

```
In [1]: #import libraries
import pandas as pd
import numpy as np
```

```
In [2]: #use pandas to read sales_data_na.csv
data = pd.read_csv("sales_data_na.csv")
data
```

Out[2]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Product_Category	Sub_Category	Product	Order_Quantity	Unit_Cost	Unit_Price	Profit	Cost	Revenue
0	11/26/2013	26	November	2013	19.0	Youth (<25)	M	Canada	British Columbia	Accessories	Bike Racks	NaN	8	45.0	120.0	590.0	NaN	950.0
1	11/26/2015	26	NaN	2015	19.0	NaN	M	Canada	British Columbia	Accessories	Bike Racks	Hitch Rack - 4-Bike	8	45.0	NaN	590.0	360.0	950.0
2	3/23/2014	23	March	2014	NaN	Adults (35-64)	M	Australia	NaN	Accessories	Bike Racks	Hitch Rack - 4-Bike	23	45.0	120.0	1366.0	1035.0	NaN
3	3/23/2016	23	March	2016	49.0	Adults (35-64)	M	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	20	45.0	120.0	1188.0	900.0	2088.0
4	5/15/2014	15	May	2014	47.0	Adults (35-64)	F	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	4	45.0	120.0	238.0	180.0	418.0
...
113032	4/12/2016	12	April	2016	41.0	Adults (35-64)	M	United Kingdom	England	Clothing	Vests	Classic Vest, S	3	24.0	64.0	112.0	72.0	184.0
113033	4/2/2014	2	April	2014	18.0	Youth (<25)	M	Australia	Queensland	Clothing	Vests	Classic Vest, M	22	24.0	64.0	655.0	528.0	1183.0
113034	4/2/2016	2	April	2016	18.0	Youth (<25)	M	Australia	Queensland	Clothing	Vests	Classic Vest, M	22	24.0	64.0	655.0	528.0	1183.0
113035	3/4/2014	4	March	2014	37.0	Adults (35-64)	F	France	Seine (Paris)	Clothing	Vests	Classic Vest, L	24	24.0	64.0	684.0	576.0	1260.0
113036	3/4/2016	4	March	2016	37.0	Adults (35-64)	F	France	Seine (Paris)	Clothing	Vests	Classic Vest, L	23	24.0	64.0	655.0	552.0	1207.0

113037 rows × 18 columns

```
In [3]: data.isnull()
```

Out[3]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Product_Category	Sub_Category	Product	Order_Quantity	Unit_Cost	Unit_Price	Profit	Cost	Revenue
0	False	False	False	False	False	False	False	False	False	False	False	True	False	False	False	False	True	False
1	False	False	True	False	False	True	False	False	False	False	False	False	False	False	True	False	False	False
2	False	False	False	False	True	False	False	False	True	False	False	False	False	False	False	False	False	True
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...
113032	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
113033	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
113034	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
113035	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
113036	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

113037 rows × 18 columns

In [4]:

data.isnull().sum()

Out[4]:

Date0
Day0
Month1
Year0
Customer_Age1
Age_Group1
Customer_Gender0
Country0
State1
Product_Category0
Sub_Category0
Product1
Order_Quantity0
Unit_Cost3
Unit_Price1
Profit3
Cost1
Revenue5
dtype: int64

In [5]:

data["Revenue"].isnull()

```
Out[5]: 0      False
        1      False
        2       True
        3      False
        4      False
        ...
113032  False
113033  False
113034  False
113035  False
113036  False
Name: Revenue, Length: 113037, dtype: bool
```

```
In [6]: # filling missing value with "No review"
        # refer step 4, apply fillna('No review') for all the fields that have >=1      null

data['Month'] = data['Month'].fillna('No review')
data['Customer_Age'] = data['Customer_Age'].fillna('No review')
data['Age_Group'] = data['Age_Group'].fillna('No review')
data['State'] = data['State'].fillna('No review')
data['Product'] = data['Product'].fillna('No review')
data['Unit_Cost'] = data['Unit_Cost'].fillna('No review')
data['Unit_Price'] = data['Unit_Price'].fillna('No review')
data['Profit'] = data['Profit'].fillna('No review')
data['Cost'] = data['Cost'].fillna('No review')
data['Revenue'] = data['Revenue'].fillna('No review')

data
```

Out [6]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Product_Category	Sub_Category	Product	Order_Quantity	Unit_Cost	Unit_Price	Profit	Cost	Revenue
0	11/26/2013	26	November	2013	19.0	Youth (<25)	M	Canada	British Columbia	Accessories	Bike Racks	No review	8	45.0	120.0	590.0	No review	950.0
1	11/26/2015	26	No review	2015	19.0	No review	M	Canada	British Columbia	Accessories	Bike Racks	Hitch Rack - 4-Bike	8	45.0	No review	590.0	360.0	950.0
2	3/23/2014	23	March	2014	No review	Adults (35-64)	M	Australia	No review	Accessories	Bike Racks	Hitch Rack - 4-Bike	23	45.0	120.0	1366.0	1035.0	No review
3	3/23/2016	23	March	2016	49.0	Adults (35-64)	M	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	20	45.0	120.0	1188.0	900.0	2088.0
4	5/15/2014	15	May	2014	47.0	Adults (35-64)	F	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	4	45.0	120.0	238.0	180.0	418.0
...
113032	4/12/2016	12	April	2016	41.0	Adults (35-64)	M	United Kingdom	England	Clothing	Vests	Classic Vest, S	3	24.0	64.0	112.0	72.0	184.0
113033	4/2/2014	2	April	2014	18.0	Youth (<25)	M	Australia	Queensland	Clothing	Vests	Classic Vest, M	22	24.0	64.0	655.0	528.0	1183.0
113034	4/2/2016	2	April	2016	18.0	Youth (<25)	M	Australia	Queensland	Clothing	Vests	Classic Vest, M	22	24.0	64.0	655.0	528.0	1183.0
113035	3/4/2014	4	March	2014	37.0	Adults (35-64)	F	France	Seine (Paris)	Clothing	Vests	Classic Vest, L	24	24.0	64.0	684.0	576.0	1260.0
113036	3/4/2016	4	March	2016	37.0	Adults (35-64)	F	France	Seine (Paris)	Clothing	Vests	Classic Vest, L	23	24.0	64.0	655.0	552.0	1207.0

113037 rows × 18 columns

In [7]: data['Month'].unique()

Out[7]: array(['November', 'No review', 'March', 'May', 'February', 'July', 'August', 'September', 'January', 'December', 'June', 'October', 'Aogust', 'April', 'Marsh', 'Jone', 'Joly'], dtype=object)

In [8]: *#Use the "replace()" function*

data['Month'].replace('Aogust','August')
data['Month'].replace('Marsh','March')
data['Month'].replace('Jone','June')
data['Month'].replace('Joly','July')

Out[8]: 0 November
1 No review
2 March
3 March
4 May

...
113032 April
113033 April
113034 April
113035 March
113036 March
Name: Month, Length: 113037, dtype: object

In [9]: *#Use the "replace()" function*

```
data['Month'].replace('Aogust','August')
data['Month'].replace('Marsh','March')
data['Month'].replace('Jone','June')
data['Month'].replace('Joly','July')
```

Out[9]:

0	November
1	No review
2	March
3	March
4	May
...	
113032	April
113033	April
113034	April
113035	March
113036	March

Name: Month, Length: 113037, dtype: object

In [10]: data['Country'].unique()

Out[10]: array(['Canada', 'Australia', 'United States', 'Germany', 'France',
 'United Kingdom'], dtype=object)

In [11]: *#check on duplicated data*
data.duplicated()

Out[11]:

0	False
1	False
2	False
3	False
4	False
...	
113032	False
113033	False
113034	False
113035	False
113036	False

Length: 113037, dtype: bool

In [12]: *#check on the total number of duplicated data*
data.duplicated().sum()

Out[12]: 1001

In [13]: *#drop the duplicated data*
data2 = data.drop_duplicates()
print(data2)

	Date	Day	Month	Year	Customer_Age	Age_Group	\
0	11/26/2013	26	November	2013	19.0	Youth (<25)	
1	11/26/2015	26	No review	2015	19.0	No review	
2	3/23/2014	23	March	2014	No review	Adults (35-64)	
3	3/23/2016	23	March	2016	49.0	Adults (35-64)	
4	5/15/2014	15	May	2014	47.0	Adults (35-64)	
...	
113032	4/12/2016	12	April	2016	41.0	Adults (35-64)	
113033	4/2/2014	2	April	2014	18.0	Youth (<25)	
113034	4/2/2016	2	April	2016	18.0	Youth (<25)	
113035	3/4/2014	4	March	2014	37.0	Adults (35-64)	
113036	3/4/2016	4	March	2016	37.0	Adults (35-64)	

	Customer_Gender	Country	State	Product_Category	\
0	M	Canada	British Columbia	Accessories	
1	M	Canada	British Columbia	Accessories	
2	M	Australia	No review	Accessories	
3	M	Australia	New South Wales	Accessories	
4	F	Australia	New South Wales	Accessories	
...	
113032	M	United Kingdom	England	Clothing	
113033	M	Australia	Queensland	Clothing	
113034	M	Australia	Queensland	Clothing	
113035	F	France	Seine (Paris)	Clothing	
113036	F	France	Seine (Paris)	Clothing	

	Sub_Category	Product	Order_Quantity	Unit_Cost	Unit_Price	\
0	Bike Racks	No review	8	45.0	120.0	
1	Bike Racks	Hitch Rack - 4-Bike	8	45.0	No review	
2	Bike Racks	Hitch Rack - 4-Bike	23	45.0	120.0	
3	Bike Racks	Hitch Rack - 4-Bike	20	45.0	120.0	
4	Bike Racks	Hitch Rack - 4-Bike	4	45.0	120.0	
...	
113032	Vests	Classic Vest, S	3	24.0	64.0	
113033	Vests	Classic Vest, M	22	24.0	64.0	
113034	Vests	Classic Vest, M	22	24.0	64.0	
113035	Vests	Classic Vest, L	24	24.0	64.0	
113036	Vests	Classic Vest, L	23	24.0	64.0	

	Profit	Cost	Revenue
0	590.0	No review	950.0
1	590.0	360.0	950.0
2	1366.0	1035.0	No review
3	1188.0	900.0	2088.0
4	238.0	180.0	418.0
...
113032	112.0	72.0	184.0
113033	655.0	528.0	1183.0
113034	655.0	528.0	1183.0
113035	684.0	576.0	1260.0
113036	655.0	552.0	1207.0

[112036 rows x 18 columns]

```
In [14]: #Use istitle() function to check on the casing
('Date').istitle()
('Day').istitle()
('Month').istitle()
('Year').istitle()
('Customer_Age').istitle()
```

```
('Age_Group').istitle()
('Customer_Gender').istitle()
('Country').istitle()
('State').istitle()
('Product_Category').istitle()
('Sub_Category').istitle()
('Product').istitle()
('Order_Quantity').istitle()
('Unit_Cost').istitle()
('Unit_Price').istitle()
('Profit').istitle()
('Cost').istitle()
('Revenue').istitle()
```

Out[14]: True

```
In [15]: #checking data types
data2.dtypes
```

```
Out[15]: Date          object
Day              int64
Month           object
Year            int64
Customer_Age     object
Age_Group        object
Customer_Gender  object
Country          object
State            object
Product_Category object
Sub_Category     object
Product          object
Order_Quantity   int64
Unit_Cost        object
Unit_Price       object
Profit           object
Cost             object
Revenue          object
dtype: object
```

```
In [16]: data2['Profit'] = pd.to_numeric(data2['Profit'],errors='coerce')
data2['Cost'] = pd.to_numeric(data2['Cost'],errors='coerce')
data2['Revenue'] = pd.to_numeric(data2['Revenue'],errors='coerce')
print(data2)
```

	Date	Day	Month	Year	Customer_Age	Age_Group \
0	11/26/2013	26	November	2013	19.0	Youth (<25)
1	11/26/2015	26	No review	2015	19.0	No review
2	3/23/2014	23	March	2014	No review	Adults (35-64)
3	3/23/2016	23	March	2016	49.0	Adults (35-64)
4	5/15/2014	15	May	2014	47.0	Adults (35-64)
...
113032	4/12/2016	12	April	2016	41.0	Adults (35-64)
113033	4/2/2014	2	April	2014	18.0	Youth (<25)
113034	4/2/2016	2	April	2016	18.0	Youth (<25)
113035	3/4/2014	4	March	2014	37.0	Adults (35-64)
113036	3/4/2016	4	March	2016	37.0	Adults (35-64)

	Customer_Gender	Country	State	Product_Category \
0	M	Canada	British Columbia	Accessories
1	M	Canada	British Columbia	Accessories
2	M	Australia	No review	Accessories
3	M	Australia	New South Wales	Accessories
4	F	Australia	New South Wales	Accessories
...
113032	M	United Kingdom	England	Clothing
113033	M	Australia	Queensland	Clothing
113034	M	Australia	Queensland	Clothing
113035	F	France	Seine (Paris)	Clothing
113036	F	France	Seine (Paris)	Clothing

	Sub_Category	Product	Order_Quantity	Unit_Cost	Unit_Price \
0	Bike Racks	No review	8	45.0	120.0
1	Bike Racks	Hitch Rack - 4-Bike	8	45.0	No review
2	Bike Racks	Hitch Rack - 4-Bike	23	45.0	120.0
3	Bike Racks	Hitch Rack - 4-Bike	20	45.0	120.0
4	Bike Racks	Hitch Rack - 4-Bike	4	45.0	120.0
...
113032	Vests	Classic Vest, S	3	24.0	64.0
113033	Vests	Classic Vest, M	22	24.0	64.0
113034	Vests	Classic Vest, M	22	24.0	64.0
113035	Vests	Classic Vest, L	24	24.0	64.0
113036	Vests	Classic Vest, L	23	24.0	64.0

	Profit	Cost	Revenue
0	590.0	NaN	950.0
1	590.0	360.0	950.0
2	1366.0	1035.0	NaN
3	1188.0	900.0	2088.0
4	238.0	180.0	418.0
...
113032	112.0	72.0	184.0
113033	655.0	528.0	1183.0
113034	655.0	528.0	1183.0
113035	684.0	576.0	1260.0
113036	655.0	552.0	1207.0

[112036 rows x 18 columns]


```
C:\Users\DELL\AppData\Local\Temp\ipykernel_21532\1291404572.py: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
data2['Profit'] = pd.to_numeric(data2['Profit'],errors='coerce')
C:\Users\DELL\AppData\Local\Temp\ipykernel_21532\1291404572.py:2: 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
data2['Cost'] = pd.to_numeric(data2['Cost'],errors='coerce')
C:\Users\DELL\AppData\Local\Temp\ipykernel_21532\1291404572.py:3: 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
data2['Revenue'] = pd.to_numeric(data2['Revenue'],errors='coerce')
```

```
In [17]: #checking data types
data2.dtypes
```

```
Out[17]: Date          object
Day             int64
Month           object
Year            int64
Customer_Age    object
Age_Group       object
Customer_Gender object
Country         object
State           object
Product_Category object
Sub_Category    object
Product         object
Order_Quantity  int64
Unit_Cost       object
Unit_Price      object
Profit          float64
Cost            float64
Revenue         float64
dtype: object
```

```
In [18]: data2['Country'].unique()
```

```
Out[18]: array(['Canada', 'Australia', 'United States', 'Germany', 'France',
               'United Kingdom'], dtype=object)
```

```
In [19]: #Grouping total revenue by country
result1 = data2.groupby('Country')['Revenue'].sum()
```

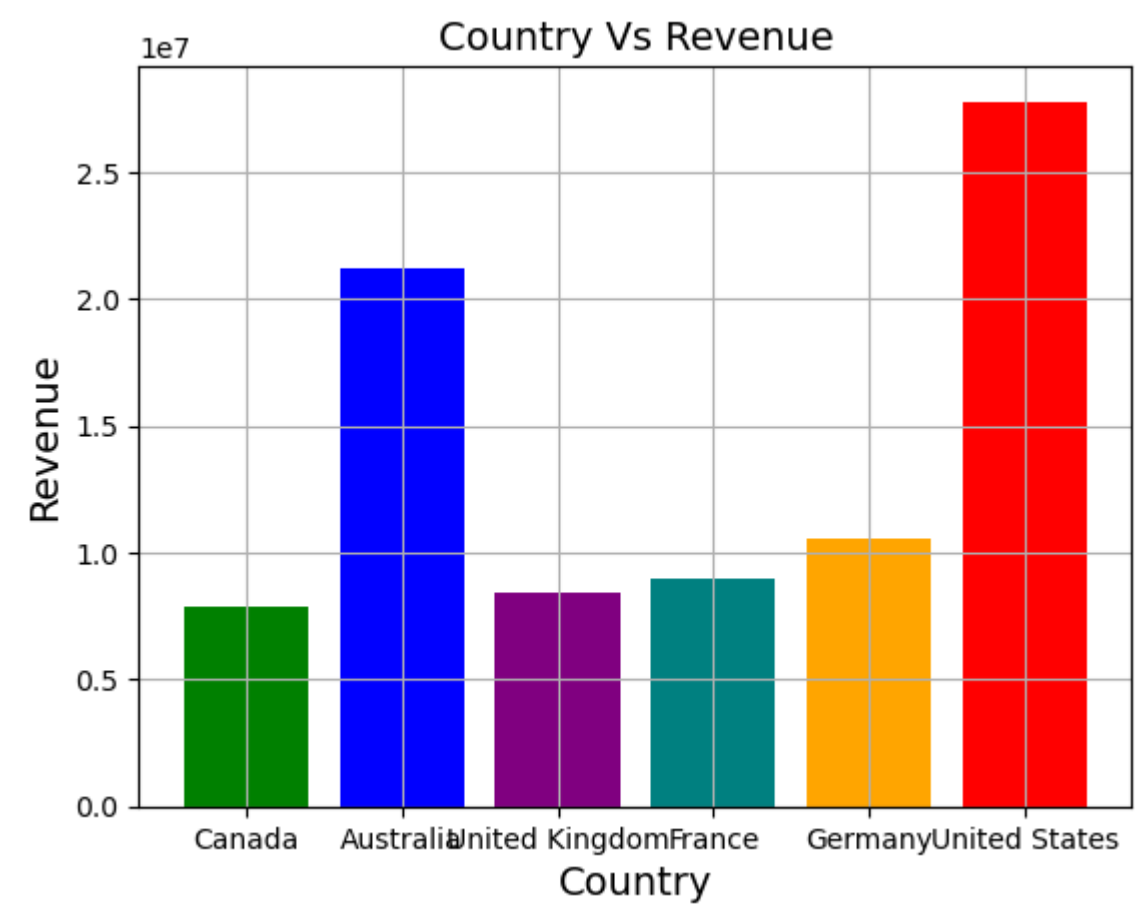
```
In [20]: print(result1)
```

```
Country
Australia    21193994.0
Canada        7906182.0
France        8414745.0
Germany       8956724.0
United Kingdom 10575628.0
United States 27772186.0
Name: Revenue, dtype: float64
```

```
In [21]: import matplotlib.pyplot as plt
import pandas as pd

Data = {'Country': ['Canada','Australia','United Kingdom','France','Germany','United States'],
        'Revenue': [7906182,21193994,8414745,8956724,10575628,27772186]}
df = pd.DataFrame(Data,columns=['Country','Revenue'])

New_Colors = ['green','blue','purple','teal','orange','red']
plt.bar(df['Country'], df['Revenue'], color=New_Colors)
plt.title('Country Vs Revenue', fontsize=14)
plt.xlabel('Country', fontsize=14)
plt.ylabel('Revenue', fontsize=14)
plt.grid(True)
plt.show()
```



```
In [22]: #Grouping total profit by country
result2 = data2.groupby('Country')['Profit'].sum()
print(result2)
```

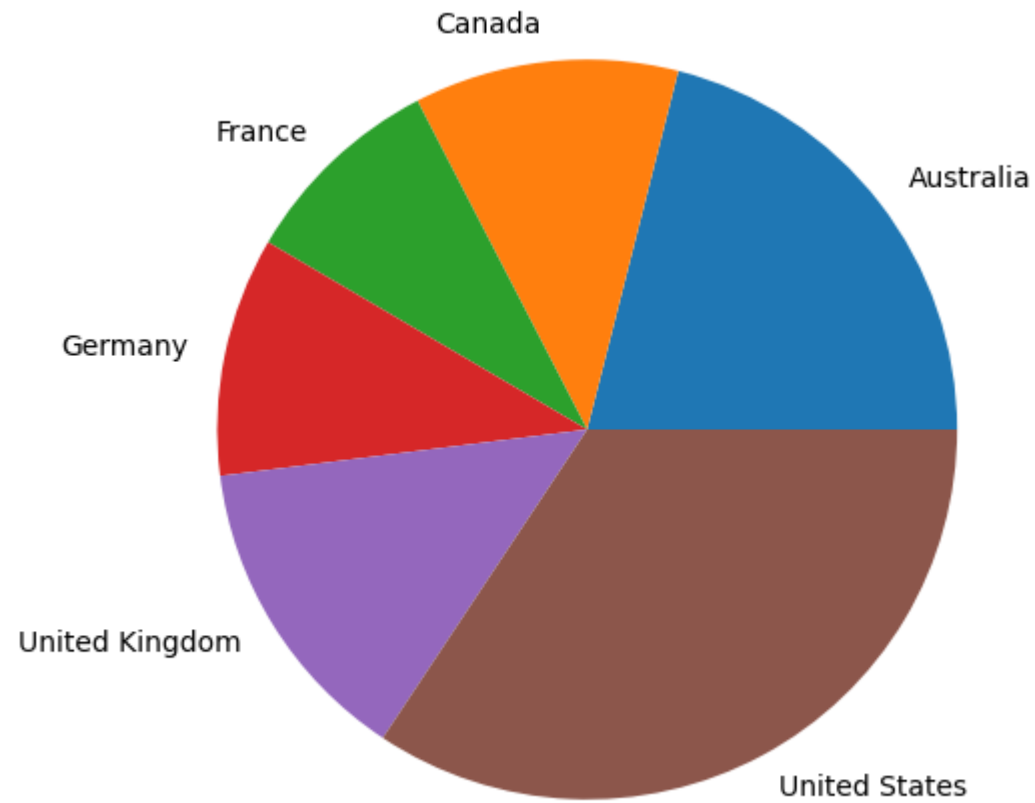
```
Country
Australia    6740522.0
Canada       3702019.0
France       2872082.0
Germany      3349991.0
United Kingdom 4383987.0
United States 10996028.0
Name: Profit, dtype: float64
```

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

Ctry = ["Australia", "Canada", "France", "Germany", "United Kingdom", "United States"]
CtryPrt = np.array([6740522, 3702019, 2872082, 3349991, 4383987, 10996028])

fig = plt.figure(figsize=(9, 6))
plt.pie(CtryPrt, labels=Ctry)

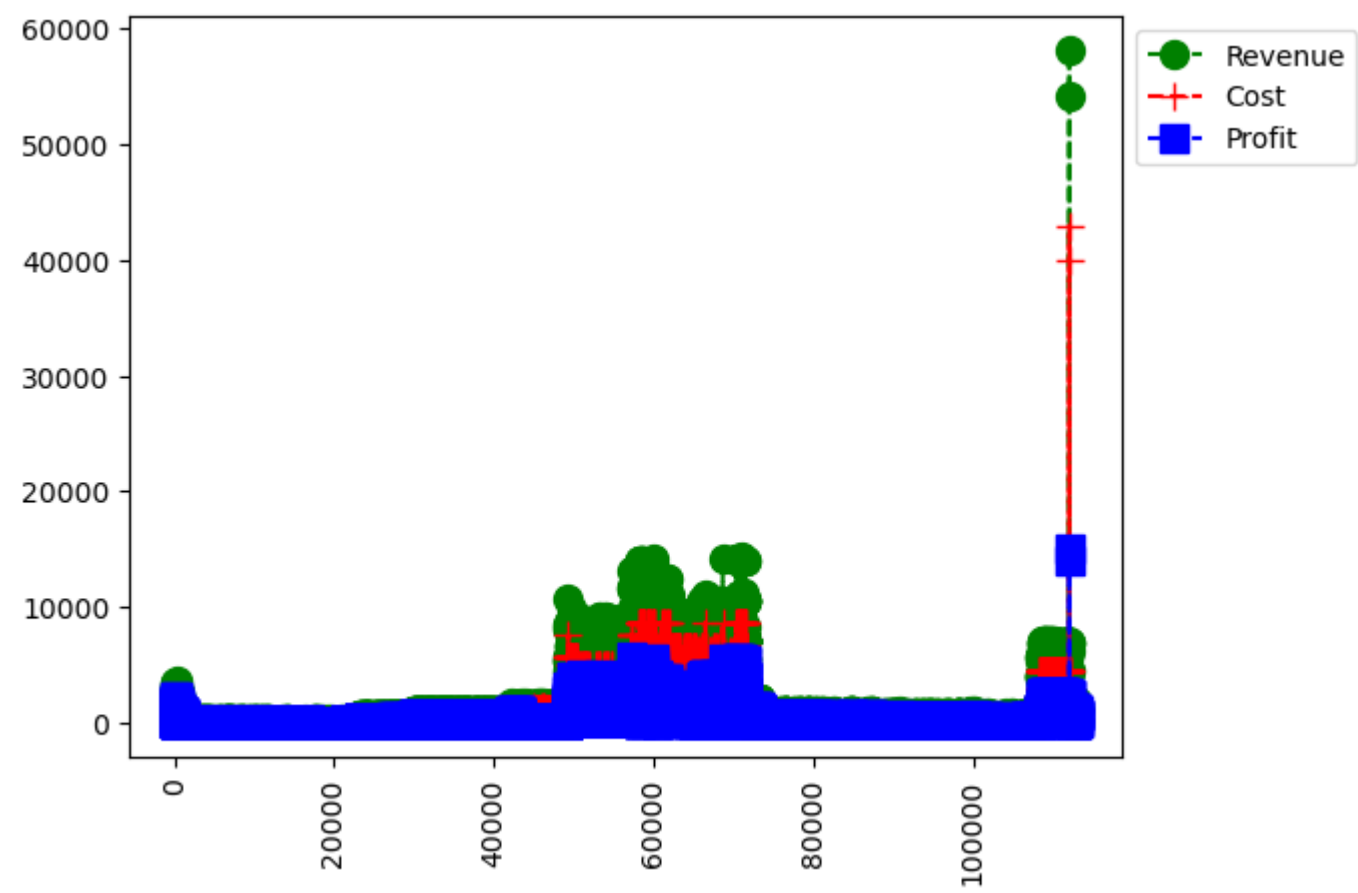
plt.show()
```



```
In [24]: import matplotlib.pyplot as plt
%matplotlib inline

plt.plot(data2['Revenue'], color="green", ls="--", marker="o", ms=10, label="Revenue") ,
plt.plot(data2['Cost'], color="red", ls="--", marker="+", ms=10, label="Cost"),
plt.plot(data2['Profit'], color="blue", ls="--", marker="s", ms=10, label="Profit")

plt.legend(loc='upper left', bbox_to_anchor=(1,1))
#to show the labels at proper location
plt.xticks(rotation="vertical")
plt.show()
```



In []: