

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

```
In [3]: df=pd.read_csv('Financials.csv',encoding='unicode_escape')
```

```
In [4]: df.shape
```

```
Out[4]: (700, 16)
```

```
In [171]: df.head()
```

```
Out[171]:
```

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts
0	Government	Canada	Carretera	None	1618.5	3.0	20.0	32370.0	0
1	Government	Germany	Carretera	None	1321.0	3.0	20.0	26420.0	0
2	Midmarket	France	Carretera	None	2178.0	3.0	15.0	32670.0	0
3	Midmarket	Germany	Carretera	None	888.0	3.0	15.0	13320.0	0
4	Midmarket	Mexico	Carretera	None	2470.0	3.0	15.0	37050.0	0

In [10]: 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
```

In [11]: df.columns=df.columns.str.strip()

In [13]: dollars = ['Units Sold', 'Manufacturing Price', 'Sale Price', 'Gross Sales', '  
for column in dollars:  
df[column] = pd.to\_numeric(df[column].replace('\\$', ''), regex=True), er

In [14]: df.head()

Out[14]:

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts
0	Government	Canada	Carretera	None	1618.5	3.0	20.0	32370.0	NaN
1	Government	Germany	Carretera	None	1321.0	3.0	20.0	26420.0	NaN
2	Midmarket	France	Carretera	None	2178.0	3.0	15.0	32670.0	NaN
3	Midmarket	Germany	Carretera	None	888.0	3.0	15.0	13320.0	NaN
4	Midmarket	Mexico	Carretera	None	2470.0	3.0	15.0	37050.0	NaN

In [15]: df['Discounts']=df['Discounts'].fillna('0')

In [16]: `df.head()`

Out[16]:

	Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Discounts
0	Government	Canada	Carretera	None	1618.5	3.0	20.0	32370.0	0
1	Government	Germany	Carretera	None	1321.0	3.0	20.0	26420.0	0
2	Midmarket	France	Carretera	None	2178.0	3.0	15.0	32670.0	0
3	Midmarket	Germany	Carretera	None	888.0	3.0	15.0	13320.0	0
4	Midmarket	Mexico	Carretera	None	2470.0	3.0	15.0	37050.0	0

In [17]: `df['Date']=pd.to_datetime(df['Date'])`

In [18]: `df.info()`

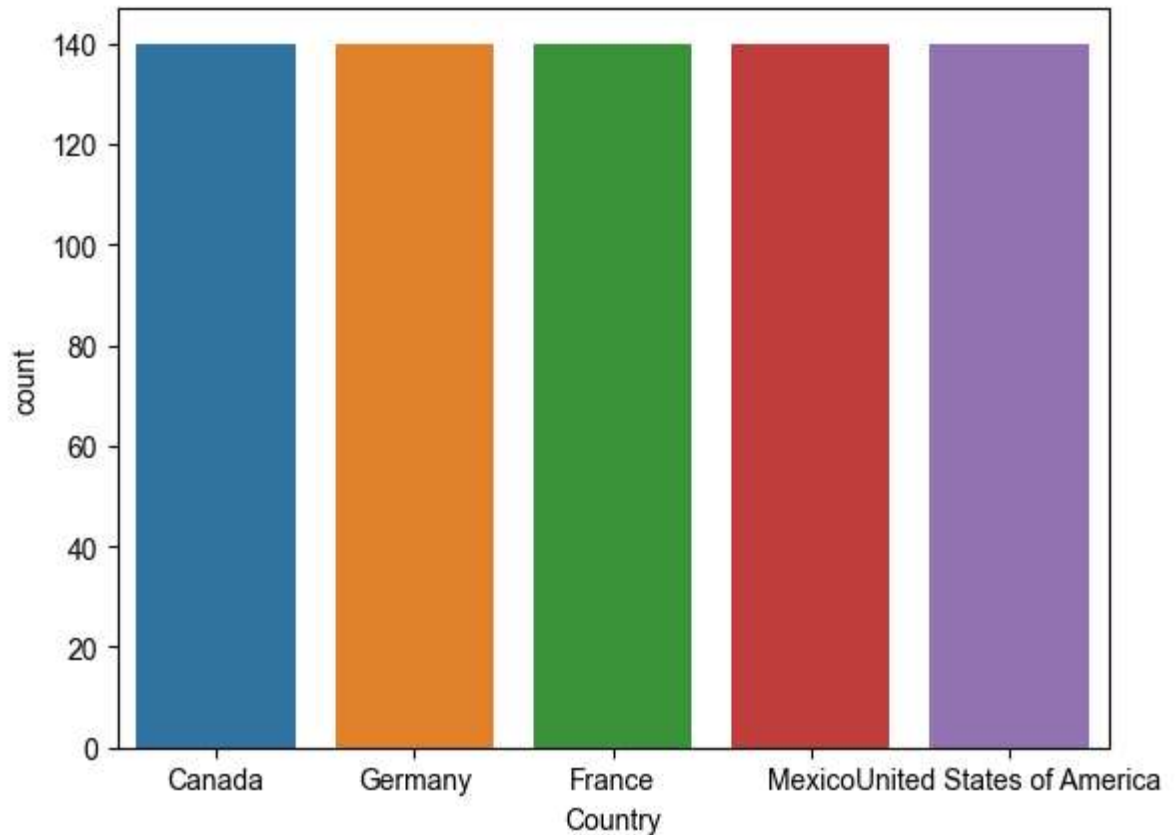
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Segment                700 non-null    object
1   Country                700 non-null    object
2   Product                700 non-null    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    object
9   Sales                  700 non-null    float64
10  COGS                   700 non-null    float64
11  Profit                 637 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(7), int64(2), object(6)
memory usage: 87.6+ KB
```

```
In [19]: df.describe()
```

Out[19]:

	Units Sold	Manufacturing Price	Sale Price	Gross Sales	Sales	COGS	
count	700.000000	700.000000	700.000000	7.000000e+02	7.000000e+02	700.000000	6
mean	1608.294286	96.477143	118.428571	1.827594e+05	1.696091e+05	145475.211429	277
min	200.000000	3.000000	7.000000	1.799000e+03	1.655080e+03	918.000000	2
25%	905.000000	5.000000	12.000000	1.739175e+04	1.592800e+04	7490.000000	36
50%	1542.500000	10.000000	20.000000	3.798000e+04	3.554020e+04	22506.250000	117
75%	2229.125000	250.000000	300.000000	2.790250e+05	2.610775e+05	245607.500000	257
max	4492.500000	260.000000	350.000000	1.207500e+06	1.159200e+06	950625.000000	2622
std	867.427859	108.602612	136.775515	2.542623e+05	2.367263e+05	203865.506118	436

```
In [20]: sns.countplot(x='Country',data=df)
sns.set(rc={'figure.figsize':(15,5)})
```



```
In [21]: country_group = df.groupby('Country').agg({'Sales': 'sum', 'Profit': 'sum', 'Units Sold': 'sum', 'COGS': 'sum'})
```

```
In [22]: country_group.head()
```

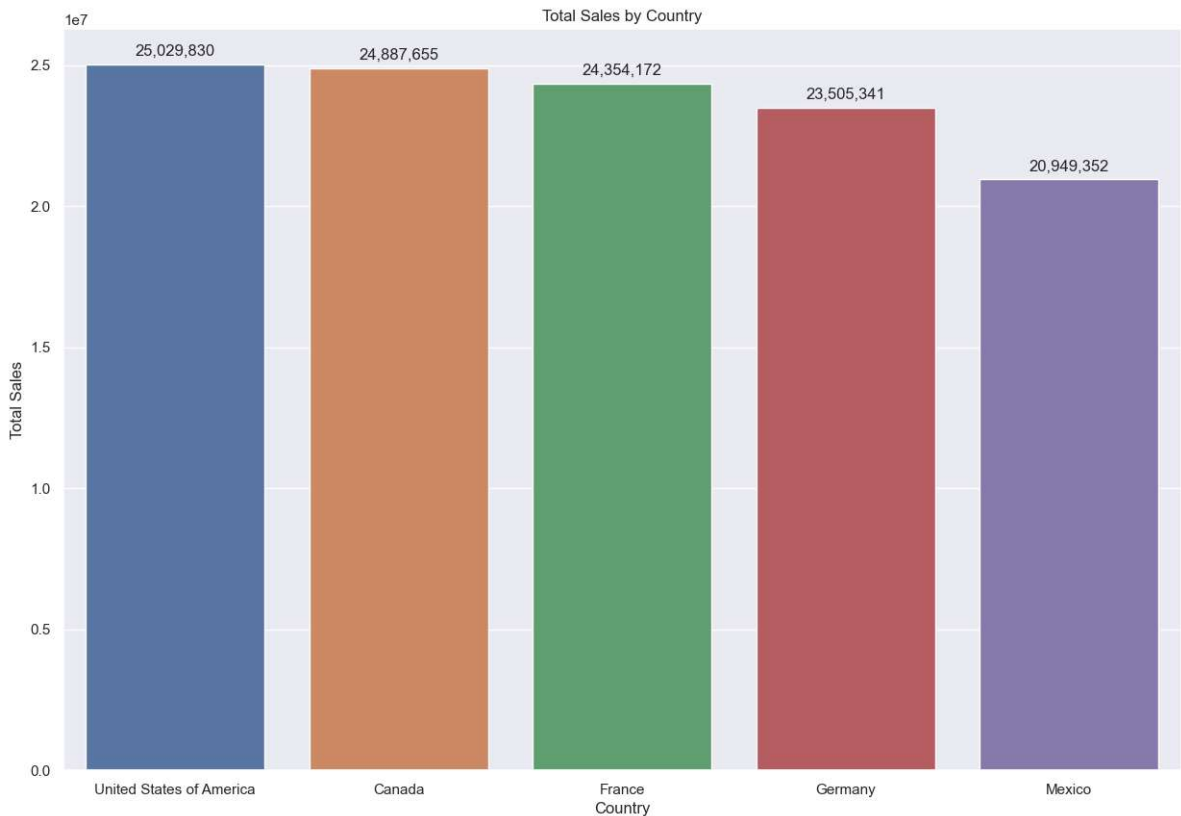
Out[22]:

	Country	Sales	Profit	Units Sold	COGS
0	Canada	24887654.89	3693717.64	247428.5	21358426.0
1	France	24354172.29	3906754.54	240931.0	20573151.5
2	Germany	23505340.82	3820885.07	201494.0	19824952.0
3	Mexico	20949352.11	3053223.11	203325.0	18041829.0
4	United States of America	25029830.18	3196443.18	232627.5	22034289.5

**total sales countrywise :**

```
In [23]: sales=df.groupby(['Country'],as_index=False)['Sales'].sum().sort_values(by='Sales')
plt.figure(figsize=(15, 10))
ax=sns.barplot(x='Country',y='Sales',data=sales)
for p in ax.patches:
    ax.annotate(f'{p.get_height():.0f}',
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset p

plt.title('Total Sales by Country')
plt.xlabel('Country')
plt.ylabel('Total Sales')
plt.show()
```



from above graph we can see that more sales from USA and lowest sales from mexico

**total profit countrywise :**

```
In [24]: profit=df.groupby(['Country'],as_index=False)['Profit'].sum().sort_values(by='
plt.figure(figsize=(10,5))
ax=sns.barplot(data=profit,x='Country',y='Profit')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.0f}',(p.get_x()+p.get_width()/2.,p.get_hei
        textcoords='offset points')
plt.title('TOTAL PROFIT BY COUNTRY')
plt.xlabel('Country')
plt.ylabel('Total Profit')
plt.show()
```



from above grap we can see that more profit in france and less profit in mexico

```
In [25]: country_group = df.groupby('Country').agg({'Sales': 'sum', 'Profit': 'sum', 'U
country_group.head()
```

Out[25]:

	Country	Sales	Profit	Units Sold	COGS
0	Canada	24887654.89	3693717.64	247428.5	21358426.0
1	France	24354172.29	3906754.54	240931.0	20573151.5
2	Germany	23505340.82	3820885.07	201494.0	19824952.0
3	Mexico	20949352.11	3053223.11	203325.0	18041829.0
4	United States of America	25029830.18	3196443.18	232627.5	22034289.5

## total analysis :

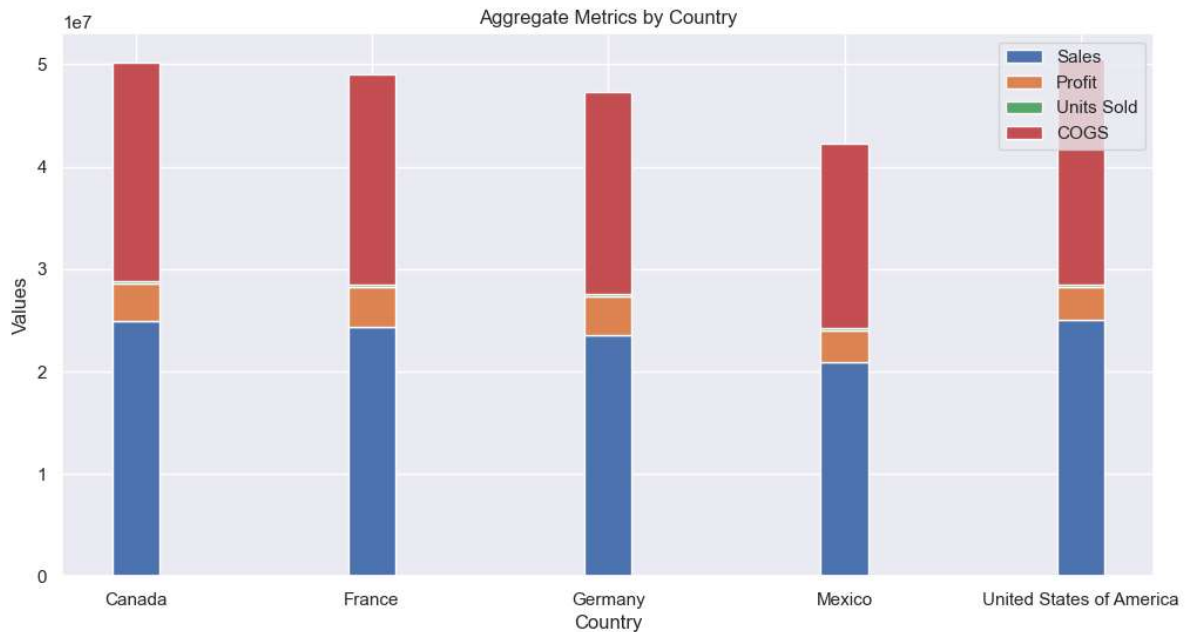
```
In [26]: country_group = df.groupby('Country').agg({'Sales': 'sum', 'Profit': 'sum', 'Units Sold': 'sum', 'COGS': 'sum'})

country_group = country_group.sort_values(by='Country')
countries = country_group['Country']
sales = country_group['Sales']
profit = country_group['Profit']
units_sold = country_group['Units Sold']
cogs = country_group['COGS']

width = 0.2
fig, ax = plt.subplots(figsize=(12, 6))
ax.bar(countries, sales, width, label='Sales')
ax.bar(countries, profit, width, label='Profit', bottom=sales)
ax.bar(countries, units_sold, width, label='Units Sold', bottom=sales+profit)
ax.bar(countries, cogs, width, label='COGS', bottom=sales+profit+units_sold)

ax.set_title('Aggregate Metrics by Country')
ax.set_xlabel('Country')
ax.set_ylabel('Values')
ax.legend()

plt.show()
```



**total sales countrywise :**

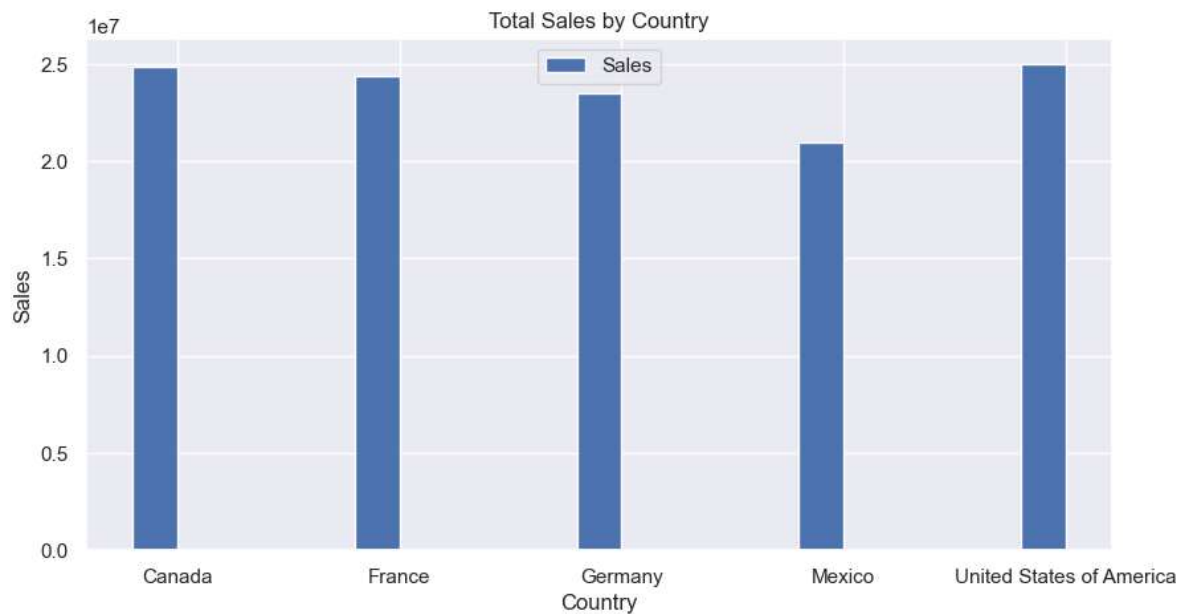


```
In [27]: country_group = df.groupby('Country').agg({'Sales': 'sum'}).reset_index()
plt.figure(figsize=(10,5))
width=0.2
x = np.arange(len(country_group['Country']))

plt.bar(x - width / 2 , country_group['Sales'], width, label='Sales')

plt.xlabel('Country')
plt.ylabel('Sales')
plt.title('Total Sales by Country')
plt.xticks(x, country_group['Country'])
plt.legend()

plt.show()
```



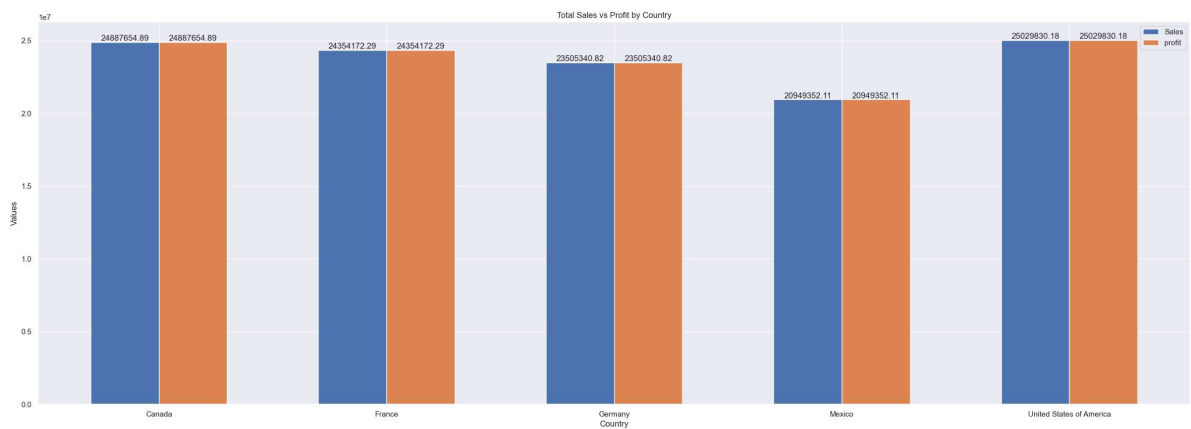
**total Sales vs Profit countrywise :**

```
In [58]: country_group = df.groupby('Country').agg({'Sales': 'sum', 'Profit': 'sum'}).re
plt.figure(figsize=(30,10))
width=0.3
x = np.arange(len(country_group['Country']))

bar1=plt.bar(x - width / 2 , country_group['Sales'], width, label='Sales')
bar2=plt.bar(x + width / 2 , country_group['Sales'], width, label='profit')

for bar, value in zip(bar1, country_group['Sales']):
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), str(value),
             ha='center', va='bottom')
for bar, value in zip(bar2, country_group['Sales']):
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), str(value),
             ha='center', va='bottom')
plt.xlabel('Country')
plt.ylabel('Values')
plt.title('Total Sales vs Profit by Country')
plt.xticks(x, country_group['Country'])
plt.legend()

plt.show()
```

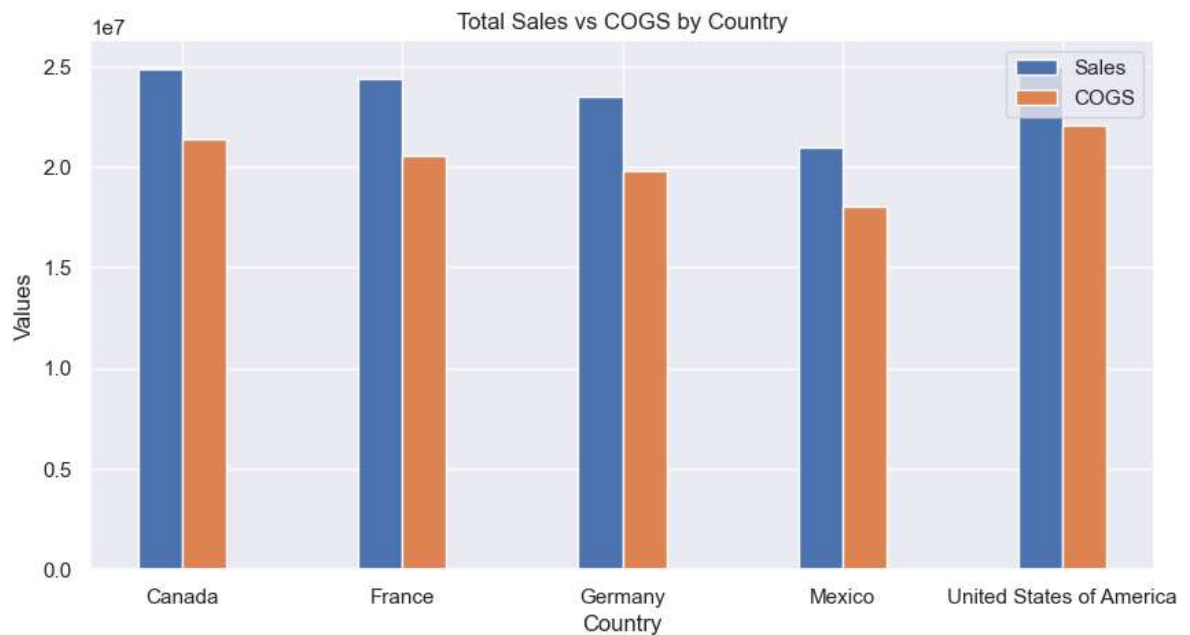


## Total Sales vs COGS by Country

```
In [54]: CGS=df.groupby(['Country']).agg({'Sales': 'sum','COGS' : 'sum'}).reset_index()
plt.figure(figsize=(10,5))
width=0.2
ax=np.arange(len(country_group['Country']))
plt.bar(x - width / 2 , CGS['Sales'], width, label='Sales')
plt.bar(x + width / 2 , CGS['COGS'], width, label='COGS' )

plt.xlabel('Country')
plt.ylabel('Values')
plt.title('Total Sales vs COGS by Country')
plt.xticks(x, CGS['Country'])
plt.legend()

plt.show()
```

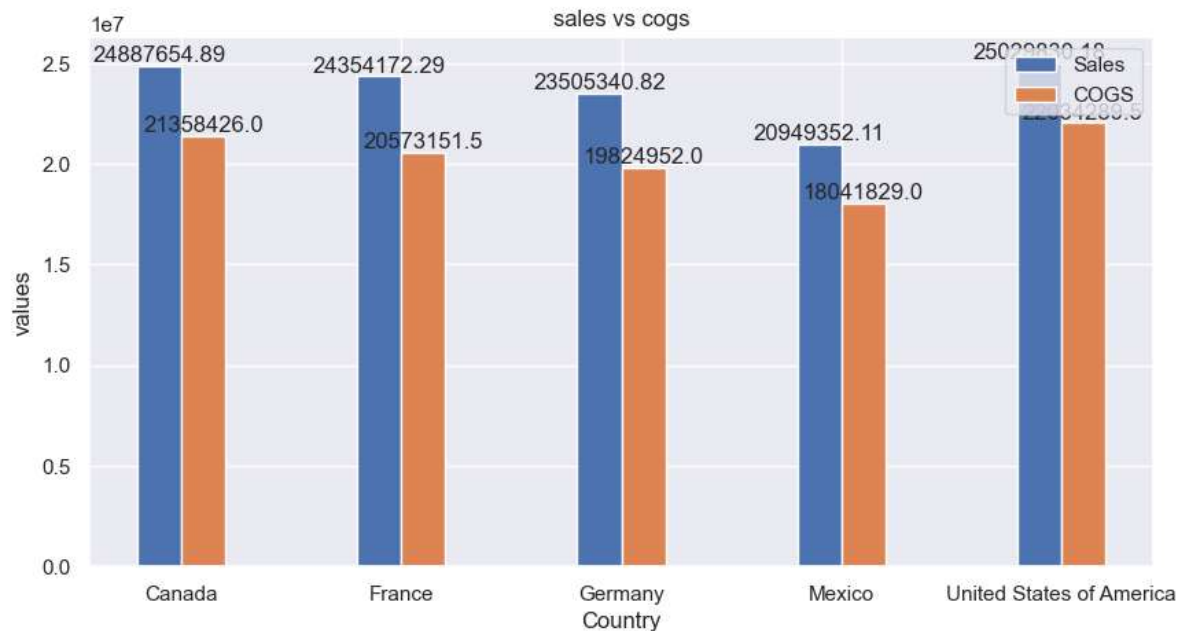


## Total Sales vs COGS by Country

```
In [38]: SL = df.groupby(['Country']).agg({'Sales':'sum','COGS':'sum'}).reset_index()
plt.figure(figsize=(10,5))
width=0.2
ax=np.arange(len(SL['Country']))
bar1=plt.bar(ax - width/2,SL['Sales'],width,label='Sales')
bar2=plt.bar(ax + width/2,SL['COGS'],width,label='COGS')

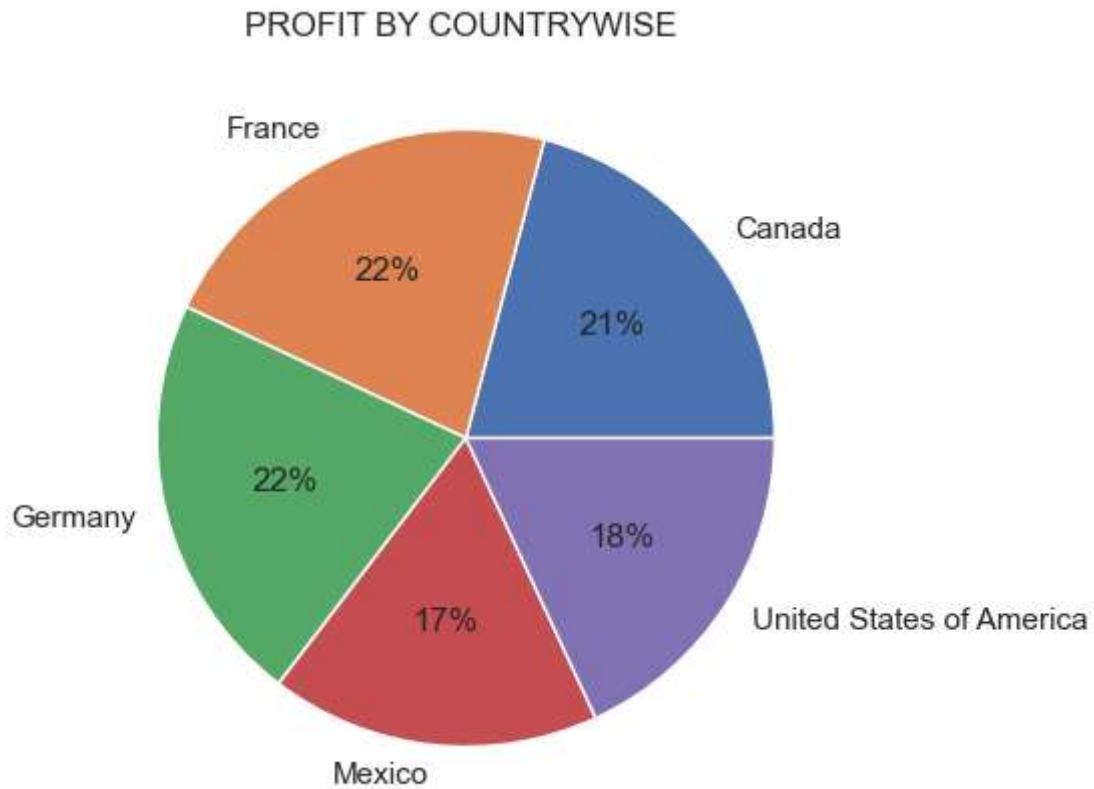
for bar, value in zip(bar1, SL['Sales']):
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), str(value),
             ha='center', va='bottom')
for bar, value in zip(bar2, SL['COGS']):
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), str(value),
             ha='center', va='bottom')

plt.xlabel('Country')
plt.ylabel('values')
plt.title('sales vs cogs')
plt.xticks(x, SL['Country'])
plt.legend()
plt.show()
```



## PROFIT BY COUNTRY WISE

```
In [61]: plt.figure(figsize=(10,5))
plt.pie(country_group['Profit'],labels=country_group['Country'],autopct='%1.0f')
plt.title('PROFIT BY COUNTRYWISE ')
plt.show()
```



Above pie chart france in higher profit and mexico is lowest profit

## PRODUCT WISE PROFIT MARGIN

```
In [122]: product = df.groupby(['Product']).agg({
    'Units Sold': 'sum',
    'Manufacturing Price': 'mean',
    'Sale Price': 'sum',
    'Gross Sales': 'sum',
    'COGS': 'sum',
    'Profit': 'sum'
})
product['margin']=product['Sale Price']-product['Manufacturing Price']

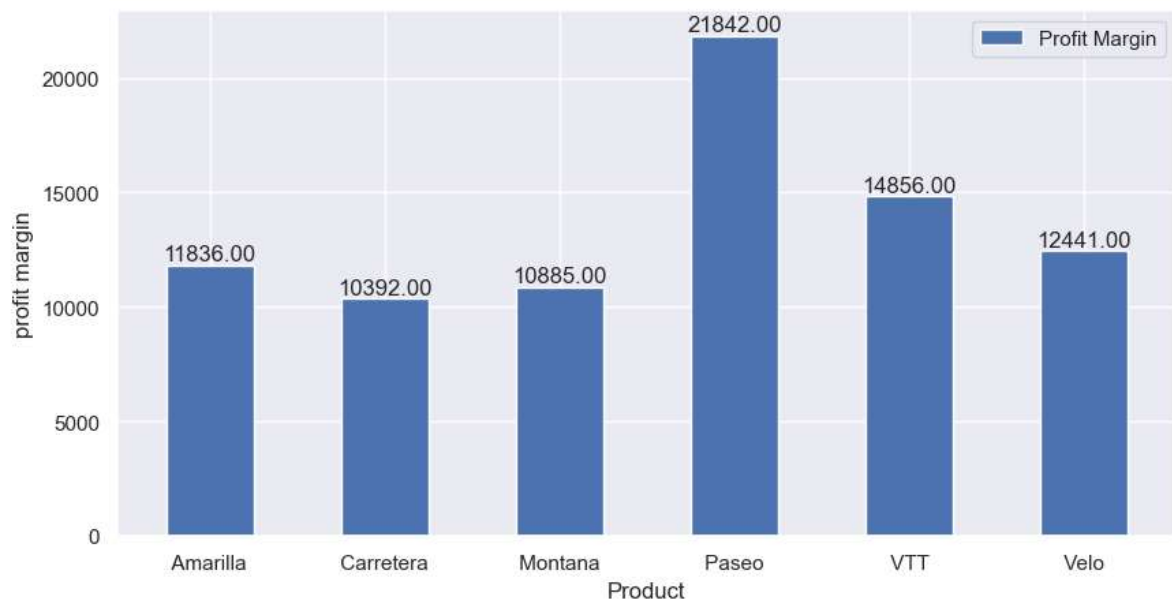
plt.figure(figsize=(10,5))
width=0.5

x = range(len(product))

bars=plt.bar(x, product['margin'], width, label='Profit Margin')

for bar, value in zip(bars, product['margin']):
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), f'{value:.2f}'
             ha='center', va='bottom')

plt.xlabel('Product')
plt.ylabel('profit margin')
plt.legend()
plt.xticks(x, product.index)
plt.show()
```

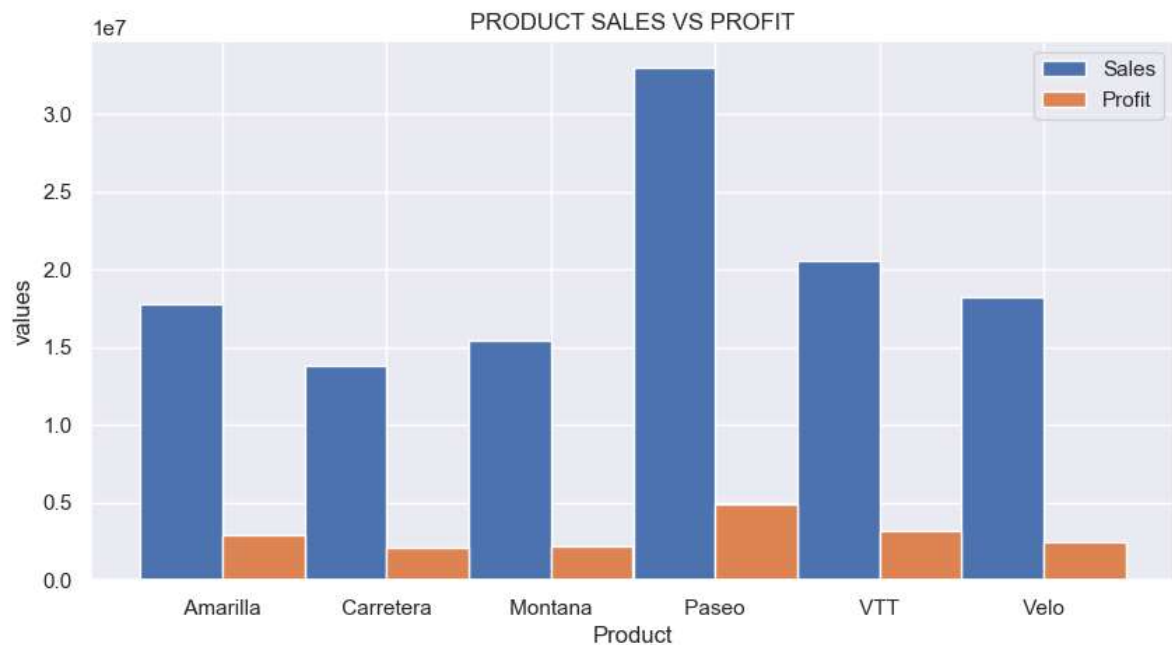


## PRODUCT SALES VS PROFIT

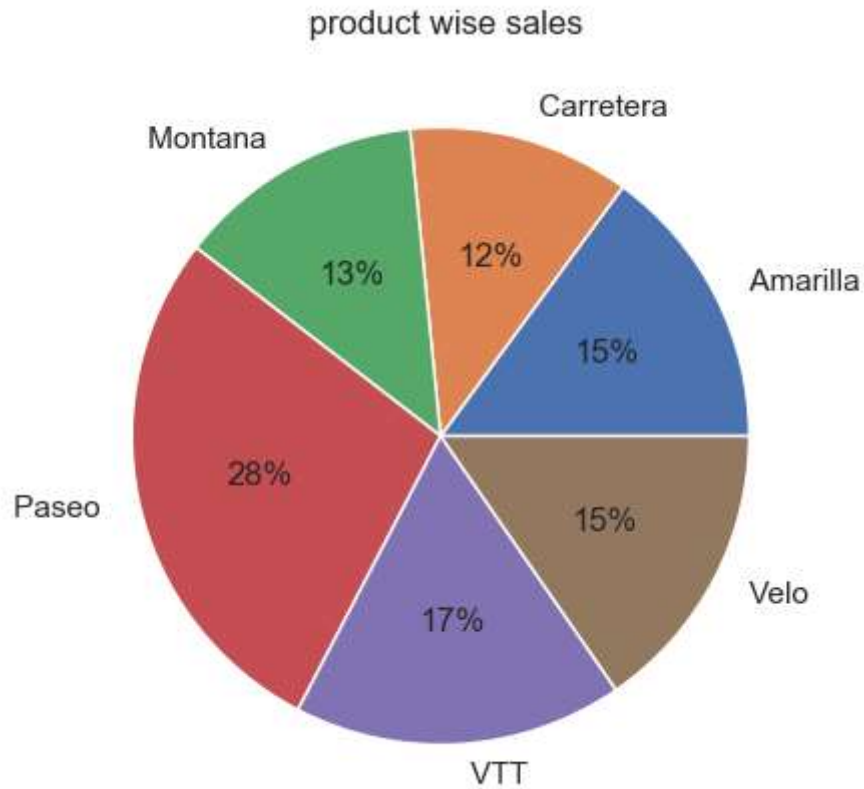
```
In [170]: AF=df.groupby(['Product']).agg({'Sales':'sum','Profit':'sum'}).reset_index()
width=0.5
plt.figure(figsize=(10,5))
x=np.arange(len(AF['Product']))
plt.bar(x-width/2,AF['Sales'],width,label='Sales')

plt.bar(x+width/2,AF['Profit'],width,label='Profit')

plt.xlabel('Product')
plt.ylabel('values')
plt.title('PRODUCT SALES VS PROFIT')
plt.xticks(x,AF['Product'])
plt.legend()
plt.show()
```



```
In [142]: plt.figure(figsize=(10,5))
plt.pie(AF['Sales'],labels=AF['Product'],autopct='%1.0f%%')
plt.title(' product wise sales')
plt.show()
```



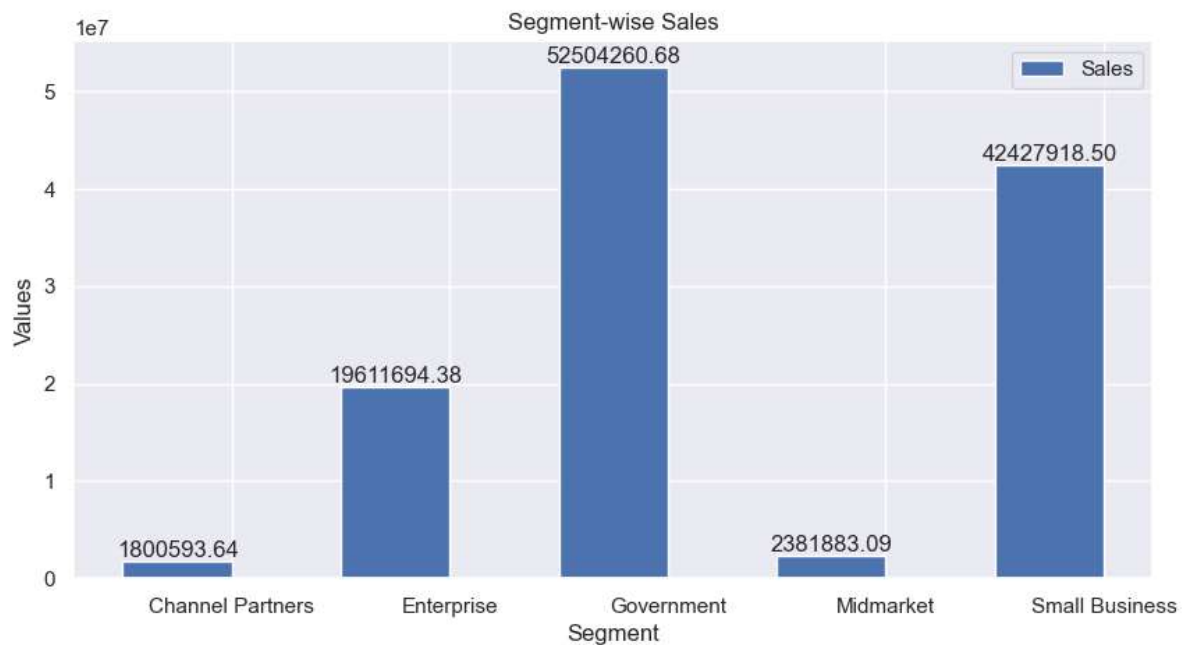
## SEGMENT WISE SALES



```
In [167]: BF = df.groupby(['Segment']).agg({'Sales': 'sum', 'Profit': 'sum'}).reset_index()

plt.figure(figsize=(10, 5))
width = 0.5
Ax = np.arange(len(BF['Segment']))

bars=plt.bar(Ax - width/2, BF['Sales'], width, label='Sales')
for bar, value in zip(bars, BF['Sales']):
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height(), f'{value:.2f}',
             ha='center', va='bottom')
plt.xlabel('Segment')
plt.ylabel('Values')
plt.title('Segment-wise Sales')
plt.xticks(Ax, BF['Segment'])
plt.legend()
plt.show()
```



**# Thank you**