```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


data = pd.read_csv('/content/E-commerce Dataset.csv')
print(data.head())
```

```
  Order_Date      Time  Aging  Customer_Id  Gender Device_Type  \
0  2018-01-02  10:56:33    8.0        37077  Female         Web
1  2018-07-24  20:41:37    2.0        59173  Female         Web
2  2018-11-08  08:38:49    8.0        41066  Female         Web
3  2018-04-18  19:28:06    7.0        50741  Female         Web
4  2018-08-13  21:18:39    9.0        53639  Female         Web

  Customer_Login_type    Product_Category            Product  Sales  Quantity  \
0              Member  Auto & Accessories  Car Media Players  140.0       1.0
1              Member  Auto & Accessories       Car Speakers  211.0       1.0
2              Member  Auto & Accessories    Car Body Covers  117.0       5.0
3              Member  Auto & Accessories    Car & Bike Care  118.0       1.0
4              Member  Auto & Accessories               Tyre  250.0       1.0

   Discount  Profit  Shipping_Cost Order_Priority Payment_method
0       0.3    46.0            4.6         Medium    credit_card
1       0.3   112.0           11.2         Medium    credit_card
2       0.1    31.2            3.1       Critical    credit_card
3       0.3    26.2            2.6           High    credit_card
4       0.3   160.0           16.0       Critical    credit_card
```

```
# Checking for missing values
print(data.isnull().sum())

# Filling or dropping missing values
data.fillna(method='ffill', inplace=True)

# Removing duplicates
data.drop_duplicates(inplace=True)

# Converting data types if necessary
#data['Order_date'] = pd.to_datetime(data['Order_date'])
```

```
Order_Date             0
Time                   0
Aging                  0
Customer_Id            0
Gender                 0
Device_Type            0
Customer_Login_type    0
Product_Category       0
Product                0
Sales                  0
Quantity               0
Discount               0
Profit                 0
Shipping_Cost          0
Order_Priority         0
Payment_method         0
dtype: int64
```

```
# Summary statistics
print(data.describe())

# Basic info
print(data.info())
```

```
              Aging   Customer_Id         Sales      Quantity      Discount  \
count  51290.000000  51290.000000  51290.000000  51290.000000  51290.000000
mean       5.255089  58155.758764    152.340632      2.502983      0.303821
std        2.959944  26032.215826     66.494793      1.511858      0.131025
min        1.000000  10000.000000     33.000000      1.000000      0.100000
25%        3.000000  35831.250000     85.000000      1.000000      0.200000
50%        5.000000  61018.000000    136.500000      2.000000      0.300000
75%        8.000000  80736.250000    218.000000      4.000000      0.400000
max       10.500000  99999.000000    250.000000      5.000000      0.500000

              Profit  Shipping_Cost
```

```
count  51290.000000   51290.000000
mean      70.407226       7.041456
std       48.729488       4.871750
min        0.500000       0.100000
25%       24.900000       2.500000
50%       59.900000       6.000000
75%      118.400000      11.800000
max      167.500000      16.800000
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Order_Date           51290 non-null  object
 1   Time                 51290 non-null  object
 2   Aging                51290 non-null  float64
 3   Customer_Id          51290 non-null  int64
 4   Gender               51290 non-null  object
 5   Device_Type          51290 non-null  object
 6   Customer_Login_type  51290 non-null  object
 7   Product_Category     51290 non-null  object
 8   Product              51290 non-null  object
 9   Sales                51290 non-null  float64
 10  Quantity             51290 non-null  float64
 11  Discount             51290 non-null  float64
 12  Profit               51290 non-null  float64
 13  Shipping_Cost        51290 non-null  float64
 14  Order_Priority       51290 non-null  object
 15  Payment_method       51290 non-null  object
dtypes: float64(6), int64(1), object(9)
memory usage: 6.3+ MB
None
```

```
# Convert Order_Date to datetime
data['Order_Date'] = pd.to_datetime(data['Order_Date'])

# Check the data types and basic info
print(data.info())

# Display summary statistics
print(data.describe())
```

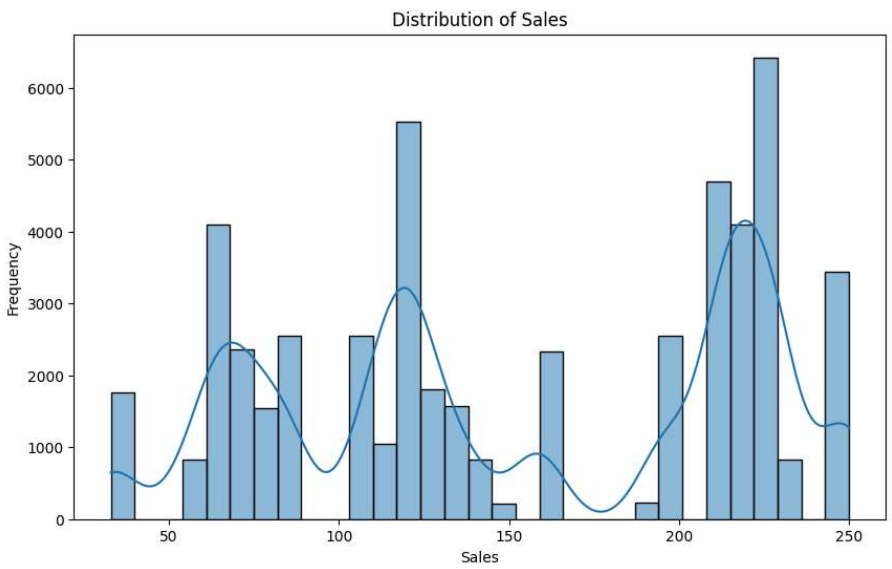
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Order_Date           51290 non-null  datetime64[ns]
 1   Time                 51290 non-null  object
 2   Aging                51289 non-null  float64
 3   Customer_Id          51290 non-null  int64
 4   Gender               51290 non-null  object
 5   Device_Type          51290 non-null  object
 6   Customer_Login_type  51290 non-null  object
 7   Product_Category     51290 non-null  object
 8   Product              51290 non-null  object
 9   Sales                51289 non-null  float64
 10  Quantity             51288 non-null  float64
 11  Discount             51289 non-null  float64
 12  Profit               51290 non-null  float64
 13  Shipping_Cost        51289 non-null  float64
 14  Order_Priority       51288 non-null  object
 15  Payment_method       51290 non-null  object
dtypes: datetime64[ns](1), float64(6), int64(1), object(8)
memory usage: 6.3+ MB
None
                          Order_Date         Aging   Customer_Id  \
count                          51290  51289.000000  51290.000000
mean   2018-07-23 11:27:05.720413184      5.255035  58155.758764
min              2018-01-01 00:00:00      1.000000  10000.000000
25%              2018-05-07 00:00:00      3.000000  35831.250000
50%              2018-07-28 00:00:00      5.000000  61018.000000
75%              2018-10-17 00:00:00      8.000000  80736.250000
max              2018-12-30 00:00:00     10.500000  99999.000000
std                              NaN      2.959948  26032.215826

              Sales      Quantity      Discount        Profit  Shipping_Cost
count  51289.000000  51288.000000  51289.000000  51290.000000   51289.000000
mean     152.340872      2.502983      0.303821     70.407226       7.041557
min       33.000000      1.000000      0.100000      0.500000       0.100000
25%       85.000000      1.000000      0.200000     24.900000       2.500000
50%      133.000000      2.000000      0.300000     59.900000       6.000000
```

```
75%      218.000000      4.000000      0.400000    118.400000      11.800000
max      250.000000      5.000000      0.500000    167.500000      16.800000
std       66.495419      1.511859      0.131027     48.729488       4.871745
```

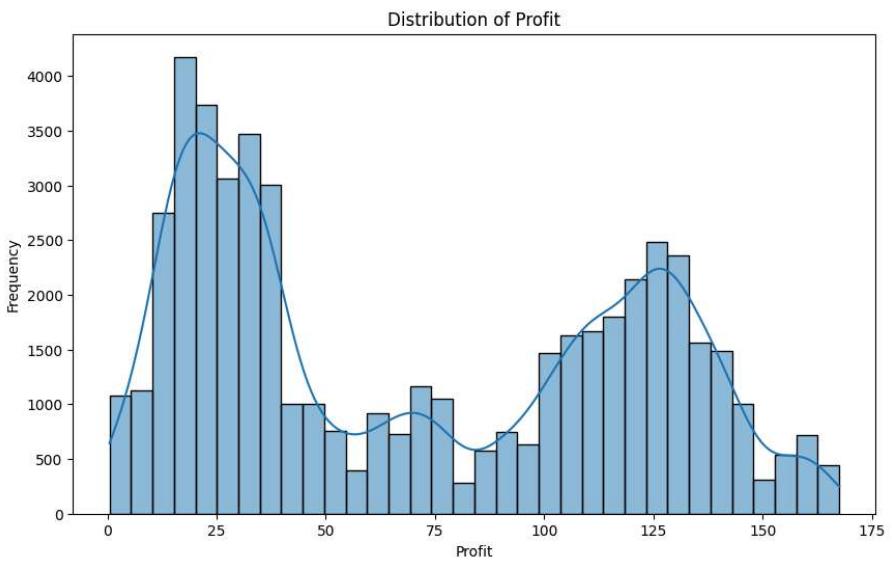
```
# Summary statistics
print(data.describe(include='all'))

# Basic info
print(data.info())
```

```
freq    28138       47632               49097            25646    2332
mean      NaN         NaN                 NaN              NaN     NaN
min       NaN         NaN                 NaN              NaN     NaN
25%       NaN         NaN                 NaN              NaN     NaN
50%       NaN         NaN                 NaN              NaN     NaN
75%       NaN         NaN                 NaN              NaN     NaN
max       NaN         NaN                 NaN              NaN     NaN
std       NaN         NaN                 NaN              NaN     NaN

                Sales      Quantity      Discount        Profit  Shipping_Cost  \
count   51289.000000  51288.000000  51289.000000  51290.000000   51289.000000
unique           NaN           NaN           NaN           NaN            NaN
top              NaN           NaN           NaN           NaN            NaN
freq             NaN           NaN           NaN           NaN            NaN
mean      152.340872      2.502983      0.303821     70.407226       7.041557
min        33.000000      1.000000      0.100000      0.500000       0.100000
25%        85.000000      1.000000      0.200000     24.900000       2.500000
50%       133.000000      2.000000      0.300000     59.900000       6.000000
75%       218.000000      4.000000      0.400000    118.400000      11.800000
max       250.000000      5.000000      0.500000    167.500000      16.800000
std        66.495419      1.511859      0.131027     48.729488       4.871745

       Order_Priority Payment_method
count           51288          51290
unique              4              5
top            Medium    credit_card
freq            29433          38137
mean              NaN            NaN
min               NaN            NaN
25%               NaN            NaN
50%               NaN            NaN
75%               NaN            NaN
max               NaN            NaN
std               NaN            NaN
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Order_Date           51290 non-null  datetime64[ns]
 1   Time                 51290 non-null  object
 2   Aging                51289 non-null  float64
 3   Customer_Id          51290 non-null  int64
 4   Gender               51290 non-null  object
 5   Device_Type          51290 non-null  object
 6   Customer_Login_type  51290 non-null  object
 7   Product_Category     51290 non-null  object
 8   Product              51290 non-null  object
 9   Sales                51289 non-null  float64
 10  Quantity             51288 non-null  float64
 11  Discount             51289 non-null  float64
 12  Profit               51290 non-null  float64
 13  Shipping_Cost        51289 non-null  float64
 14  Order_Priority       51288 non-null  object
 15  Payment_method       51290 non-null  object
dtypes: datetime64[ns](1), float64(6), int64(1), object(8)
memory usage: 6.3+ MB
None
```

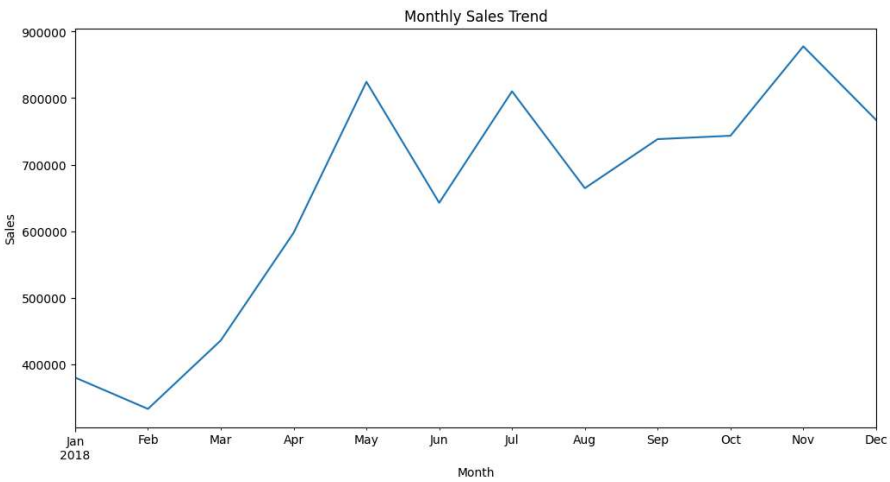
```
# Distribution of Sales
plt.figure(figsize=(10, 6))
sns.histplot(data['Sales'], kde=True)
plt.title('Distribution of Sales')
plt.xlabel('Sales')
plt.ylabel('Frequency')
plt.show()
```



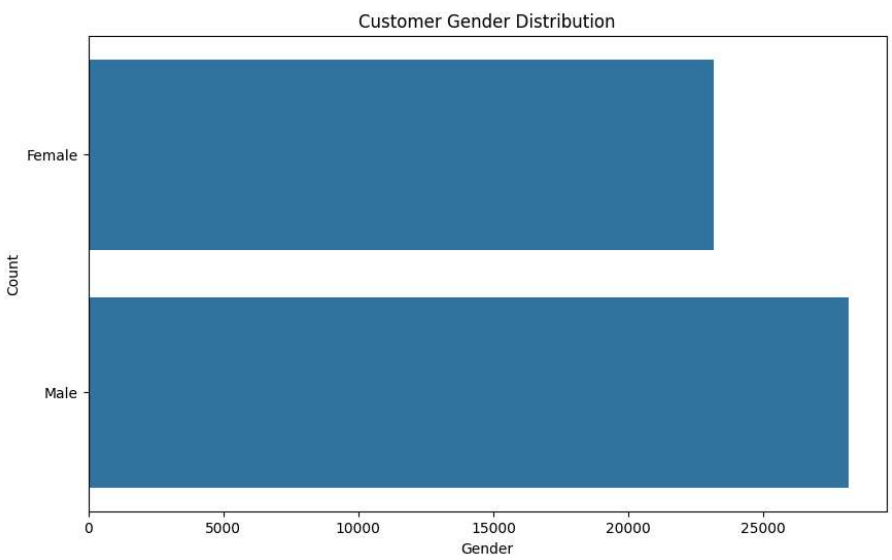

```
# Distribution of Profit
plt.figure(figsize=(10, 6))
sns.histplot(data['Profit'], kde=True)
plt.title('Distribution of Profit')
plt.xlabel('Profit')
plt.ylabel('Frequency')
plt.show()
```



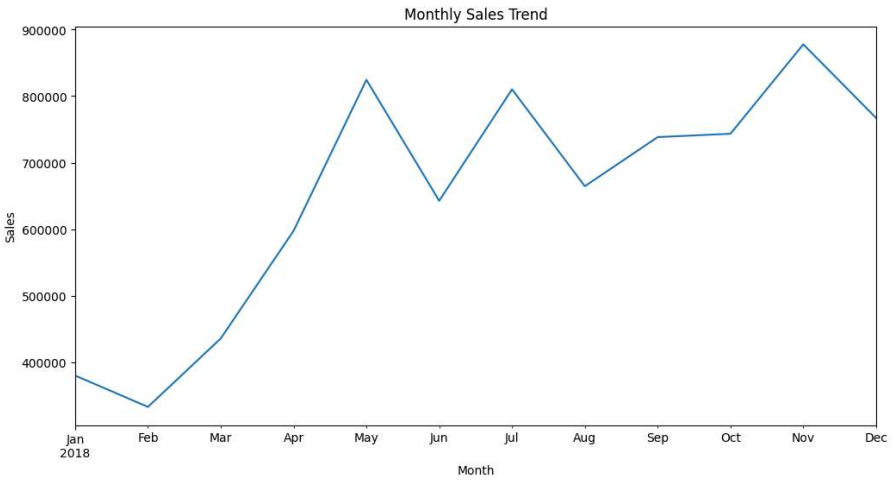
```
monthly_sales = data.set_index('Order_Date').resample('M')['Sales'].sum()

# Plotting the monthly sales trend
plt.figure(figsize=(12, 6))
monthly_sales.plot(kind='line')
plt.title('Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.show()
```



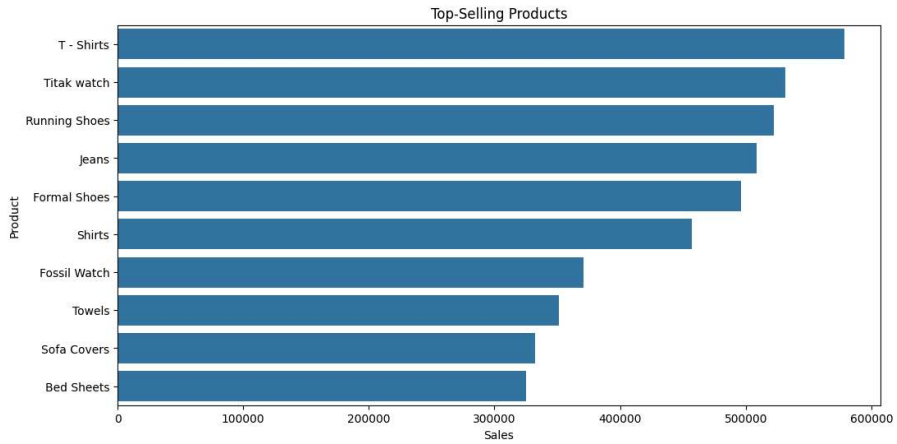

```
# Distribution of Gender
plt.figure(figsize=(10, 6))
sns.countplot(data['Gender'])
plt.title('Customer Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```




```
# Group by month and sum sales
monthly_sales = data.set_index('Order_Date').resample('M')['Sales'].sum()

# Plotting the monthly sales trend
plt.figure(figsize=(12, 6))
monthly_sales.plot(kind='line')
plt.title('Monthly Sales Trend')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.show()
```

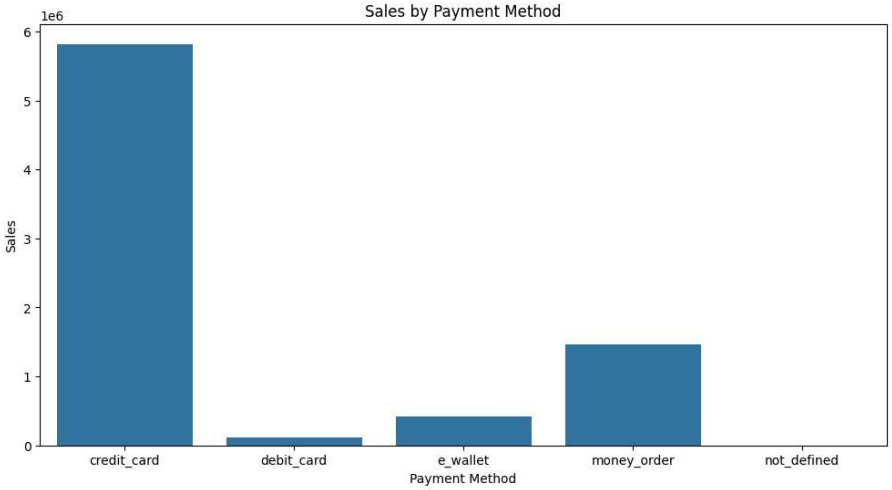


```
top_products = data.groupby('Product').sum(numeric_only=True)['Sales'].sort_values(ascending=False).head(10)

# Plotting top-selling products
plt.figure(figsize=(12, 6))
sns.barplot(x=top_products.values, y=top_products.index)
plt.title('Top-Selling Products')
plt.xlabel('Sales')
plt.ylabel('Product')
plt.show()
```




```
plt.figure(figsize=(10, 6))
sns.countplot(data['Gender'])
plt.title('Customer Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```

```
payment_sales = data.groupby('Payment_method').sum(numeric_only=True)['Sales']

# Plotting sales by payment method
plt.figure(figsize=(12, 6))
sns.barplot(x=payment_sales.index, y=payment_sales.values)
plt.title('Sales by Payment Method')
plt.xlabel('Payment Method')
plt.ylabel('Sales')
plt.show()
```




```
device_sales = data.groupby('Device_Type').sum(numeric_only=True)['Sales']

# Plotting sales by device type
plt.figure(figsize=(12, 6))
sns.barplot(x=device_sales.index, y=device_sales.values)
plt.title('Sales by Device Type')
plt.xlabel('Device Type')
plt.ylabel('Sales')
plt.show()
```