You: URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DA0101EN-Coursera/laptop_pricing_dataset_mod1.csv"

Assistant: Thanks for sharing the URL. It points to a CSV file likely containing a laptop pricing dataset used in IBM's data analysis course on Coursera.

Would you like help with any of the following?
- Loading and reading the dataset
- Exploring or visualizing the data
- Performing data cleaning or preprocessing
- Running some analysis

Let me know how you'd like to proceed!

You: Write a Python code that can perform the following tasks.

Read the CSV file, located on a given file path, into a Pandas data frame, assuming that the first rows of the file are the headers for the data.

Assistant:

```
import pandas as pd

url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DA0101EN-Coursera/laptop_pricing_dataset_mod1.csv"
df = pd.read_csv(url)
print(df.head())
```

You: Write a Python code that identifies the columns with missing values in a pandas data frame.

Assistant:

```
missing_columns = df.columns[df.isnull().any()]
for col in missing_columns:
    print(f"Column '{col}' has {df[col].isnull().sum()} missing values.")
```

You: Write a Python code to replace the missing values in a pandas data frame...

Assistant:

```
if 'Screen_Size_cm' in df.columns:
    mode_value = df['Screen_Size_cm'].mode()[0]
    df['Screen_Size_cm'].fillna(mode_value, inplace=True)

if 'Weight_kg' in df.columns:
    mean_value = df['Weight_kg'].mean()
    df['Weight_kg'].fillna(mean_value, inplace=True)
```

You: Write a Python code snippet to change the data type of the attributes "Screen_Size_cm"

and "Weight_kg" of a data frame to float.

```
Assistant:
for col in ['Screen_Size_cm', 'Weight_kg']:
    if col in df.columns:
        df[col] = df[col].astype(float)
```

You: Write a Python code to modify the contents under the following attributes...

```
Assistant:
if 'Screen_Size_cm' in df.columns:
    df['Screen_Size_inch'] = df['Screen_Size_cm'] / 2.54
    df.drop('Screen_Size_cm', axis=1, inplace=True)

if 'Weight_kg' in df.columns:
    df['Weight_pounds'] = df['Weight_kg'] * 2.20462
    df.drop('Weight_kg', axis=1, inplace=True)
```

You: Write a Python code to normalize the content under the attribute "CPU_frequency"...

```
Assistant:
if 'CPU_frequency' in df.columns:
    max_value = df['CPU_frequency'].max()
    if max_value != 0:
        df['CPU_frequency'] = df['CPU_frequency'] / max_value
```

You: Write a Python code to perform the following tasks (One-hot encoding for 'Screen')

```
Assistant:
if 'Screen' in df.columns:
    df1 = pd.get_dummies(df['Screen'], prefix='Screen')
    df = pd.concat([df, df1], axis=1)
    df.drop('Screen', axis=1, inplace=True)
```

You: Generate a Python code that converts the values under Price from USD to Euros.

```
Assistant:
exchange_rate = 0.85
if 'Price' in df.columns:
    df['Price'] = df['Price'] * exchange_rate
```

You: Perform min-max normalization on the CPU_frequency parameter.

```
Assistant:
if 'CPU_frequency' in df.columns:
    min_val = df['CPU_frequency'].min()
    max_val = df['CPU_frequency'].max()
    if max_val != min_val:
        df['CPU_frequency'] = (df['CPU_frequency'] - min_val) / (max_val - min_val)
    else:
        df['CPU_frequency'] = 0.0
```