In [1]:
```
import pandas as pd
```

In [2]:
```
df= pd.read_csv('Financials.csv')
df.head()
```

Out[2]:

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	D
0	Government	Canada	Carretera	None	$1,618.50	$3.00	$20.00	$32,370.00	
1	Government	Germany	Carretera	None	$1,321.00	$3.00	$20.00	$26,420.00	
2	Midmarket	France	Carretera	None	$2,178.00	$3.00	$15.00	$32,670.00	
3	Midmarket	Germany	Carretera	None	$888.00	$3.00	$15.00	$13,320.00	
4	Midmarket	Mexico	Carretera	None	$2,470.00	$3.00	$15.00	$37,050.00	

In [3]:
```
df.tail()
```

Out[3]:

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales
695	Small Business	France	Amarilla	High	$2,475.00	$260.00	$300.00	$7,42,500.00
696	Small Business	Mexico	Amarilla	High	$546.00	$260.00	$300.00	$1,63,800.00
697	Government	Mexico	Montana	High	$1,368.00	$5.00	$7.00	$9,576.00
698	Government	Canada	Paseo	High	$723.00	$10.00	$7.00	$5,061.00
699	Channel Partners	United States of America	VTT	High	$1,806.00	$250.00	$12.00	$21,672.00

Data cleaning

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
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
```

In [5]:
```
df.columns = df.columns.str.strip()
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
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
```

In [7]:
```
df.dtypes
```

Out[7]:
```
Segment                 object
Country                 object
Product                 object
Discount Band           object
Units Sold              object
Manufacturing Price     object
Sale Price              object
Gross Sales             object
Discounts               object
Sales                   object
COGS                    object
Profit                  object
Date                    object
Month Number             int64
Month Name              object
Year                     int64
dtype: object
```

In [8]:
```
dollor=['Units Sold','Manufacturing Price','Sale Price','Gross Sales','Disc
for column in dollor:
    df[column] = df[column].replace('[/$]','',regex=True)
```

In [9]:
```
df.head()
```

Out[9]:

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Disc
0	Government	Canada	Carretera	None	1,618.50	3.00	20.00	32,370.00	
1	Government	Germany	Carretera	None	1,321.00	3.00	20.00	26,420.00	
2	Midmarket	France	Carretera	None	2,178.00	3.00	15.00	32,670.00	
3	Midmarket	Germany	Carretera	None	888.00	3.00	15.00	13,320.00	
4	Midmarket	Mexico	Carretera	None	2,470.00	3.00	15.00	37,050.00	

In [10]:
```
df['Discounts'] = df['Discounts'].replace('-', '0' , regex =True)
```

In [11]:
```
df.head()
```

Out[11]:

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Disc
0	Government	Canada	Carretera	None	1,618.50	3.00	20.00	32,370.00	
1	Government	Germany	Carretera	None	1,321.00	3.00	20.00	26,420.00	
2	Midmarket	France	Carretera	None	2,178.00	3.00	15.00	32,670.00	
3	Midmarket	Germany	Carretera	None	888.00	3.00	15.00	13,320.00	
4	Midmarket	Mexico	Carretera	None	2,470.00	3.00	15.00	37,050.00	

In [12]:
```
for columns in dollor:
    df[columns]= df[columns].replace(',','', regex=True)
```

In [13]:
```
df.head()
```

Out[13]:

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discou
0	Government	Canada	Carretera	None	1618.50	3.00	20.00	32370.00	
1	Government	Germany	Carretera	None	1321.00	3.00	20.00	26420.00	
2	Midmarket	France	Carretera	None	2178.00	3.00	15.00	32670.00	
3	Midmarket	Germany	Carretera	None	888.00	3.00	15.00	13320.00	
4	Midmarket	Mexico	Carretera	None	2470.00	3.00	15.00	37050.00	

In [14]:
```
df['Units Sold']=df['Units Sold'].astype(float)
df['Manufacturing Price']=df['Manufacturing Price'].astype(float)
df['Sale price']=df['Sale Price'].astype(float)
df['Gross Sales']=df['Gross Sales'].astype(float)
df['Discounts']=df['Discounts'].astype(float)
df['Sales']=df['Sales'].astype(float)
df['COGS']=df['COGS'].astype(float)
```

In [15]:
```
(df['Profit']==' -    ').sum()
```

Out[15]: 5

In [16]:
```
df = df[df['Profit'] != ' -    ']
```

In [17]:
```
df['Profit']= df['Profit'].str.strip()
```

In [18]:
```
df['Profit']= df['Profit'].replace(r'[()]','', regex = True)
```

In [19]:
```
df['Profit'] = df['Profit'].astype(float)
```

In [20]:
```
df.dtypes
```

Out[20]:
```
Segment                 object
Country                 object
Product                 object
Discount Band           object
Units Sold             float64
Manufacturing Price    float64
Sale Price              object
Gross Sales            float64
Discounts              float64
Sales                  float64
COGS                   float64
Profit                 float64
Date                    object
Month Number             int64
Month Name              object
Year                     int64
Sale price             float64
dtype: object
```

In [21]:
```
df['Date']= pd.to_datetime(df['Date'])
```

In [22]:
```
df.dtypes
```

Out[22]:
```
Segment                        object
Country                        object
Product                        object
Discount Band                  object
Units Sold                    float64
Manufacturing Price           float64
Sale Price                     object
Gross Sales                   float64
Discounts                     float64
Sales                         float64
COGS                          float64
Profit                        float64
Date                   datetime64[ns]
Month Number                    int64
Month Name                     object
Year                            int64
Sale price                    float64
dtype: object
```

In [23]:
```
df= df.drop(['Month Number','Month Name', 'Year'], axis=1)
```

In [24]:
```
df.dtypes
```

Out[24]:
```
Segment                    object
Country                    object
Product                    object
Discount Band              object
Units Sold                float64
Manufacturing Price       float64
Sale Price                 object
Gross Sales               float64
Discounts                 float64
Sales                     float64
COGS                      float64
Profit                    float64
Date                datetime64[ns]
Sale price                float64
dtype: object
```

In [25]:
```
df.head()
```

Out[25]:

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discount
0	Government	Canada	Carretera	None	1618.5	3.0	20.00	32370.0	0
1	Government	Germany	Carretera	None	1321.0	3.0	20.00	26420.0	0
2	Midmarket	France	Carretera	None	2178.0	3.0	15.00	32670.0	0
3	Midmarket	Germany	Carretera	None	888.0	3.0	15.00	13320.0	0
4	Midmarket	Mexico	Carretera	None	2470.0	3.0	15.00	37050.0	0

Visualization

In [26]:
```
import matplotlib.pyplot as plt
import numpy as np
```

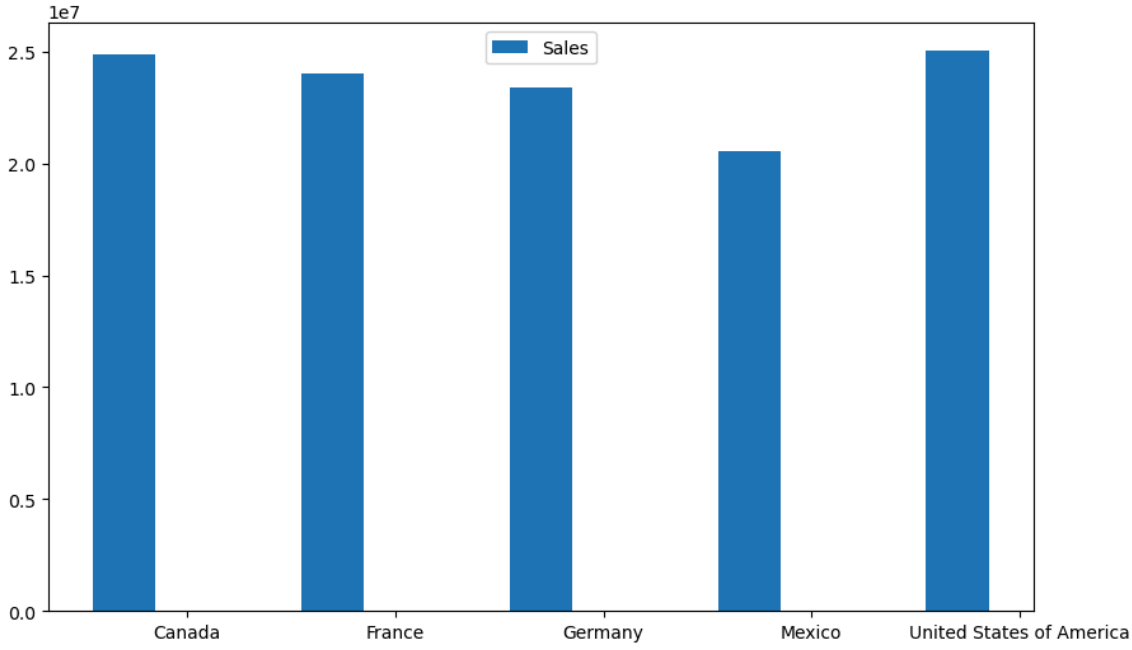
In [27]:
```
group=df.groupby('Country').agg({'Sales':'sum', 'Profit':'sum', 'Units Sold
```

In [28]:
```
group.head(10)
```

Out[28]:

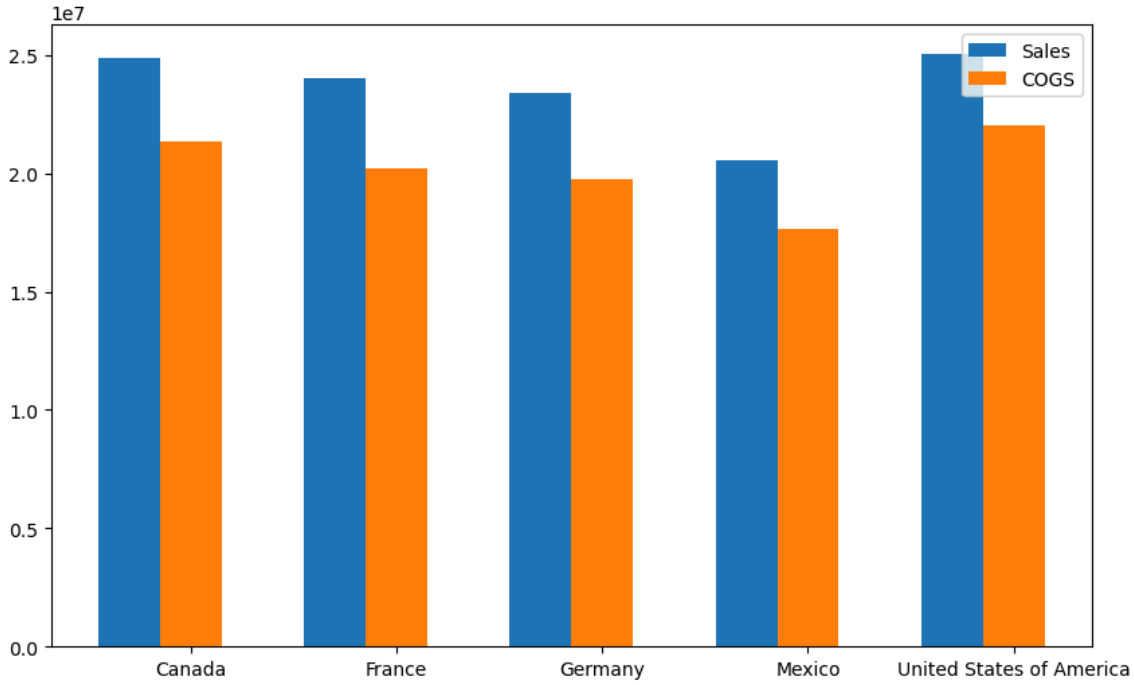
	Country	Sales	Profit	Units Sold	COGS
0	Canada	24887654.89	3858206.39	247428.5	21358426.0
1	France	23995612.29	4032488.29	237943.0	20214591.5
2	Germany	23409940.82	3961381.32	200699.0	19729552.0
3	Mexico	20547352.11	3198923.11	199975.0	17639829.0
4	United States of America	25029830.18	3397345.68	232627.5	22034289.5

In [29]:
```
bar_width = 0.3
plt.figure(figsize=(10,6))
#calculate the x positions for the bars
x = np.arange(len(group['Country']))
plt.bar(x-bar_width, group['Sales'], bar_width, label = 'Sales')

plt.xticks(x, group['Country'])
plt.legend()
plt.show()
```



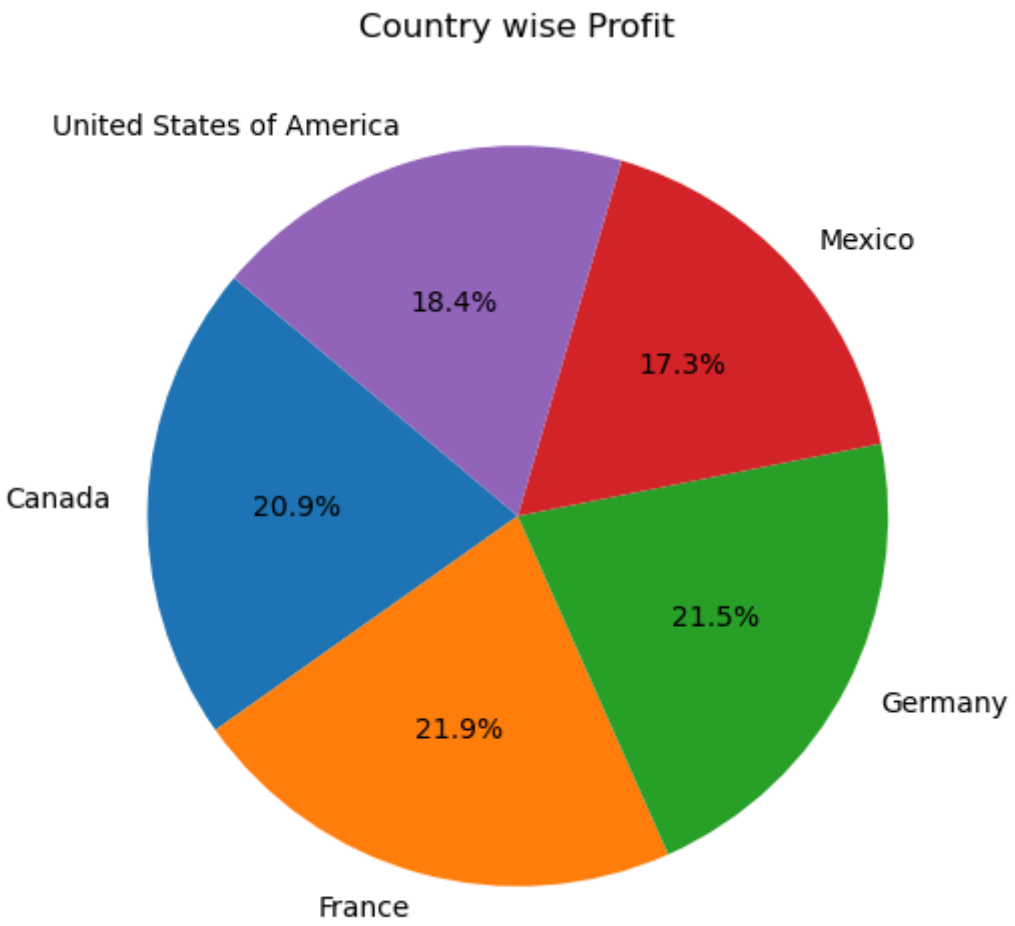
we can see that highest sales is noticed in USA and Canada and Lowest sale is recorded in Mexico

In [30]:
```
plt.figure(figsize=(10,6))
x = np.arange(len(group['Country']))
plt.bar(x-bar_width, group['Sales'], bar_width, label='Sales')
plt.bar(x, group['COGS'], bar_width, label='COGS')
# plt.bar(x+bar_width, group['Units Sold'], bar_width, label='Units Sold')

# plt.xlabel('Country')
# plt.ylabel('Values')
plt.xticks(x, group['Country'])
plt.legend()
plt.show()
```



We can see that sales is more than the COGS(Cost of good sale) it means company is in profit

In [31]:
```
plt.figure(figsize=(10,6))

plt.pie(group['Profit'], labels= group['Country'], autopct='%1.1f%%', starta
plt.title('Country wise Profit')
plt.show()
```



Product wise analysis

In [32]:
```
plt.figure(figsize=(10,6))

plt.pie(group_product['Profit'], labels=group_product['Product'], autopct='%
plt.title('Product wise Profit')
plt.show()
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Cell In[32], line 3
      1 plt.figure(figsize=(10,6))
----> 3 plt.pie(group_product['Profit'], labels=group_product['Product'],
autopct='%1.1f%%', startangle=140)
      4 plt.title('Product wise Profit')
      5 plt.show()

NameError: name 'group_product' is not defined


<Figure size 1000x600 with 0 Axes>
```

From here we can see that Paseo has highest profit and Montana have low profit

In [ ]:
```
plt.figure(figsize=(10,6))
x = np.arange(len(group_product['Product']))
plt.bar(x-bar_width, group_product['Gross Sales'], bar_width, label='Sales'
plt.bar(x, group_product['COGS'], bar_width, label='COGS')
# plt.bar(x+bar_width, group['Units Sold'], bar_width, label='Units Sold')

# plt.xlabel('Country')
# plt.ylabel('Values')
plt.xticks(x, group_product['Product'])
plt.legend()
plt.show()
```

Sales of Paseo is much more than other Products and Carretera has the low sales

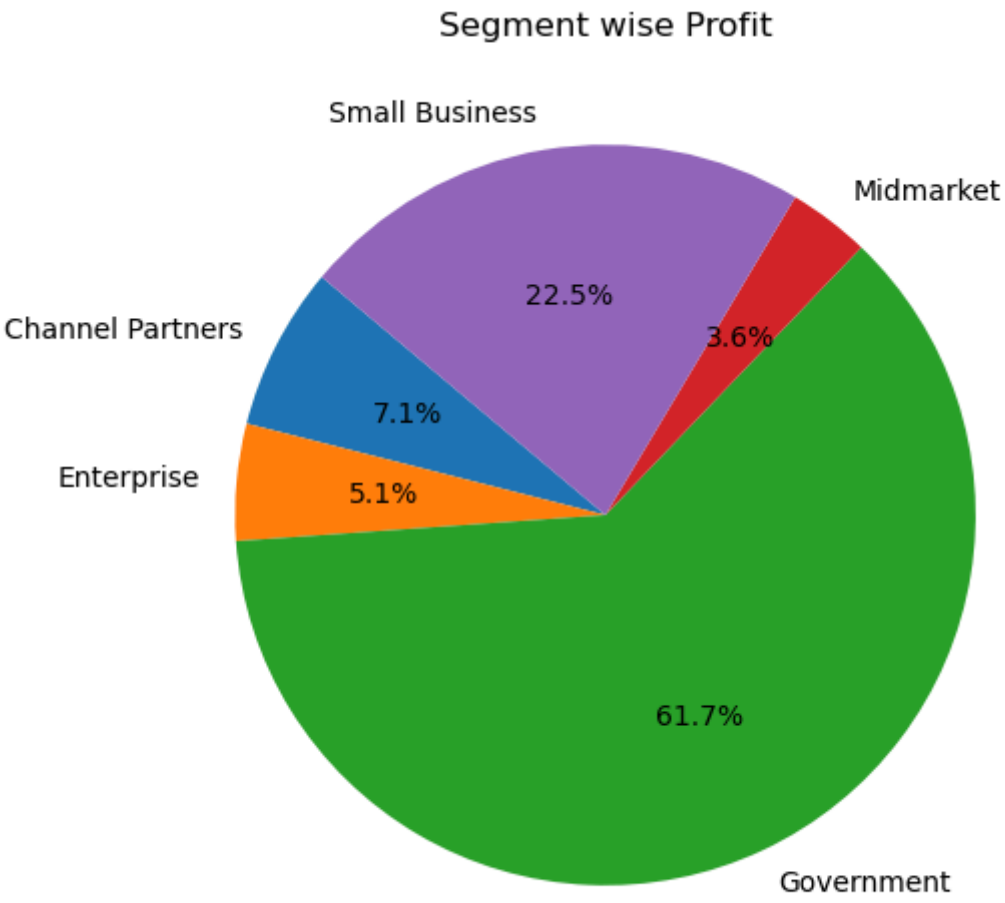
In [ ]:
```
bar_width = 0.3
plt.figure(figsize=(10,6))
# Calculate the x positions for the bars
x = np.arange(len(group_product['Product']))
plt.bar(x-bar_width, group_product['Units Sold'], bar_width, label='Units S

plt.xticks(x, group_product['Product'])
plt.legend()
plt.show()
```

From here we can clearly see that Paseo is the product which sells maximum

Sector wise analysis

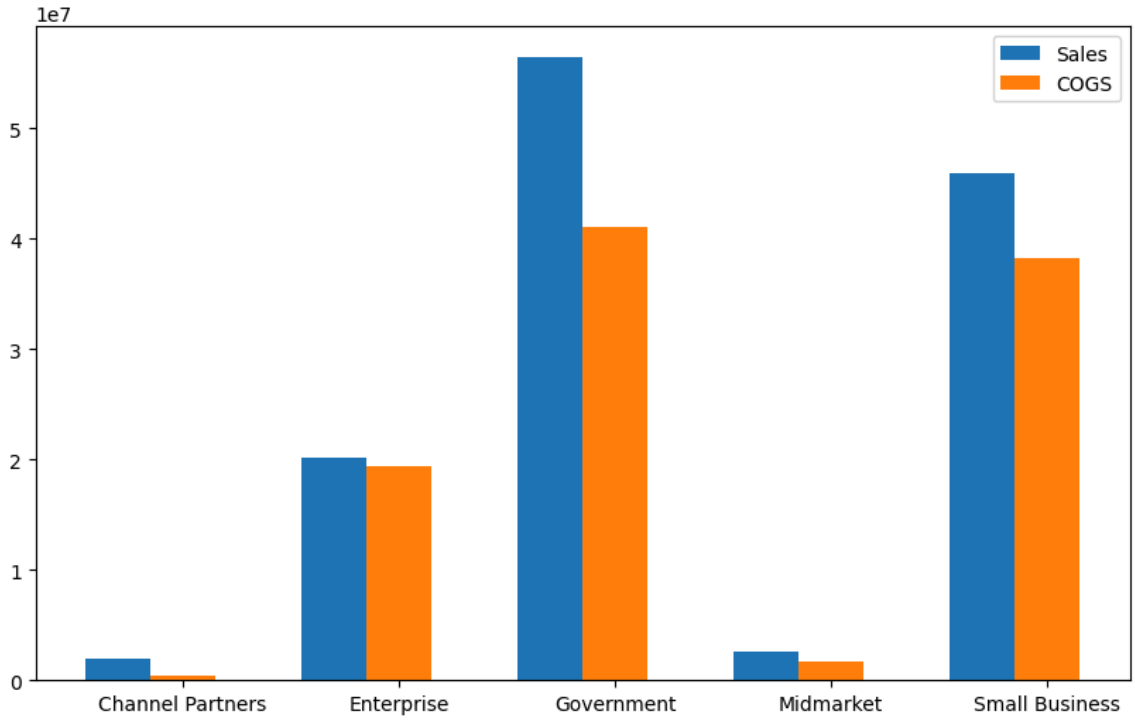
In [33]:
```
group_segment=df.groupby('Segment').agg({'Units Sold':'sum', 'Gross Sales':
```

In [34]:
```
plt.figure(figsize=(10,6))

plt.pie(group_segment['Profit'], labels=group_segment['Segment'], autopct='%
plt.title('Segment wise Profit')
plt.show()
```



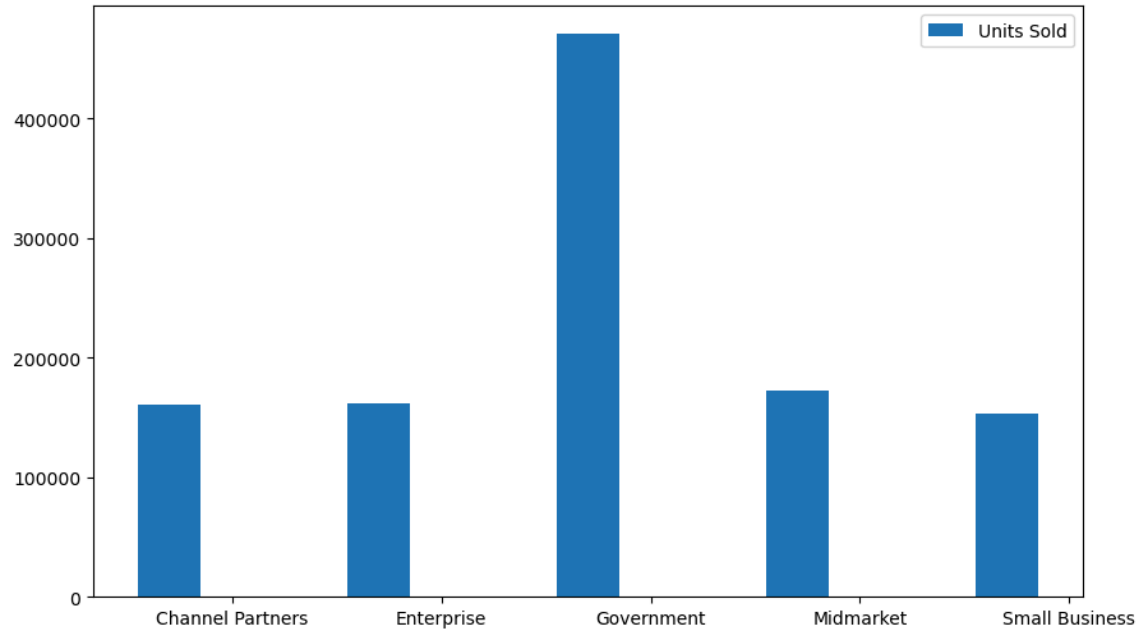
Goverment segment gives most profit and Midmarket gives least profit

In [35]:
```
plt.figure(figsize=(10,6))
x = np.arange(len(group_segment['Segment']))
plt.bar(x-bar_width, group_segment['Gross Sales'], bar_width, label='Sales'
plt.bar(x, group_segment['COGS'], bar_width, label='COGS')
# plt.bar(x+bar_width, group['Units Sold'], bar_width, label='Units Sold')

# plt.xlabel('Country')
# plt.ylabel('Values')
plt.xticks(x, group_segment['Segment'])
plt.legend()
plt.show()
```



In the Enterprise sector cost of sales and sales is almost same so we can say that margin in that sector could be low and in goverment sector difference between Sales and COGS is more means this sector has more margin

In [36]:
```
bar_width = 0.3
plt.figure(figsize=(10,6))
# Calculate the x positions for the bars
x = np.arange(len(group_segment['Segment']))
plt.bar(x-bar_width, group_segment['Units Sold'], bar_width, label='Units S
plt.xticks(x,group_segment['Segment'])
plt.legend()
plt.show()
```



more units are sold in goverment sector then mid market and least units are sold in small business

In [ ]: