Sales Data

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

df = pd.read_csv("Sales.csv")

df.head()
```

```
      Segment  Country     Product   Discount Band   Units Sold   \
0  Government   Canada   Carretera            None    $1,618.50
1  Government  Germany   Carretera            None    $1,321.00
2   Midmarket   France   Carretera            None    $2,178.00
3   Midmarket  Germany   Carretera            None      $888.00
4   Midmarket   Mexico   Carretera            None    $2,470.00

   Manufacturing Price   Sale Price   Gross Sales   Discounts
Sales   \
0                $3.00       $20.00    $32,370.00        $-
$32,370.00
1                $3.00       $20.00    $26,420.00        $-
$26,420.00
2                $3.00       $15.00    $32,670.00        $-
$32,670.00
3                $3.00       $15.00    $13,320.00        $-
$13,320.00
4                $3.00       $15.00    $37,050.00        $-
$37,050.00

          COGS        Profit         Date  Month Number  Month Name
Year
0   $16,185.00    $16,185.00   01/01/2014             1     January
2014
1   $13,210.00    $13,210.00   01/01/2014             1     January
2014
2   $21,780.00    $10,890.00   01/06/2014             6        June
2014
3    $8,880.00     $4,440.00   01/06/2014             6        June
2014
4   $24,700.00    $12,350.00   01/06/2014             6        June
2014
```

```
df.tail()
```

```
              Segment                    Country    Product   Discount
Band   \
695    Small Business                     France   Amarilla
```

```
High
696    Small Business                      Mexico   Amarilla
High
697        Government                      Mexico    Montana
High
698        Government                      Canada      Paseo
High
699  Channel Partners  United States of America         VTT
High

     Units Sold   Manufacturing Price   Sale Price     Gross Sales   \
695   $2,475.00               $260.00      $300.00    $7,42,500.00
696     $546.00               $260.00      $300.00    $1,63,800.00
697   $1,368.00                 $5.00        $7.00       $9,576.00
698     $723.00                $10.00        $7.00       $5,061.00
699   $1,806.00               $250.00       $12.00      $21,672.00

         Discounts            Sales             COGS         Profit
Date  \
695   $1,11,375.00    $6,31,125.00    $6,18,750.00    $12,375.00
01/03/2014
696     $24,570.00    $1,39,230.00    $1,36,500.00     $2,730.00
01/10/2014
697      $1,436.40       $8,139.60       $6,840.00     $1,299.60
01/02/2014
698        $759.15       $4,301.85       $3,615.00       $686.85
01/04/2014
699      $3,250.80      $18,421.20       $5,418.00    $13,003.20
01/05/2014

     Month Number  Month Name   Year
695             3       March   2014
696            10     October   2014
697             2    February   2014
698             4       April   2014
699             5         May   2014
```

Removing Dollar Sign

```
df[' Units Sold '] = df[' Units Sold '].str.replace('[$,]', '',
regex=True).str.strip()
df[' Manufacturing Price '] = df[' Manufacturing Price
'].str.replace('[$,]', '', regex=True).str.strip()
df[' Gross Sales '] = df[' Gross Sales '].str.replace('[$,]', '',
regex=True).str.strip()
df[' Discounts '] = df[' Discounts '].str.replace('[$,]', '',
regex=True).str.strip()
df[' COGS '] = df[' COGS '].str.replace('[$,]', '',
```

```
regex=True).str.strip()
df[' Profit '] = df[' Profit '].str.replace('[$,]', '',
regex=True).str.strip()
df[' Sale Price '] = df[' Sale Price '].str.replace('[$,]', '',
regex=True).str.strip()
df['  Sales '] = df['  Sales '].str.replace('[$,]', '',
regex=True).str.strip()
```

Removing Spaces

```
df[' Units Sold '] = df[' Units Sold '].str.replace(' ', '',
regex=False)
df[' Manufacturing Price '] = df[' Manufacturing Price
'].str.replace(' ', '', regex=False)
df[' Gross Sales '] = df[' Gross Sales '].str.replace(' ', '',
regex=False)
df[' Discounts '] = df[' Discounts '].str.replace(' ', '',
regex=False)
df[' COGS '] = df[' COGS '].str.replace(' ', '', regex=False)
df[' Profit '] = df[' Profit '].str.replace(' ', '', regex=False)
df[' Sale Price '] = df[' Sale Price '].str.replace(' ', '',
regex=False)
df['  Sales '] = df['  Sales '].str.replace(' ', '', regex=False)
df[' Month Name '] = df[' Month Name '].str.replace(' ', '',
regex=False)
```

Checking Column Data Type

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Segment                700 non-null    object
 1   Country                700 non-null    object
 2    Product               700 non-null    object
 3    Discount Band         700 non-null    object
 4    Units Sold            700 non-null    object
 5    Manufacturing Price   700 non-null    object
 6    Sale Price            700 non-null    object
 7    Gross Sales           700 non-null    object
 8    Discounts             700 non-null    object
 9     Sales                700 non-null    object
 10   COGS                   700 non-null    object
```

```
 11   Profit                 700 non-null    object
 12  Date                    700 non-null    object
 13  Month Number            700 non-null    int64
 14   Month Name             700 non-null    object
 15  Year                    700 non-null    int64
dtypes: int64(2), object(14)
memory usage: 87.6+ KB
```

```
print(df.columns)
```

```
Index(['Segment', 'Country', ' Product ', ' Discount Band ', ' Units
Sold ',
       ' Manufacturing Price ', ' Sale Price ', ' Gross Sales ', '
Discounts ',
       '  Sales ', ' COGS ', ' Profit ', 'Date', 'Month Number',
       ' Month Name ', 'Year'],
      dtype='object')
```

Convert Column Data Type

```
df = df.astype({
     ' Units Sold ' : 'float',
    ' Manufacturing Price ' : 'float',
    ' Sale Price ' : 'float',
    ' Gross Sales ' : 'float',
    '  Sales ' : 'float',
    ' COGS ' : 'float',
    ' Discounts ': 'float',
     ' Profit ' : 'float'
})

df['Date'] = pd.to_datetime(df['Date'])

df[' Discounts '] = df[' Discounts '].replace('-', 0)
df[' Profit '] = df[' Profit '].replace('-', 0)

df[' Profit '] = df[' Profit '].str.replace(r'[()]', '', regex=True)

df.tail()
```

```
               Segment                    Country    Product   Discount
Band   \
695    Small Business                     France   Amarilla
High
696    Small Business                     Mexico   Amarilla
High
697        Government                     Mexico    Montana
High
698        Government                     Canada      Paseo
```

```
High
699  Channel Partners  United States of America        VTT
High

      Units Sold    Manufacturing Price    Sale Price    Gross Sales
\
695        2475.0                  260.0         300.0       742500.0

696         546.0                  260.0         300.0       163800.0

697        1368.0                    5.0           7.0         9576.0

698         723.0                   10.0           7.0         5061.0

699        1806.0                  250.0          12.0        21672.0


     Discounts      Sales      COGS    Profit        Date  Month
Number  \
695   111375.00  631125.00  618750.0  12375.00 2014-01-03
3
696    24570.00  139230.00  136500.0   2730.00 2014-01-10
10
697     1436.40    8139.60    6840.0   1299.60 2014-01-02
2
698      759.15    4301.85    3615.0    686.85 2014-01-04
4
699     3250.80   18421.20    5418.0  13003.20 2014-01-05
5

     Month Name   Year
695        March  2014
696      October  2014
697     February  2014
698        April  2014
699          May  2014

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Segment               700 non-null    object
 1   Country               700 non-null    object
 2    Product              700 non-null    object
 3    Discount Band        700 non-null    object
 4    Units Sold           700 non-null    float64
 5    Manufacturing Price  700 non-null    float64
```

```
 6    Sale Price            700 non-null    float64
 7    Gross Sales           700 non-null    float64
 8    Discounts             700 non-null    float64
 9     Sales                700 non-null    float64
 10   COGS                  700 non-null    float64
 11   Profit                695 non-null    float64
 12  Date                   700 non-null    datetime64[ns]
 13  Month Number           700 non-null    int64
 14   Month Name            700 non-null    object
 15  Year                   700 non-null    int64
dtypes: datetime64[ns](1), float64(8), int64(2), object(5)
memory usage: 87.6+ KB
```

```
df['Sales'] = df['Sales'].astype('int')
```

```
df.describe()
```

```
        Units Sold    Manufacturing Price    Sale Price    Gross Sales
\
count    700.000000             700.000000    700.000000
7.000000e+02
mean    1608.294286              96.477143    118.428571
1.827594e+05
min      200.000000               3.000000      7.000000
1.799000e+03
25%      905.000000               5.000000     12.000000
1.739175e+04
50%     1542.500000              10.000000     20.000000
3.798000e+04
75%     2229.125000             250.000000    300.000000
2.790250e+05
max     4492.500000             260.000000    350.000000
1.207500e+06
std      867.427859             108.602612    136.775515
2.542623e+05

          Discounts         Sales           COGS         Profit   \
count     700.000000  7.000000e+02     700.000000     695.000000
mean    13150.354671  1.696091e+05  145475.211429   26544.380993
min         0.000000  1.655080e+03     918.000000     285.600000
25%       800.320000  1.592800e+04    7490.000000    4013.630000
50%      2585.250000  3.554020e+04   22506.250000   11135.600000
75%     15956.347500  2.610775e+05  245607.500000   23918.500000
max    149677.500000  1.159200e+06  950625.000000  262200.000000
std     22962.928760  2.367263e+05  203865.506118   41515.104658

                      Date  Month Number         Year
count                  700    700.000000   700.000000
mean   2013-10-08 15:36:00      7.900000  2013.750000
min    2013-01-09 00:00:00      1.000000  2013.000000
```

```
25%    2013-10-04 12:00:00      5.750000  2013.750000
50%    2014-01-05 12:00:00      9.000000  2014.000000
75%    2014-01-09 06:00:00     10.250000  2014.000000
max    2014-01-12 00:00:00     12.000000  2014.000000
std                    NaN      3.377321     0.433322
```

```
df.isnull().sum()
```

```
Segment                  0
Country                  0
 Product                 0
 Discount Band           0
 Units Sold              0
 Manufacturing Price     0
 Sale Price              0
 Gross Sales             0
 Discounts               0
  Sales                  0
 COGS                    0
 Profit                  5
Date                     0
Month Number             0
 Month Name              0
Year                     0
dtype: int64
```

Replace Null Values

```
df[' Profit '] = df[' Profit '].replace(np.nan,26544.380993)
```

```
df.isnull().sum()
```

```
Segment                  0
Country                  0
 Product                 0
 Discount Band           0
 Units Sold              0
 Manufacturing Price     0
 Sale Price              0
 Gross Sales             0
 Discounts               0
  Sales                  0
 COGS                    0
 Profit                  0
Date                     0
Month Number             0
 Month Name              0
```

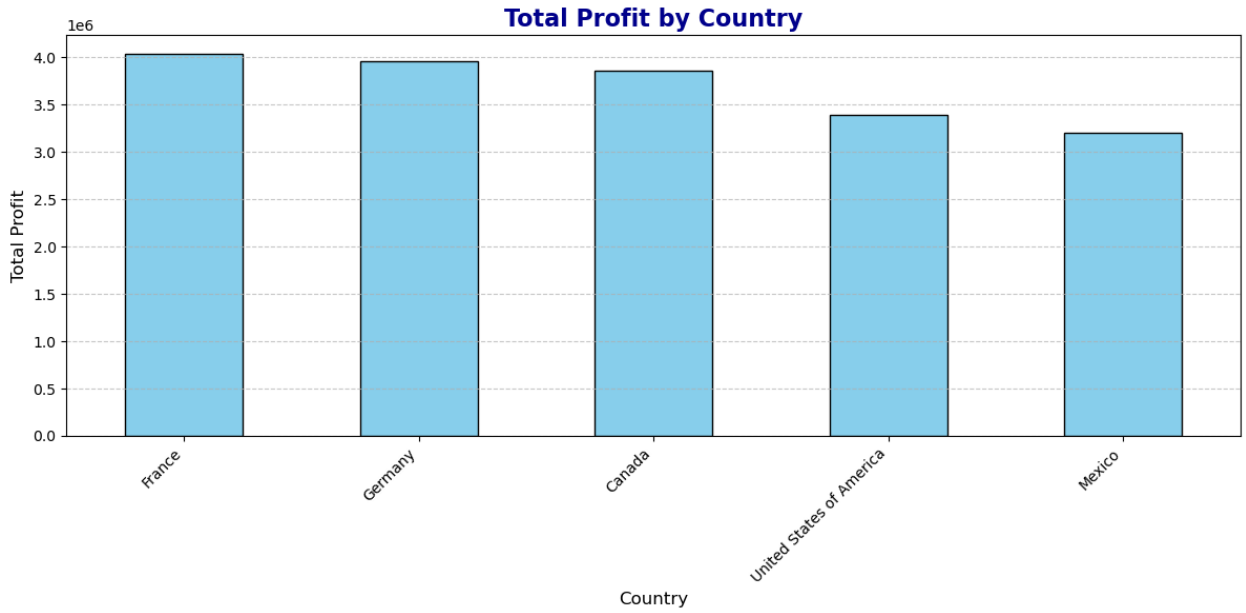
```
Year                         0
dtype: int64
```

```
df.columns
```

```
Index(['Segment', 'Country', ' Product ', ' Discount Band ', ' Units
Sold ',
       ' Manufacturing Price ', ' Sale Price ', ' Gross Sales ', '
Discounts ',
       '  Sales ', ' COGS ', ' Profit ', 'Date', 'Month Number',
       ' Month Name ', 'Year'],
      dtype='object')
```

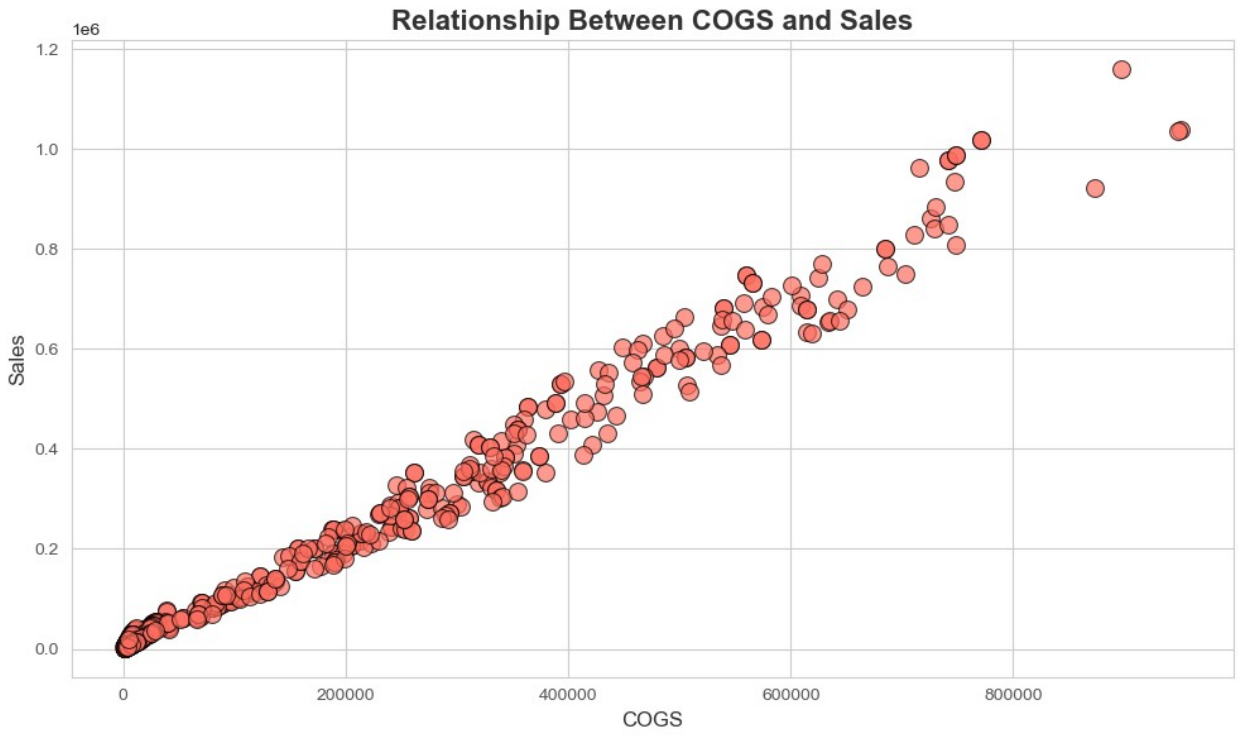
Profit by Country

```
profit_by_country = df.groupby('Country')[' Profit
'].sum().sort_values(ascending=False)


plt.figure(figsize=(12, 6))
profit_by_country.plot(kind='bar',color='skyblue',edgecolor='black'
)


plt.title('Total Profit by Country', fontsize=16, fontweight='bold',
color='darkblue')
plt.xlabel('Country', fontsize=12)
plt.ylabel('Total Profit', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)

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



Sales vs COGS scatter plot

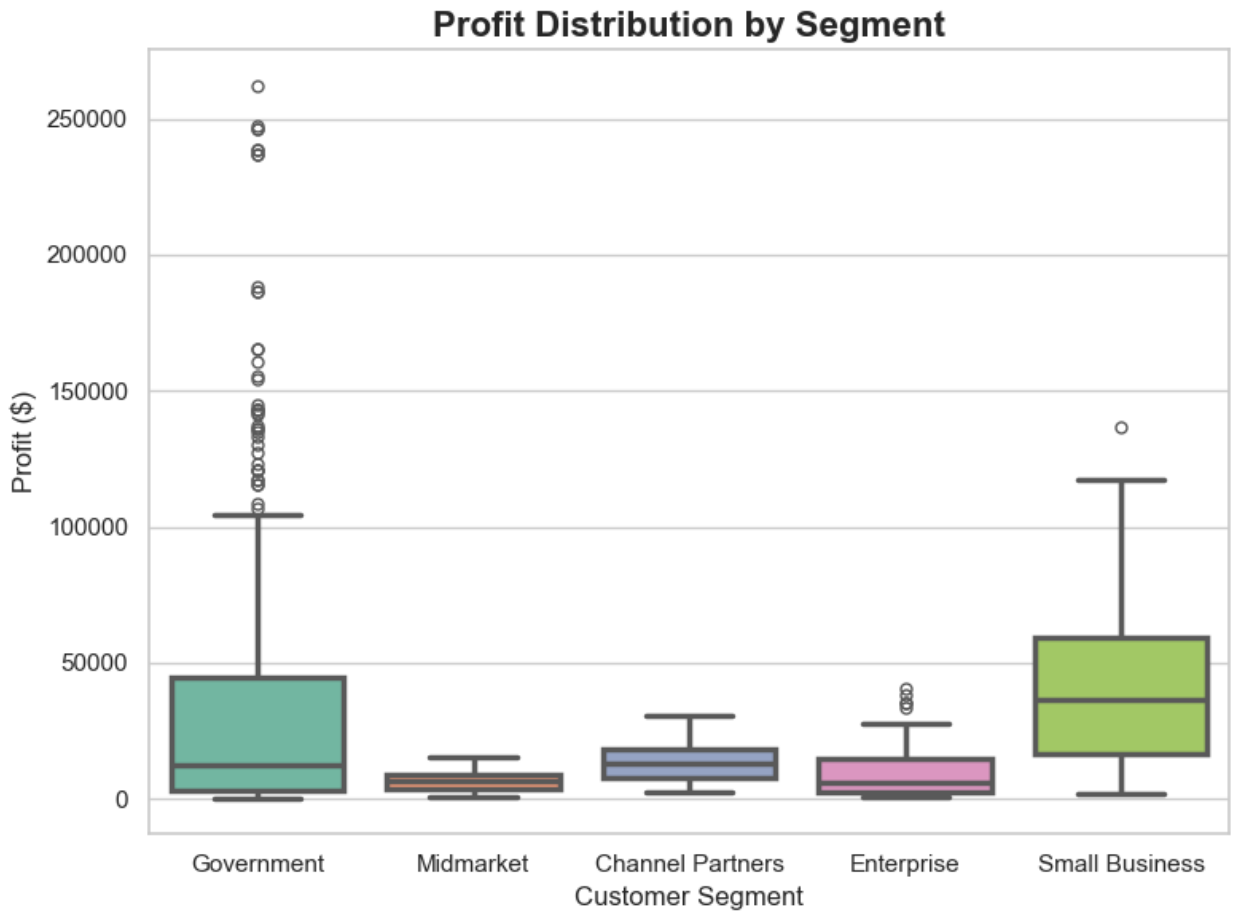
```
sns.set_style("whitegrid")


plt.figure(figsize=(10, 6))


sns.scatterplot(data=df,x=' COGS ',y='  Sales ',color='#FF6F61',
edgecolor='black',s=100,  alpha=0.7
)

plt.title("Relationship Between COGS and Sales", fontsize=16,
fontweight='bold', color='#333333')
plt.xlabel("COGS", fontsize=12, color='#333333')
plt.ylabel("Sales", fontsize=12, color='#333333')

plt.xticks(fontsize=10, color='#555555')
plt.yticks(fontsize=10, color='#555555')

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



Boxplot of Profit by Segment

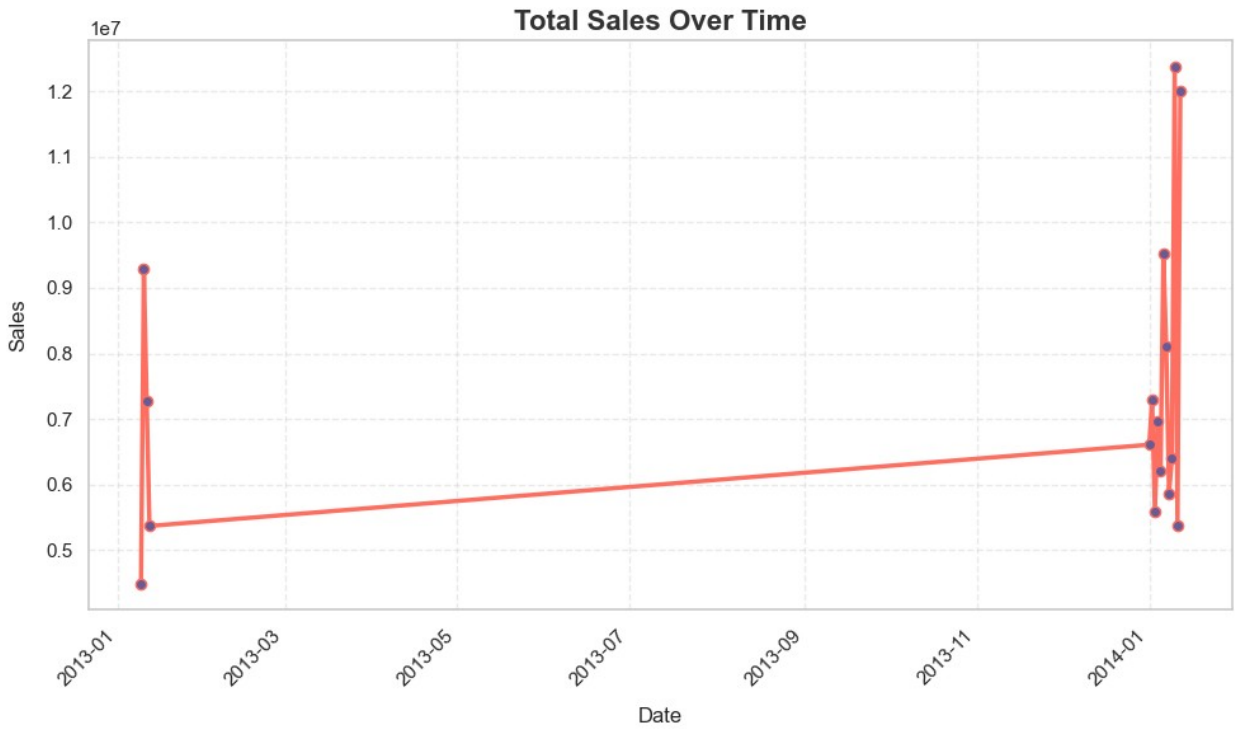
```
sns.set_theme(style="whitegrid")

plt.figure(figsize=(8, 6))
sns.boxplot(data=df, x='Segment',y=' Profit ',
palette='Set2',linewidth=2.5,fliersize=5
)

plt.title('Profit Distribution by Segment', fontsize=16,
fontweight='bold')
plt.xlabel('Customer Segment', fontsize=12)
plt.ylabel('Profit ($)', fontsize=12)

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

```
C:\Users\akhil\AppData\Local\Temp\ipykernel_12244\2900962614.py: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.boxplot(data=df, x='Segment',y=' Profit ',
palette='Set2',linewidth=2.5,fliersize=5
```



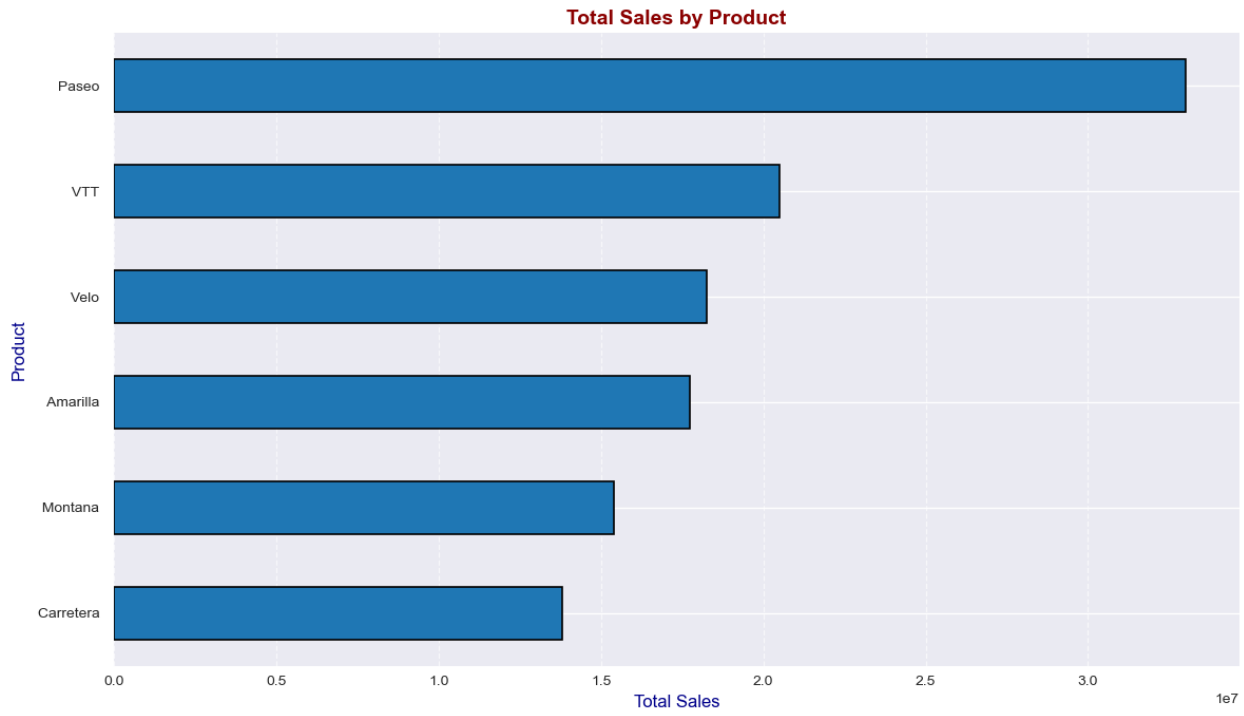
Total Sales over time

```
plt.figure(figsize=(10, 6))
df.groupby('Date')['  Sales
'].sum().plot(kind='line',color='#FF6F61',linewidth=2.5,marker='o',mar
kersize=6,markerfacecolor='#6B5B95',linestyle='-')

plt.title('Total Sales Over Time', fontsize=16, fontweight='bold',
color='#333333')
plt.xlabel('Date', fontsize=12, labelpad=10)
plt.ylabel('Sales', fontsize=12, labelpad=10)
plt.grid(True, linestyle='--', alpha=0.4)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



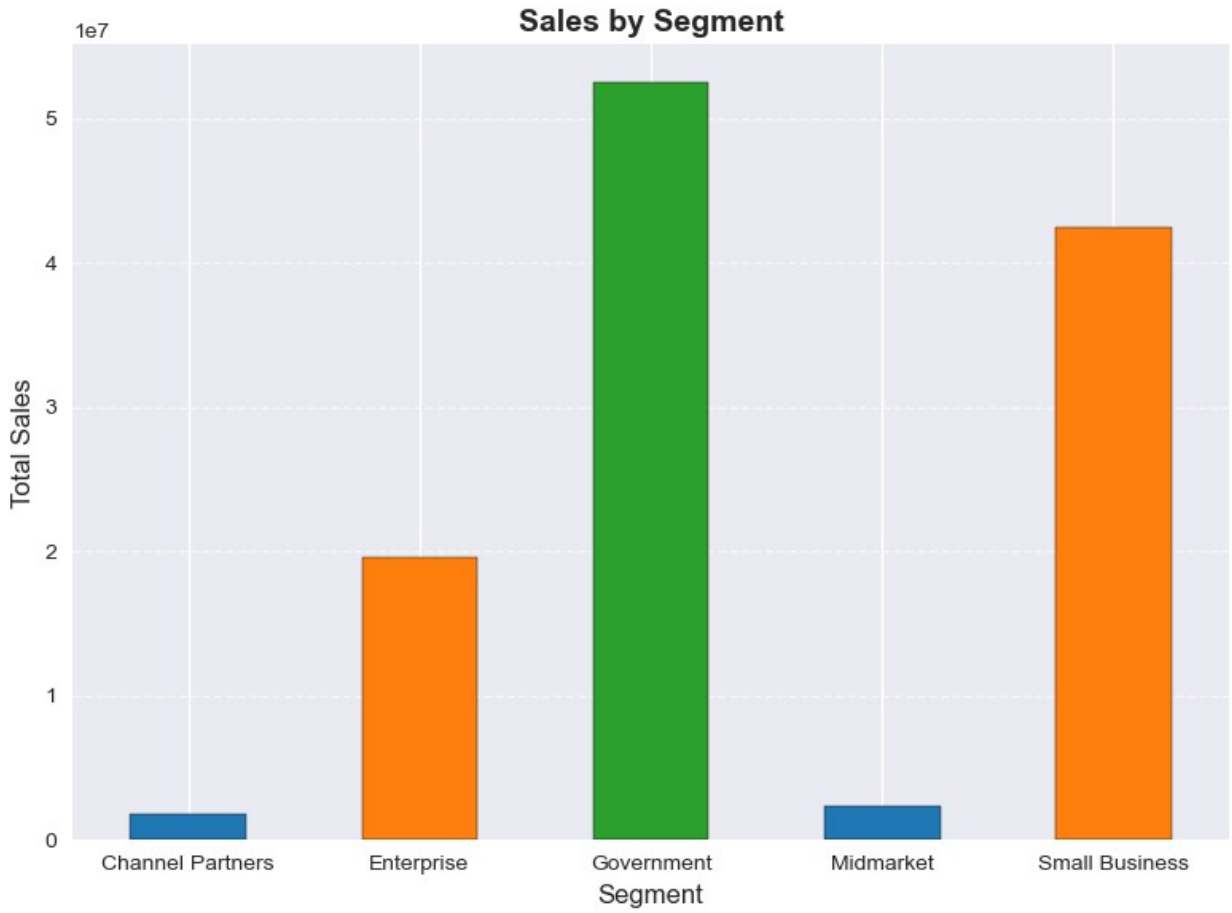
Total Sales by Product

```
plt.style.use('seaborn-v0_8')

# Plot
df.groupby(' Product ')['  Sales
'].sum().sort_values().plot(kind='barh',figsize=(12,
7),color='#1f77b4',  edgecolor='black',linewidth=1.2,
title='Total Sales by Product')

plt.xlabel('Total Sales', fontsize=12, color='darkblue')
plt.ylabel('Product', fontsize=12, color='darkblue')
plt.title('Total Sales by Product', fontsize=14, fontweight='bold',
color='darkred')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



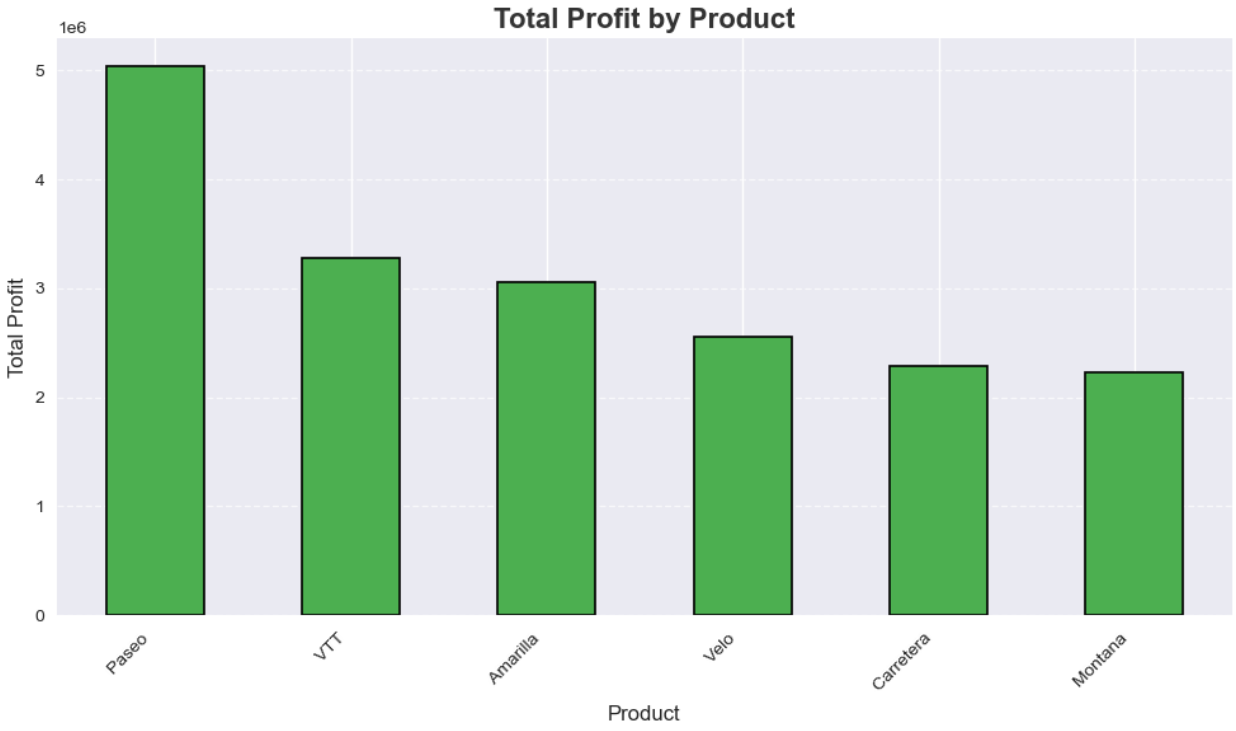
Total Sales by Segment

```
ax = df.groupby('Segment')['  Sales
'].sum().plot(kind='bar',color=['#1f77b4', '#ff7f0e', '#2ca02c'],
edgecolor='black',figsize=(8, 6),
title='Sales by Segment')

ax.set_xlabel('Segment', fontsize=12)
ax.set_ylabel('Total Sales', fontsize=12)
ax.set_title('Sales by Segment', fontsize=14, fontweight='bold')
ax.grid(axis='y', linestyle='--', alpha=0.7)
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```



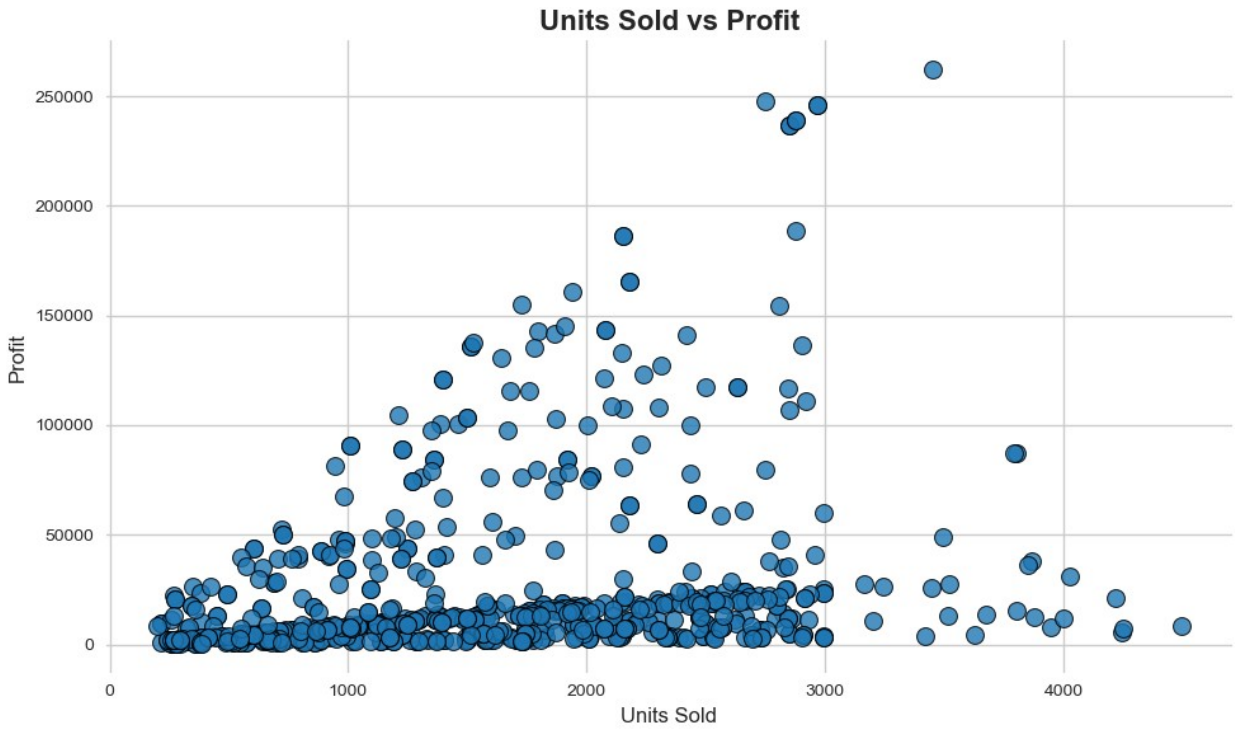
Profit by Product

```
plt.figure(figsize=(10, 6))
df.groupby(' Product ')[' Profit
'].sum().sort_values(ascending=False).plot(kind='bar',
color='#4CAF50',   edgecolor='black',linewidth=1.2,
title='Total Profit by Product')

plt.title('Total Profit by Product', fontsize=16, fontweight='bold',
color='#333333')
plt.xlabel('Product', fontsize=12)
plt.ylabel('Total Profit', fontsize=12)
plt.xticks(rotation=45, ha='right', fontsize=10)
plt.yticks(fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

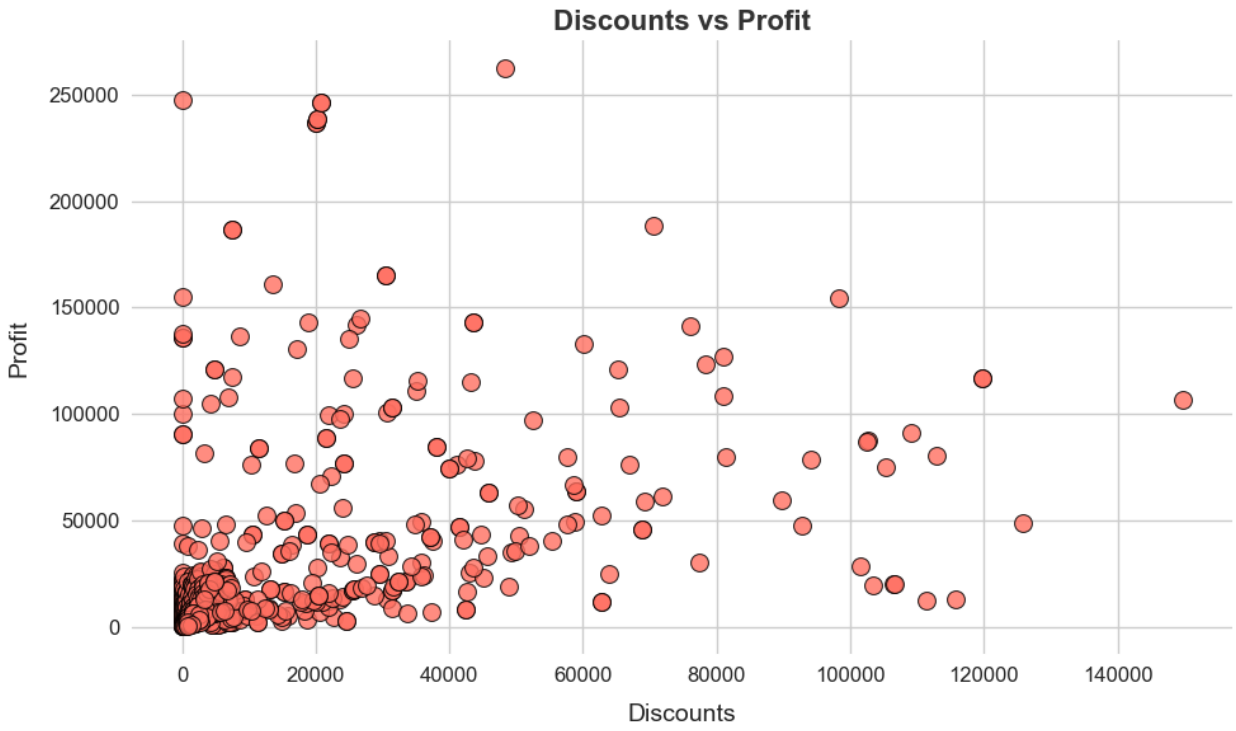


Profit vs Units Sold

```
plt.figure(figsize=(10, 6))
sns.set_style("whitegrid")


sns.scatterplot(data=df,x=' Units Sold ',y=' Profit
',color="#1f77b4",s=100,edgecolor='black',alpha=0.8)

plt.title("Units Sold vs Profit", fontsize=16, fontweight='bold')
plt.xlabel("Units Sold", fontsize=12)
plt.ylabel("Profit", fontsize=12)

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




```
sns.set_style("whitegrid")


plt.figure(figsize=(10, 6))
sns.scatterplot( data=df,x=' Discounts ',y=' Profit ',color='#FF6F61',
edgecolor='black', s=100 , alpha=0.8)

plt.title('Discounts vs Profit', fontsize=16, fontweight='bold',
color='#333333')
plt.xlabel('Discounts', fontsize=14, labelpad=10)
plt.ylabel('Profit', fontsize=14, labelpad=10)

plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

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

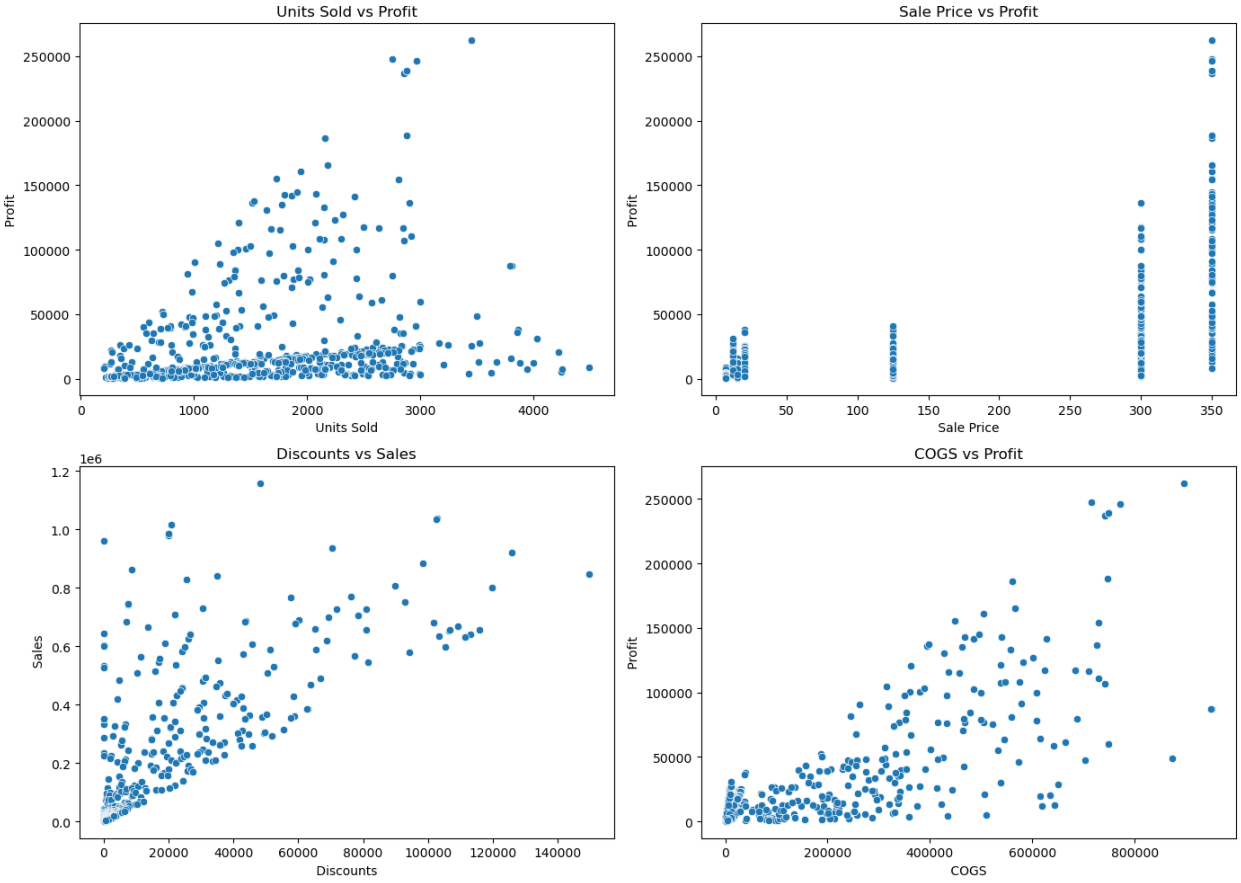



```
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")

sns.boxplot(data=df, x='Country', y=' Profit ', palette='Set2',
linewidth=2, fliersize=4 )

plt.title('Profit Distribution by Country', fontsize=14,
fontweight='bold', color='darkblue')
plt.xlabel('Country', fontsize=12, fontweight='bold')
plt.ylabel('Profit', fontsize=12, fontweight='bold')
plt.xticks(rotation=90, fontsize=10)
plt.yticks(fontsize=10)
plt.tight_layout()
plt.show()
```

```
C:\Users\akhil\AppData\Local\Temp\ipykernel_12244\807970354.py:5:
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.boxplot(
```

Sub Plots

```
fig, axes = plt.subplots(2, 2, figsize=(14, 10))

# Plot 1
sns.scatterplot(ax=axes[0, 0], data=df, x=' Units Sold ', y=' Profit
')
axes[0, 0].set_title('Units Sold vs Profit')

# Plot 2
sns.scatterplot(ax=axes[0, 1], data=df, x=' Sale Price ', y=' Profit
')
axes[0, 1].set_title('Sale Price vs Profit')

# Plot 3
sns.scatterplot(ax=axes[1, 0], data=df, x=' Discounts ', y='  Sales ')
axes[1, 0].set_title('Discounts vs Sales')

# Plot 4
sns.scatterplot(ax=axes[1, 1], data=df, x=' COGS ', y=' Profit ')
axes[1, 1].set_title('COGS vs Profit')

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


Units Sold vs Profit
Profit
Units Sold
Sale Price vs Profit
Profit
Sale Price
Discounts vs Sales
1e6
Sales
Discounts
COGS vs Profit
Profit
COGS