In [ ]:
```python
import io
```

In [ ]:
```python
import pandas as pd
```

In [ ]:
```python
import requests
```

In [ ]:
```python
import numpy as np
```

1. import the dataset using pandas from url

In [9]:
```python
url = pd.read_csv("https://raw.githubusercontent.com/SR1608/Datasets/main/covid-data.csv")
url
```

Out[9]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty
0	AFG	Asia	Afghanistan	31/12/19	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
1	AFG	Asia	Afghanistan	01/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
2	AFG	Asia	Afghanistan	02/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
3	AFG	Asia	Afghanistan	03/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
4	AFG	Asia	Afghanistan	04/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
57389	NaN	NaN	International	13/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	NaN
57390	NaN	NaN	International	14/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	NaN
57391	NaN	NaN	International	15/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	NaN
57392	NaN	NaN	International	16/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	NaN
57393	NaN	NaN	International	17/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	NaN

57394 rows × 49 columns

2. high level data understanding

a. Find no. of rows & columns in the dataset b. Data types of columns. c. Info & describe of data in dataframe.

In [10]:
```python
url.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
```

3. low level data understanding

a. Find count of unique values in location column.

In [23]:
```python
url['location'].value_counts()
```

Out[23]:
```
United Kingdom       323
San Marino           323
Estonia              323
Greece               323
Brazil               323
                    ...
Hong Kong             72
Solomon Islands       33
Wallis and Futuna     32
Marshall Islands      20
Vanuatu                7
Name: location, Length: 216, dtype: int64
```

b. Find which continent has maximum frequency using values counts.

In [14]:
```python
url['continent'].value_counts(dropna=False)
```

Out[14]:
```
Europe           14828
Africa           13637
Asia             13528
North America     9116
South America     3404
Oceania           2235
NaN                646
Name: continent, dtype: int64
```

Europe has the highest frequency of 14828

c. Find maximum & mean value in 'total_cases'.

In [26]:
```python
column_name="total_cases"
column_sum=url[column_name].sum()
a=column_sum/57393
print("The maximum and mean of total cases is: ",a)
```

```
The maximum and mean of total cases is:  157169.88057777082
```

e. Find which continent has maximum human development index

In [34]:
```python
url.groupby('continent')['human_development_index'].sum()
```

Out[34]:
```
continent
Africa            7222.777
Asia              9661.617
Europe           10817.325
North America     4494.381
Oceania            984.648
South America     2386.556
Name: human_development_index, dtype: float64
```

Continent with highest human development index is Europe that is 10817.325

f. Find which continent has minimum 'gdp_per_capita'.

In [35]:
```python
url.groupby('continent')['gdp_per_capita'].sum()
```

Out[35]:
```
continent
Africa           7.392128e+07
Asia             3.011561e+08
Europe           4.313814e+08
North America    1.536159e+08
Oceania          2.972208e+07
South America    4.378286e+07
Name: gdp_per_capita, dtype: float64
```

Continent with minimum gdp per capita is North America

4. Filter the dataframe with only this columns

['continent','location','date','total_cases','total_deaths','gdp_per_capita','human_development_index'] and update the data frame.

In [37]:
```python
df.filter(items=['continent','location','date','total_cases','total_deaths','gdp_per_capita','human_development_index'])
```

Out[37]:

	continent	location	date	total_cases	total_deaths	gdp_per_capita	human_development_index
0	Asia	Afghanistan	31/12/19	NaN	NaN	1803.987	0.498
1	Asia	Afghanistan	01/01/20	NaN	NaN	1803.987	0.498
2	Asia	Afghanistan	02/01/20	NaN	NaN	1803.987	0.498
3	Asia	Afghanistan	03/01/20	NaN	NaN	1803.987	0.498
4	Asia	Afghanistan	04/01/20	NaN	NaN	1803.987	0.498
...	...	...	...	...	...	...	...
57389	NaN	International	13/11/20	696.0	7.0	NaN	NaN
57390	NaN	International	14/11/20	696.0	7.0	NaN	NaN
57391	NaN	International	15/11/20	696.0	7.0	NaN	NaN
57392	NaN	International	16/11/20	696.0	7.0	NaN	NaN
57393	NaN	International	17/11/20	696.0	7.0	NaN	NaN

57394 rows × 7 columns

5. Data Cleaning

a. Remove all duplicates observations

In [6]:
```python
url.sort_values("date", inplace = True)

url.drop_duplicates(subset ="date",
                     keep = False, inplace = True)
url
```

Out[6]:

iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardiovasc_d

0 rows × 49 columns

we think there are no repeated obsevation in the dataframe

b. Find missing values in all columns

In [15]:
```python
import pandas as pd
import numpy as np
url = pd.read_csv("https://raw.githubusercontent.com/SR1608/Datasets/main/covid-data.csv")
url.fillna(method ='pad')
```

Out[15]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty
0	AFG	Asia	Afghanistan	31/12/19	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
1	AFG	Asia	Afghanistan	01/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
2	AFG	Asia	Afghanistan	02/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
3	AFG	Asia	Afghanistan	03/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
4	AFG	Asia	Afghanistan	04/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
57389	OWID_WRL	Africa	International	13/11/20	696.0	-9.0	-1.286	7.0	1.0	0.143	...	15469.207	10.0
57390	OWID_WRL	Africa	International	14/11/20	696.0	-9.0	-1.286	7.0	1.0	0.143	...	15469.207	10.0
57391	OWID_WRL	Africa	International	15/11/20	696.0	-9.0	-1.286	7.0	1.0	0.143	...	15469.207	10.0
57392	OWID_WRL	Africa	International	16/11/20	696.0	-9.0	-1.286	7.0	1.0	0.143	...	15469.207	10.0
57393	OWID_WRL	Africa	International	17/11/20	696.0	-9.0	-1.286	7.0	1.0	0.143	...	15469.207	10.0

57394 rows × 49 columns

c. Remove all observations where continent column value is missing

In [17]:
```python
mod_url = url.dropna( how='any',
                    subset=['continent'])
print("Modified Dataframe : ")
display(mod_url)
```

```
Modified Dataframe :
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty
0	AFG	Asia	Afghanistan	31/12/19	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
1	AFG	Asia	Afghanistan	01/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
2	AFG	Asia	Afghanistan	02/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
3	AFG	Asia	Afghanistan	03/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
4	AFG	Asia	Afghanistan	04/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	1803.987	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
56743	ZWE	Africa	Zimbabwe	13/11/20	8696.0	29.0	36.000	255.0	0.0	1.000	...	1899.775	21.4
56744	ZWE	Africa	Zimbabwe	14/11/20	8765.0	69.0	42.000	257.0	2.0	1.000	...	1899.775	21.4
56745	ZWE	Africa	Zimbabwe	15/11/20	8786.0	21.0	41.143	257.0	0.0	0.857	...	1899.775	21.4
56746	ZWE	Africa	Zimbabwe	16/11/20	8786.0	0.0	36.429	257.0	0.0	0.571	...	1899.775	21.4
56747	ZWE	Africa	Zimbabwe	17/11/20	8897.0	111.0	48.000	257.0	0.0	0.429	...	1899.775	21.4

56748 rows × 49 columns

d. Fill all missing values with 0

In [19]:
```python
import pandas as pd
import numpy as np
url = pd.read_csv("https://raw.githubusercontent.com/SR1608/Datasets/main/covid-data.csv")
url.fillna(0)
```

Out[19]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty
0	AFG	Asia	Afghanistan	31/12/19	0.0	0.0	0.0	0.0	0.0	0.0	...	1803.987	0.0
1	AFG	Asia	Afghanistan	01/01/20	0.0	0.0	0.0	0.0	0.0	0.0	...	1803.987	0.0
2	AFG	Asia	Afghanistan	02/01/20	0.0	0.0	0.0	0.0	0.0	0.0	...	1803.987	0.0
3	AFG	Asia	Afghanistan	03/01/20	0.0	0.0	0.0	0.0	0.0	0.0	...	1803.987	0.0
4	AFG	Asia	Afghanistan	04/01/20	0.0	0.0	0.0	0.0	0.0	0.0	...	1803.987	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
57389	0	0	International	13/11/20	696.0	0.0	0.0	7.0	0.0	0.0	...	0.000	0.0
57390	0	0	International	14/11/20	696.0	0.0	0.0	7.0	0.0	0.0	...	0.000	0.0
57391	0	0	International	15/11/20	696.0	0.0	0.0	7.0	0.0	0.0	...	0.000	0.0
57392	0	0	International	16/11/20	696.0	0.0	0.0	7.0	0.0	0.0	...	0.000	0.0
57393	0	0	International	17/11/20	696.0	0.0	0.0	7.0	0.0	0.0	...	0.000	0.0

57394 rows × 49 columns

6. Date time format :

a. Convert date column in datetime format using pandas.to_datetime

In [6]:
```python
import pandas as pd
url = pd.read_csv("https://raw.githubusercontent.com/SR1608/Datasets/main/covid-data.csv")
a=url['date'] = pd.to_datetime(url['date'])
display(a)
```

```
0       2019-12-31
1       2020-01-01
2       2020-02-01
3       2020-03-01
4       2020-04-01
           ...
57389   2020-11-13
57390   2020-11-14
57391   2020-11-15
57392   2020-11-16
57393   2020-11-17
Name: date, Length: 57394, dtype: datetime64[ns]
```

b. Create new column month after extracting month data from date column.

In [10]:
```python
import pandas as pd
url = pd.read_csv("https://raw.githubusercontent.com/SR1608/Datasets/main/covid-data.csv")
url['month'] = pd.DatetimeIndex(url['date']).month
url
```

Out[10]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death
0	AFG	Asia	Afghanistan	31/12/19	NaN	0.0	NaN	NaN	0.0	NaN	...	NaN	59
1	AFG	Asia	Afghanistan	01/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	NaN	59
2	AFG	Asia	Afghanistan	02/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	NaN	59
3	AFG	Asia	Afghanistan	03/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	NaN	59
4	AFG	Asia	Afghanistan	04/01/20	NaN	0.0	NaN	NaN	0.0	NaN	...	NaN	59
...	...	...	...	...	...	...	...	...	...	...	...	...	
57389	NaN	NaN	International	13/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	
57390	NaN	NaN	International	14/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	
57391	NaN	NaN	International	15/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	
57392	NaN	NaN	International	16/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	
57393	NaN	NaN	International	17/11/20	696.0	NaN	NaN	7.0	NaN	NaN	...	NaN	

57394 rows × 50 columns

7. Data Aggregation:

a. Find max value in all columns using groupby function on 'continent' column Tip: use reset_index() after applying groupby.

In [8]:
```python
import pandas as pd
url = pd.read_csv("https://raw.githubusercontent.com/SR1608/Datasets/main/covid-data.csv")
grouped_df = url.groupby("continent")
maximums = grouped_df.max()
maximums = maximums.reset_index()
print(maximums)
```

```
       continent iso_code                      location      date  \
0         Africa      ZWE                      Zimbabwe  31/12/19
1           Asia      YEM                         Yemen  31/12/19
2         Europe      VAT                       Vatican  31/12/19
3  North America      VIR  United States Virgin Islands  31/12/19
4        Oceania      WLF             Wallis and Futuna  31/12/19
5  South America      VEN                     Venezuela  31/12/19

   total_cases  new_cases  new_cases_smoothed  total_deaths  new_deaths  \
0     752269.0    13944.0           12583.714       20314.0       572.0
1    8874290.0    97894.0           93198.571      130519.0      2003.0
2    1991233.0    86852.0           54868.571       52147.0      2004.0
3   11205486.0   184813.0          156419.143      247220.0      4928.0
4      27750.0     1384.0             551.714         907.0        59.0
5    5876464.0    69074.0           46393.000      166014.0      3935.0

   new_deaths_smoothed  ...  gdp_per_capita  extreme_poverty  \
0              297.429  ...       26382.287             77.6
1             1168.000  ...      116935.600             30.3
2             1101.000  ...       94277.965              5.7
3             2715.143  ...       54225.446             23.5
4               22.000  ...       44648.710             25.1
5             1096.714  ...       22767.037              7.1

   cardiovasc_death_rate  diabetes_prevalence  female_smokers  male_smokers  \
0                525.432                22.02             9.7          65.8
1                724.417                17.72            26.9          78.1
2                539.849                10.08            44.0          58.3
3                430.548                17.11            19.1          53.3
4                561.494                30.53            23.5          48.8
5                373.159                12.54            34.2          42.9

   handwashing_facilities  hospital_beds_per_thousand  life_expectancy  \
0                  89.827                        6.30            76.88
1                  98.999                       13.05            84.86
2                  97.719                       13.80            86.75
3                  90.650                        5.80            83.92
4                  82.502                        3.84            83.44
5                  80.635                        5.00            81.44

   human_development_index
0                    0.797
1                    0.933
2                    0.953
3                    0.926
4                    0.939
5                    0.843

[6 rows x 49 columns]
```

b. Store the result in a new dataframe named 'df_groupby'. (Use df_groupby dataframe for all further analysis)

In [12]:
```python
import pandas as pd
df_groupby=pd.DataFrame(maximums)
display(df_groupby)
```

	continent	iso_code	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	gdp_per_capita	extreme_poverty	cardio
0	Africa	ZWE	Zimbabwe	31/12/19	752269.0	13944.0	12583.714	20314.0	572.0	297.429	...	26382.287	77.6	
1	Asia	YEM	Yemen	31/12/19	8874290.0	97894.0	93198.571	130519.0	2003.0	1168.000	...	116935.600	30.3	
2	Europe	VAT	Vatican	31/12/19	1991233.0	86852.0	54868.571	52147.0	2004.0	1101.000	...	94277.965	5.7	
3	North America	VIR	United States Virgin Islands	31/12/19	11205486.0	184813.0	156419.143	247220.0	4928.0	2715.143	...	54225.446	23.5	
4	Oceania	WLF	Wallis and Futuna	31/12/19	27750.0	1384.0	551.714	907.0	59.0	22.000	...	44648.710	25.1	
5	South America	VEN	Venezuela	31/12/19	5876464.0	69074.0	46393.000	166014.0	3935.0	1096.714	...	22767.037	7.1	

6 rows × 49 columns

8. Feature Engineering :

a. Create a new feature 'total_deaths_to_total_cases' by ratio of 'total_deaths' column to 'total_cases'

In [14]:
```python
df_groupby["total_deaths_to_total_cases"]=(df_groupby["total_deaths"]/df_groupby["total_cases"])
display(df_groupby)
```

	continent	iso_code	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	extreme_poverty	cardiovasc_death_rate
0	Africa	ZWE	Zimbabwe	31/12/19	752269.0	13944.0	12583.714	20314.0	572.0	297.429	...	77.6	525.432
1	Asia	YEM	Yemen	31/12/19	8874290.0	97894.0	93198.571	130519.0	2003.0	1168.000	...	30.3	724.417
2	Europe	VAT	Vatican	31/12/19	1991233.0	86852.0	54868.571	52147.0	2004.0	1101.000	...	5.7	539.849
3	North America	VIR	United States Virgin Islands	31/12/19	11205486.0	184813.0	156419.143	247220.0	4928.0	2715.143	...	23.5	430.548
4	Oceania	WLF	Wallis and Futuna	31/12/19	27750.0	1384.0	551.714	907.0	59.0	22.000	...	25.1	561.494
5	South America	VEN	Venezuela	31/12/19	5876464.0	69074.0	46393.000	166014.0	3935.0	1096.714	...	7.1	373.159

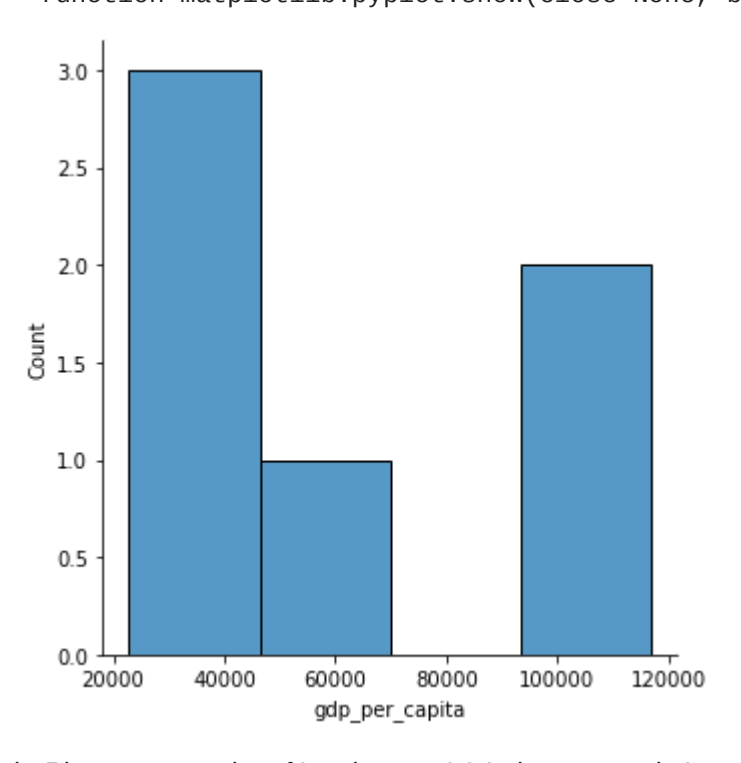
6 rows × 50 columns

9. Data Visualization :

a. Perform Univariate analysis on 'gdp_per_capita' column by plotting histogram using seaborn dist plot.

In [26]:
```python
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
data=df_groupby['gdp_per_capita']
sns.displot(data)
plt.show
```

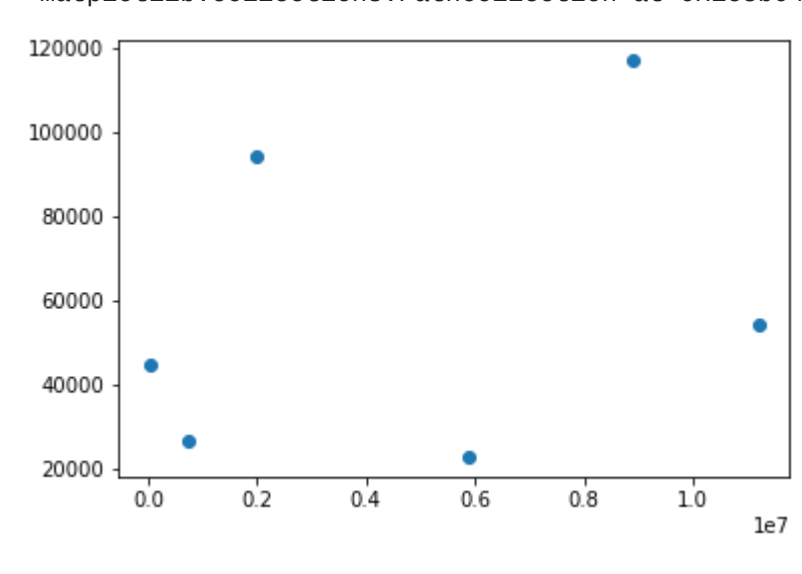
Out[26]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



b. Plot a scatter plot of 'total_cases' & 'gdp_per_capita'

In [28]:
```python
import matplotlib.pyplot as plt
plt.scatter(df_groupby["total_cases"], df_groupby["gdp_per_capita"])
```

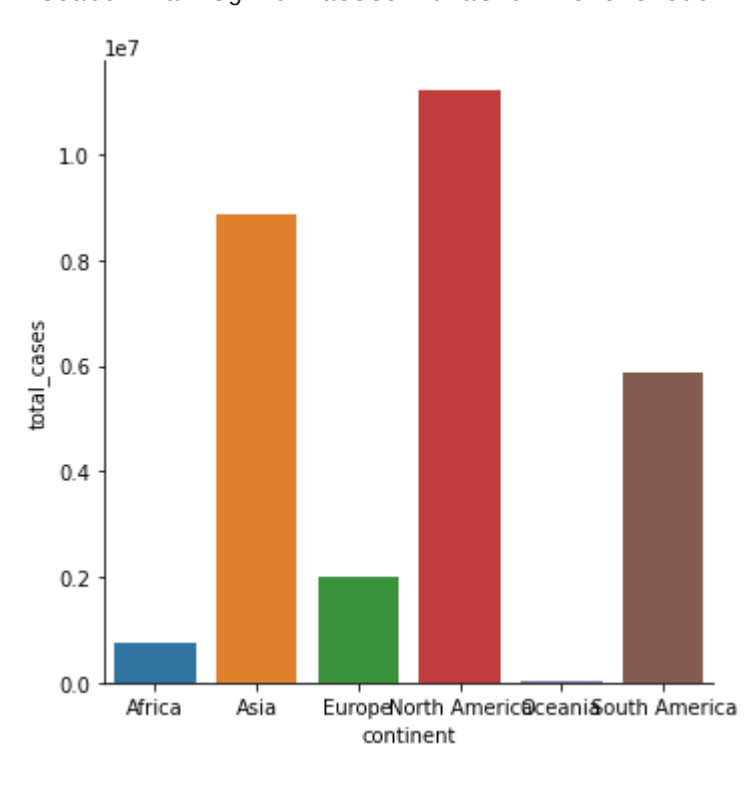
Out[28]:
```
<matplotlib.collections.PathCollection at 0x1c8b04b1370>
```



d. Plot a bar plot of 'continent' column with 'total_cases' . Tip : using kind='bar' in seaborn catplot

In [23]:
```python
import seaborn as sns
sns.catplot(x="continent",y="total_cases",kind="bar",data=df_groupby,)
```

Out[23]:
```
<seaborn.axisgrid.FacetGrid at 0x24e2c7876d0>
```



10.Save the df_groupby dataframe in your local drive using pandas.to_csv function .

In [4]:
```python
import pandas as pd
url = pd.read_csv("https://raw.githubusercontent.com/SR1608/Datasets/main/covid-data.csv")
grouped_df = url.groupby("continent")
maximums = grouped_df.max()
maximums = maximums.reset_index()
df_groupby=pd.DataFrame(maximums)
df_groupby.to_csv('df.csv')
```

In [ ]: