```
from google.colab import files
import pandas as pd

# Upload file
uploaded = files.upload()

# Load the uploaded file into a DataFrame
for fn in uploaded.keys():
    print(f'User uploaded file "{fn}" with length {len(uploaded[fn])} bytes')

# Assuming the uploaded file is a CSV
df = pd.read_csv(next(iter(uploaded)))
df.head()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Details (1).csv to Details (1).csv
User uploaded file "Details (1).csv" with length 63384 bytes
```

	Order ID	Amount	Profit	Quantity	Category	Sub-Category	PaymentMode
0	B-25681	1096	658	7	Electronics	Electronic Games	COD
1	B-26055	5729	64	14	Furniture	Chairs	EMI
2	B-25955	2927	146	8	Furniture	Bookcases	EMI
3	B-26093	2847	712	8	Electronics	Printers	Credit Card
4	B-25602	2617	1151	4	Electronics	Phones	Credit Card

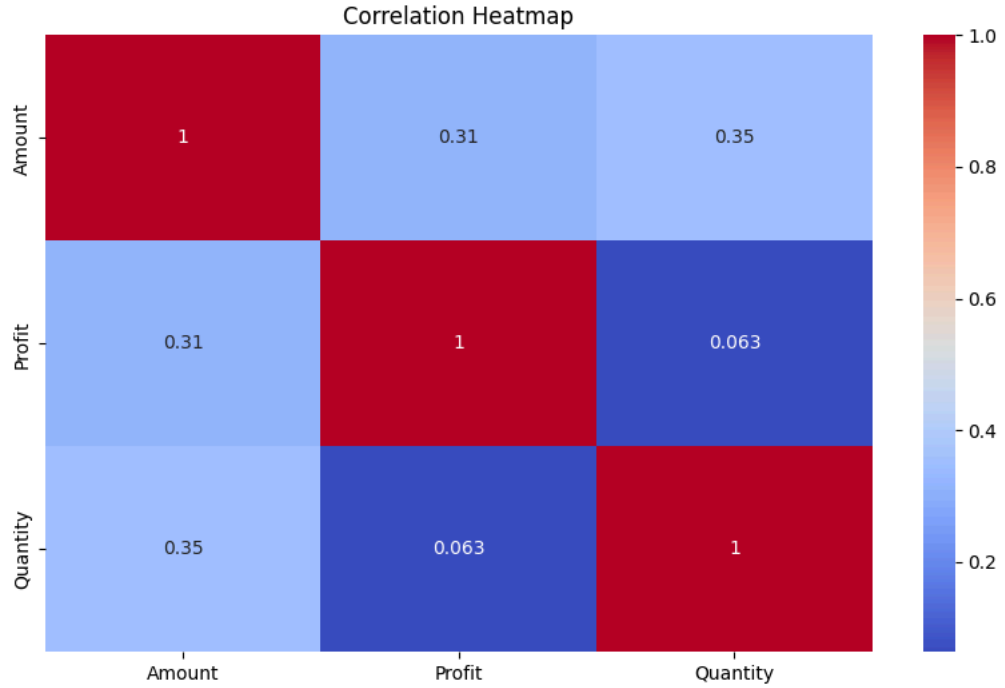
```
# Display shape and column names
print("Shape of Dataset:", df.shape)
print("Column Names:", df.columns.tolist())

# Check data types
print("\nData Types:\n", df.dtypes)

# Summary statistics
df.describe()
```

```
Shape of Dataset: (1500, 7)
Column Names: ['Order ID', 'Amount', 'Profit', 'Quantity', 'Category', 'Sub-Category', 'PaymentMode']

Data Types:
 Order ID        object
Amount           int64
Profit           int64
Quantity         int64
Category        object
Sub-Category    object
PaymentMode     object
dtype: object
```

	Amount	Profit	Quantity
count	1500.000000	1500.00000	1500.000000
mean	291.847333	24.64200	3.743333
std	461.924620	168.55881	2.184942
min	4.000000	-1981.00000	1.000000
25%	47.750000	-12.00000	2.000000
50%	122.000000	8.00000	3.000000
75%	326.250000	38.00000	5.000000
max	5729.000000	1864.00000	14.000000

```
# Check for missing values
df.isnull().sum()

# Fill missing numeric values with mean
df.fillna(df.mean(numeric_only=True), inplace=True)

# Or drop rows with missing values
# df.dropna(inplace=True)


import seaborn as sns
import matplotlib.pyplot as plt

# Correlation heatmap
plt.figure(figsize=(10,6))
```

```
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```




```
df.to_csv('cleaned_dataset.csv', index=False)
files.download('cleaned_dataset.csv')
```