```
import pandas as pd
import numpy as np

# Read the CSV file from the URL
data = pd.read_csv("https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv")

# Display the first few rows
print("First few rows:")
print(data.head())

# Provide data information
print("\nData information:")
print(data.info())

# Display the last few rows
print("\nLast few rows:")
print(data.tail())

# Calculate summary statistics
print("\nSummary statistics:")
print(data.describe())
```

```
First few rows:
   Column1  Order ID       Product Category               Product  \
0        0    295665  Laptops and Computers    Macbook Pro Laptop
1        1    295666        Home Appliances    LG Washing Machine
2        2    295667        Charging Cables  USB-C Charging Cable
3        3    295668               Monitors      27in FHD Monitor
4        4    295669        Charging Cables  USB-C Charging Cable

   Quantity Ordered  Price Each        Order Date  \
0                 1     1700.00  30-12-2019 00:01
1                 1      600.00  29-12-2019 07:03
2                 1       11.95  12-12-2019 18:21
3                 1      149.99  22-12-2019 15:13
4                 1       11.95  18-12-2019 12:38

                        Purchase Address  Month    Sales            City  \
0  136 Church St, New York City, NY 10001     12  1700.00   New York City
1     562 2nd St, New York City, NY 10001     12   600.00   New York City
2    277 Main St, New York City, NY 10001     12    11.95   New York City
3     410 6th St, San Francisco, CA 94016     12   149.99   San Francisco
4           43 Hill St, Atlanta, GA 30301     12    11.95         Atlanta

   Hour Time of Day
0     0       Night
1     7     Morning
2    18     Evening
3    15   Afternoon
4    12   Afternoon

Data information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 185950 entries, 0 to 185949
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Column1           185950 non-null  int64
 1   Order ID          185950 non-null  int64
 2   Product Category  185950 non-null  object
 3   Product           185950 non-null  object
 4   Quantity Ordered  185950 non-null  int64
 5   Price Each        185950 non-null  float64
 6   Order Date        185950 non-null  object
 7   Purchase Address  185950 non-null  object
 8   Month             185950 non-null  int64
 9   Sales             185950 non-null  float64
 10  City              185950 non-null  object
 11  Hour              185950 non-null  int64
 12  Time of Day       185950 non-null  object
dtypes: float64(2), int64(5), object(6)
memory usage: 18.4+ MB
None

Last few rows:
        Column1  Order ID Product Category                 Product  \
185945    13617    222905         Batterie  AAA Batteries (4-pack)
185946    13618    222906         Monitors        27in FHD Monitor
185947    13619    222907  Charging Cables    USB-C Charging Cable
185948    13620    222908  Charging Cables    USB-C Charging Cable
```

```
import pandas as pd

# Load the dataset
url = 'https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv'
df = pd.read_csv(url)

# Null Data Identification
null_values = df.isnull().sum()
print("Null Data Identification:")
print(null_values)

# Null Data Imputation
# Assuming you want to fill missing values in the 'Product' column with 'Unknown'
df['Product'].fillna('Unknown', inplace=True)

# Assuming you want to fill missing values in the 'Sales' column with the median value
median_sales = df['Sales'].median()
df['Sales'].fillna(median_sales, inplace=True)

# Null Data Removal
# Assuming you want to drop rows with any missing values
cleaned_df = df.dropna()

# Display the cleaned DataFrame
print("\nCleaned DataFrame:")
print(cleaned_df.head())
```

```
Null Data Identification:
Column1             0
Order ID            0
Product Category    0
Product             0
Quantity Ordered    0
Price Each          0
Order Date          0
Purchase Address    0
Month               0
Sales               0
City                0
Hour                0
Time of Day         0
dtype: int64

Cleaned DataFrame:
   Column1  Order ID       Product Category               Product  \
0        0    295665  Laptops and Computers    Macbook Pro Laptop
1        1    295666        Home Appliances    LG Washing Machine
2        2    295667        Charging Cables  USB-C Charging Cable
3        3    295668               Monitors      27in FHD Monitor
4        4    295669        Charging Cables  USB-C Charging Cable

   Quantity Ordered  Price Each        Order Date  \
0                 1     1700.00  30-12-2019 00:01
1                 1      600.00  29-12-2019 07:03
2                 1       11.95  12-12-2019 18:21
3                 1      149.99  22-12-2019 15:13
4                 1       11.95  18-12-2019 12:38

                          Purchase Address  Month    Sales            City  \
0  136 Church St, New York City, NY 10001     12  1700.00   New York City
1     562 2nd St, New York City, NY 10001     12   600.00   New York City
2    277 Main St, New York City, NY 10001     12    11.95   New York City
3     410 6th St, San Francisco, CA 94016     12   149.99   San Francisco
4           43 Hill St, Atlanta, GA 30301     12    11.95         Atlanta

   Hour Time of Day
0     0       Night
1     7     Morning
2    18     Evening
3    15   Afternoon
4    12   Afternoon
```

```
import pandas as pd

# Load the dataset
url = 'https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv'
df = pd.read_csv(url)

# Data Integrity Check: Data Consistency Verification

# Example 1: Checking if the 'Order ID' column is unique
unique_one = df['Order ID'].is_unique
print("Is 'Order ID' unique across the dataset?", unique_one)

# Example 2: Verifying consistency between 'Quantity' and 'Price Each' columns
# Calculate total price for each row and compare it with 'Quantity' * 'Price Each'
total_price_consistent = (df['Quantity Ordered'] * df['Price Each'] == df['Sales']).all()
print("Are 'Quantity Ordered' * 'Price Each' consistent with 'Sales'?", total_price_consistent)

# Example 3: Checking for consistency across different columns
# For example, verifying if 'Order Date' is in a consistent format
date_consistent = pd.to_datetime(df['Order Date'], errors='coerce').notna().all()
print("Is 'Order Date' consistent in format?", date_consistent)
```

```
Is 'Order ID' unique across the dataset? False
Are 'Quantity Ordered' * 'Price Each' consistent with 'Sales'? False
<ipython-input-5-fe3d8c0ef89f>:20: UserWarning: Parsing dates in %d-%m-%Y %H:%M format when dayfirst=False (the default) was specifi
  date_consistent = pd.to_datetime(df['Order Date'], errors='coerce').notna().all()
Is 'Order Date' consistent in format? True
```

```
import pandas as pd

# Load the dataset
url = 'https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv'
df = pd.read_csv(url)

# Reshaping Rows and Columns: Transposing Data

# Transpose the DataFrame using .T attribute
transposed_df = df.T

# Display the transposed DataFrame
print("Transposed DataFrame:")
print(transposed_df.head())
```

```
Transposed DataFrame:
                                    0                   1       \
Column1                             0                   1
Order ID                       295665              295666
Product Category  Laptops and Computers     Home Appliances
Product              Macbook Pro Laptop  LG Washing Machine
Quantity Ordered                    1                   1

                                   2                 3       \
Column1                            2                 3
Order ID                      295667            295668
Product Category       Charging Cables          Monitors
Product           USB-C Charging Cable  27in FHD Monitor
Quantity Ordered                   1                 1

                                   4                      5       \
Column1                            4                      5
Order ID                      295669                 295670
Product Category       Charging Cables               Batterie
Product           USB-C Charging Cable  AA Batteries (4-pack)
Quantity Ordered                   1                      1

                                   6                     7       \
Column1                            6                     7
Order ID                      295671                295672
Product Category       Charging Cables       Charging Cables
Product           USB-C Charging Cable  USB-C Charging Cable

Quantity Ordered                   1                     2

                                         8                       9       ...  \
Column1                                  8                       9  ...
Order ID                            295673                  295674  ...
Product Category             Audio Devices                Batterie  ...
Product           Bose SoundSport Headphones  AAA Batteries (4-pack)  ...
Quantity Ordered                         1                       4  ...

                                    185940                    185941  \
Column1                              13612                     13613
Order ID                            222901                    222902
```

```
Product Category                Batterie           Charging Cables
Product           AAA Batteries (4-pack)  Lightning Charging Cable
Quantity Ordered                       1                         1

                                  185942                    185943  \
Column1                            13614                     13615
Order ID                          222903                    222903
Product Category  Phones and Accessories           Charging Cables
Product                           iPhone  Lightning Charging Cable
Quantity Ordered                       1                         1

                                 185944                  185945  \
Column1                           13616                   13617
Order ID                         222904                  222905
Product Category  Laptops and Computers                Batterie
Product              Macbook Pro Laptop  AAA Batteries (4-pack)
Quantity Ordered                      1                       1
```

```
import pandas as pd

# Create the first dataset (original dataset)
df1 = pd.read_csv('https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv')

# Display the first dataset
print("First Dataset:")
print(df1.head())

# Create the second dataset
data = {
    'Order ID': [295670, 295671, 295672],
    'Product': ['Keyboard', 'Mouse', 'Monitor'],
    'Quantity Ordered': [1, 1, 1],
    'Price Each': [50, 20, 200],
    'Order Date': ['2020-01-01', '2020-01-02', '2020-01-03'],
    'Purchase Address': ['123 Main St', '456 Elm St', '789 Oak St']
}
df2 = pd.DataFrame(data)

# Display the second dataset
print("\nSecond Dataset:")
print(df2)

# Combining Datasets: Merging based on a common column
merged_df = pd.merge(df1, df2, on='Order ID', how='inner')

# Display the merged DataFrame
print("\nMerged DataFrame:")
print(merged_df)
```

```
First Dataset:
   Column1  Order ID       Product Category               Product  \
0        0    295665  Laptops and Computers    Macbook Pro Laptop
1        1    295666        Home Appliances    LG Washing Machine
2        2    295667        Charging Cables  USB-C Charging Cable
3        3    295668               Monitors      27in FHD Monitor
4        4    295669        Charging Cables  USB-C Charging Cable

   Quantity Ordered  Price Each        Order Date  \
0                 1     1700.00  30-12-2019 00:01
1                 1      600.00  29-12-2019 07:03
2                 1       11.95  12-12-2019 18:21
3                 1      149.99  22-12-2019 15:13
4                 1       11.95  18-12-2019 12:38

                          Purchase Address  Month    Sales            City  \
0  136 Church St, New York City, NY 10001     12  1700.00   New York City
1     562 2nd St, New York City, NY 10001     12   600.00   New York City
2    277 Main St, New York City, NY 10001     12    11.95   New York City
3     410 6th St, San Francisco, CA 94016     12   149.99   San Francisco
4           43 Hill St, Atlanta, GA 30301     12    11.95         Atlanta

   Hour Time of Day
0     0       Night
1     7     Morning
2    18     Evening
3    15   Afternoon
4    12   Afternoon

Second Dataset:
   Order ID   Product  Quantity Ordered  Price Each  Order Date  \
0    295670  Keyboard                 1          50  2020-01-01
1    295671     Mouse                 1          20  2020-01-02
2    295672   Monitor                 1         200  2020-01-03

  Purchase Address
```

```
0      123 Main St
1       456 Elm St
2       789 Oak St

Merged DataFrame:
   Column1  Order ID Product Category              Product_x  \
0        5    295670         Batterie  AA Batteries (4-pack)
1        6    295671  Charging Cables   USB-C Charging Cable
2        7    295672  Charging Cables   USB-C Charging Cable

   Quantity Ordered_x  Price Each_x      Order Date_x  \
0                   1          3.84  31-12-2019 22:58
1                   1         11.95  16-12-2019 15:10
2                   2         11.95  13-12-2019 09:29

                       Purchase Address_x  Month  Sales            City  \
0  200 Jefferson St, New York City, NY 10001     12   3.84   New York City
1            928 12th St, Portland, OR 97035     12  11.95        Portland
2           813 Hickory St, Dallas, TX 75001     12  23.90          Dallas

   Hour Time of Day Product_y  Quantity Ordered_y  Price Each_y Order Date_y  \
0    22     Evening  Keyboard                   1            50   2020-01-01
```

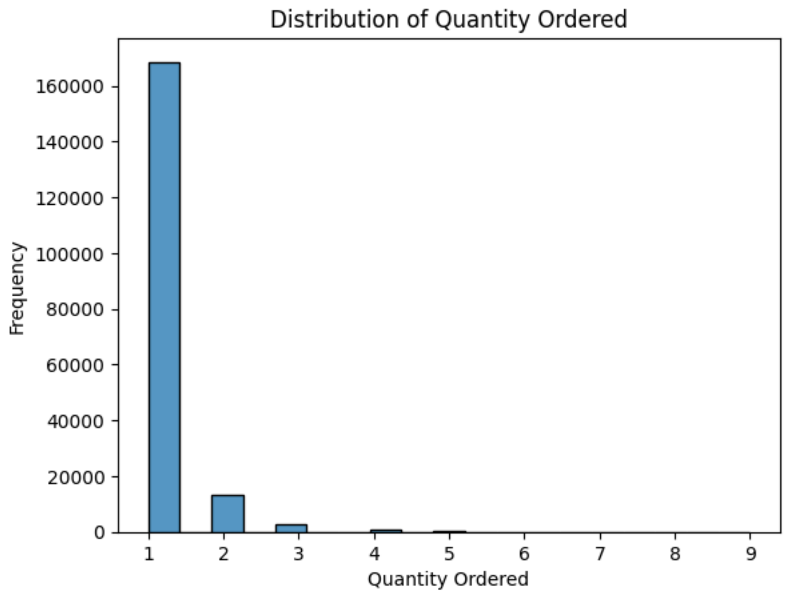
```
import pandas as pd

# Load the dataset directly from the URL
df = pd.read_csv('https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv')

# Display the first few rows of the dataset
print("First Few Rows of the Dataset:")
print(df.head())

# Grouping Data: Grouping dataset rows based on specific criteria
grouped_data = df.groupby('Product')

# Aggregating Data: Computing summary statistics for grouped data
summary_statistics = grouped_data.agg({
    'Quantity Ordered': 'sum',   # Total quantity ordered
    'Price Each': 'mean',        # Average price each
    'Sales': 'sum'               # Total sales
})

# Displaying the summary statistics
print("\nSummary Statistics for Grouped Data:")
print(summary_statistics)
```

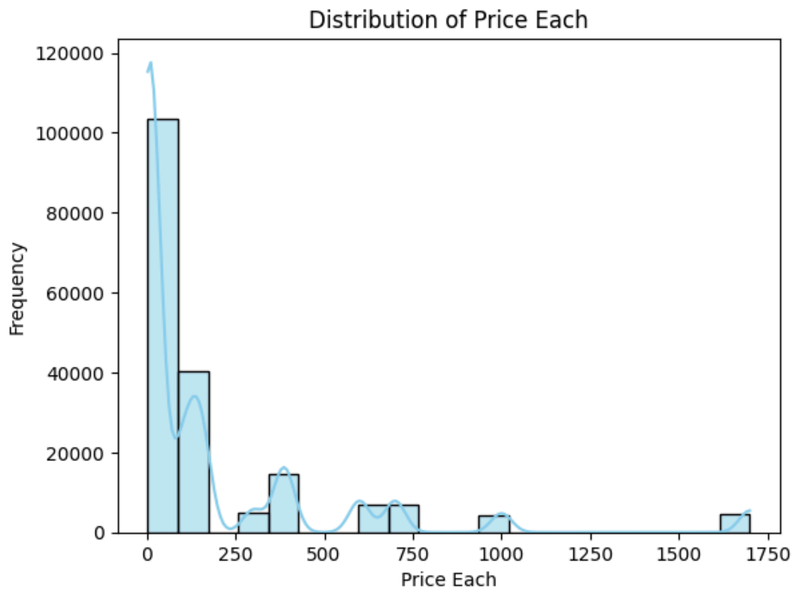
```
First Few Rows of the Dataset:
   Column1  Order ID       Product Category               Product  \
0        0    295665  Laptops and Computers    Macbook Pro Laptop
1        1    295666        Home Appliances    LG Washing Machine
2        2    295667        Charging Cables  USB-C Charging Cable
3        3    295668               Monitors      27in FHD Monitor
4        4    295669        Charging Cables  USB-C Charging Cable

   Quantity Ordered  Price Each        Order Date  \
0                 1     1700.00  30-12-2019 00:01
1                 1      600.00  29-12-2019 07:03
2                 1       11.95  12-12-2019 18:21
3                 1      149.99  22-12-2019 15:13
4                 1       11.95  18-12-2019 12:38

                      Purchase Address  Month    Sales            City  \
0  136 Church St, New York City, NY 10001     12  1700.00   New York City
1     562 2nd St, New York City, NY 10001     12   600.00   New York City
2    277 Main St, New York City, NY 10001     12    11.95   New York City
3     410 6th St, San Francisco, CA 94016     12   149.99   San Francisco
4           43 Hill St, Atlanta, GA 30301     12    11.95         Atlanta

   Hour Time of Day
0     0       Night
1     7     Morning
2    18     Evening
3    15   Afternoon
4    12   Afternoon

Summary Statistics for Grouped Data:
                            Quantity Ordered  Price Each       Sales
Product
20in Monitor                            4129      109.99   454148.71
27in 4K Gaming Monitor                  6244      389.99  2435097.56
27in FHD Monitor                        7550      149.99  1132424.50
34in Ultrawide Monitor                  6199      379.99  2355558.01
AA Batteries (4-pack)                  27635        3.84   106118.40
AAA Batteries (4-pack)                 31017        2.99    92740.83
Apple Airpods Headphones               15661      150.00  2349150.00
Bose SoundSport Headphones             13457       99.99  1345565.43
```

```
Flatscreen TV                        4819      300.00  1445700.00
Google Phone                         5532      600.00  3319200.00
LG Dryer                              646      600.00   387600.00
LG Washing Machine                    666      600.00   399600.00
Lightning Charging Cable            23217       14.95   347094.15
Macbook Pro Laptop                   4728     1700.00  8037600.00
ThinkPad Laptop                      4130      999.99  4129958.70
USB-C Charging Cable                23975       11.95   286501.25
Vareebadd Phone                      2068      400.00   827200.00
Wired Headphones                    20557       11.99   246478.43
iPhone                               6849      700.00  4794300.00
```

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

# Load the dataset directly from the URL
df = pd.read_csv('https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv')

# Univariate Analysis: Analyzing individual variables
print("\n UNIVARIATE ANALYSIS\n")

# 1. Distribution of Quantity Ordered
sns.histplot(df['Quantity Ordered'])
plt.title('Distribution of Quantity Ordered')
plt.xlabel('Quantity Ordered')
plt.ylabel('Frequency')
plt.show()

# 2. Distribution of Price Each
sns.histplot(df['Price Each'], kde=True, bins=20, color='skyblue')
plt.title('Distribution of Price Each')
plt.xlabel('Price Each')
plt.ylabel('Frequency')
plt.show()
```

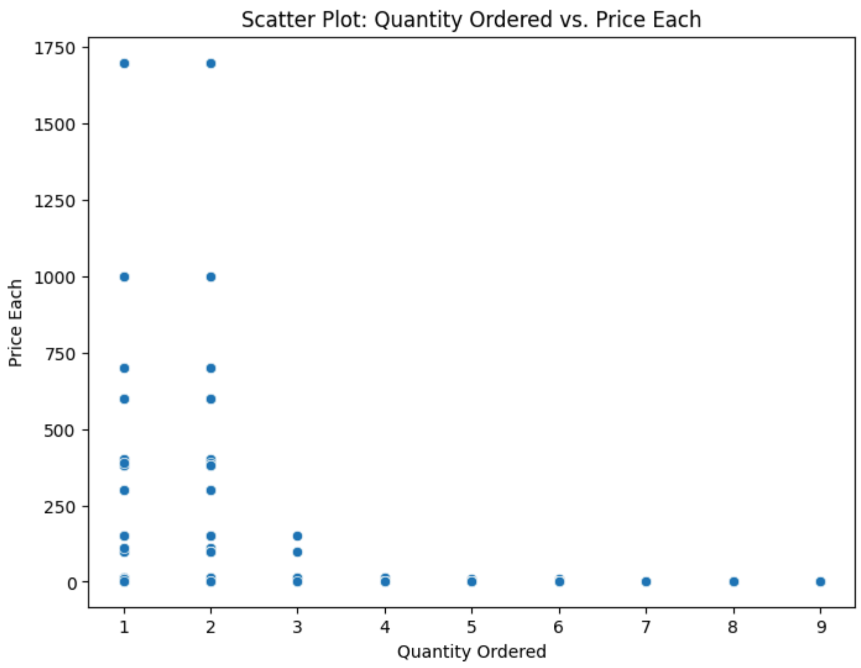
```
UNIVARIATE ANALYSIS
```

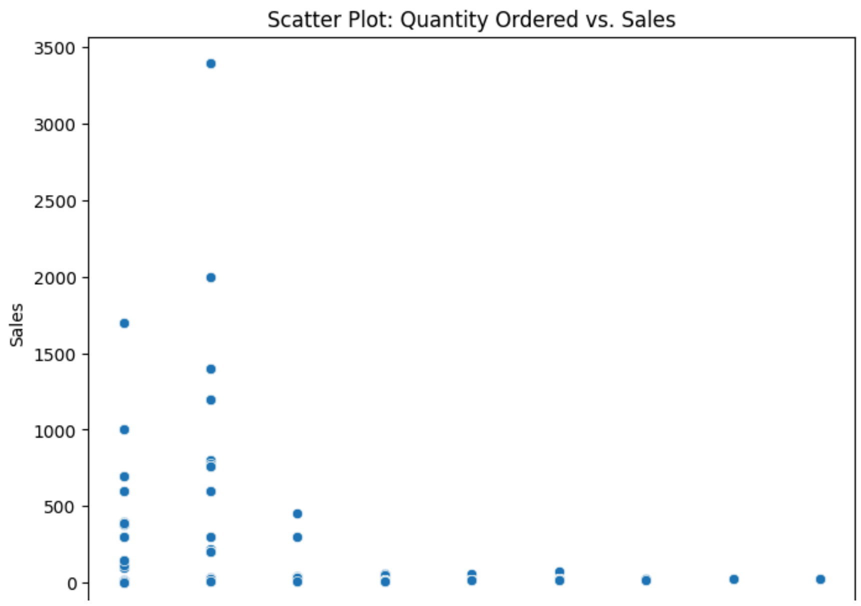




```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset directly from the URL
df = pd.read_csv('https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv')

# Bivariate Analysis: Analyzing the relationship between two variables
print("\n BIVARIATE ANALYSIS")
# 1. Scatter plot for Quantity Ordered vs. Price Each
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Quantity Ordered', y='Price Each')
plt.title('Scatter Plot: Quantity Ordered vs. Price Each')
plt.xlabel('Quantity Ordered')
plt.ylabel('Price Each')
plt.show()

# 2. Scatter plot for Quantity Ordered vs. Sales
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Quantity Ordered', y='Sales')
plt.title('Scatter Plot: Quantity Ordered vs. Sales')
plt.xlabel('Quantity Ordered')
plt.ylabel('Sales')
plt.show()

# 3. Scatter plot for Price Each vs. Sales
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Price Each', y='Sales')
plt.title('Scatter Plot: Price Each vs. Sales')
plt.xlabel('Price Each')
plt.ylabel('Sales')
plt.show()
```

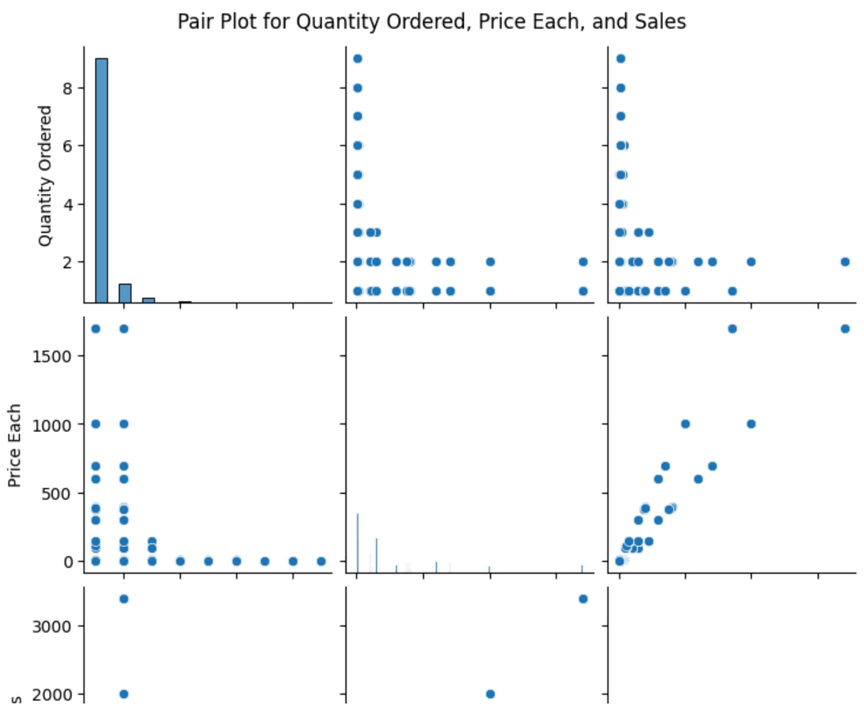
```
BIVARIATE ANALYSIS
```





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

# Load the dataset directly from the URL
df = pd.read_csv('https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv')

# Multivariate Analysis: Analyzing relationships between multiple variables
print("\n MULTIVARIATE ANALYSIS \n")
# Pair plot for selected variables
selected_variables = ['Quantity Ordered', 'Price Each', 'Sales']
sns.pairplot(df[selected_variables])
plt.suptitle('Pair Plot for Quantity Ordered, Price Each, and Sales', y=1.02)
plt.show()
```

```
MULTIVARIATE ANALYSIS
```



```
import pandas as pd

# Load the dataset
url = 'https://raw.githubusercontent.com/iamnaofil/E-commerce-Sales-Analysis/main/Sales%20Data%20Analysis.csv'
df = pd.read_csv(url)

# Assuming 'Order ID' is the unique identifier for orders and 'City' represents the location information
# Aggregate user interactions based on 'Order ID' and 'City' to create user profiles
user_profiles = df.groupby(['Order ID', 'City']).agg({
    'Sales': 'sum',                 # Total sales
    'Quantity Ordered': 'sum',      # Total quantity ordered
    # Add more interactions as needed
})
```