'Iraq',
       'Ireland', 'Isle of Man', 'Israel', 'Italy', 'Jamaica', 'Japan',
       'Jersey', 'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati', 'Kosovo',
       'Kuwait', 'Kyrgyzstan', 'Laos', 'Latvia', 'Lebanon', 'Lesotho',
       'Liberia', 'Libya', 'Liechtenstein', 'Lithuania', 'Low income',
       'Lower middle income', 'Luxembourg', 'Macao', 'Madagascar',
       'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta',
       'Marshall Islands', 'Mauritania', 'Mauritius', 'Mexico',
       'Micronesia (country)', 'Moldova', 'Monaco', 'Mongolia',
       'Montenegro', 'Montserrat', 'Morocco', 'Mozambique', 'Myanmar',
       'Namibia', 'Nauru', 'Nepal', 'Netherlands', 'New Caledonia',
       'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'Niue',
       'North America', 'North Korea', 'North Macedonia',
       'Northern Cyprus', 'Northern Mariana Islands', 'Norway', 'Oceania',
       'Oman', 'Pakistan', 'Palau', 'Palestine', 'Panama',
       'Papua New Guinea', 'Paraguay', 'Peru', 'Philippines', 'Pitcairn',
       'Poland', 'Portugal', 'Puerto Rico', 'Qatar', 'Romania', 'Russia',
       'Rwanda', 'Saint Helena', 'Saint Kitts and Nevis', 'Saint Lucia',
       'Saint Pierre and Miquelon', 'Saint Vincent and the Grenadines',
       'Samoa', 'San Marino', 'Sao Tome and Principe', 'Saudi Arabia',
       'Senegal', 'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore',
       'Sint Maarten (Dutch part)', 'Slovakia', 'Slovenia',
       'Solomon Islands', 'Somalia', 'South Africa', 'South America',
       'South Korea', 'South Sudan', 'Spain', 'Sri Lanka', 'Sudan',
       'Suriname', 'Sweden', 'Switzerland', 'Syria', 'Taiwan',
       'Tajikistan', 'Tanzania', 'Thailand', 'Timor', 'Togo', 'Tokelau',
       'Tonga', 'Trinidad and Tobago', 'Tunisia', 'Turkey',
       'Turkmenistan', 'Turks and Caicos Islands', 'Tuvalu', 'Uganda',
       'Ukraine', 'United Arab Emirates', 'United Kingdom',
       'United States', 'United States Virgin Islands',
       'Upper middle income', 'Uruguay', 'Uzbekistan', 'Vanuatu',
       'Vatican', 'Venezuela', 'Vietnam', 'Wallis and Futuna',
       'Western Sahara', 'World', 'Yemen', 'Zambia', 'Zimbabwe'],
      dtype=object)
```

In [45]:
```
df['location'].nunique()
```

Out[45]:
```
244
```

Selecting the 'World' data

In [46]:
```
df_world = df[df.location == 'World']
df_world
```

Out[46]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	tot
186716	OWID_WRL	NaN	World	2020-01-22	557.0	0.0	NaN	
186717	OWID_WRL	NaN	World	2020-01-23	657.0	100.0	NaN	
186718	OWID_WRL	NaN	World	2020-01-24	944.0	287.0	NaN	
186719	OWID_WRL	NaN	World	2020-01-25	1437.0	493.0	NaN	
186720	OWID_WRL	NaN	World	2020-01-26	2120.0	683.0	NaN	
...	...	...	...	...	...	...	...	
187569	OWID_WRL	NaN	World	2022-05-24	526707203.0	611748.0	539597.429	
187570	OWID_WRL	NaN	World	2022-05-25	527349857.0	642654.0	525451.571	
187571	OWID_WRL	NaN	World	2022-05-26	527839364.0	494015.0	511715.429	
187572	OWID_WRL	NaN	World	2022-05-27	528431653.0	592289.0	509904.571	
187573	OWID_WRL	NaN	World	2022-05-28	528720932.0	289279.0	497477.000	

858 rows × 67 columns

Finding the date of the maximum number of deaths all over the world

In [47]:
```
df_world[df_world['new_deaths'] ==df_world['new_deaths'] .max()]['date']
```

Out[47]:
```
187080    2021-01-20
Name: date, dtype: object
```

Creating a summary table for the most recent date all over the world

In [48]:
```
df_world_recent=df_world[df_world['date']==df_world['date'].max()]
df_world_recent
```

Out[48]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	tot
187573	OWID_WRL	NaN	World	2022-05-28	528720932.0	289279.0	497477.0	

1 rows × 67 columns

In [40]:
```
df_world_recent.groupby('date')[['total_cases','new_cases','new_deaths','total_de
```

Out[40]:

date	total_cases	new_cases	new_deaths	total_deaths
2022-05-23	526091236.0	511252.0	1436.0	6278676.0

Calculating the percentage of confirmed cases regarding the world population

In [41]:
```
df_world_ratio= df_world['total_cases'].max()/ df_world['population'].max()
df_world_ratio
```

Out[41]:
```
0.0668055270381475
```

VISUALLIZING WORLD DATA

Confirmed cases (Total Cases) all over the world

In [14]:
```
df_world_over_time=df_world.groupby(["date"])[['total_cases','new_cases','new_dea
df_world_over_time
```

Out[14]:

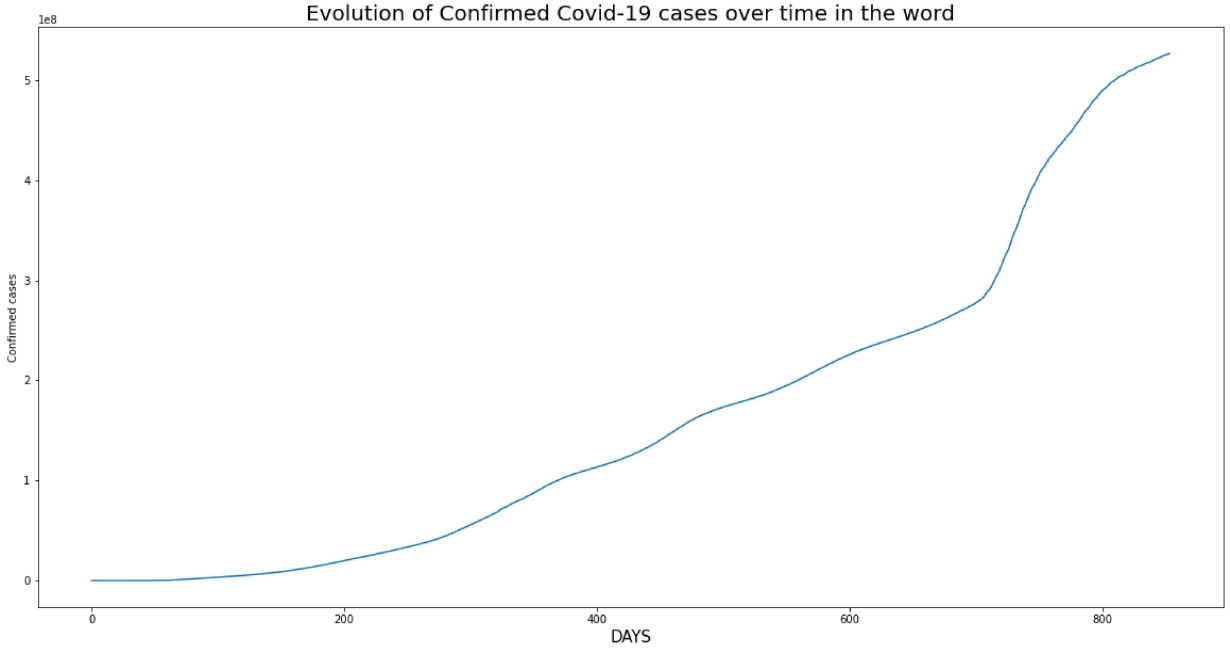
	date	total_cases	new_cases	new_deaths	total_deaths
0	2020-01-22	557.0	0.0	0.0	17.0
1	2020-01-23	657.0	100.0	1.0	18.0
2	2020-01-24	944.0	287.0	8.0	26.0
3	2020-01-25	1437.0	493.0	16.0	42.0
4	2020-01-26	2120.0	683.0	14.0	56.0
...	...	...	...	...	...
849	2022-05-20	524896482.0	573104.0	1972.0	6275490.0
850	2022-05-21	525272754.0	376272.0	1117.0	6276605.0
851	2022-05-22	525610716.0	337962.0	640.0	6277245.0
852	2022-05-23	526095455.0	514392.0	1432.0	6278676.0
853	2022-05-24	526707203.0	611748.0	1753.0	6280424.0

854 rows × 5 columns

Using line-plot

In [18]:
```
plt.figure(figsize=(20,10))
plt.plot(df_world_over_time.index,df_world_over_time['total_cases'])
plt.title('Evolution of Confirmed Covid-19 cases over time in the world',fontsize
plt.xlabel('DAYS',fontsize=15)
plt.ylabel('Confirmed cases',fontsize=10)
```

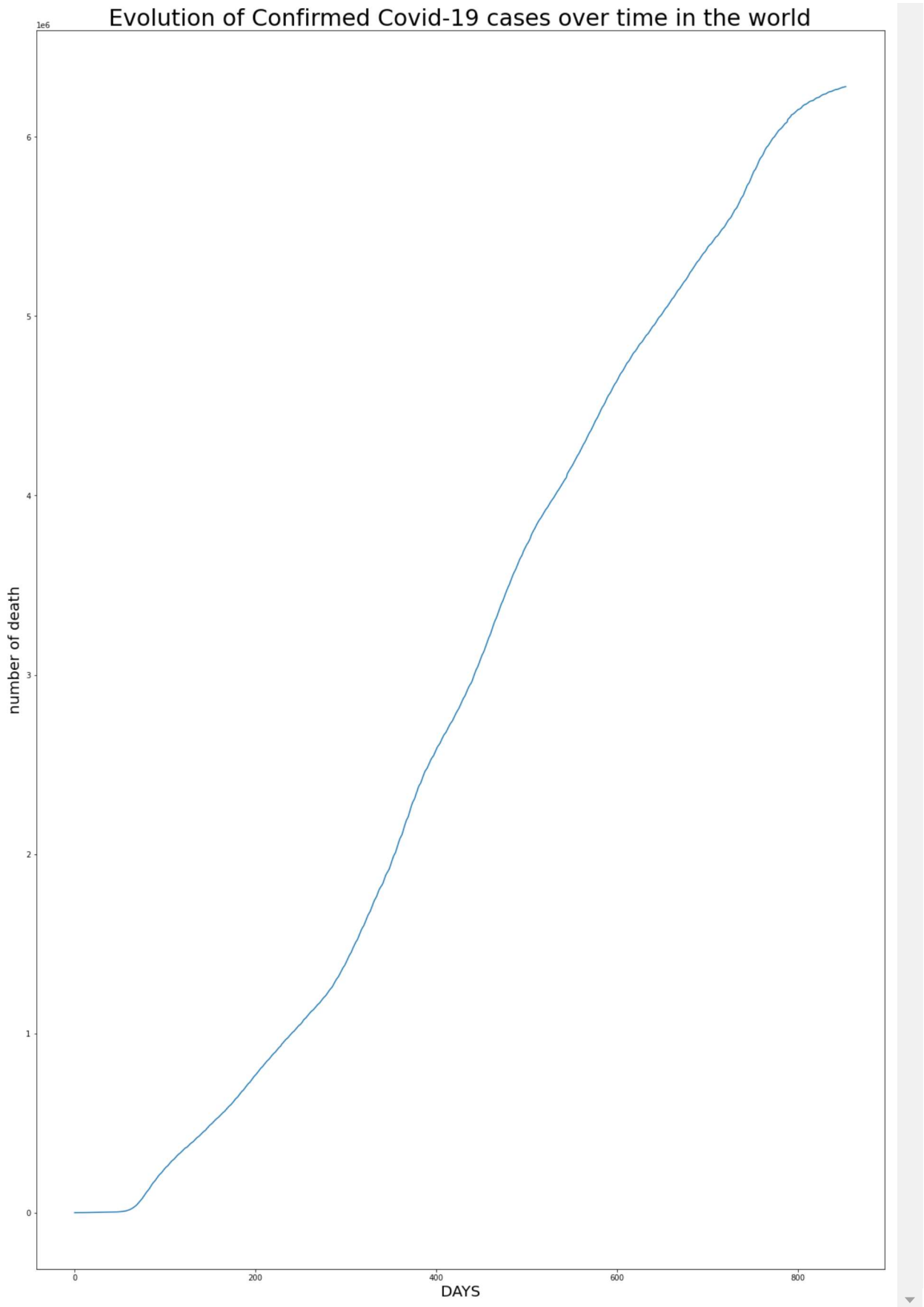
Out[18]:
```
Text(0, 0.5, 'Confirmed cases')
```



Total deaths cases evolution over time

In [22]:
```
plt.figure(figsize=(20,30))
plt.plot(df_world_over_time.index,df_world_over_time['total_deaths'])
plt.title('Evolution of Confirmed Covid-19 cases over time in the world',fontsize
plt.xlabel('DAYS',fontsize=20)
plt.ylabel('number of death',fontsize=20)
```

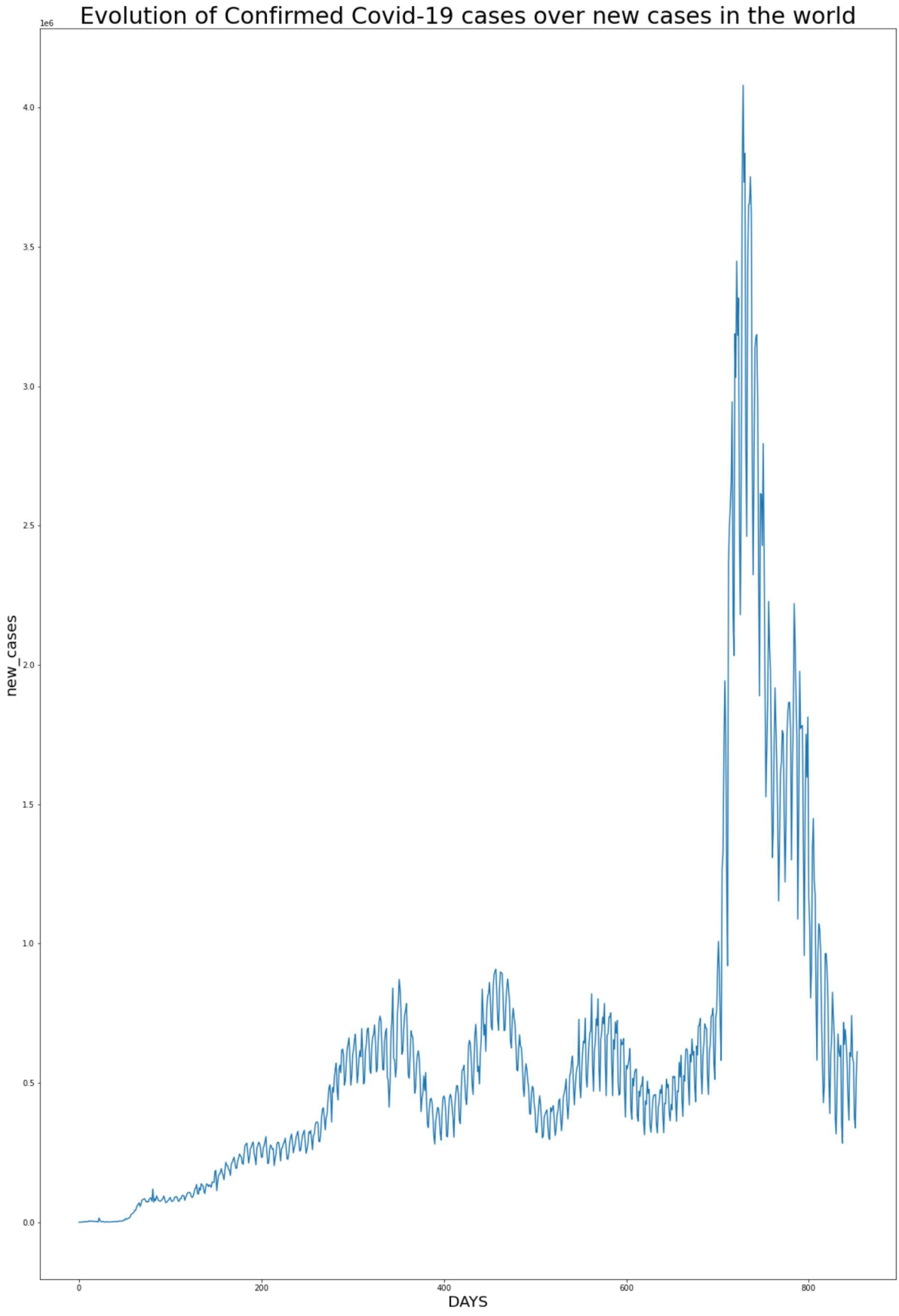
Out[22]:
```
Text(0, 0.5, 'number of death')
```



New cases all over the world

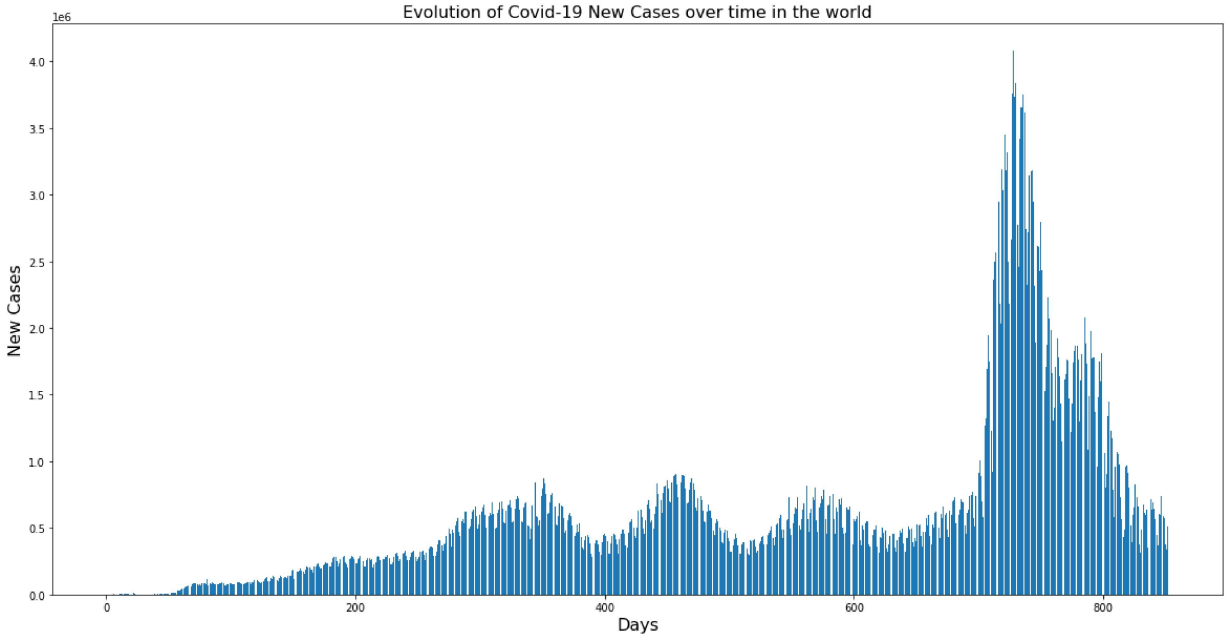
In [23]:
```
plt.figure(figsize=(20,30))
plt.plot(df_world_over_time.index,df_world_over_time['new_cases'])
plt.title('Evolution of Confirmed Covid-19 cases over new cases in the world',for
plt.xlabel('DAYS',fontsize=20)
plt.ylabel('new_cases',fontsize=20)
```

Out[23]:
```
Text(0, 0.5, 'new_cases')
```

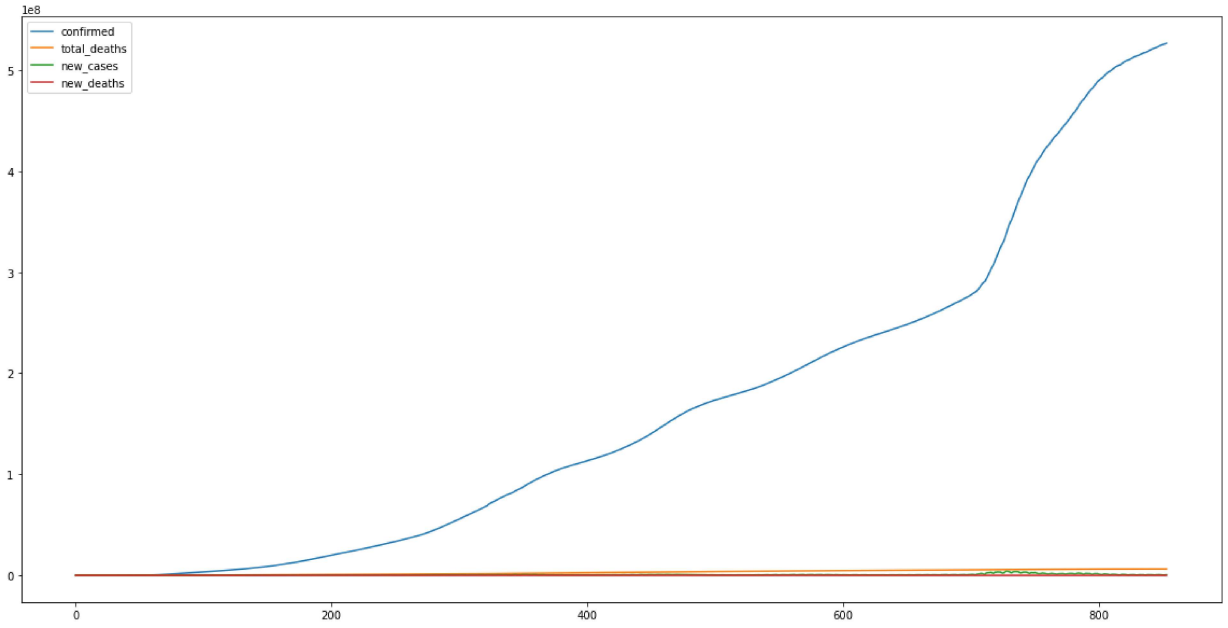

Evolution of Confirmed Covid-19 cases over new cases in the world
1e6
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
new_cases
0
200
400
600
800
DAYS

In [24]:
```
plt.figure(figsize=(20,10))
plt.bar(df_world_over_time.index, df_world_over_time['new_cases'])
plt.title('Evolution of Covid-19 New Cases over time in the world', fontsize=16)
plt.xlabel('Days', fontsize=16)
plt.ylabel('New Cases', fontsize=16)
```

Out[24]:
```
Text(0, 0.5, 'New Cases')
```



In [27]:
```
plt.figure(figsize=(20,10))
plt.plot(df_world_over_time.index,df_world_over_time['total_cases'],label='confir
plt.plot(df_world_over_time.index,df_world_over_time['total_deaths'],label='total
plt.plot(df_world_over_time.index,df_world_over_time['new_cases'],label='new_case
plt.plot(df_world_over_time.index,df_world_over_time['new_deaths'],label='new_dea
plt.legend(loc=0)
plt.show()
```



EXPLORING EGYPT DATA

Let's first get 'Egypt' data

In [49]:
```
df_egypt = df[df['location'] == 'Egypt']
df_egypt
```

Out[49]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_d
49335	EGY	Africa	Egypt	2020-02-14	1.0	1.0	NaN	
49336	EGY	Africa	Egypt	2020-02-15	1.0	0.0	NaN	
49337	EGY	Africa	Egypt	2020-02-16	1.0	0.0	NaN	
49338	EGY	Africa	Egypt	2020-02-17	1.0	0.0	NaN	
49339	EGY	Africa	Egypt	2020-02-18	1.0	0.0	NaN	
...	...	...	...	...	...	...	...	
50165	EGY	Africa	Egypt	2022-05-24	515645.0	0.0	0.0	24
50166	EGY	Africa	Egypt	2022-05-25	515645.0	0.0	0.0	24
50167	EGY	Africa	Egypt	2022-05-26	515645.0	0.0	0.0	24
50168	EGY	Africa	Egypt	2022-05-27	515645.0	0.0	0.0	24
50169	EGY	Africa	Egypt	2022-05-28	515645.0	0.0	0.0	24

835 rows × 67 columns

Creating a summary table for the most recent 'Egypt' data

In [51]:
```
df_egypt[['total_cases','new_cases','total_deaths','new_deaths','date']]
```

Out[51]:

	total_cases	new_cases	total_deaths	new_deaths	date
49335	1.0	1.0	NaN	NaN	2020-02-14
49336	1.0	0.0	NaN	NaN	2020-02-15
49337	1.0	0.0	NaN	NaN	2020-02-16
49338	1.0	0.0	NaN	NaN	2020-02-17
49339	1.0	0.0	NaN	NaN	2020-02-18
...	...	...	...	...	...
50165	515645.0	0.0	24704.0	NaN	2022-05-24
50166	515645.0	0.0	24704.0	NaN	2022-05-25
50167	515645.0	0.0	24704.0	NaN	2022-05-26
50168	515645.0	0.0	24704.0	NaN	2022-05-27
50169	515645.0	0.0	24704.0	NaN	2022-05-28

835 rows × 5 columns

In [52]:
```
df_egypt_recent=df_egypt[df_egypt['date'] == df_egypt.date.max()]
df_egypt_recent
```

Out[52]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_d
50169	EGY	Africa	Egypt	2022-05-28	515645.0	0.0	0.0	24

1 rows × 67 columns

In [53]:
```
df_egypt_recent[['date','total_cases','new_cases','total_deaths','new_deaths']]
```

Out[53]:

	date	total_cases	new_cases	total_deaths	new_deaths
50169	2022-05-28	515645.0	0.0	24704.0	NaN

Calculating the maximum values of 'Egypt' data

In [54]:
```
df_egypt[['total_cases','new_cases','total_deaths','new_deaths']].max()
```

Out[54]:
```
total_cases     515645.0
new_cases         5516.0
total_deaths     24704.0
new_deaths         132.0
dtype: float64
```

The highest date recorded for death in Egypt

In [55]:
```
df_egypt[df_egypt['new_deaths'] == df_egypt['new_deaths'].max()]['date']
```

Out[55]:
```
49975    2021-11-15
Name: date, dtype: object
```

The average value(s) of daily-recorded data in Egypt

In [56]:
```
df_egypt['new_cases'].mean()
```

Out[56]:
```
617.5389221556886
```

In [ ]:
```

```