In [2]:
```
import pandas as pd
```

In [5]:
```
covid_df = pd.read_csv("/content/worldometer_data.csv")
covid_df.head() # pandas.dataframe.tail()
```

Out[5]:

	Country/Region	Continent	Population	TotalCases	NewCases	TotalDeaths	NewDeaths	Tota
0	USA	North America	3.311981e+08	5032179	NaN	162804.0	NaN	
1	Brazil	South America	2.127107e+08	2917562	NaN	98644.0	NaN	
2	India	Asia	1.381345e+09	2025409	NaN	41638.0	NaN	
3	Russia	Europe	1.459409e+08	871894	NaN	14606.0	NaN	
4	South Africa	Africa	5.938157e+07	538184	NaN	9604.0	NaN	

Exploration of Dataset

In [42]:
```
covid_df.info()  # provides information about the DataFrame, including data types an
covid_df.shape  # returns the number of rows and columns in the DataFrame
covid_df.columns  # returns the column names
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209 entries, 0 to 208
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Country/Region    209 non-null    object 
 1   Continent         208 non-null    object 
 2   Population        208 non-null    float64
 3   TotalCases        209 non-null    int64  
 4   NewCases          4 non-null      float64
 5   TotalDeaths       188 non-null    float64
 6   NewDeaths         3 non-null      float64
 7   TotalRecovered    205 non-null    float64
 8   NewRecovered      3 non-null      float64
 9   ActiveCases       205 non-null    float64
 10  Serious,Critical  122 non-null    float64
 11  Tot Cases/1M pop  208 non-null    float64
 12  Deaths/1M pop     187 non-null    float64
 13  TotalTests        191 non-null    float64
 14  Tests/1M pop      191 non-null    float64
 15  WHO Region        184 non-null    object 
dtypes: float64(12), int64(1), object(3)
memory usage: 26.2+ KB
```

Out[42]:
```
Index(['Country/Region', 'Continent', 'Population', 'TotalCases', 'NewCases',
       'TotalDeaths', 'NewDeaths', 'TotalRecovered', 'NewRecovered',
       'ActiveCases', 'Serious,Critical', 'Tot Cases/1M pop', 'Deaths/1M pop',
       'TotalTests', 'Tests/1M pop', 'WHO Region'],
      dtype='object')
```

In [43]:
```
covid_df.describe()  # generates descriptive statistics for numerical columns
```

Out[43]:

	Population	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalRecovered	NewF
count	2.080000e+02	2.090000e+02	4.000000	188.000000	3.000000	2.050000e+02	
mean	3.041549e+07	9.171850e+04	1980.500000	3792.590426	300.000000	5.887898e+04	17
std	1.047661e+08	4.325867e+05	3129.611424	15487.184877	451.199512	2.566984e+05	21
min	8.010000e+02	1.000000e+01	20.000000	1.000000	1.000000	7.000000e+00	
25%	9.663140e+05	7.120000e+02	27.500000	22.000000	40.500000	3.340000e+02	4
50%	7.041972e+06	4.491000e+03	656.000000	113.000000	80.000000	2.178000e+03	9
75%	2.575614e+07	3.689600e+04	2609.000000	786.000000	449.500000	2.055300e+04	25
max	1.381345e+09	5.032179e+06	6590.000000	162804.000000	819.000000	2.576668e+06	41

Acessing Data

In [14]:
```
print(covid_df.loc[7])  # returns a specific row as a Series based on the row index

print(50*"-")
covid_df.loc[8:16, 'TotalDeaths']  # returns a specific range of rows for a column
```

```
Country/Region             Chile
Continent          South America
Population           19132514.0
TotalCases               366671
NewCases                    NaN
TotalDeaths               9889.0
NewDeaths                   NaN
TotalRecovered          340168.0
NewRecovered                NaN
ActiveCases             16614.0
Serious,Critical         1358.0
Tot Cases/1M pop        19165.0
Deaths/1M pop             517.0
TotalTests             1760615.0
Tests/1M pop             92022.0
WHO Region              Americas
Name: 7, dtype: object
--------------------------------------------------
```

Out[14]:
```
8     11939.0
9     28500.0
10    17976.0
11    46413.0
12     3055.0
13     6035.0
14     3306.0
15    35187.0
16     5798.0
Name: TotalDeaths, dtype: float64
```

In [46]:
```
print(covid_df.iloc[1])

print(50*"-")

covid_df.iloc[1:3, 1:6]
```

```
Country/Region             Brazil
Continent           South America
Population            212710692.0
TotalCases                2917562
NewCases                      NaN
TotalDeaths               98644.0
NewDeaths                     NaN
TotalRecovered          2047660.0
NewRecovered                  NaN
ActiveCases              771258.0
Serious,Critical           8318.0
Tot Cases/1M pop          13716.0
Deaths/1M pop               464.0
TotalTests              13206188.0
Tests/1M pop               62085.0
WHO Region                Americas
Name: 1, dtype: object
--------------------------------------------------
```

Out[46]:

	Continent	Population	TotalCases	NewCases	TotalDeaths
1	South America	2.127107e+08	2917562	NaN	98644.0
2	Asia	1.381345e+09	2025409	NaN	41638.0

Filtering data

In [31]:
```
covid_df["TotalCases"].max
```

Out[31]:
```
<bound method NDFrame._add_numeric_operations.<locals>.max of 0      5032179
1      2917562
2      2025409
3       871894
4       538184
        ...
204         13
205         13
206         13
207         12
208         10
Name: TotalCases, Length: 209, dtype: int64>
```

In [32]:
```
covid_df[covid_df['TotalCases'] > 2025409]  # filters rows based on a condition

covid_df.query('TotalCases > 2025409')  # alternative way to filter rows based on a
```

Out[32]:

	Country/Region	Continent	Population	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalR
0	USA	North America	331198130.0	5032179	NaN	162804.0	NaN	
1	Brazil	South America	212710692.0	2917562	NaN	98644.0	NaN	

Handling missing data:

In [35]:
```
covid_df
```

Out[35]:

	Country/Region	Continent	Population	TotalCases	NewCases	TotalDeaths	NewDeaths	Tc
0	USA	North America	3.311981e+08	5032179	NaN	162804.0	NaN	
1	Brazil	South America	2.127107e+08	2917562	NaN	98644.0	NaN	
2	India	Asia	1.381345e+09	2025409	NaN	41638.0	NaN	
3	Russia	Europe	1.459409e+08	871894	NaN	14606.0	NaN	
4	South Africa	Africa	5.938157e+07	538184	NaN	9604.0	NaN	
...	...	...	...	...	...	...	...	
204	Montserrat	North America	4.992000e+03	13	NaN	1.0	NaN	
205	Caribbean Netherlands	North America	2.624700e+04	13	NaN	NaN	NaN	
206	Falkland Islands	South America	3.489000e+03	13	NaN	NaN	NaN	
207	Vatican City	Europe	8.010000e+02	12	NaN	NaN	NaN	
208	Western Sahara	Africa	5.986820e+05	10	NaN	1.0	NaN	

209 rows × 16 columns

In [47]:
```
covid_df.sort_values('NewCases', ascending=True)
```

Out[47]:

	Country/Region	Continent	Population	TotalCases	NewCases	TotalDeaths	NewDeaths	Tot
72	S. Korea	Asia	51273732.0	14519	20.0	303.0	1.0	
146	Jamaica	North America	2962478.0	958	30.0	12.0	NaN	
28	Bolivia	South America	11688459.0	86423	1282.0	3465.0	80.0	
5	Mexico	North America	129066160.0	462690	6590.0	50517.0	819.0	
0	USA	North America	331198130.0	5032179	NaN	162804.0	NaN	
...	...	...	...	...	...	...	...	
204	Montserrat	North America	4992.0	13	NaN	1.0	NaN	
205	Caribbean Netherlands	North America	26247.0	13	NaN	NaN	NaN	
206	Falkland Islands	South America	3489.0	13	NaN	NaN	NaN	
207	Vatican City	Europe	801.0	12	NaN	NaN	NaN	
208	Western Sahara	Africa	598682.0	10	NaN	1.0	NaN	

209 rows × 16 columns

In [ ]:

Logging and REGEX

In [37]:
```
import logging
```

In [38]:
```
logging.basicConfig(level=logging.DEBUG, format='%(asctime)s - %(levelname)s - %(mes
```

In [39]:
```
logging.debug('This is a debug message')
logging.info('This is an info message')
logging.warning('This is a warning message')
logging.error('This is an error message')
logging.critical('This is a critical message')
```

```
WARNING:root:This is a warning message
ERROR:root:This is an error message
CRITICAL:root:This is a critical message
```

In [49]:
```
import re

pattern = r'the'  # regular expression pattern
text = 'The quick brown fox jumps over the lazy dog'
match = re.search(pattern, text)  # search for the pattern in the text
if match:
    print('Pattern found')
else:
    print('Pattern not found')
```

```
Pattern found
```

In [50]:
```
pattern = r'(\d+)-(\d+)-(\d+)'  # pattern to match a date in the format "yyyy-mm-dd"
text = 'Today is 2023-07-07'
match = re.search(pattern, text)
if match:
    year = match.group(1)
    month = match.group(2)
    day = match.group(3)
    print(f'Date: {year}-{month}-{day}')
else:
    print('Date not found')
```

```
Date: 2023-07-07
```

In [ ]:
```

```