In [1]:
```
# Importing Important libraries
import numpy as np
import pandas as pd
```

In [2]:
```
# 1. Import the dataset using Pandas from above mentioned url .
Url = 'https://raw.githubusercontent.com/SR1608/Datasets/main/covid-data.csv'
dataset = pd.read_csv(Url)
dataset.head()
```

Out[2]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_
0	AFG	Asia	Afghanistan	31/12/19	NaN	0.0	NaN	
1	AFG	Asia	Afghanistan	01/01/20	NaN	0.0	NaN	
2	AFG	Asia	Afghanistan	02/01/20	NaN	0.0	NaN	
3	AFG	Asia	Afghanistan	03/01/20	NaN	0.0	NaN	
4	AFG	Asia	Afghanistan	04/01/20	NaN	0.0	NaN	

5 rows × 49 columns

In [3]:
```
#2. High Level Data Understanding :
# a . Find no . of rows & columns in the dataset

print('Rows :',dataset.shape[0])
print('Columns :',dataset.shape[1])
```

```
Rows : 57394
Columns : 49
```

In [4]:
```python
# b . Data types of columns .

print(dataset.dtypes)
```

```
iso_code                                object
continent                               object
location                                object
date                                    object
total_cases                            float64
new_cases                              float64
new_cases_smoothed                     float64
total_deaths                           float64
new_deaths                             float64
new_deaths_smoothed                    float64
total_cases_per_million                float64
new_cases_per_million                  float64
new_cases_smoothed_per_million         float64
total_deaths_per_million               float64
new_deaths_per_million                 float64
new_deaths_smoothed_per_million        float64
reproduction_rate                      float64
icu_patients                           float64
icu_patients_per_million               float64
hosp_patients                          float64
hosp_patients_per_million              float64
weekly_icu_admissions                  float64
weekly_icu_admissions_per_million      float64
weekly_hosp_admissions                 float64
weekly_hosp_admissions_per_million     float64
total_tests                            float64
new_tests                              float64
total_tests_per_thousand               float64
new_tests_per_thousand                 float64
new_tests_smoothed                     float64
new_tests_smoothed_per_thousand        float64
tests_per_case                         float64
positive_rate                          float64
stringency_index                       float64
population                             float64
population_density                     float64
median_age                             float64
aged_65_older                          float64
aged_70_older                          float64
gdp_per_capita                         float64
extreme_poverty                        float64
cardiovasc_death_rate                  float64
diabetes_prevalence                    float64
female_smokers                         float64
male_smokers                           float64
handwashing_facilities                 float64
hospital_beds_per_thousand             float64
life_expectancy                        float64
human_development_index                float64
dtype: object
```

In [5]:
```
# c . Info & describe of data in dataframe .
print(dataset.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57394 entries, 0 to 57393
Data columns (total 49 columns):
 #   Column                               Non-Null Count  Dtype  
---  ------                               --------------  -----  
 0   iso_code                             57071 non-null  object 
 1   continent                            56748 non-null  object 
 2   location                             57394 non-null  object 
 3   date                                 57394 non-null  object 
 4   total_cases                          53758 non-null  float64
 5   new_cases                            56465 non-null  float64
 6   new_cases_smoothed                   55652 non-null  float64
 7   total_deaths                         44368 non-null  float64
 8   new_deaths                           56465 non-null  float64
 9   new_deaths_smoothed                  55652 non-null  float64
 10  total_cases_per_million              53471 non-null  float64
 11  new_cases_per_million                56401 non-null  float64
 12  new_cases_smoothed_per_million       55587 non-null  float64
 13  total_deaths_per_million             44096 non-null  float64
 14  new_deaths_per_million               56401 non-null  float64
 15  new_deaths_smoothed_per_million      55587 non-null  float64
 16  reproduction_rate                    37696 non-null  float64
 17  icu_patients                         4490 non-null   float64
 18  icu_patients_per_million             4490 non-null   float64
 19  hosp_patients                        5005 non-null   float64
 20  hosp_patients_per_million            5005 non-null   float64
 21  weekly_icu_admissions                357 non-null    float64
 22  weekly_icu_admissions_per_million    357 non-null    float64
 23  weekly_hosp_admissions               645 non-null    float64
 24  weekly_hosp_admissions_per_million   645 non-null    float64
 25  total_tests                          22017 non-null  float64
 26  new_tests                            21787 non-null  float64
 27  total_tests_per_thousand             22017 non-null  float64
 28  new_tests_per_thousand               21787 non-null  float64
 29  new_tests_smoothed                   24612 non-null  float64
 30  new_tests_smoothed_per_thousand      24612 non-null  float64
 31  tests_per_case                       22802 non-null  float64
 32  positive_rate                        23211 non-null  float64
 33  stringency_index                     47847 non-null  float64
 34  population                           57071 non-null  float64
 35  population_density                   54371 non-null  float64
 36  median_age                           51034 non-null  float64
 37  aged_65_older                        50265 non-null  float64
 38  aged_70_older                        50768 non-null  float64
 39  gdp_per_capita                       50367 non-null  float64
 40  extreme_poverty                      33571 non-null  float64
 41  cardiovasc_death_rate                51013 non-null  float64
 42  diabetes_prevalence                  52881 non-null  float64
 43  female_smokers                       39669 non-null  float64
 44  male_smokers                         39156 non-null  float64
 45  handwashing_facilities               24176 non-null  float64
 46  hospital_beds_per_thousand           45936 non-null  float64
 47  life_expectancy                      56336 non-null  float64
 48  human_development_index            49247 non-null  float64
dtypes: float64(45), object(4)
memory usage: 21.5+ MB
None
```

In [6]:
```
# describe
print(dataset.describe(include = 'all'))
```

```
       iso_code continent     location     date   total_cases      new_cases  \
count     57071     56748        57394    57394  5.375800e+04   56465.000000
unique      215         6          216      323           NaN            NaN
top         AFG    Europe  Afghanistan 30/10/20           NaN            NaN
freq        323     14828          323      215           NaN            NaN
mean        NaN       NaN          NaN      NaN  1.677974e+05    1953.576941
std         NaN       NaN          NaN      NaN  1.693038e+06   18269.650340
min         NaN       NaN          NaN      NaN  1.000000e+00   -8261.000000
25%         NaN       NaN          NaN      NaN  1.800000e+02       0.000000
50%         NaN       NaN          NaN      NaN  2.070000e+03      14.000000
75%         NaN       NaN          NaN      NaN  2.235675e+04     235.000000
max         NaN       NaN          NaN      NaN  5.515465e+07  646281.000000

        new_cases_smoothed  total_deaths    new_deaths  new_deaths_smoothed  \
count         55652.000000  4.436800e+04  56465.000000         55652.000000
unique                 NaN           NaN           NaN                  NaN
top                    NaN           NaN           NaN                  NaN
freq                   NaN           NaN           NaN                  NaN
mean           1920.431953  6.858639e+03     47.054317            46.835439
std           17777.391785  5.578081e+04    390.853776           378.272794
min            -552.000000  1.000000e+00  -1918.000000          -232.143000
25%               0.857000  1.300000e+01      0.000000             0.000000
50%              19.429000  8.400000e+01      0.000000             0.286000
75%             245.286000  7.270000e+02      4.000000             4.000000
max          584981.857000  1.328537e+06  10600.000000          9027.714000

        ...  gdp_per_capita  extreme_poverty  cardiovasc_death_rate  \
count   ...    50367.000000     33571.000000           51013.000000
unique  ...             NaN              NaN                    NaN
top     ...             NaN              NaN                    NaN
freq    ...             NaN              NaN                    NaN
mean    ...    20620.172071        12.435453             252.646642
std     ...    20310.999832        19.427924             117.522344
min     ...      661.240000         0.100000              79.370000
25%     ...     5321.444000         0.500000             156.139000
50%     ...    13913.839000         2.000000             238.339000
75%     ...    31400.840000        18.100000             318.991000
max     ...   116935.600000        77.600000             724.417000

        diabetes_prevalence  female_smokers  male_smokers  \
count          52881.000000    39669.000000  39156.000000
unique                  NaN             NaN           NaN
top                     NaN             NaN           NaN
freq                    NaN             NaN           NaN
mean               8.070269       10.741569     32.642686
std                4.189605       10.470743     13.453566
min                0.990000        0.100000      7.700000
25%                5.310000        1.900000     21.400000
50%                7.110000        6.400000     31.400000
75%               10.390000       19.600000     40.900000
max               30.530000       44.000000     78.100000
```

```
        handwashing_facilities  hospital_beds_per_thousand  life_expectancy  \
count             24176.000000                45936.000000     56336.000000   
unique                     NaN                         NaN              NaN   
top                        NaN                         NaN              NaN   
freq                       NaN                         NaN              NaN   
mean                 52.089636                    3.089724        73.937780   
std                  31.645306                    2.513193         7.397016   
min                   1.188000                    0.100000        53.280000   
25%                  21.222000                    1.300000        69.870000   
50%                  52.232000                    2.500000        75.345000   
75%                  83.741000                    4.200000        79.380000   
max                  98.999000                   13.800000        86.750000   

        human_development_index  
count              49247.000000  
unique                      NaN  
top                         NaN  
freq                        NaN  
mean                   0.722223  
std                    0.153261  
min                    0.354000  
25%                    0.601000  
50%                    0.752000  
75%                    0.847000  
max                    0.953000  

[11 rows x 49 columns]
```

In [7]:
```python
# describe
print(dataset.describe(include = 'all'))
```

```
       iso_code continent     location      date  total_cases      new_cases  \
count     57071     56748        57394     57394  5.375800e+04   56465.000000
unique      215         6          216       323          NaN            NaN
top         AFG    Europe  Afghanistan  30/10/20          NaN            NaN
freq        323     14828          323       215          NaN            NaN
mean        NaN       NaN          NaN       NaN  1.677974e+05    1953.576941
std         NaN       NaN          NaN       NaN  1.693038e+06   18269.650340
min         NaN       NaN          NaN       NaN  1.000000e+00   -8261.000000
25%         NaN       NaN          NaN       NaN  1.800000e+02       0.000000
50%         NaN       NaN          NaN       NaN  2.070000e+03      14.000000
75%         NaN       NaN          NaN       NaN  2.235675e+04     235.000000
max         NaN       NaN          NaN       NaN  5.515465e+07  646281.000000

        new_cases_smoothed  total_deaths    new_deaths  new_deaths_smoothed  \
count         55652.000000  4.436800e+04  56465.000000         55652.000000
unique                 NaN           NaN           NaN                  NaN
top                    NaN           NaN           NaN                  NaN
freq                   NaN           NaN           NaN                  NaN
mean           1920.431953  6.858639e+03     47.054317            46.835439
std           17777.391785  5.578081e+04    390.853776           378.272794
min            -552.000000  1.000000e+00  -1918.000000          -232.143000
25%               0.857000  1.300000e+01      0.000000             0.000000
50%              19.429000  8.400000e+01      0.000000             0.286000
75%             245.286000  7.270000e+02      4.000000             4.000000
max          584981.857000  1.328537e+06  10600.000000          9027.714000

        ...  gdp_per_capita  extreme_poverty  cardiovasc_death_rate  \
count   ...    50367.000000     33571.000000           51013.000000
unique  ...             NaN              NaN                    NaN
top     ...             NaN              NaN                    NaN
freq    ...             NaN              NaN                    NaN
mean    ...    20620.172071        12.435453             252.646642
std     ...    20310.999832        19.427924             117.522344
min     ...      661.240000         0.100000              79.370000
25%     ...     5321.444000         0.500000             156.139000
50%     ...    13913.839000         2.000000             238.339000
75%     ...    31400.840000        18.100000             318.991000
max     ...   116935.600000        77.600000             724.417000

        diabetes_prevalence  female_smokers  male_smokers  \
count          52881.000000    39669.000000  39156.000000
unique                  NaN             NaN           NaN
top                     NaN             NaN           NaN
freq                    NaN             NaN           NaN
mean               8.070269       10.741569     32.642686
std                4.189605       10.470743     13.453566
min                0.990000        0.100000      7.700000
25%                5.310000        1.900000     21.400000
50%                7.110000        6.400000     31.400000
75%               10.390000       19.600000     40.900000
max               30.530000       44.000000     78.100000
```

```
        handwashing_facilities  hospital_beds_per_thousand  life_expectancy  \
count             24176.000000                45936.000000     56336.000000
unique                     NaN                         NaN              NaN
top                        NaN                         NaN              NaN
freq                       NaN                         NaN              NaN
mean                 52.089636                    3.089724        73.937780
std                  31.645306                    2.513193         7.397016
min                   1.188000                    0.100000        53.280000
25%                  21.222000                    1.300000        69.870000
50%                  52.232000                    2.500000        75.345000
75%                  83.741000                    4.200000        79.380000
max                  98.999000                   13.800000        86.750000

        human_development_index
count              49247.000000
unique                      NaN
top                         NaN
freq                        NaN
mean                   0.722223
std                    0.153261
min                    0.354000
25%                    0.601000
50%                    0.752000
75%                    0.847000
max                    0.953000

[11 rows x 49 columns]
```

In [8]:
```
#  b . Find which continent has maximum frequency using values
dataset['continent'].value_counts().head()
```

Out[8]:
```
Europe           14828
Africa           13637
Asia             13528
North America     9116
South America     3404
Name: continent, dtype: int64
```

In [9]:
```
# c . Find maximum & mean value in ' total_cases ' .
print('Maximum value :',dataset['total_cases'].max())
print('Mean value    :',dataset['total_cases'].mean())
```

```
Maximum value : 55154651.0
Mean value    : 167797.3688753302
```

In [10]:
```
#  d . Find 25 % , 50 % & 75 % quartile value in ' total_deaths ' .
print(" 25% : ", dataset['total_deaths'].quantile(0.25))
print(" 50% : ", dataset['total_deaths'].quantile(0.50))
print(" 75% : ", dataset['total_deaths'].quantile(0.75))
```

```
 25% :  13.0
 50% :  84.0
 75% :  727.0
```

In [11]:
```
# e . Find which continent has maximum ' human_development_index ' .
dataset[dataset["human_development_index"] == dataset["human_development_index"].
```

Out[11]:
```
38632    Europe
38633    Europe
38634    Europe
38635    Europe
38636    Europe
Name: continent, dtype: object
```

In [12]:
```
# f . Find which continent has minimum ' gdp_per_capita ' .
dataset[dataset['gdp_per_capita']== dataset['gdp_per_capita'].min()].head()['cont
```

Out[12]:
```
10259    Africa
10260    Africa
10261    Africa
10262    Africa
10263    Africa
Name: continent, dtype: object
```

In [13]:
```
# 4 question
dataset_updated = dataset.filter(items = ['continent','location','date','total_ca
dataset_updated.head()
```

Out[13]:

	continent	location	date	total_cases	total_deaths	gdp_per_capita	human_development_i
0	Asia	Afghanistan	31/12/19	NaN	NaN	1803.987	(
1	Asia	Afghanistan	01/01/20	NaN	NaN	1803.987	(
2	Asia	Afghanistan	02/01/20	NaN	NaN	1803.987	(
3	Asia	Afghanistan	03/01/20	NaN	NaN	1803.987	(
4	Asia	Afghanistan	04/01/20	NaN	NaN	1803.987	(

In [14]:
```
# DATA CLEANING
#  a . Remove all duplicates observations

dataset_duplicated = dataset.copy()
```

In [15]:
```
dataset_duplicated = dataset_duplicated.set_index('continent')
dataset_duplicated.duplicated().sum()
```

Out[15]:
```
0
```

In [16]:
```
dataset_duplicated.shape
```

Out[16]:
```
(57394, 48)
```

In [17]:
```
# b Find missing values in all columns
dataset.isnull()
```

Out[17]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_d
0	False	False	False	False	True	False	True	
1	False	False	False	False	True	False	True	
2	False	False	False	False	True	False	True	
3	False	False	False	False	True	False	True	
4	False	False	False	False	True	False	True	
...	...	...	...	...	...	...	...	
57389	True	True	False	False	False	True	True	
57390	True	True	False	False	False	True	True	
57391	True	True	False	False	False	True	True	
57392	True	True	False	False	False	True	True	
57393	True	True	False	False	False	True	True	

57394 rows × 49 columns

In [18]:
```python
# c . Remove all observations where continent column value is missing
dataset.dropna(subset = 'continent')
```

Out[18]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	tc
0	AFG	Asia	Afghanistan	31/12/19	NaN	0.0	NaN	
1	AFG	Asia	Afghanistan	01/01/20	NaN	0.0	NaN	
2	AFG	Asia	Afghanistan	02/01/20	NaN	0.0	NaN	
3	AFG	Asia	Afghanistan	03/01/20	NaN	0.0	NaN	
4	AFG	Asia	Afghanistan	04/01/20	NaN	0.0	NaN	
...	...	...	...	...	...	...	...	
56743	ZWE	Africa	Zimbabwe	13/11/20	8696.0	29.0	36.000	
56744	ZWE	Africa	Zimbabwe	14/11/20	8765.0	69.0	42.000	
56745	ZWE	Africa	Zimbabwe	15/11/20	8786.0	21.0	41.143	
56746	ZWE	Africa	Zimbabwe	16/11/20	8786.0	0.0	36.429	
56747	ZWE	Africa	Zimbabwe	17/11/20	8897.0	111.0	48.000	

56748 rows × 49 columns

In [19]:
```python
#d. Fill all missing values with 0
fill_value = dataset.copy()
fill_value = fill_value.fillna(0)
fill_value
```

Out[19]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	t
0	AFG	Asia	Afghanistan	31/12/19	0.0	0.0	0.0	
1	AFG	Asia	Afghanistan	01/01/20	0.0	0.0	0.0	
2	AFG	Asia	Afghanistan	02/01/20	0.0	0.0	0.0	
3	AFG	Asia	Afghanistan	03/01/20	0.0	0.0	0.0	
4	AFG	Asia	Afghanistan	04/01/20	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	
57389	0	0	International	13/11/20	696.0	0.0	0.0	
57390	0	0	International	14/11/20	696.0	0.0	0.0	
57391	0	0	International	15/11/20	696.0	0.0	0.0	
57392	0	0	International	16/11/20	696.0	0.0	0.0	
57393	0	0	International	17/11/20	696.0	0.0	0.0	

57394 rows × 49 columns

In [20]:
```
#DATE TIME FORMATE
#a . Convert date column in datetime format using pandas.to_datetime
dataset['date'] = pd.to_datetime(dataset['date'])
dataset.dtypes['date']
```

Out[20]:
```
dtype('<M8[ns]')
```

In [21]:
```
#b . Create new column month after extracting month data from date column
dataset['month'] = dataset['date'].dt.month
dataset.dtypes
```

Out[21]:
```
iso_code                                    object
continent                                   object
location                                    object
date                                datetime64[ns]
total_cases                                float64
new_cases                                  float64
new_cases_smoothed                         float64
total_deaths                               float64
new_deaths                                 float64
new_deaths_smoothed                        float64
total_cases_per_million                    float64
new_cases_per_million                      float64
new_cases_smoothed_per_million             float64
total_deaths_per_million                   float64
new_deaths_per_million                     float64
new_deaths_smoothed_per_million            float64
reproduction_rate                          float64
icu_patients                               float64
icu_patients_per_million                   float64
hosp_patients                              float64
hosp_patients_per_million                  float64
weekly_icu_admissions                      float64
weekly_icu_admissions_per_million          float64
weekly_hosp_admissions                     float64
weekly_hosp_admissions_per_million         float64
total_tests                                float64
new_tests                                  float64
total_tests_per_thousand                   float64
new_tests_per_thousand                     float64
new_tests_smoothed                         float64
new_tests_smoothed_per_thousand            float64
tests_per_case                             float64
positive_rate                              float64
stringency_index                           float64
population                                 float64
population_density                         float64
median_age                                 float64
aged_65_older                              float64
aged_70_older                              float64
gdp_per_capita                             float64
extreme_poverty                            float64
cardiovasc_death_rate                      float64
diabetes_prevalence                        float64
female_smokers                             float64
male_smokers                               float64
handwashing_facilities                     float64
hospital_beds_per_thousand                 float64
life_expectancy                            float64
human_development_index                    float64
month                                        int64
dtype: object
```

In [22]:
```
#DATA AGGRIGATION
#a . Find max value in all columns using groupby function on ' continent ' columr
df_groupby = dataset.groupby('continent').max()
df_groupby
```

Out[22]:

	iso_code	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths
continent							
Africa	ZWE	Zimbabwe	2020-12-11	752269.0	13944.0	12583.714	20314.0
Asia	YEM	Yemen	2020-12-11	8874290.0	97894.0	93198.571	130519.0
Europe	VAT	Vatican	2020-12-11	1991233.0	86852.0	54868.571	52147.0
North America	VIR	United States Virgin Islands	2020-12-11	11205486.0	184813.0	156419.143	247220.0
Oceania	WLF	Wallis and Futuna	2020-12-11	27750.0	1384.0	551.714	907.0
South America	VEN	Venezuela	2020-12-11	5876464.0	69074.0	46393.000	166014.0

6 rows × 49 columns

In [23]:
```
#b . Store the result in a new dataframe named ' df_groupby '
df_groupby = df_groupby.reset_index()
df_groupby
```

Out[23]:

	continent	iso_code	location	date	total_cases	new_cases	new_cases_smoothed	total_deat
0	Africa	ZWE	Zimbabwe	2020-12-11	752269.0	13944.0	12583.714	2031
1	Asia	YEM	Yemen	2020-12-11	8874290.0	97894.0	93198.571	13051
2	Europe	VAT	Vatican	2020-12-11	1991233.0	86852.0	54868.571	5214
3	North America	VIR	United States Virgin Islands	2020-12-11	11205486.0	184813.0	156419.143	24722
4	Oceania	WLF	Wallis and Futuna	2020-12-11	27750.0	1384.0	551.714	90
5	South America	VEN	Venezuela	2020-12-11	5876464.0	69074.0	46393.000	16601

6 rows × 50 columns

In [24]:
```
#FEATURE ENGINEERING
#a . Create a new feature ' total_deaths_to_total_cases ' by ratio of ' total_dea
df_groupby['total_deaths_to_total_cases'] = (df_groupby['total_deaths']/df_groupb
df_groupby
```

Out[24]:

	continent	iso_code	location	date	total_cases	new_cases	new_cases_smoothed	total_deat
0	Africa	ZWE	Zimbabwe	2020-12-11	752269.0	13944.0	12583.714	20314
1	Asia	YEM	Yemen	2020-12-11	8874290.0	97894.0	93198.571	130519
2	Europe	VAT	Vatican	2020-12-11	1991233.0	86852.0	54868.571	52147
3	North America	VIR	United States Virgin Islands	2020-12-11	11205486.0	184813.0	156419.143	247220
4	Oceania	WLF	Wallis and Futuna	2020-12-11	27750.0	1384.0	551.714	907
5	South America	VEN	Venezuela	2020-12-11	5876464.0	69074.0	46393.000	166014

6 rows × 51 columns

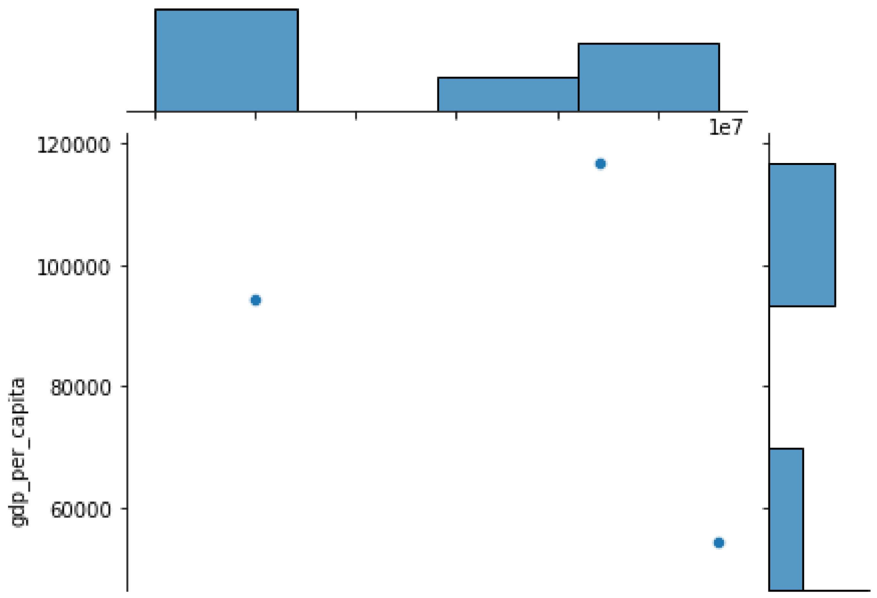
In [ ]:
```
#DATA VISUVALIZATION
#a . Perform Univariate analysis on ' gdp_per_capita ' column by plotting histogr

import seaborn as sns
sns.displot(df_groupby['gdp_per_capita'])
```

In [26]:
```
#b . Plot a scatter plot of ' total_cases ' & ' gdp_per_capita'¶

sns.jointplot(data=df_groupby,x = 'total_cases', y = 'gdp_per_capita', kind = 'sc
```

Out[26]:
```
<seaborn.axisgrid.JointGrid at 0x1ee335098b0>
```



In [ ]:
```
#c . Plot Pairplot on df_groupby dataset .

sns.pairplot(data = df_groupby)
```

In [ ]:
```
#d . Plot a bar plot of ' continent ' column with ' total_cases ' .

sns.barplot(x = 'continent' , y = 'total_cases', data = 'bar')
```

In [ ]:
```
#10.Save the df_groupby dataframe in your local drive using pandas.to_csv functio

df_groupby.to_csv('df_groupby.csv')
```

In [ ]: