In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [3]:
```
pip install pandas openpyxl
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-pack
ages (2.2.2)
Requirement already satisfied: openpyxl in /usr/local/lib/python3.11/dist-pa
ckages (3.1.5)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/di
st-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyth
on3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dis
t-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/d
ist-packages (from pandas) (2025.2)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.11/dist-
packages (from openpyxl) (2.0.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-pa
ckages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

In [5]:
```
df2 = pd.read_excel("ADIDAS.xlsx", engine="openpyxl")

df2.to_csv("ADIDAS.xlsx", index=False, header=True)
```

In [6]:
```
df2
```

Out[6]:

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Ur |
|---|---|---|---|---|---|---|---|
| **0** | NaN | NaN | Adidas Sales Database | NaN | NaN | NaN | |
| **1** | NaN | NaN | NaN | NaN | NaN | NaN | |
| **2** | NaN | NaN | NaN | NaN | NaN | NaN | |
| **3** | NaN | Retailer | Retailer ID | Invoice Date | Region | State | |
| **4** | NaN | Foot Locker | 1185732 | 2020-01-01 00:00:00 | Northeast | New York | |
| **...** | ... | ... | ... | ... | ... | ... | |
| **9647** | NaN | Foot Locker | 1185732 | 2021-01-24 00:00:00 | Northeast | New Hampshire | Ma |
| **9648** | NaN | Foot Locker | 1185732 | 2021-01-24 00:00:00 | Northeast | New Hampshire | Ma |
| **9649** | NaN | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Ma |
| **9650** | NaN | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Ma |
| **9651** | NaN | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Ma |

9652 rows × 14 columns

In [7]:

```
df2.columns = ["index",
    "Retailer", "Retailer ID", "Invoice Date", "Region", "State", "City",
    "Product", "Price per Unit", "Units Sold", "Total Sales",
    "Operating Profit", "Operating Margin", "Sales Method"
]
```

In [8]:

```
df2
```

Out[8]:

| | index | Retailer | Retailer ID | Invoice Date | Region | State | City | Pro |
|---|---|---|---|---|---|---|---|---|
| **0** | NaN | NaN | Adidas Sales Database | NaN | NaN | NaN | NaN | |
| **1** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| **2** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| **3** | NaN | Retailer | Retailer ID | Invoice Date | Region | State | City | Pr |
| **4** | NaN | Foot Locker | 1185732 | 2020-01-01 00:00:00 | Northeast | New York | New York | S Foo |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **9647** | NaN | Foot Locker | 1185732 | 2021-01-24 00:00:00 | Northeast | New Hampshire | Manchester | Ap |
| **9648** | NaN | Foot Locker | 1185732 | 2021-01-24 00:00:00 | Northeast | New Hampshire | Manchester | Wo Ap |
| **9649** | NaN | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Manchester | S Foo |
| **9650** | NaN | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Manchester | At Foo |
| **9651** | NaN | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Manchester | Wo S Foo |

9652 rows × 14 columns

In [9]:
```python
print("Dataset shape:", df2.shape)
```

```
Dataset shape: (9652, 14)
```

In [10]:
```python
df2.drop("index", axis=1, inplace=True)
```

In [11]:
```python
df2
```

Out[11]:

| | Retailer | Retailer ID | Invoice Date | Region | State | City | Product |
|---|---|---|---|---|---|---|---|
| **0** | NaN | Adidas Sales Database | NaN | NaN | NaN | NaN | NaN |
| **1** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **2** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **3** | Retailer | Retailer ID | Invoice Date | Region | State | City | Product |
| **4** | Foot Locker | 1185732 | 2020-01-01 00:00:00 | Northeast | New York | New York | Men's Street Footwear |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **9647** | Foot Locker | 1185732 | 2021-01-24 00:00:00 | Northeast | New Hampshire | Manchester | Men's Apparel |
| **9648** | Foot Locker | 1185732 | 2021-01-24 00:00:00 | Northeast | New Hampshire | Manchester | Women's Apparel |
| **9649** | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Manchester | Men's Street Footwear |
| **9650** | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Manchester | Men's Athletic Footwear |
| **9651** | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Manchester | Women's Street Footwear |

9652 rows × 13 columns

In [12]:
```
df = df2.copy()
```

In [13]:
```
df
```

Out[13]:

| | Retailer | Retailer ID | Invoice Date | Region | State | City | Product |
|---|---|---|---|---|---|---|---|
| **0** | NaN | Adidas Sales Database | NaN | NaN | NaN | NaN | NaN |
| **1** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **2** | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **3** | Retailer | Retailer ID | Invoice Date | Region | State | City | Product |
| **4** | Foot Locker | 1185732 | 2020-01-01 00:00:00 | Northeast | New York | New York | Men's Street Footwear |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **9647** | Foot Locker | 1185732 | 2021-01-24 00:00:00 | Northeast | New Hampshire | Manchester | Men's Apparel |
| **9648** | Foot Locker | 1185732 | 2021-01-24 00:00:00 | Northeast | New Hampshire | Manchester | Women's Apparel |
| **9649** | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Manchester | Men's Street Footwear |
| **9650** | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Manchester | Men's Athletic Footwear |
| **9651** | Foot Locker | 1185732 | 2021-02-22 00:00:00 | Northeast | New Hampshire | Manchester | Women's Street Footwear |

9652 rows × 13 columns

In [14]:
```
df.dtypes
```

Out[14]:

| | 0 |
|---|---|
| **Retailer** | object |
| **Retailer ID** | object |
| **Invoice Date** | object |
| **Region** | object |
| **State** | object |
| **City** | object |
| **Product** | object |
| **Price per Unit** | object |
| **Units Sold** | object |
| **Total Sales** | object |
| **Operating Profit** | object |
| **Operating Margin** | object |
| **Sales Method** | object |

**dtype:** object

In [30]:

```python
numeric_cols = [
    "Price per Unit", "Units Sold", "Total Sales",
    "Operating Profit", "Operating Margin"
]

for col in numeric_cols:
    df[col] = pd.to_numeric(df[col].astype(str).str.replace(',', '').str.str

# Verify the result
print(df.dtypes)
```

```
Retailer                    object
Retailer ID                 object
Invoice Date        datetime64[ns]
Region                      object
State                       object
City                        object
Product                     object
Price per Unit             float64
Units Sold                   int64
Total Sales                float64
Operating Profit           float64
Operating Margin           float64
Sales Method                object
dtype: object
```

```
<ipython-input-30-f53f300c1a82>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[col] = pd.to_numeric(df[col].astype(str).str.replace(',', '').str.strip
())
<ipython-input-30-f53f300c1a82>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[col] = pd.to_numeric(df[col].astype(str).str.replace(',', '').str.strip
())
<ipython-input-30-f53f300c1a82>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[col] = pd.to_numeric(df[col].astype(str).str.replace(',', '').str.strip
())
<ipython-input-30-f53f300c1a82>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[col] = pd.to_numeric(df[col].astype(str).str.replace(',', '').str.strip
())
<ipython-input-30-f53f300c1a82>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[col] = pd.to_numeric(df[col].astype(str).str.replace(',', '').str.strip
())
```

In [31]:
```
df["Invoice Date"] = pd.to_datetime(df["Invoice Date"], errors="coerce")
```

```
<ipython-input-31-63e3fb4dbc3c>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df["Invoice Date"] = pd.to_datetime(df["Invoice Date"], errors="coerce")
```

In [32]:
```
df.isna().sum()
```

```
Out[32]:
```

| | 0 |
|---|---|
| **Retailer** | 0 |
| **Retailer ID** | 0 |
| **Invoice Date** | 0 |
| **Region** | 0 |
| **State** | 0 |
| **City** | 0 |
| **Product** | 0 |
| **Price per Unit** | 0 |
| **Units Sold** | 0 |
| **Total Sales** | 0 |
| **Operating Profit** | 0 |
| **Operating Margin** | 0 |
| **Sales Method** | 0 |

**dtype:** int64

```
In [33]: df.dropna(inplace=True)
         df.reset_index(drop=True, inplace=True)
```

```
<ipython-input-33-5110ff77c3d8>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.dropna(inplace=True)
```

```
In [34]: df.isna().sum()
```

```
Out[34]:
```

| | 0 |
|---|---|
| **Retailer** | 0 |
| **Retailer ID** | 0 |
| **Invoice Date** | 0 |
| **Region** | 0 |
| **State** | 0 |
| **City** | 0 |
| **Product** | 0 |
| **Price per Unit** | 0 |
| **Units Sold** | 0 |
| **Total Sales** | 0 |
| **Operating Profit** | 0 |
| **Operating Margin** | 0 |
| **Sales Method** | 0 |

**dtype:** int64

```
In [35]: df.duplicated().sum()
```

```
Out[35]: np.int64(0)
```

```
In [36]: df['Units Sold'] =df['Units Sold'].astype(int)
```

```
<ipython-input-36-8466fce0d082>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Units Sold'] =df['Units Sold'].astype(int)
```

```
In [37]: print(df.dtypes)
```

```
Retailer                    object
Retailer ID                 object
Invoice Date        datetime64[ns]
Region                      object
State                       object
City                        object
Product                     object
Price per Unit             float64
Units Sold                   int64
Total Sales                float64
Operating Profit           float64
Operating Margin           float64
Sales Method                object
dtype: object
```

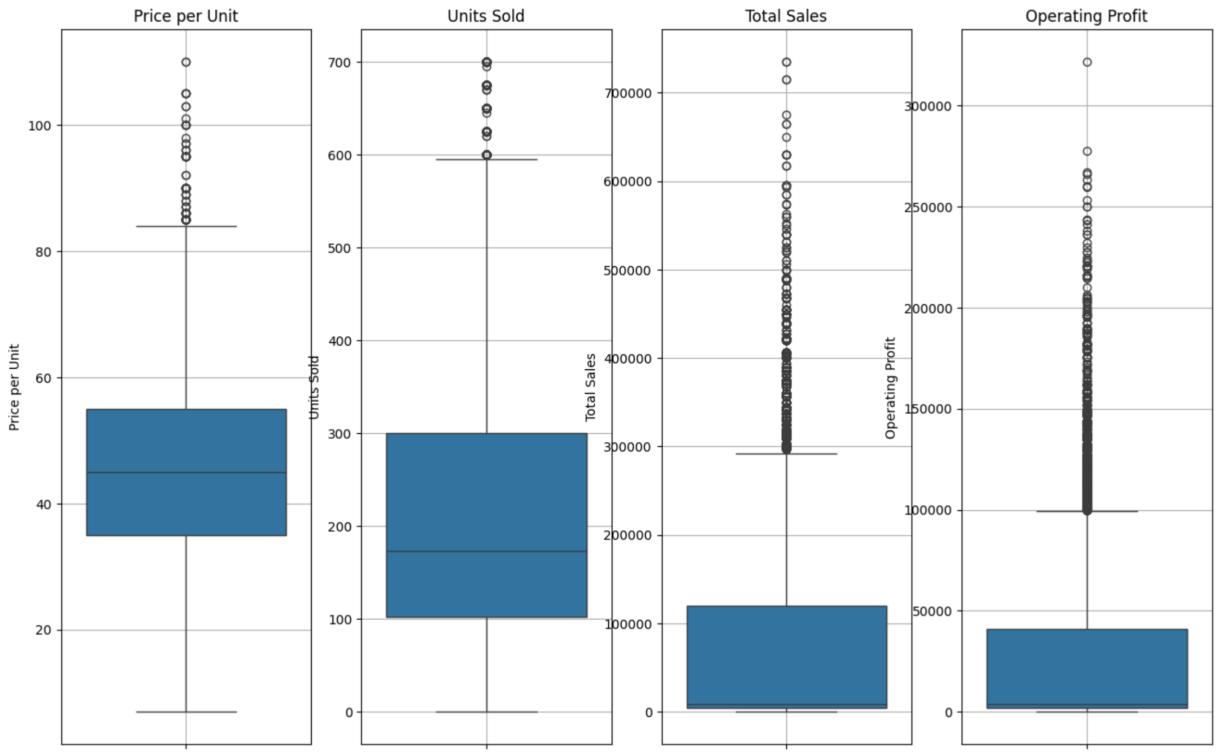
In [38]:
```
df.describe(include=[np.number]).style.background_gradient(cmap="YlGnBu")
```

Out[38]:

| | Price per Unit | Units Sold | Total Sales | Operating Profit | Operating Margin |
|---|---|---|---|---|---|
| **count** | 9140.000000 | 9140.000000 | 9140.000000 | 9140.000000 | 9140.000000 |
| **mean** | 44.720460 | 224.164223 | 72956.687637 | 26628.911264 | 0.425287 |
| **std** | 14.594924 | 165.526529 | 111750.442367 | 41158.473480 | 0.096851 |
| **min** | 7.000000 | 0.000000 | 0.000000 | 0.000000 | 0.100000 |
| **25%** | 35.000000 | 102.000000 | 4070.250000 | 1831.765000 | 0.350000 |
| **50%** | 45.000000 | 173.000000 | 8614.500000 | 3936.120000 | 0.410000 |
| **75%** | 55.000000 | 300.000000 | 120000.000000 | 40781.250000 | 0.500000 |
| **max** | 110.000000 | 700.000000 | 735000.000000 | 321750.000000 | 0.800000 |

In [39]:
```
import seaborn as sns
import matplotlib.pyplot as plt

cols_to_plot = df.select_dtypes(include='number').columns
fig, axes = plt.subplots(1, 4, figsize=(16, 10))

for i, col in enumerate(cols_to_plot):
  sns.boxplot(data=df, y=col, ax=axes[i])
  axes[i].set_title(col)
  axes[i].grid(True)

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

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-39-efeb47626901> in <cell line: 0>()
      6
      7 for i, col in enumerate(cols_to_plot):
----> 8   sns.boxplot(data=df, y=col, ax=axes[i])
      9    # vertical boxplot
     10   axes[i].set_title(col)

IndexError: index 4 is out of bounds for axis 0 with size 4
```



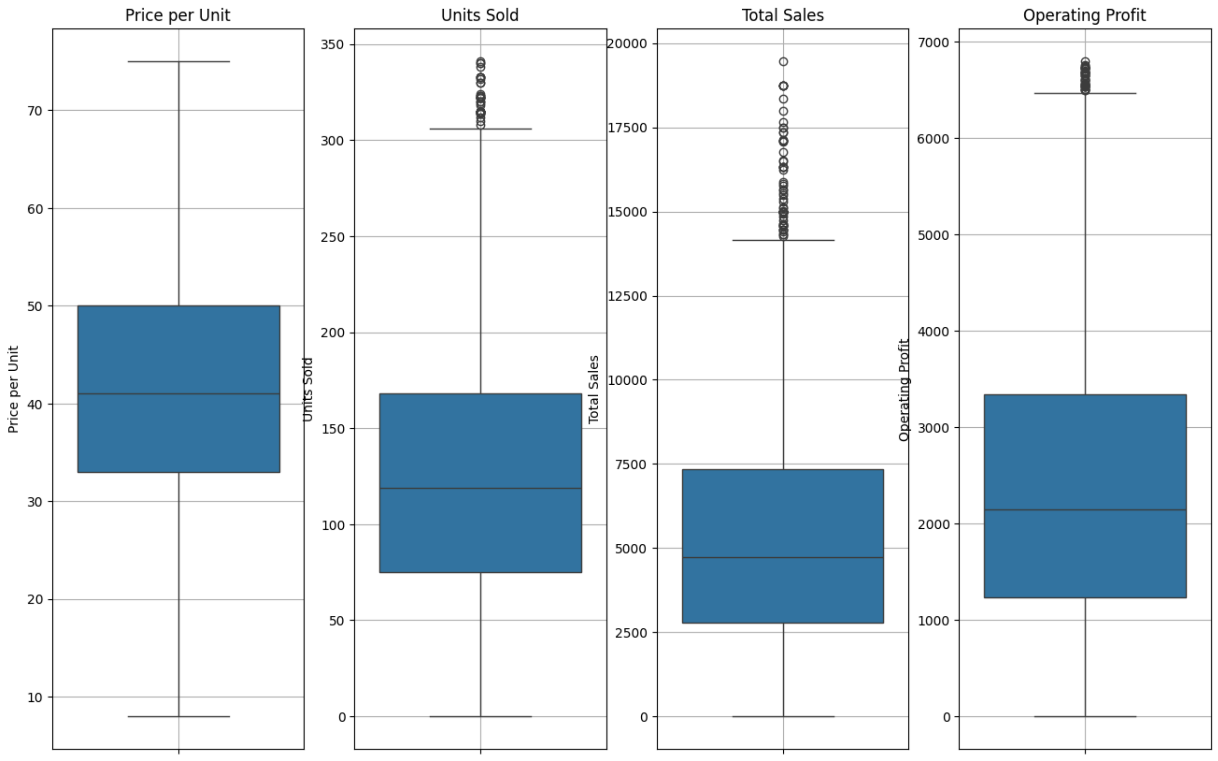
In [48]:
```python
# Select numeric columns
numeric_cols = df.select_dtypes(include='number').columns

for col in numeric_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
    print(f"{col}: {len(outliers)} outliers")
```

```
Price per Unit: 0 outliers
Units Sold: 39 outliers
Total Sales: 64 outliers
Operating Profit: 42 outliers
Operating Margin: 0 outliers
```

In [45]:
```python
for col in numeric_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Filter out the outliers
    df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]
```

In [46]:
```
import seaborn as sns
import matplotlib.pyplot as plt

cols_to_plot = df.select_dtypes(include='number').columns
fig, axes = plt.subplots(1, 4, figsize=(16, 10))

for i, col in enumerate(cols_to_plot):
  sns.boxplot(data=df, y=col, ax=axes[i])
  axes[i].set_title(col)
  axes[i].grid(True)

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

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-46-efeb47626901> in <cell line: 0>()
      6
      7 for i, col in enumerate(cols_to_plot):
----> 8   sns.boxplot(data=df, y=col, ax=axes[i])
      9    # vertical boxplot
     10   axes[i].set_title(col)

IndexError: index 4 is out of bounds for axis 0 with size 4
```



In [47]:
```
df.describe(include=np.number).style.background_gradient(cmap="YlGnBu")
```

Out[47]:

| | Price per Unit | Units Sold | Total Sales | Operating Profit | Operating Margin |
|---|---|---|---|---|---|
| **count** | 5379.000000 | 5379.000000 | 5379.000000 | 5379.000000 | 5379.000000 |
| **mean** | 41.416992 | 126.783975 | 5353.549916 | 2434.308987 | 0.458305 |
| **std** | 12.743527 | 64.228787 | 3250.398982 | 1490.017459 | 0.078460 |
| **min** | 8.000000 | 0.000000 | 0.000000 | 0.000000 | 0.240000 |
| **25%** | 33.000000 | 75.000000 | 2794.000000 | 1238.800000 | 0.400000 |
| **50%** | 41.000000 | 119.000000 | 4719.000000 | 2147.000000 | 0.460000 |
| **75%** | 50.000000 | 168.000000 | 7344.000000 | 3340.950000 | 0.510000 |
| **max** | 75.000000 | 341.000000 | 19462.000000 | 6794.280000 | 0.670000 |

In [49]:
```
df
```

Out[49]:

| | Retailer | Retailer ID | Invoice Date | Region | State | City | Product |
|---|---|---|---|---|---|---|---|
| **238** | Foot Locker | 1185732 | 2021-11-06 | Northeast | Pennsylvania | Philadelphia | Women's Athletic Footwear |
| **525** | Foot Locker | 1185732 | 2020-11-01 | Midwest | Minnesota | Minneapolis | Women's Athletic Footwear |
| **531** | Foot Locker | 1185732 | 2020-11-07 | Midwest | Minnesota | Minneapolis | Women's Athletic Footwear |
| **717** | Foot Locker | 1185732 | 2021-05-30 | Midwest | Nebraska | Omaha | Women's Athletic Footwear |
| **723** | Foot Locker | 1185732 | 2021-06-05 | Midwest | Nebraska | Omaha | Women's Athletic Footwear |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **9135** | Foot Locker | 1185732 | 2021-01-24 | Northeast | New Hampshire | Manchester | Men's Apparel |
| **9136** | Foot Locker | 1185732 | 2021-01-24 | Northeast | New Hampshire | Manchester | Women's Apparel |
| **9137** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Men's Street Footwear |
| **9138** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Men's Athletic Footwear |
| **9139** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Women's Street Footwear |

5379 rows × 13 columns

In [50]:
```
df['Product'].unique()
```

Out[50]:
```
array(["Women's Athletic Footwear", "Women's Street Footwear",
       "Men's Apparel", "Men's Athletic Footwear", "Women's Apparel",
       "Men's Street Footwear"], dtype=object)
```

In [51]:
```
df['Retailer'].unique()
```

Out[51]:
```
array(['Foot Locker', 'Amazon', 'Sports Direct', 'West Gear', 'Walmart',
       "Kohl's"], dtype=object)
```

In [52]:
```
string_cols = ['Retailer','Region', 'State', 'City', 'Product' , 'Sales Meth

for col in string_cols:
    df[col] = df[col].str.title()
```

feature engineering

```
In [53]: if 'Year' not in df.columns:
             df['Year'] = pd.to_datetime(df['Invoice Date']).dt.year
```

```
In [57]: profit_per_year = df.groupby('Year')['Operating Profit'].sum().reset_index()
         profit_per_year.columns = ['Year', 'Total_Profit_Year']
         profit_per_year
```

Out[57]:

|   | Year | Total_Profit_Year |
|---|---|---|
| **0** | 2020 | 2225195.15 |
| **1** | 2021 | 10868952.89 |

```
In [54]: profit_per_city = df.groupby('City')['Operating Profit'].sum().reset_index()
         profit_per_city.columns = ['City', 'Total_Profit_City']
```

```
In [55]: profit_per_city
```

Out[55]:

| | City | Total_Profit_City |
|---|---|---|
| **0** | Albany | 221212.42 |
| **1** | Albuquerque | 393185.41 |
| **2** | Anchorage | 158133.36 |
| **3** | Atlanta | 383372.41 |
| **4** | Baltimore | 106154.62 |
| **5** | Billings | 171634.44 |
| **6** | Birmingham | 314550.81 |
| **7** | Boise | 408981.76 |
| **8** | Boston | 202238.37 |
| **9** | Burlington | 329441.49 |
| **10** | Charleston | 354436.60 |
| **11** | Charlotte | 238761.80 |
| **12** | Cheyenne | 180864.81 |
| **13** | Chicago | 129568.61 |
| **14** | Columbus | 185938.08 |
| **15** | Dallas | 406765.70 |
| **16** | Denver | 197350.08 |
| **17** | Des Moines | 108879.87 |
| **18** | Detroit | 198413.31 |
| **19** | Fargo | 104257.22 |
| **20** | Hartford | 240920.79 |
| **21** | Honolulu | 156921.23 |
| **22** | Houston | 481015.05 |
| **23** | Indianapolis | 112943.50 |
| **24** | Jackson | 344771.57 |
| **25** | Knoxville | 373057.92 |
| **26** | Las Vegas | 372583.36 |
| **27** | Little Rock | 275542.36 |
| **28** | Los Angeles | 480560.81 |
| **29** | Louisville | 134130.23 |
| **30** | Manchester | 332666.37 |
| **31** | Miami | 237755.83 |
| **32** | Milwaukee | 111367.59 |

| | City | Total_Profit_City |
|---|---|---|
| **33** | Minneapolis | 95279.76 |
| **34** | New Orleans | 314058.48 |
| **35** | New York | 513679.95 |
| **36** | Newark | 123537.22 |
| **37** | Oklahoma City | 233158.19 |
| **38** | Omaha | 118703.41 |
| **39** | Orlando | 349857.20 |
| **40** | Philadelphia | 232903.23 |
| **41** | Phoenix | 316602.08 |
| **42** | Portland | 450539.18 |
| **43** | Providence | 191367.26 |
| **44** | Richmond | 332514.68 |
| **45** | Salt Lake City | 223225.48 |
| **46** | San Francisco | 487892.46 |
| **47** | Seattle | 162689.26 |
| **48** | Sioux Falls | 98680.71 |
| **49** | St. Louis | 128584.36 |
| **50** | Wichita | 121509.79 |
| **51** | Wilmington | 150987.56 |

In [59]:
```
df = df.merge(profit_per_city, on='City', how='left')
df = df.merge(profit_per_year, on='Year', how='left')
```

In [61]:
```
df.drop(columns=['Total_Profit_City_y'], inplace=True)
```

In [62]:
```
df.drop(columns=['Total_Profit_Year_y'], inplace=True)
```

In [63]:
```
df
```

Out[63]:

| | Retailer | Retailer ID | Invoice Date | Region | State | City | Product |
|---|---|---|---|---|---|---|---|
| **0** | Foot Locker | 1185732 | 2021-11-06 | Northeast | Pennsylvania | Philadelphia | Women'S Athletic Footwear |
| **1** | Foot Locker | 1185732 | 2020-11-01 | Midwest | Minnesota | Minneapolis | Women'S Athletic Footwear |
| **2** | Foot Locker | 1185732 | 2020-11-07 | Midwest | Minnesota | Minneapolis | Women'S Athletic Footwear |
| **3** | Foot Locker | 1185732 | 2021-05-30 | Midwest | Nebraska | Omaha | Women'S Athletic Footwear |
| **4** | Foot Locker | 1185732 | 2021-06-05 | Midwest | Nebraska | Omaha | Women'S Athletic Footwear |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **5374** | Foot Locker | 1185732 | 2021-01-24 | Northeast | New Hampshire | Manchester | Men'S Apparel |
| **5375** | Foot Locker | 1185732 | 2021-01-24 | Northeast | New Hampshire | Manchester | Women'S Apparel |
| **5376** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Men'S Street Footwear |
| **5377** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Men'S Athletic Footwear |
| **5378** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Women'S Street Footwear |

5379 rows × 16 columns

In [64]:
```
df['Profit per Unit'] = df['Operating Profit'] / df['Units Sold']
```

In [85]:
```
total_profit_per_retailer = df.groupby('Retailer')['Total Sales'].sum().rese
total_profit_per_retailer.columns = ['Retailer', 'Total Sales']


total_profit_per_retailer.sort_values(by='Total Sales', ascending=False, inp
total_profit_per_retailer
```

Out[85]:

| | Retailer | Total Sales |
|---|---|---|
| 5 | West Gear | 7230922.0 |
| 1 | Foot Locker | 6880035.0 |
| 3 | Sports Direct | 6157224.0 |
| 2 | Kohl'S | 3425253.0 |
| 4 | Walmart | 2609507.0 |
| 0 | Amazon | 2493804.0 |

In [87]:
```
total_sales_per_product = df.groupby('Product')['Total Sales'].sum().reset_i
total_sales_per_product.columns = ['Product', 'Total Sales']

total_sales_per_product.sort_values(by='Total Sales', ascending=False, inpla
total_sales_per_product
```

Out[87]:

| | Product | Total Sales |
|---|---|---|
| 2 | Men'S Street Footwear | 6048163.0 |
| 1 | Men'S Athletic Footwear | 5317221.0 |
| 5 | Women'S Street Footwear | 4622625.0 |
| 3 | Women'S Apparel | 4585653.0 |
| 4 | Women'S Athletic Footwear | 4144763.0 |
| 0 | Men'S Apparel | 4078320.0 |

In [88]:
```
total_sales_overall = df['Total Sales'].sum()
total_sales_per_product['Percentage'] = (total_sales_per_product['Total Sale
total_sales_per_product
```

Out[88]:

| | Product | Total Sales | Percentage |
|---|---|---|---|
| 2 | Men'S Street Footwear | 6048163.0 | 21.002940 |
| 1 | Men'S Athletic Footwear | 5317221.0 | 18.464660 |
| 5 | Women'S Street Footwear | 4622625.0 | 16.052596 |
| 3 | Women'S Apparel | 4585653.0 | 15.924206 |
| 4 | Women'S Athletic Footwear | 4144763.0 | 14.393165 |
| 0 | Men'S Apparel | 4078320.0 | 14.162434 |

In [90]:
```
df['Month'] = df['Invoice Date'].dt.month
df['Month']
```

Out[90]:

|  | Month |
|---|---|
| 0 | 11 |
| 1 | 11 |
| 2 | 11 |
| 3 | 5 |
| 4 | 6 |
| ... | ... |
| 5374 | 1 |
| 5375 | 1 |
| 5376 | 2 |
| 5377 | 2 |
| 5378 | 2 |

5379 rows × 1 columns

**dtype:** int32

In [91]:
```
Sales_per_month = df.groupby('Month')['Total Sales'].sum().reset_index()
Sales_per_month.columns = ['Month', 'Total Sales']
Sales_per_month.sort_values(by='Total Sales', ascending=False, inplace=True)
Sales_per_month
```

Out[91]:

|  | Month | Total Sales |
|---|---|---|
| 7 | 8 | 2879159.0 |
| 4 | 5 | 2710090.0 |
| 11 | 12 | 2702920.0 |
| 8 | 9 | 2480200.0 |
| 0 | 1 | 2446583.0 |
| 3 | 4 | 2444918.0 |
| 6 | 7 | 2347186.0 |
| 10 | 11 | 2287022.0 |
| 1 | 2 | 2214024.0 |
| 5 | 6 | 2146453.0 |
| 9 | 10 | 2088972.0 |
| 2 | 3 | 2049218.0 |

visualizations

In [92]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

In [96]:
```
df
```

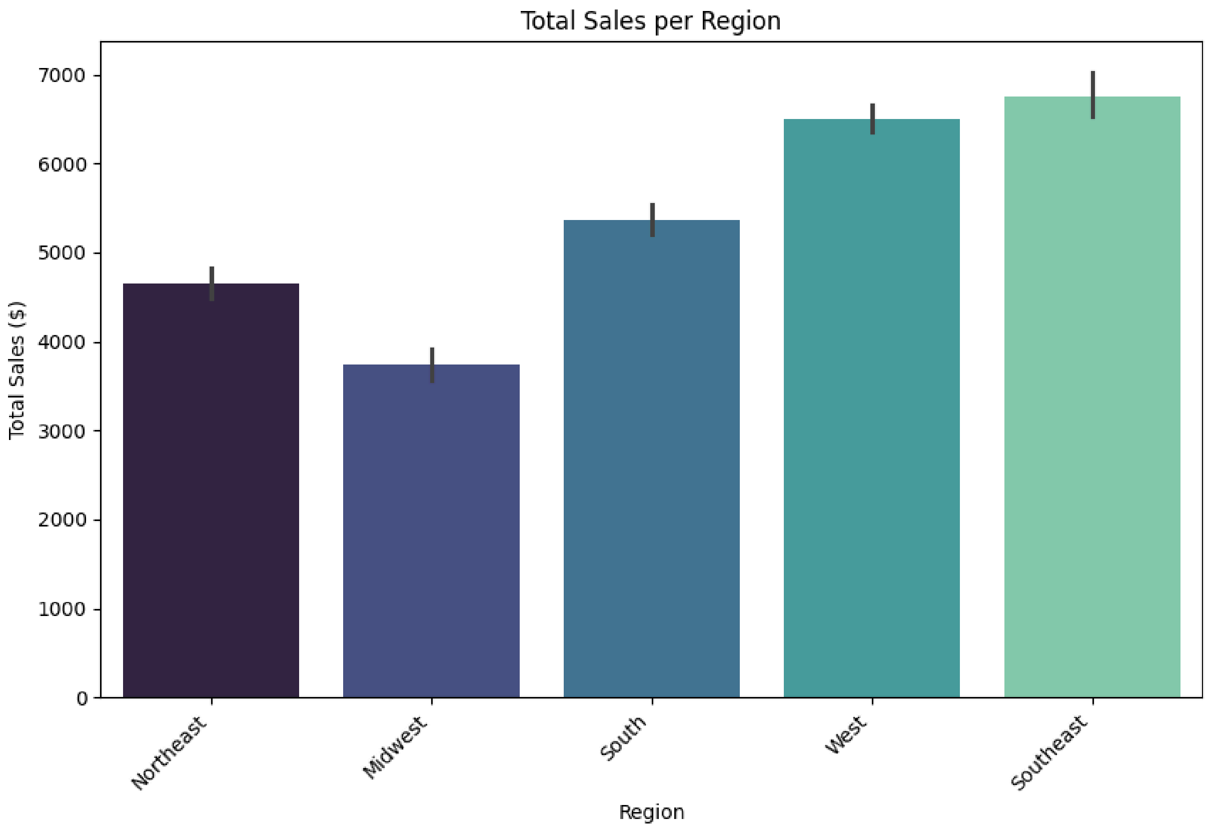
Out[96]:

| | Retailer | Retailer ID | Invoice Date | Region | State | City | Product |
|---|---|---|---|---|---|---|---|
| **0** | Foot Locker | 1185732 | 2021-11-06 | Northeast | Pennsylvania | Philadelphia | Women'S Athletic Footwear |
| **1** | Foot Locker | 1185732 | 2020-11-01 | Midwest | Minnesota | Minneapolis | Women'S Athletic Footwear |
| **2** | Foot Locker | 1185732 | 2020-11-07 | Midwest | Minnesota | Minneapolis | Women'S Athletic Footwear |
| **3** | Foot Locker | 1185732 | 2021-05-30 | Midwest | Nebraska | Omaha | Women'S Athletic Footwear |
| **4** | Foot Locker | 1185732 | 2021-06-05 | Midwest | Nebraska | Omaha | Women'S Athletic Footwear |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **5374** | Foot Locker | 1185732 | 2021-01-24 | Northeast | New Hampshire | Manchester | Men'S Apparel |
| **5375** | Foot Locker | 1185732 | 2021-01-24 | Northeast | New Hampshire | Manchester | Women'S Apparel |
| **5376** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Men'S Street Footwear |
| **5377** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Men'S Athletic Footwear |
| **5378** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Women'S Street Footwear |

5379 rows × 19 columns

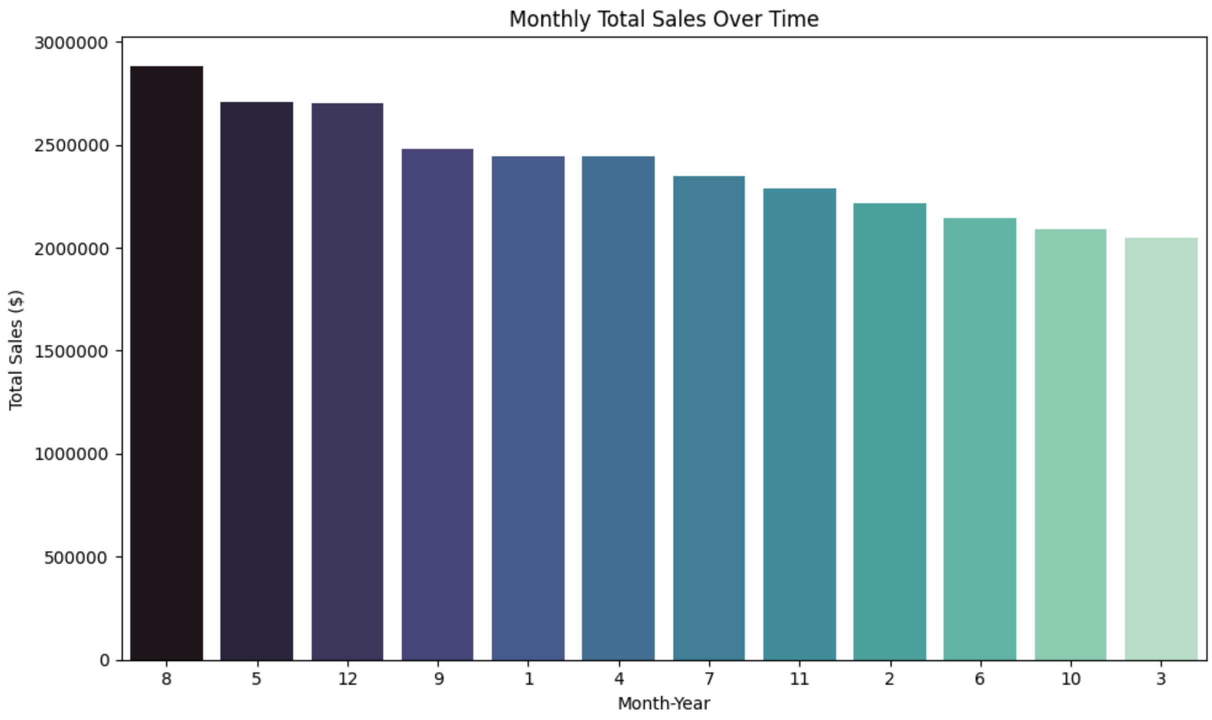
In [172…
```
plt.figure(figsize=(10,6))
plt.ticklabel_format(style='plain')
sns.barplot(x='Region', y='Total Sales', data=df, palette='mako')
plt.title('Total Sales per Region')
plt.xlabel('Region')
plt.ylabel('Total Sales ($)')
plt.xticks(rotation=45, ha='right')
plt.show()
```

```
<ipython-input-172-56aa08ddcd7a>:3: FutureWarning:



Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.
```



```
In [126…  plt.figure(figsize=(10, 6))
          plt.ticklabel_format(style='plain')
          month_order_desc = Sales_per_month.sort_values(by='Total Sales', ascending=F
          sns.barplot(x='Month', y='Total Sales', data=Sales_per_month, palette='mako'
          plt.title('Monthly Total Sales Over Time')
          plt.xlabel('Month-Year')
          plt.ylabel('Total Sales ($)')
          plt.xticks(rotation=0)
          plt.tight_layout()
          plt.savefig('monthly_sales_bar_chart.png')
          plt.tight_layout()
          plt.show()
```

```
<ipython-input-126-d66e4cad074e>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x='Month', y='Total Sales', data=Sales_per_month, palette='mak
o', order=month_order_desc )
```



In [127…

```
total_sales_per_product.sort_values(by='Total Sales' , ascending=False)
```

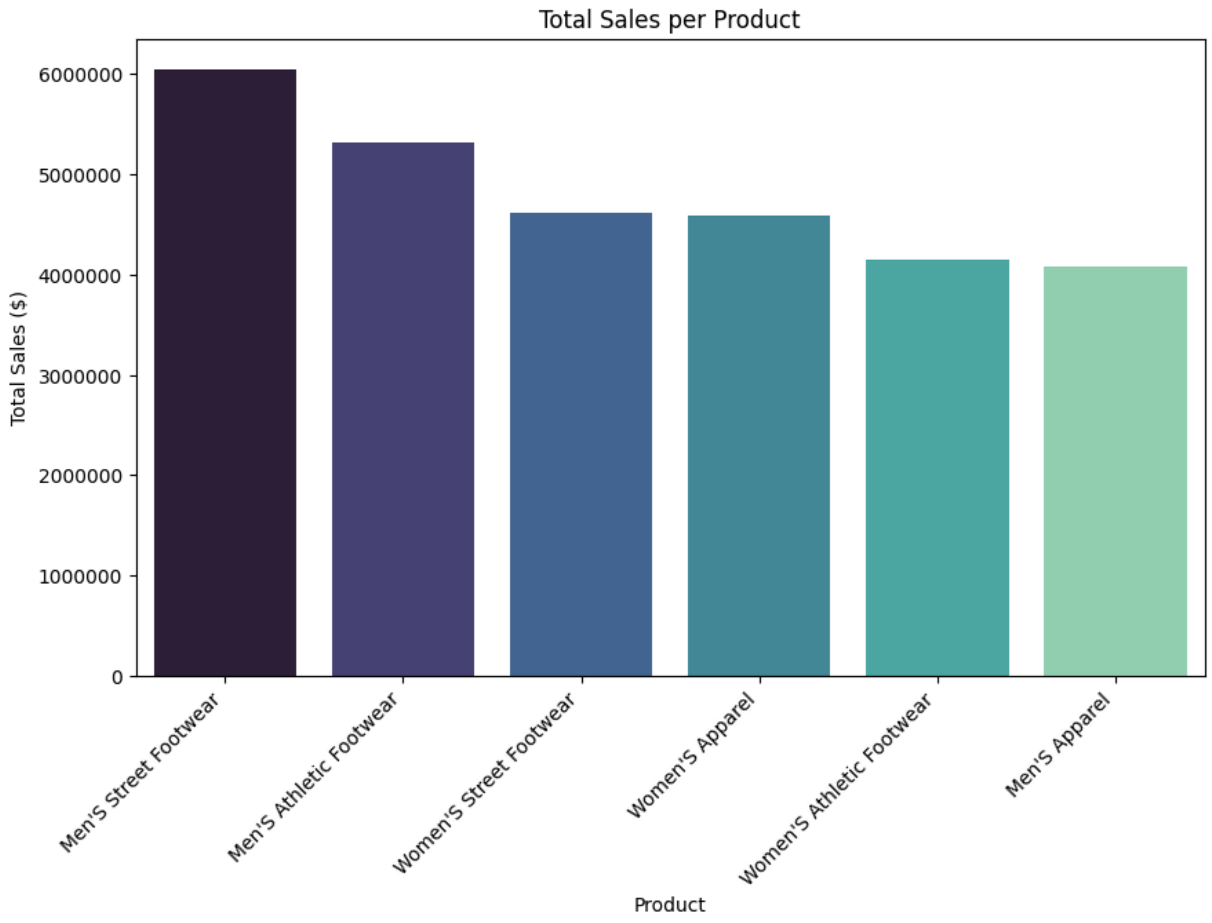
Out[127…

| | Product | Total Sales | Percentage |
|---|---|---|---|
| **2** | Men'S Street Footwear | 6048163.0 | 21.002940 |
| **1** | Men'S Athletic Footwear | 5317221.0 | 18.464660 |
| **5** | Women'S Street Footwear | 4622625.0 | 16.052596 |
| **3** | Women'S Apparel | 4585653.0 | 15.924206 |
| **4** | Women'S Athletic Footwear | 4144763.0 | 14.393165 |
| **0** | Men'S Apparel | 4078320.0 | 14.162434 |

In [137…

```
plt.figure(figsize=(10,6))
plt.ticklabel_format(style='plain')
sns.barplot(x='Product', y='Total Sales', data=total_sales_per_product, pale
plt.title('Total Sales per Product')
plt.xlabel('Product')
plt.ylabel('Total Sales ($)')
plt.xticks(rotation=45, ha='right')
plt.show()
```

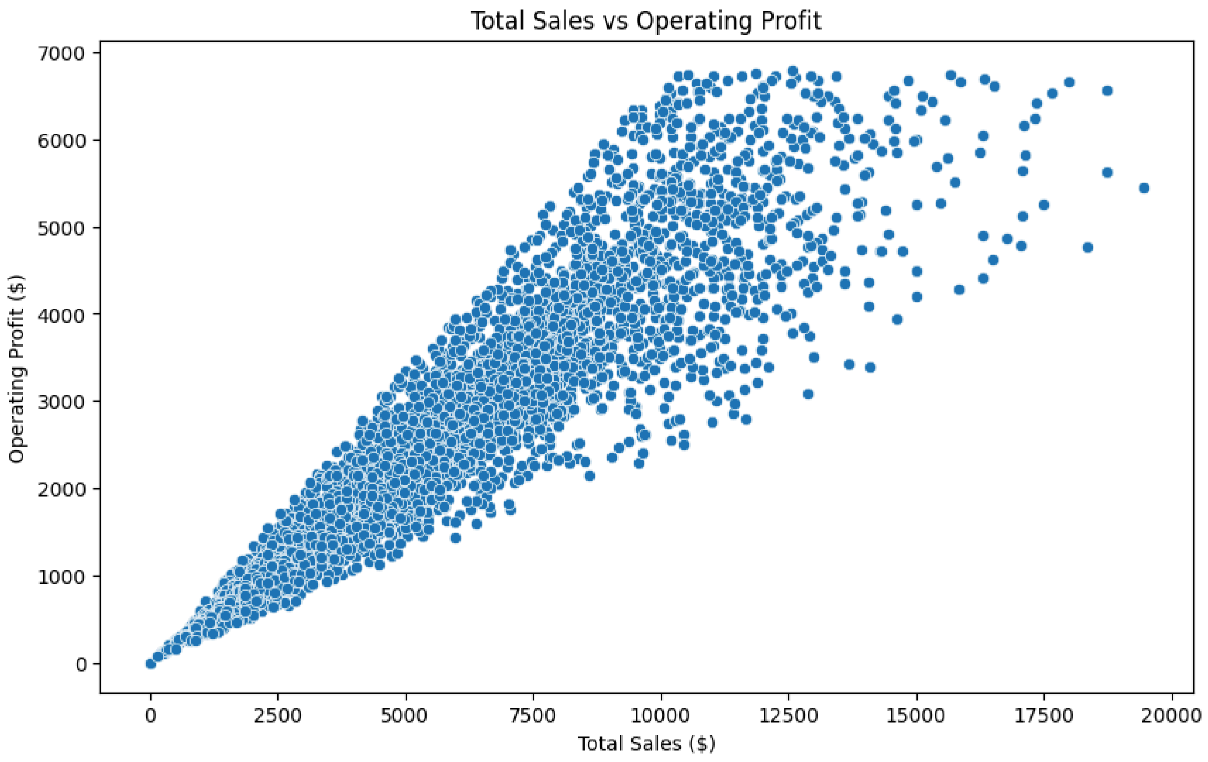
```
<ipython-input-137-bc28adf24212>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x='Product', y='Total Sales', data=total_sales_per_product, pa
lette='mako')
```



In [138…
```
plt.figure(figsize=(10,6))
plt.ticklabel_format(style='plain')
sns.scatterplot(x='Total Sales' , y ='Operating Profit', data=df , palette='
plt.title('Total Sales vs Operating Profit')
plt.xlabel('Total Sales ($)')
plt.ylabel('Operating Profit ($)')
```

```
<ipython-input-138-9e29c8293631>:3: UserWarning: Ignoring `palette` because
no `hue` variable has been assigned.
  sns.scatterplot(x='Total Sales' , y ='Operating Profit', data=df , palette
='mako')
```

Out[138…
```
Text(0, 0.5, 'Operating Profit ($)')
```


Total Sales vs Operating Profit
Operating Profit ($)
Total Sales ($)
7000
6000
5000
4000
3000
2000
1000
0
0
2500
5000
7500
10000
12500
15000
17500
20000

```
In [173…  df
```

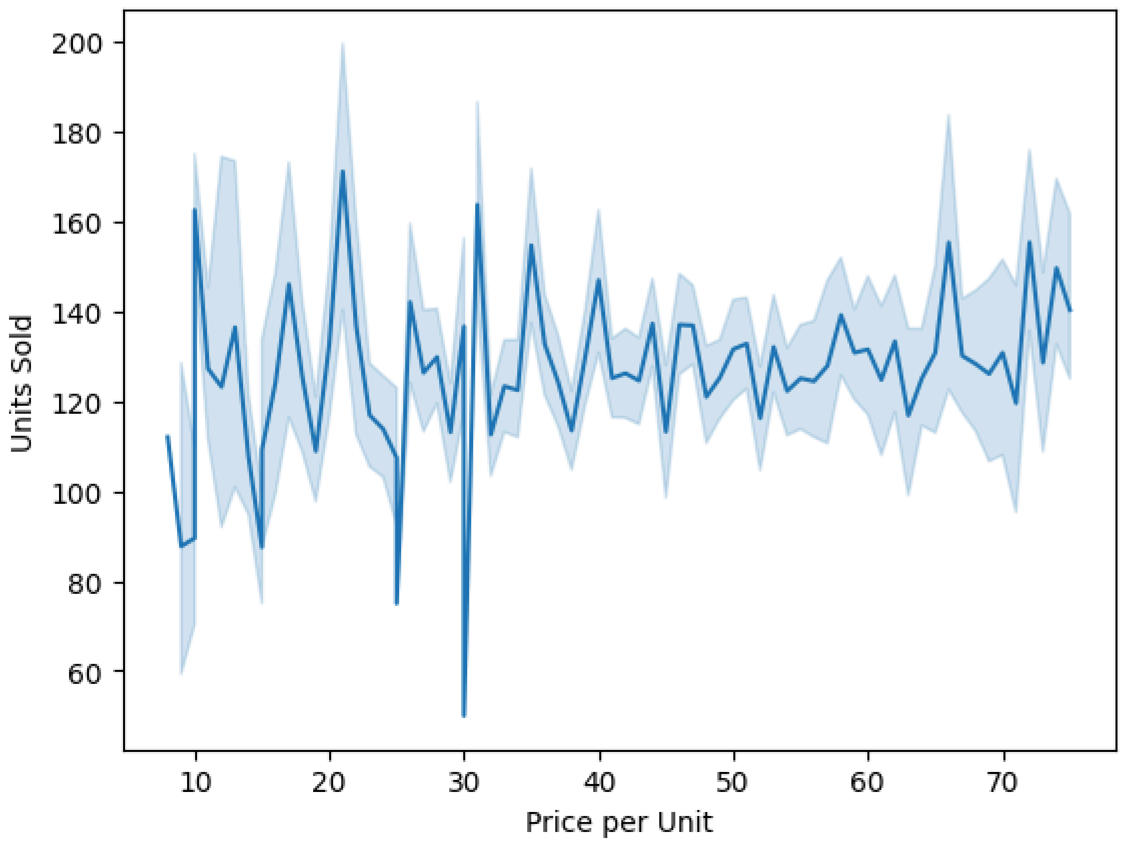
Out[173…

| | Retailer | Retailer ID | Invoice Date | Region | State | City | Product |
|---|---|---|---|---|---|---|---|
| **0** | Foot Locker | 1185732 | 2021-11-06 | Northeast | Pennsylvania | Philadelphia | Women'S Athletic Footwear |
| **1** | Foot Locker | 1185732 | 2020-11-01 | Midwest | Minnesota | Minneapolis | Women'S Athletic Footwear |
| **2** | Foot Locker | 1185732 | 2020-11-07 | Midwest | Minnesota | Minneapolis | Women'S Athletic Footwear |
| **3** | Foot Locker | 1185732 | 2021-05-30 | Midwest | Nebraska | Omaha | Women'S Athletic Footwear |
| **4** | Foot Locker | 1185732 | 2021-06-05 | Midwest | Nebraska | Omaha | Women'S Athletic Footwear |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **5374** | Foot Locker | 1185732 | 2021-01-24 | Northeast | New Hampshire | Manchester | Men'S Apparel |
| **5375** | Foot Locker | 1185732 | 2021-01-24 | Northeast | New Hampshire | Manchester | Women'S Apparel |
| **5376** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Men'S Street Footwear |
| **5377** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Men'S Athletic Footwear |
| **5378** | Foot Locker | 1185732 | 2021-02-22 | Northeast | New Hampshire | Manchester | Women'S Street Footwear |

5379 rows × 20 columns

In [178…

```
sns.lineplot(x='Price per Unit' , y ='Units Sold' , data=df)
plt.show()
```



In [180…

```
df.to_csv('Adidas.csv', index=False)
```

In [181…

```
from google.colab import files
files.download('data.csv')
```