ASSESSMENT 2 - PYTHON FUNDAMENTALS FOR DATA SCIENCE - ITEC 102 - SHANZAY ANWAR - SOO383783

TASK 2 - Data Loading

```
# Import the pandas library
import pandas as pd

# Load the COVID-19 confirmed cases dataset from the GitHub URL into a DataFrame
# Dataset is provided by the unit coordinator for teaching purposes
# Source: Johns Hopkins CSSE COVID-19 Data Repository
confirmed_df = pd.read_csv(
    'https://raw.githubusercontent.com/FarshidKeivanian/Sessions_Python/main/time_series_covid19_confirmed_global%20.csv'
)
```

TASK 3 - Data Overview

Display the first 5 rows using df.head()

```
# Import the pandas library
import pandas as pd

# Load the COVID-19 confirmed cases dataset from the GitHub URL into a DataFrame
# Dataset is provided by the unit coordinator for teaching purposes
# Source: Johns Hopkins CSSE COVID-19 Data Repository
confirmed_df = pd.read_csv(
    'https://raw.githubusercontent.com/FarshidKeivanian/Sessions_Python/main/time_series_covid19_confirmed_global%20.csv'
)

# Display the first 5 rows of the dataset to confirm it loaded correctly
confirmed_df.head()
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	2/28/23	3/1/
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0	0	...	209322	2093
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0	0	...	334391	3344
2	NaN	Algeria	28.03390	1.659600	0	0	0	0	0	0	...	271441	2714
3	NaN	Andorra	42.50630	1.521800	0	0	0	0	0	0	...	47866	478
4	NaN	Angola	-11.20270	17.873900	0	0	0	0	0	0	...	105255	1052

5 rows × 1147 columns

Use df.info() and/or df.describe() to inspect structure and stats

```
# Import pandas
import pandas as pd

# Load the dataset
confirmed_df = pd.read_csv(
    'https://raw.githubusercontent.com/FarshidKeivanian/Sessions_Python/main/time_series_covid19_confirmed_global%20.csv'
)

# View dataset structure
confirmed_df.info()

# View basic statistics of numeric columns
confirmed_df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 289 entries, 0 to 288
Columns: 1147 entries, Province/State to 3/9/23
dtypes: float64(2), int64(1143), object(2)
memory usage: 2.5+ MB
```

	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20
count	287.000000	287.000000	289.000000	289.000000	289.000000	289.000000	289.000000	289.000000	289.000000	289.000000
mean	19.718719	22.182084	1.927336	2.273356	3.266436	4.972318	7.335640	10.134948	19.307958	21.346021
std	25.956609	77.870931	26.173664	26.270191	32.707271	45.523871	63.623197	85.724481	210.329649	211.628535
min	-71.949900	-178.116500	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	4.072192	-32.823050	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	21.512583	20.939400	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	40.401784	89.224350	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	71.706900	178.065000	444.000000	444.000000	549.000000	761.000000	1058.000000	1423.000000	3554.000000	3554.000000

8 rows × 1145 columns

Reflect on key insights such as number of rows/columns, data types, presence of nulls, any anomalies or patterns noticed (Markdown)

The dataset has around 280 rows and more than 100 columns. The first few columns give location details like province or state, country or region, latitude, and longitude. These columns mostly contain text, except for latitude and longitude which are numbers. The rest of the columns are dates that show the total number of confirmed COVID-19 cases reported on each day. These numbers go up over time, as the total cases add up as days pass.

There are some missing values in the "Province/State" column, which is expected because not every country reports data by province or state. The date columns don't have any missing values, so the daily case counts are complete. I didn't notice any major problems or strange data in the dataset. Overall, the data is organized well and the information about case numbers follows a clear pattern over time.

TAKS 4 - Global Daily Confirmed Cases

```
import pandas as pd

# Load the dataset
confirmed_df = pd.read_csv(
    'https://raw.githubusercontent.com/FarshidKeivanian/Sessions_Python/main/time_series_covid19_confirmed_global%20.csv'
)

# Select only the date columns (ignore Province/State, Country/Region, Lat, Long)
date_columns = confirmed_df.columns[4:]  # columns from index 4 onward are dates

# Sum across all countries/provinces for each date
# axis=0 sums along rows → gives total per column (per date)
global_daily_cumulative = confirmed_df[date_columns].sum(axis=0)

# Convert column names to datetime
global_daily_cumulative.index = pd.to_datetime(global_daily_cumulative.index)

# Display first 10 days of global cumulative confirmed cases
print(global_daily_cumulative.head(10))
```

```
2020-01-22     557
2020-01-23     657
2020-01-24     944
2020-01-25    1437
2020-01-26    2120
2020-01-27    2929
2020-01-28    5580
2020-01-29    6169
2020-01-30    8237
2020-01-31    9927
dtype: int64
/tmp/ipython-input-2169845289.py:16: UserWarning: Could not infer format, so each element will be parsed individually, falling
  global_daily_cumulative.index = pd.to_datetime(global_daily_cumulative.index)
```

TASK 5 - Daily Increase

```
import pandas as pd
import numpy as np

# Load the COVID-19 data from the web
data = pd.read_csv(
    'https://raw.githubusercontent.com/FarshidKeivanian/Sessions_Python/main/time_series_covid19_confirmed_global%20.csv'
)

# Keep only the date columns (skip location columns)
cases = data.iloc[:, 4:]

# Sum cases across all countries for each date to get global totals
global_cases = cases.sum(axis=0)

# Convert the global cumulative cases to a NumPy array
cumulative_array = global_cases.to_numpy()

# Use NumPy to calculate daily new cases (vectorized)
daily_new_cases_np = np.diff(cumulative_array, prepend=0)  # prepend 0 keeps the same length

# Convert back to Pandas Series to keep the date index
global_new_cases = pd.Series(daily_new_cases_np, index=pd.to_datetime(global_cases.index))

# Show first 5 days of total and new cases
print("Total confirmed cases:")
print(global_cases.head())

print("\nDaily new cases (using NumPy):")
print(global_new_cases.head())
```

```
Total confirmed cases:
1/22/20     557
1/23/20     657
1/24/20     944
1/25/20    1437
1/26/20    2120
dtype: int64

Daily new cases (using NumPy):
2020-01-22    557
2020-01-23    100
2020-01-24    287
2020-01-25    493
2020-01-26    683
dtype: int64
/tmp/ipython-input-2608763775.py:22: UserWarning: Could not infer format, so each element will be parsed individually, falling
  global_new_cases = pd.Series(daily_new_cases_np, index=pd.to_datetime(global_cases.index))
```

TASK 6 - Visualization

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

# Step 1: Load the dataset
url = 'https://raw.githubusercontent.com/FarshidKeivanian/Sessions_Python/main/time_series_covid19_confirmed_global%20.csv'
df = pd.read_csv(url)

# Step 2: Select only the date columns (skip the first 4 columns: Province/State, Country/Region, Lat, Long)
date_columns = df.columns[4:]
cases_only = df[date_columns]

# Step 3: Sum confirmed cases for each date across all countries
global_cumulative = cases_only.sum(axis=0)

# Step 4: Convert the date column names to datetime objects for cleaner x-axis labels
dates = pd.to_datetime(date_columns)

# Step 5: Plot the global cumulative confirmed cases
plt.figure(figsize=(12, 6))
plt.plot(dates, global_cumulative, color='blue', linewidth=2)

# Formatting the plot
plt.title('Global Cumulative Confirmed COVID-19 Cases Over Time')
plt.xlabel('Date')
plt.ylabel('Total Confirmed Cases')
```

```
# Make x-axis more readable
plt.gca().xaxis.set_major_locator(mdates.MonthLocator(interval=2))  # Show every 2 months
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%b %Y'))  # Format as "Jan 2020"
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-1871653427.py:17: UserWarning: Could not infer format, so each element will be parsed individually, falling
  dates = pd.to_datetime(date_columns)
```