Project 1 - Amazon Sales Data

In [79]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')                                    # i
```

In [80]:
```
df = pd.read_csv("data1/Amazon_Sales_data.csv")                      # i
```

In [81]:
```
print('Rows: {} Columns: {}'.format(df.shape[0], df.shape[1]))       # de
```

```
Rows: 100 Columns: 14
```

In [82]:
```
df.head(10)                                                  #first 10 data entry from d
```

Out[82]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	U Co
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014	8102	9.33	6.
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.
5	Australia and Oceania	Solomon Islands	Baby Food	Online	C	2/4/2015	547995746	2/21/2015	2974	255.28	159.
6	Sub-Saharan Africa	Angola	Household	Offline	M	4/23/2011	135425221	4/27/2011	4187	668.27	502.
7	Sub-Saharan Africa	Burkina Faso	Vegetables	Online	H	7/17/2012	871543967	7/27/2012	8082	154.06	90.
8	Sub-Saharan Africa	Republic of the Congo	Personal Care	Offline	M	7/14/2015	770463311	8/25/2015	6070	81.73	56.
9	Sub-Saharan Africa	Senegal	Cereal	Online	H	4/18/2014	616607081	5/30/2014	6593	205.70	117

In [83]:
```
df.tail(15)                                                    #last 15 data entry
```

Out[83]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Un Pric
85	North America	Mexico	Personal Care	Offline	L	2/17/2012	430915820	3/20/2012	6422	81.7
86	Sub-Saharan Africa	Sao Tome and Principe	Beverages	Offline	C	1/16/2011	180283772	1/21/2011	8829	47.4
87	Sub-Saharan Africa	The Gambia	Baby Food	Offline	M	2/3/2014	494747245	3/20/2014	5559	255.2
88	Middle East and North Africa	Kuwait	Fruits	Online	M	4/30/2012	513417565	5/18/2012	522	9.3
89	Europe	Slovenia	Beverages	Offline	C	10/23/2016	345718562	11/25/2016	4660	47.4
90	Sub-Saharan Africa	Sierra Leone	Office Supplies	Offline	H	12/6/2016	621386563	12/14/2016	948	651.2
91	Australia and Oceania	Australia	Beverages	Offline	H	7/7/2014	240470397	7/11/2014	9389	47.4
92	Middle East and North Africa	Azerbaijan	Office Supplies	Online	M	6/13/2012	423331391	7/24/2012	2021	651.2
93	Europe	Romania	Cosmetics	Online	H	11/26/2010	660643374	12/25/2010	7910	437.2
94	Central America and the Caribbean	Nicaragua	Beverages	Offline	C	2/8/2011	963392674	3/21/2011	8156	47.4
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	9/3/2011	888	109.2
96	Asia	Malaysia	Fruits	Offline	L	11/11/2011	810711038	12/28/2011	6267	9.3
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	6/1/2016	728815257	6/29/2016	1485	154.0
98	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	8/8/2015	5767	81.7
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2/10/2012	665095412	2/15/2012	5367	668.2

In [84]:
```
df.info()                                                    #all information regardin
#Observations:
#1.There are in total 100 samples in the amazon_sales_data set
#2.There are both categorical and numerical attributes in the dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Region          100 non-null    object
 1   Country         100 non-null    object
 2   Item Type       100 non-null    object
 3   Sales Channel   100 non-null    object
 4   Order Priority  100 non-null    object
 5   Order Date      100 non-null    object
 6   Order ID        100 non-null    int64
 7   Ship Date       100 non-null    object
 8   Units Sold      100 non-null    int64
 9   Unit Price      100 non-null    float64
 10  Unit Cost       100 non-null    float64
 11  Total Revenue   100 non-null    float64
 12  Total Cost      100 non-null    float64
 13  Total Profit    100 non-null    float64
dtypes: float64(5), int64(2), object(7)
memory usage: 11.1+ KB
```

In [85]:
```
df.nunique()                                                 #finding out no. of unique v
```

Out[85]:
```
Region              7
Country            76
Item Type          12
Sales Channel       2
Order Priority      4
Order Date        100
Order ID          100
Ship Date          99
Units Sold         99
Unit Price         12
Unit Cost          12
Total Revenue     100
Total Cost        100
Total Profit      100
dtype: int64
```

In [86]:
```
for i, col in enumerate(df.columns):
    print(df.columns[i],":", df[str(col)].unique(), '\n')
```

```
5
 '4/23/2011' '7/17/2012' '7/14/2015' '4/18/2014' '6/24/2011' '8/2/2014'
 '1/13/2017' '2/8/2017' '2/19/2014' '4/23/2012' '11/19/2016' '4/1/2015'
 '12/30/2010' '7/31/2012' '5/14/2014' '7/31/2015' '6/30/2016' '9/8/2014'
 '5/7/2016' '5/22/2017' '10/13/2014' '5/7/2010' '7/18/2014' '5/26/2012'
 '9/17/2012' '12/29/2013' '10/27/2015' '1/16/2015' '2/25/2017' '5/8/2017'
 '11/22/2011' '1/14/2017' '4/1/2012' '2/16/2012' '3/11/2017' '2/6/2010'
 '6/7/2012' '10/6/2012' '11/14/2015' '3/29/2016' '12/31/2016' '12/23/2010'
 '10/14/2014' '1/11/2012' '2/2/2010' '8/18/2013' '3/25/2013' '11/26/2011'
 '9/17/2013' '6/8/2012' '6/30/2010' '2/23/2015' '1/5/2012' '4/7/2014'
 '6/9/2013' '6/26/2013' '11/7/2011' '10/30/2010' '10/13/2013' '10/11/2013'
 '7/8/2012' '7/25/2016' '10/24/2010' '4/25/2015' '4/23/2013' '8/14/2015'
 '5/26/2011' '5/20/2017' '7/5/2013' '11/6/2014' '10/28/2014' '9/15/2011'
 '5/29/2012' '7/20/2013' '10/21/2012' '9/18/2012' '11/15/2016' '1/4/2011'
 '3/18/2012' '2/17/2012' '1/16/2011' '2/3/2014' '4/30/2012' '10/23/2016'
 '12/6/2016' '7/7/2014' '6/13/2012' '11/26/2010' '2/8/2011' '7/26/2011'
 '11/11/2011' '6/1/2016' '7/30/2015' '2/10/2012']

Order ID : [669165933 963881480 341417157 514321792 115456712 547995746 135425221
 871543967 770463311 616607081 814711606 939825713 187310731 522840487
```

numerical attributes

In [87]:
```
num_attributes = df.select_dtypes(include=['int'])
print(num_attributes.columns)                                    # Ider
```

```
Index(['Order ID', 'Units Sold'], dtype='object')
```

In [88]:
```
df[['Units Sold', 'Unit Price', 'Unit Cost', 'Total Revenue', 'Total Cost', 'Total Pr
# Using describe function on dataframe for getting basic stats of numerical dataset
```

Out[88]:

	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
count	100.000000	100.000000	100.000000	1.000000e+02	1.000000e+02	1.000000e+02
mean	5128.710000	276.761300	191.048000	1.373488e+06	9.318057e+05	4.416820e+05
std	2794.484562	235.592241	188.208181	1.460029e+06	1.083938e+06	4.385379e+05
min	124.000000	9.330000	6.920000	4.870260e+03	3.612240e+03	1.258020e+03
25%	2836.250000	81.730000	35.840000	2.687212e+05	1.688680e+05	1.214436e+05
50%	5382.500000	179.880000	107.275000	7.523144e+05	3.635664e+05	2.907680e+05
75%	7369.000000	437.200000	263.330000	2.212045e+06	1.613870e+06	6.358288e+05
max	9925.000000	668.270000	524.960000	5.997055e+06	4.509794e+06	1.719922e+06

categorical_attributes

In [89]:
```
categorical_attributes = df.select_dtypes(include=['object'])
print(categorical_attributes.columns)
```

```
Index(['Region', 'Country', 'Item Type', 'Sales Channel', 'Order Priority',
       'Order Date', 'Ship Date'],
      dtype='object')
```

In [90]:
```
categorical_attributes.describe()
```

Out[90]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Ship Date
count	100	100	100	100	100	100	100
unique	7	76	12	2	4	100	99
top	Sub-Saharan Africa	The Gambia	Clothes	Offline	H	5/28/2010	11/17/2010
freq	36	4	13	50	30	1	2

In [91]:
```
# Changing the data type of different column for model training and analysis
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Ship Date'] = pd.to_datetime(df['Ship Date'])


df['Region'] = df['Region'].astype(str)
df['Country'] = df['Country'].astype(str)
df['Item Type'] = df['Item Type'].astype(str)
df['Sales Channel'] = df['Sales Channel'].astype(str)
df['Order Priority'] = df['Order Priority'].astype(str)
```

In [92]:
```
# Adding extra column to dataframe which contain only month, year and month with year
df['Order_Month'] = df['Order Date'].dt.month
df['Order_Year'] = df['Order Date'].dt.year
df['Order_Date_MonthYear'] = df['Order Date'].dt.strftime('%Y-%m')
df = df.drop(columns=['Order Date'])
```

In [93]:
```
df.tail(10)
```

Out[93]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Rev
90	Sub-Saharan Africa	Sierra Leone	Office Supplies	Offline	H	621386563	2016-12-14	948	651.21	524.96	6173
91	Australia and Oceania	Australia	Beverages	Offline	H	240470397	2014-07-11	9389	47.45	31.79	4455
92	Middle East and North Africa	Azerbaijan	Office Supplies	Online	M	423331391	2012-07-24	2021	651.21	524.96	13160
93	Europe	Romania	Cosmetics	Online	H	660643374	2010-12-25	7910	437.20	263.33	34582
94	Central America and the Caribbean	Nicaragua	Beverages	Offline	C	963392674	2011-03-21	8156	47.45	31.79	3870
95	Sub-Saharan Africa	Mali	Clothes	Online	M	512878119	2011-09-03	888	109.28	35.84	970
96	Asia	Malaysia	Fruits	Offline	L	810711038	2011-12-28	6267	9.33	6.92	584
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	728815257	2016-06-29	1485	154.06	90.93	2287
98	North America	Mexico	Personal Care	Offline	M	559427106	2015-08-08	5767	81.73	56.67	4713
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	665095412	2012-02-15	5367	668.27	502.54	35866

In [94]:
```
pd.isnull(df).sum()              # Checking out total null value in the all the colum
```

Out[94]:
```
Region                0
Country               0
Item Type             0
Sales Channel         0
Order Priority        0
Order ID              0
Ship Date             0
Units Sold            0
Unit Price            0
Unit Cost             0
Total Revenue         0
Total Cost            0
Total Profit          0
Order_Month           0
Order_Year            0
Order_Date_MonthYear  0
dtype: int64
```

In [95]:
```
# Display total values of all country
pd.set_option('display.max_rows', None)
df.Country.value_counts()
```

Out[95]:

The Gambia	4
Sierra Leone	3
Sao Tome and Principe	3
Mexico	3
Australia	3
Djibouti	3
Switzerland	2
Myanmar	2
Norway	2
Turkmenistan	2
Cameroon	2
Bulgaria	2
Honduras	2
Azerbaijan	2
Libya	2
Rwanda	2
Mali	2
Gabon	1
Belize	1
Haiti	1
Lithuania	1
San Marino	1
United Kingdom	1
Austria	1
Fiji	1
Madagascar	1
Cote d'Ivoire	1
Tuvalu	1
Democratic Republic of the Congo	1
Zambia	1
Malaysia	1
Nicaragua	1
Romania	1
Slovenia	1
Kuwait	1
Kenya	1
Iran	1
Pakistan	1
Lebanon	1
Spain	1
Samoa	1
Monaco	1
Laos	1
Saudi Arabia	1
Federated States of Micronesia	1
Slovakia	1
Lesotho	1
Albania	1
Russia	1
Solomon Islands	1
Angola	1
Burkina Faso	1
Republic of the Congo	1
Senegal	1
Kyrgyzstan	1
Cape Verde	1
Bangladesh	1
Mongolia	1
Sri Lanka	1
East Timor	1
Portugal	1
New Zealand	1
Moldova	1
France	1
Kiribati	1

```
South Sudan                             1
Costa Rica                              1
Syria                                   1
Brunei                                  1
Niger                                   1
Grenada                                 1
Comoros                                 1
Iceland                                 1
Macedonia                               1
Mauritania                              1
Mozambique                              1
Name: Country, dtype: int64
```

In [96]:
```
df.rename(columns={"Order Priority": "Order_Priority", "Item Type" : "Item_Type", "Sa
                   "Order ID": "Order_ID", "Units Sold" : "Units_Sold", "Unit Price"
                  "Total Revenue" : "Total_Revenue", "Total Cost" : "Total_Cost", "Tot
```

In [97]:
```
df.head(5)
```

Out[97]:

	Region	Country	Item_Type	Sales_Channel	Order_Priority	Order_ID	Ship_Date	Units_Sold	Unit
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	669165933	2010-06-27	9925	
1	Central America and the Caribbean	Grenada	Cereal	Online	C	963881480	2012-09-15	2804	
2	Europe	Russia	Office Supplies	Offline	L	341417157	2014-05-08	1779	
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	514321792	2014-07-05	8102	
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	115456712	2013-02-06	5062	

In [98]:
```
df.Order_Priority.value_counts()
```

Out[98]:
```
H    30
L    27
C    22
M    21
Name: Order_Priority, dtype: int64
```

In [99]:
```
df.Sales_Channel.value_counts()
```

Out[99]:
```
Offline    50
Online     50
Name: Sales_Channel, dtype: int64
```

In [100]:
```
df.Order_Year.value_counts()
```

Out[100]:
```
2012    22
2014    15
2013    12
2011    12
2015    11
2010    10
2016    10
2017     8
Name: Order_Year, dtype: int64
```

In [101]:
```
df.groupby(['Sales_Channel', 'Item_Type', 'Region']).size()
```

Out[101]:

Sales_Channel	Item_Type	Region	
Offline	Baby Food	Australia and Oceania	1
		Europe	1
		Sub-Saharan Africa	1
	Beverages	Australia and Oceania	1
		Central America and the Caribbean	1
		Europe	2
		Sub-Saharan Africa	2
	Cereal	Australia and Oceania	1
		Sub-Saharan Africa	2
	Clothes	Australia and Oceania	1
		Central America and the Caribbean	1
		Europe	1
		Middle East and North Africa	1
		Sub-Saharan Africa	2
	Cosmetics	Asia	1
		Central America and the Caribbean	1
		Europe	2
		Middle East and North Africa	1
		Sub-Saharan Africa	2
	Fruits	Asia	1
		Sub-Saharan Africa	1
	Household	Asia	2
		Central America and the Caribbean	1
		Europe	1
		North America	1
		Sub-Saharan Africa	3
	Office Supplies	Europe	2
		Sub-Saharan Africa	4
	Personal Care	Asia	1
		Central America and the Caribbean	1
		North America	2
		Sub-Saharan Africa	3
	Vegetables	Asia	1
		Sub-Saharan Africa	1
Online	Baby Food	Australia and Oceania	1
		Europe	3
	Beverages	Australia and Oceania	1
		Sub-Saharan Africa	1
	Cereal	Central America and the Caribbean	1
		Middle East and North Africa	1
		Sub-Saharan Africa	2
	Clothes	Asia	2
		Europe	2
		Middle East and North Africa	1
		Sub-Saharan Africa	2
	Cosmetics	Australia and Oceania	1
		Europe	3
		Middle East and North Africa	2
	Fruits	Australia and Oceania	2
		Middle East and North Africa	3
		Sub-Saharan Africa	3
	Household	Europe	1
	Meat	Australia and Oceania	1
		Sub-Saharan Africa	1
	Office Supplies	Asia	2
		Australia and Oceania	1
		Europe	1
		Middle East and North Africa	1
		Sub-Saharan Africa	1
	Personal Care	Europe	2
		Sub-Saharan Africa	1
	Snacks	Central America and the Caribbean	1
		Sub-Saharan Africa	2
	Vegetables	Asia	1

```
                                Europe                          1
                                Sub-Saharan Africa              2
dtype: int64
```

In [102]:
```
df.groupby(['Item_Type', 'Total_Revenue', "Order_Year"]).size()
```

Out[102]:

Item_Type	Total_Revenue	Order_Year	
Baby Food	324971.44	2015	1
	759202.72	2015	1
	1212580.00	2013	1
	1419101.52	2014	1
	1901836.00	2014	1
	2198981.92	2012	1
	2533654.00	2010	1
Beverages	221117.00	2016	1
	243133.80	2014	1
	257653.50	2015	1
	272410.45	2011	1
	387002.20	2011	1
	418936.05	2011	1
	445033.55	2014	1
	445508.05	2014	1
Cereal	140287.40	2013	1
	197883.40	2016	1
	435466.90	2012	1
	576782.80	2012	1
	835759.10	2013	1
	1356180.10	2014	1
	1780539.20	2017	1
Clothes	97040.64	2011	1
	182825.44	2012	1
	247956.32	2010	1
	380512.96	2012	1
	455479.04	2014	1
	600821.44	2016	1
	648030.40	2015	1
	668356.48	2010	1
	802333.76	2015	1
	856973.76	2014	1
	861563.52	2012	1
	902980.64	2017	1
	1082418.40	2010	1
Cosmetics	745426.00	2013	1
	793518.00	2017	1
	1244708.40	2015	1
	1957344.40	2013	1
	2836990.80	2016	1
	3039414.40	2016	1
	3154398.00	2014	1
	3162704.80	2010	1
	3458252.00	2010	1
	3786589.20	2012	1
	3876652.40	2016	1
	4220728.80	2013	1
	4324782.40	2013	1
Fruits	4870.26	2012	1
	6279.09	2015	1
	20404.71	2014	1
	35304.72	2011	1
	50363.34	2014	1
	54319.26	2010	1
	58471.11	2011	1
	71253.21	2013	1
	75591.66	2014	1
	89623.98	2013	1
Household	188452.14	2012	1
	1583799.90	2012	1
	2559474.10	2010	1
	2798046.49	2011	1
	3015902.51	2012	1
	3586605.09	2012	1

```
                    4647149.58     2014          1
                    5513227.50     2015          1
                    5997054.98     2017          1
Meat                2011149.63     2017          1
                    2492526.12     2012          1
Office Supplies     617347.08      2016          1
                    824431.86      2012          1
                    1158502.59     2014          1
                    1316095.41     2012          1
                    1904138.04     2015          1
                    2251232.97     2011          1
                    2596374.27     2012          1
                    3262562.10     2013          1
                    3296425.02     2013          1
                    3593376.78     2011          1
                    4368316.68     2012          1
                    5396577.27     2010          1
Personal Care       22312.29       2010          1
                    173676.25      2013          1
                    246415.95      2017          1
                    400558.73      2014          1
                    414371.10      2016          1
                    471336.91      2015          1
                    496101.10      2015          1
                    523807.57      2017          1
                    524870.06      2012          1
                    707454.88      2012          1
Snacks              339490.50      2016          1
                    623289.30      2011          1
                    1117953.66     2017          1
Vegetables          19103.44       2011          1
                    26344.26       2012          1
                    228779.10      2016          1
                    574951.92      2011          1
                    994765.42      2012          1
                    1245112.92     2012          1
dtype: int64
```

In [103]:
```
df['Item_Type'].value_counts(normalize=True)*100
```

Out[103]:
```
Clothes            13.0
Cosmetics          13.0
Office Supplies    12.0
Fruits             10.0
Personal Care      10.0
Household           9.0
Beverages           8.0
Baby Food           7.0
Cereal              7.0
Vegetables          6.0
Snacks              3.0
Meat                2.0
Name: Item_Type, dtype: float64
```

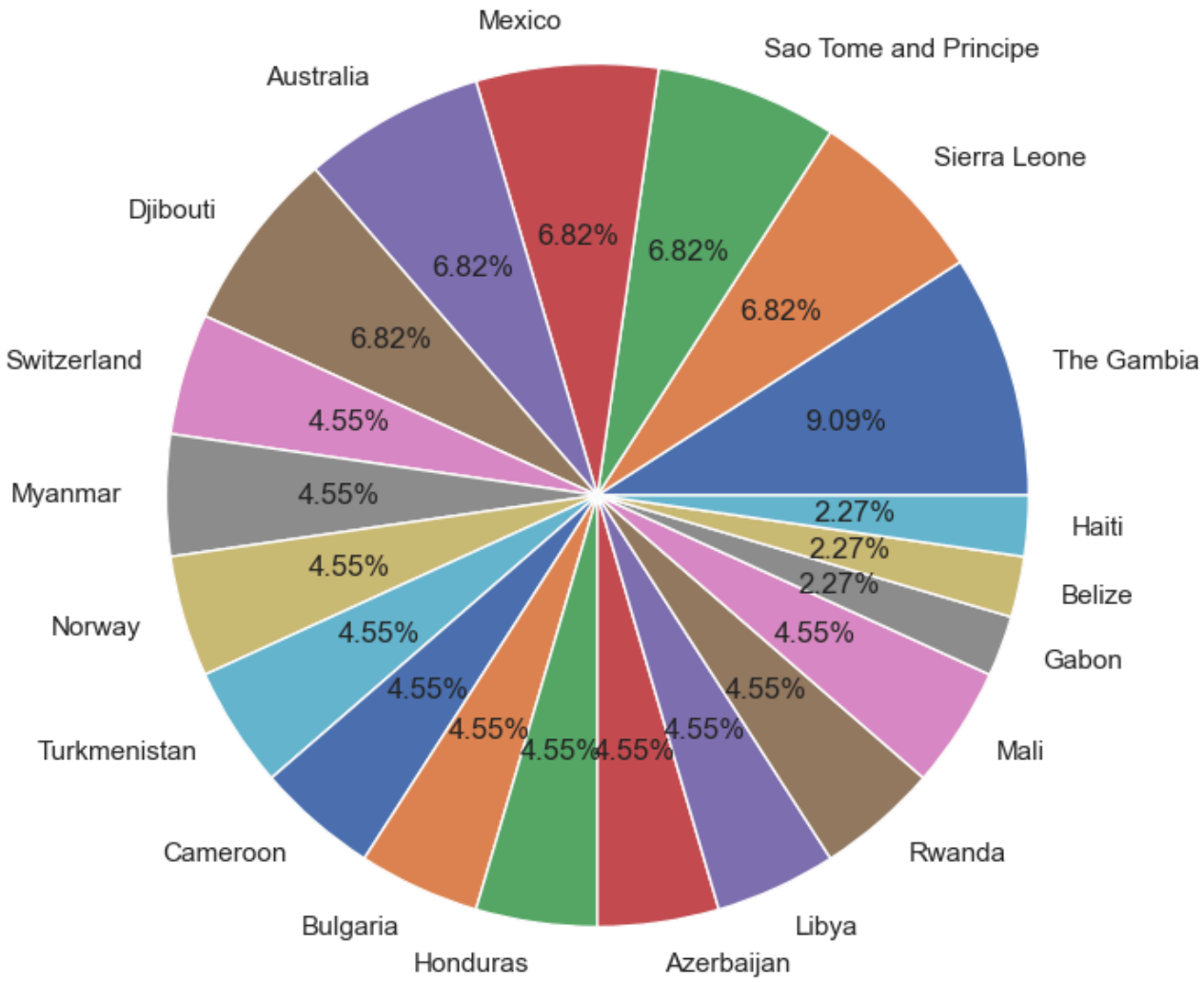
In [106]:
```
df.sum(axis = 0, skipna = True)
```

Out[106]:
```
Region                 Australia and OceaniaCentral America and the C...
Country                TuvaluGrenadaRussiaSao Tome and PrincipeRwanda...
Item_Type              Baby FoodCerealOffice SuppliesFruitsOffice Sup...
Sales_Channel          OfflineOnlineOfflineOnlineOfflineOnlineOffline...
Order_Priority         HCLCLCMHMHHHLHCMMCLLLHLHLHMLCLMCCHMLLMLMHMHHHH...
Order_ID                                                55502041236
Units_Sold                                                  512871
Unit_Price                                                27676.13
Unit_Cost                                                  19104.8
Total_Revenue                                          137348768.31
Total_Cost                                              93180569.91
Total_Profit                                             44168198.4
Order_Month                                                    626
Order_Year                                                  201323
Order_Date_MonthYear   2010-052012-082014-052014-062013-022015-022011...
dtype: object
```

In [107]:
```
df.sort_values(by = ['Order_Year', 'Order_Month'])
```

_Type	Sales_Channel	Order_Priority	Order_ID	Ship_Date	Units_Sold	Unit_Price	Unit_Cost	Total_Revenue	Total_C
metics	Online	M	382392299	2010-02-25	7234	437.20	263.33	3162704.80	1904929
Clothes	Online	C	385383069	2010-03-18	2269	109.28	35.84	247956.32	81320
y Food	Offline	H	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243
Fruits	Online	L	686048400	2010-05-10	5822	9.33	6.92	54319.26	40288
Clothes	Offline	C	647876489	2010-08-01	9905	109.28	35.84	1082418.40	354995

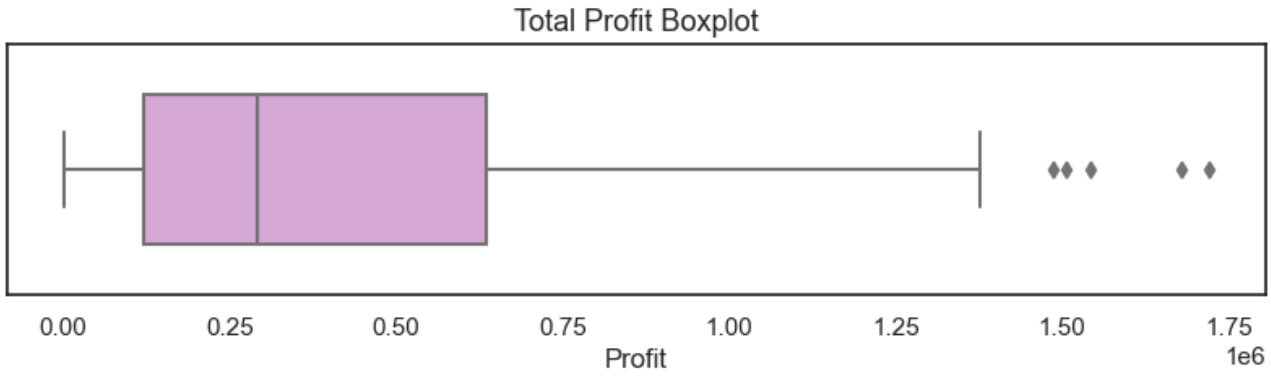
In [108]:
```
country_names = df.Country.value_counts().index
country_val = df.Country.value_counts().values
# Pie Chart for top 20 country
fig,ax = plt.subplots(figsize=(10,8))
ax.pie(country_val[:20],labels=country_names[:20],autopct='%1.2f%%')
plt.show()
```



Outliers detection

Box Plot of Total Profit

In [109]:
```
sns.set(style='white')
fig, ax = plt.subplots(figsize=(10, 2))
sns.boxplot(df['Total_Profit'], color="plum", width=.6)

plt.title('Total Profit Boxplot', fontsize=13)
plt.xlabel('Profit')
plt.show()
```



The function 'detect_outliers' takes in two arguments: a pandas DataFrame 'dataframe' and a column name 'column'. It aims to detect outliers in the specified column using the Z-score method.

The Z-score method assumes that the data follows a normal distribution and detects outliers as values that fall outside a certain number of standard deviations from the mean. In this case, the threshold is set to 2 standard deviations.

The function first calculates the mean and standard deviation of the column. It then loops through each value in the column and calculates its Z-score. If the absolute Z-score is greater than the threshold, the function appends the index of the outlier to a list and prints the corresponding row of the DataFrame. Finally, the function returns the list of outlier indices.

In [110]:
```
def detect_outliers(dataframe, column):
    threshold = 2     ## 2rd standard deviation
    mean = np.mean(column)
    std = np.std(column)
    outliers = []

    for i, value in enumerate(column):
        z_score = (value - mean) / std
        if np.abs(z_score) > threshold:
            outliers.append(i)
            print(dataframe.loc[i])

    return outliers
```

In [111]:
```
outliers = detect_outliers(df, df["Total_Profit"])
```

```
Region                 Central America and the Caribbean
Country                                         Honduras
Item_Type                                        Household
Sales_Channel                                     Offline
Order_Priority                                          H
Order_ID                                        522840487
Ship_Date                             2017-02-13 00:00:00
Units_Sold                                           8974
Unit_Price                                         668.27
Unit_Cost                                          502.54
Total_Revenue                                 5997054.98
Total_Cost                                    4509793.96
Total_Profit                                  1487261.02
Order_Month                                            2
Order_Year                                          2017
Order_Date_MonthYear                                2017-02
Name: 13, dtype: object
Region                                 Europe
Country                            Switzerland
```

In [112]:
```
#Print rows where outlier is present for the Total Profit column value
print(outliers)
```

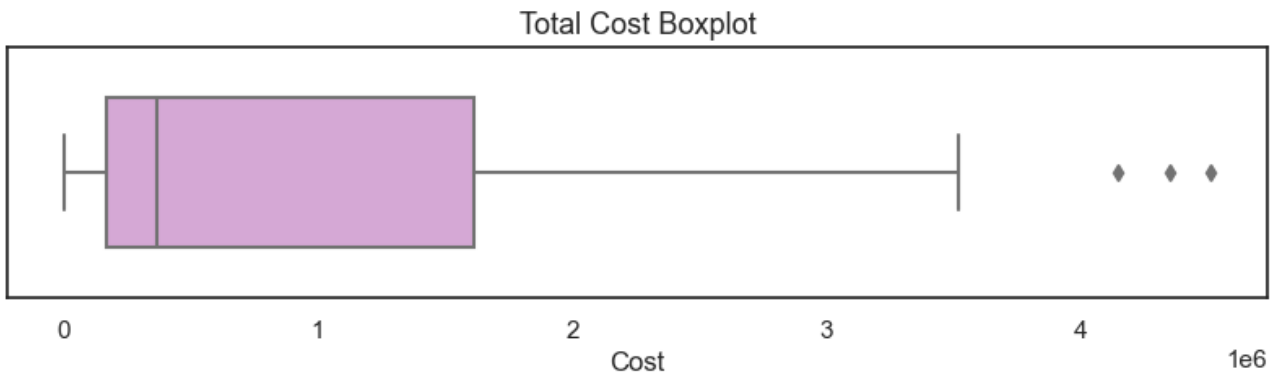
```
[13, 30, 33, 46, 74, 79, 93]
```

In [113]:
```
list_length = len(outliers)

# Print the number of values in the list
print("The list has", list_length, "outliers in Total Profit column of dataframe data
```

```
The list has 7 outliers in Total Profit column of dataframe data
```

Box Plot of Total Cost

In [114]:
```
sns.set(style='white')
fig, ax = plt.subplots(figsize=(10, 2))
sns.boxplot(df['Total_Cost'], color="plum", width=.6)

plt.title('Total Cost Boxplot', fontsize=13)
plt.xlabel('Cost')
plt.show()
```



In [115]:
```
def detect_outliers(dataframe, column):
    threshold = 2     ## 3rd standard deviation
    mean = np.mean(column)
    std = np.std(column)
    outliers = []

    for i, value in enumerate(column):
        z_score = (value - mean) / std
        if np.abs(z_score) > threshold:
            outliers.append(i)
            print(dataframe.loc[i])

    return outliers
```

In [116]:
```
outliers = detect_outliers(df, df["Total_Cost"])
```

```
Region                  Central America and the Caribbean
Country                                          Honduras
Item_Type                                         Household
Sales_Channel                                      Offline
Order_Priority                                           H
Order_ID                                         522840487
Ship_Date                          2017-02-13 00:00:00
Units_Sold                                          8974
Unit_Price                                        668.27
Unit_Cost                                         502.54
Total_Revenue                                  5997054.98
Total_Cost                                     4509793.96
Total_Profit                                   1487261.02
Order_Month                                             2
Order_Year                                           2017
Order_Date_MonthYear                                2017-02
Name: 13, dtype: object
Region                                 Asia
Country                             Myanmar
Item_Type                         Household
Sales_Channel                       Offline
Order_Priority                            H
Order_ID                          177713572
Ship_Date                2015-03-01 00:00:00
Units_Sold                             8250
Unit_Price                           668.27
Unit_Cost                            502.54
Total_Revenue                     5513227.5
Total_Cost                        4145955.0
Total_Profit                      1367272.5
Order_Month                               1
Order_Year                             2015
Order_Date_MonthYear                 2015-01
Name: 33, dtype: object
Region                                 Asia
Country                              Brunei
Item_Type                   Office Supplies
Sales_Channel                        Online
Order_Priority                            L
Order_ID                          320009267
Ship_Date                2012-05-08 00:00:00
Units_Sold                             6708
Unit_Price                           651.21
Unit_Cost                            524.96
Total_Revenue                    4368316.68
Total_Cost                       3521431.68
Total_Profit                      846885.0
Order_Month                               4
Order_Year                             2012
Order_Date_MonthYear                 2012-04
Name: 38, dtype: object
Region                               Europe
Country                           Lithuania
Item_Type                   Office Supplies
Sales_Channel                       Offline
Order_Priority                            H
Order_ID                          166460740
Ship_Date                2010-11-17 00:00:00
Units_Sold                             8287
Unit_Price                           651.21
Unit_Cost                            524.96
Total_Revenue                    5396577.27
Total_Cost                       4350343.52
Total_Profit                     1046233.75
Order_Month                              10
```

```
Order_Year                          2010
Order_Date_MonthYear               2010-10
Name: 68, dtype: object
Region                       North America
Country                             Mexico
Item_Type                        Household
Sales_Channel                      Offline
Order_Priority                           C
Order_ID                         986435210
Ship_Date              2014-12-12 00:00:00
Units_Sold                            6954
Unit_Price                          668.27
Unit_Cost                           502.54
Total_Revenue                  4647149.58
Total_Cost                     3494663.16
Total_Profit                   1152486.42
Order_Month                            11
Order_Year                           2014
Order_Date_MonthYear               2014-11
Name: 75, dtype: object
```

In [117]:
```
# Print rows where outlier is present for the Total Cost column value
print(outliers)
```

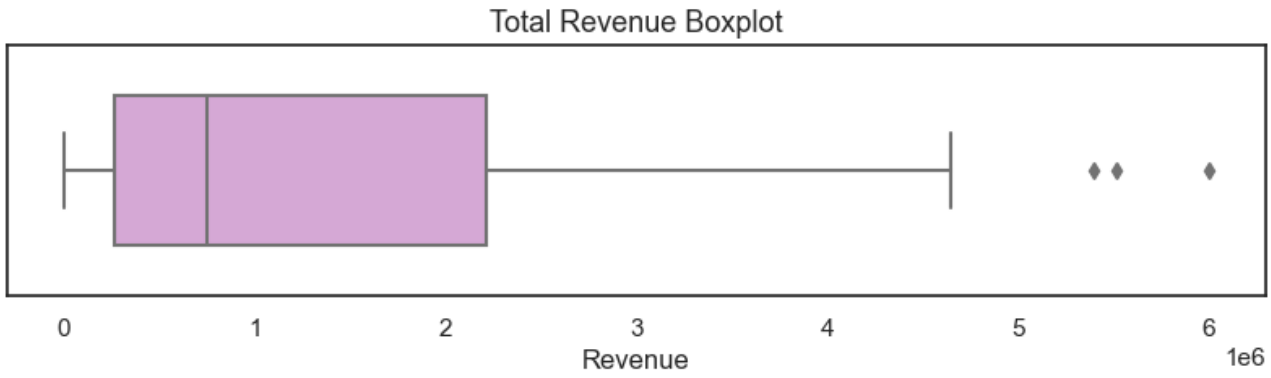
```
[13, 33, 38, 68, 75]
```

In [118]:
```
list_length = len(outliers)

# Print the number of values in the list
print("The list has", list_length, "outliers in Total_Cost column of dataframe data "
```

```
The list has 5 outliers in Total_Cost column of dataframe data
```

Box Plot of Total Revenue

In [119]:
```
sns.set(style='white')
fig, ax = plt.subplots(figsize=(10, 2))
sns.boxplot(df['Total_Revenue'], color="plum", width=.6)

plt.title('Total Revenue Boxplot', fontsize=13)
plt.xlabel('Revenue')
plt.show()
```



In [120]:
```
def detect_outliers(dataframe, column):
    threshold = 2     ## 3rd standard deviation
    mean = np.mean(column)
    std = np.std(column)
    outliers = []

    for i, value in enumerate(column):
        z_score = (value - mean) / std
        if np.abs(z_score) > threshold:
            outliers.append(i)
            print(dataframe.loc[i])

    return outliers
```

In [121]:
```
outliers = detect_outliers(df, df["Total_Revenue"])
```

```
Region                  Central America and the Caribbean
Country                                          Honduras
Item_Type                                         Household
Sales_Channel                                      Offline
Order_Priority                                           H
Order_ID                                         522840487
Ship_Date                          2017-02-13 00:00:00
Units_Sold                                          8974
Unit_Price                                        668.27
Unit_Cost                                         502.54
Total_Revenue                                  5997054.98
Total_Cost                                     4509793.96
Total_Profit                                   1487261.02
Order_Month                                             2
Order_Year                                           2017
Order_Date_MonthYear                                2017-02
Name: 13, dtype: object
Region                                 Asia
Country                             Myanmar
Item_Type                         Household
Sales_Channel                       Offline
Order_Priority                            H
Order_ID                          177713572
Ship_Date                2015-03-01 00:00:00
Units_Sold                             8250
Unit_Price                           668.27
Unit_Cost                            502.54
Total_Revenue                     5513227.5
Total_Cost                        4145955.0
Total_Profit                      1367272.5
Order_Month                               1
Order_Year                             2015
Order_Date_MonthYear                 2015-01
Name: 33, dtype: object
Region                                 Asia
Country                              Brunei
Item_Type                   Office Supplies
Sales_Channel                        Online
Order_Priority                            L
Order_ID                          320009267
Ship_Date                2012-05-08 00:00:00
Units_Sold                             6708
Unit_Price                           651.21
Unit_Cost                            524.96
Total_Revenue                    4368316.68
Total_Cost                       3521431.68
Total_Profit                      846885.0
Order_Month                               4
Order_Year                             2012
Order_Date_MonthYear                 2012-04
Name: 38, dtype: object
Region                               Europe
Country                           Lithuania
Item_Type                   Office Supplies
Sales_Channel                       Offline
Order_Priority                            H
Order_ID                          166460740
Ship_Date                2010-11-17 00:00:00
Units_Sold                             8287
Unit_Price                           651.21
Unit_Cost                            524.96
Total_Revenue                    5396577.27
Total_Cost                       4350343.52
Total_Profit                     1046233.75
Order_Month                              10
```

```
Order_Year                              2010
Order_Date_MonthYear                 2010-10
Name: 68, dtype: object
Region                  Middle East and North Africa
Country                                     Pakistan
Item_Type                                   Cosmetics
Sales_Channel                                Offline
Order_Priority                                     L
Order_ID                                   231145322
Ship_Date                        2013-08-16 00:00:00
Units_Sold                                      9892
Unit_Price                                     437.2
Unit_Cost                                     263.33
Total_Revenue                               4324782.4
Total_Cost                                 2604860.36
Total_Profit                               1719922.04
Order_Month                                        7
Order_Year                                      2013
Order_Date_MonthYear                         2013-07
Name: 74, dtype: object
Region                           North America
Country                                 Mexico
Item_Type                             Household
Sales_Channel                          Offline
Order_Priority                               C
Order_ID                             986435210
Ship_Date                  2014-12-12 00:00:00
Units_Sold                                6954
Unit_Price                              668.27
Unit_Cost                               502.54
Total_Revenue                         4647149.58
Total_Cost                            3494663.16
Total_Profit                          1152486.42
Order_Month                                 11
Order_Year                                2014
Order_Date_MonthYear                    2014-11
Name: 75, dtype: object
```

In [122]:
```
# Print rows where outlier is present for the Total Revenue column value
print(outliers)
```

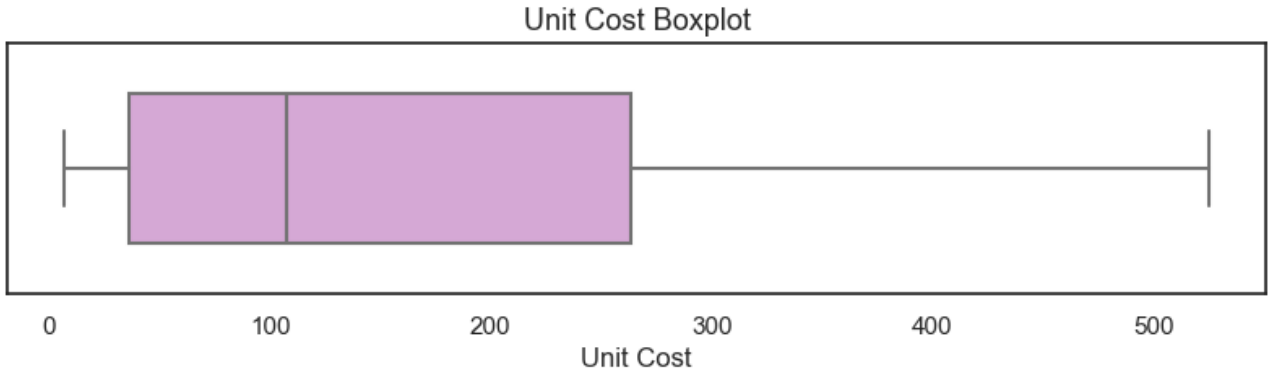
```
[13, 33, 38, 68, 74, 75]
```

In [123]:
```
list_length = len(outliers)

# Print the number of values in the list
print("The list has", list_length, "outliers in Total_Revenue column of dataframe dat
```

```
The list has 6 outliers in Total_Revenue column of dataframe data
```

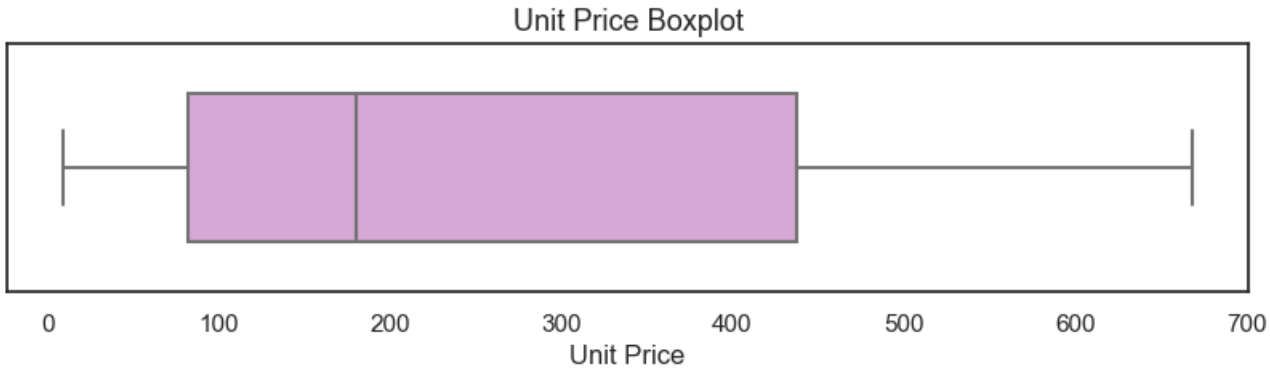
Box Plot of Unit Cost

In [124]:
```
sns.set(style='white')
fig, ax = plt.subplots(figsize=(10, 2))
sns.boxplot(df['Unit_Cost'], color="plum", width=.6)
plt.title('Unit Cost Boxplot', fontsize=13)
plt.xlabel('Unit Cost')
plt.show()
```



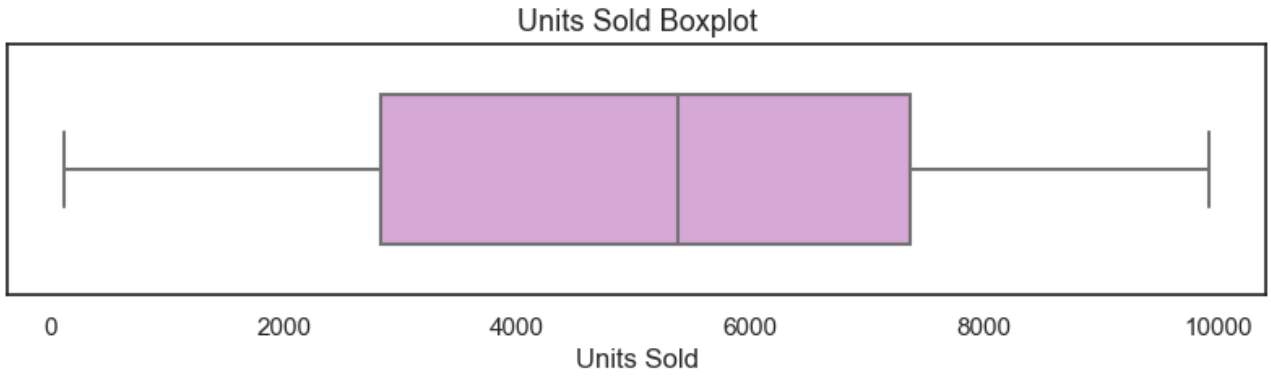
Box Plot of Unit Price

In [125]:
```
# sns.set(style='white')
fig, ax = plt.subplots(figsize=(10, 2))
sns.boxplot(df['Unit_Price'], color="plum", width=.6)
plt.title('Unit Price Boxplot', fontsize=13)
plt.xlabel('Unit Price')
plt.show()
```

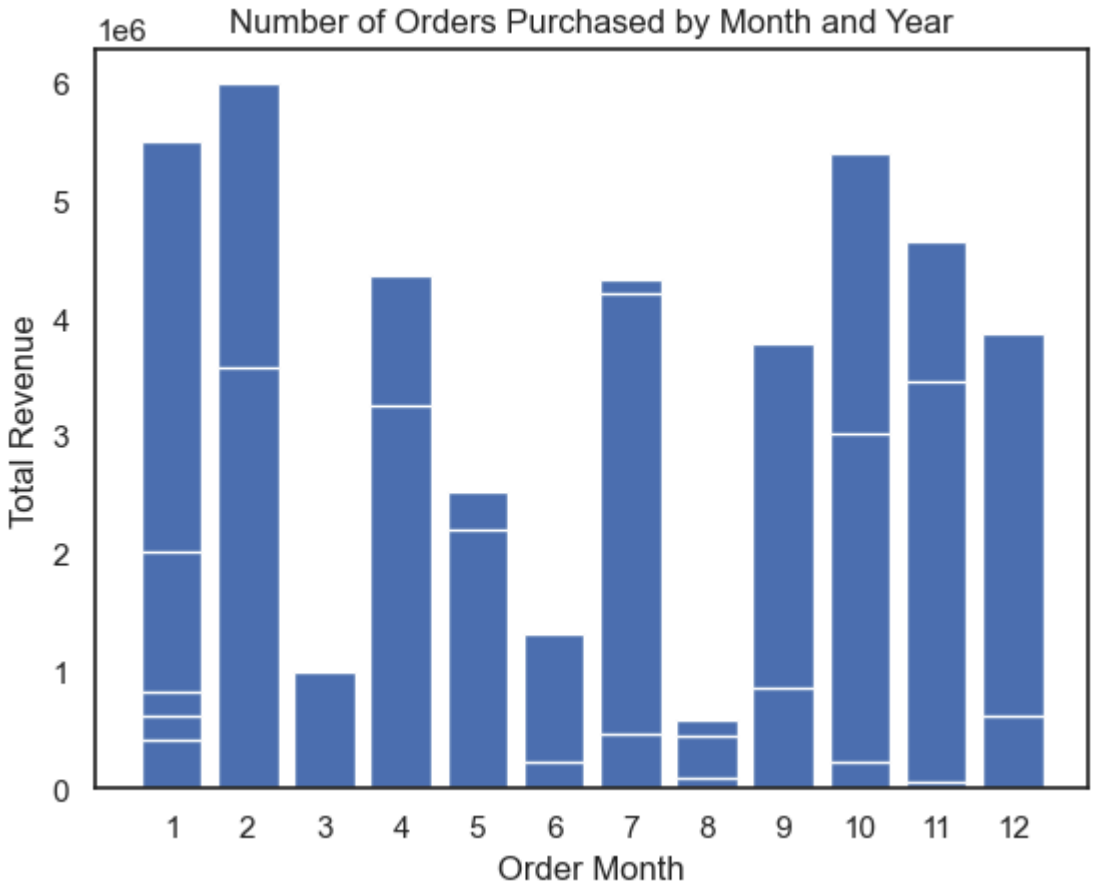


Box Plot of Unit Sold

In [126]:
```
sns.set(style='white')
fig, ax = plt.subplots(figsize=(10, 2))
sns.boxplot(df['Units_Sold'], color="plum", width=.6)

plt.title('Units Sold Boxplot', fontsize=13)
plt.xlabel('Units Sold')
plt.show()
```

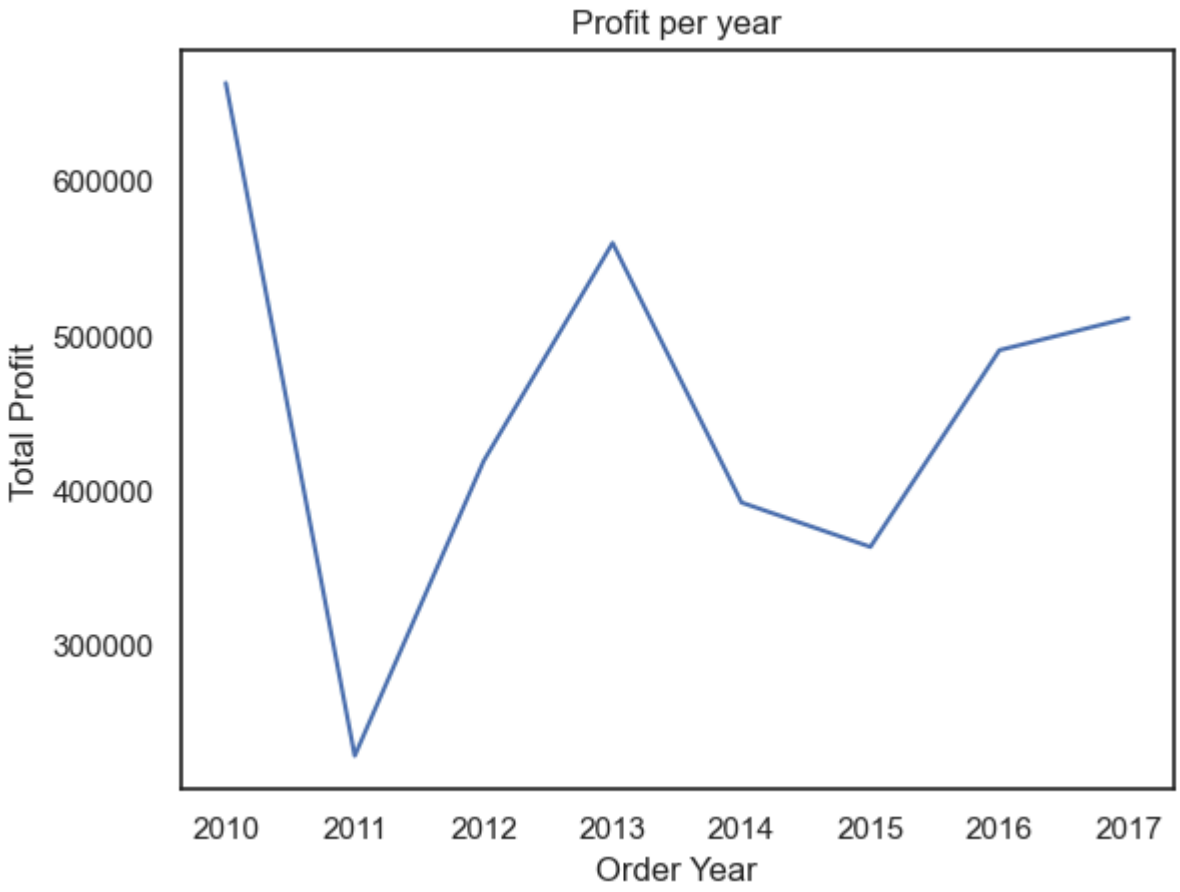


In [127]:
```
# Creating a bar chart for Total Revenue and Order Month
plt.bar(df['Order_Month'], df['Total_Revenue'])

# Set the chart title and axis labels
plt.title('Number of Orders Purchased by Month and Year')
plt.xticks([1,2,3,4,5,6,7,8,9,10,11,12])
plt.xlabel('Order Month')
plt.ylabel('Total Revenue')

# Rotate the x-axis labels for better readability

# Display the chart
plt.show()
```

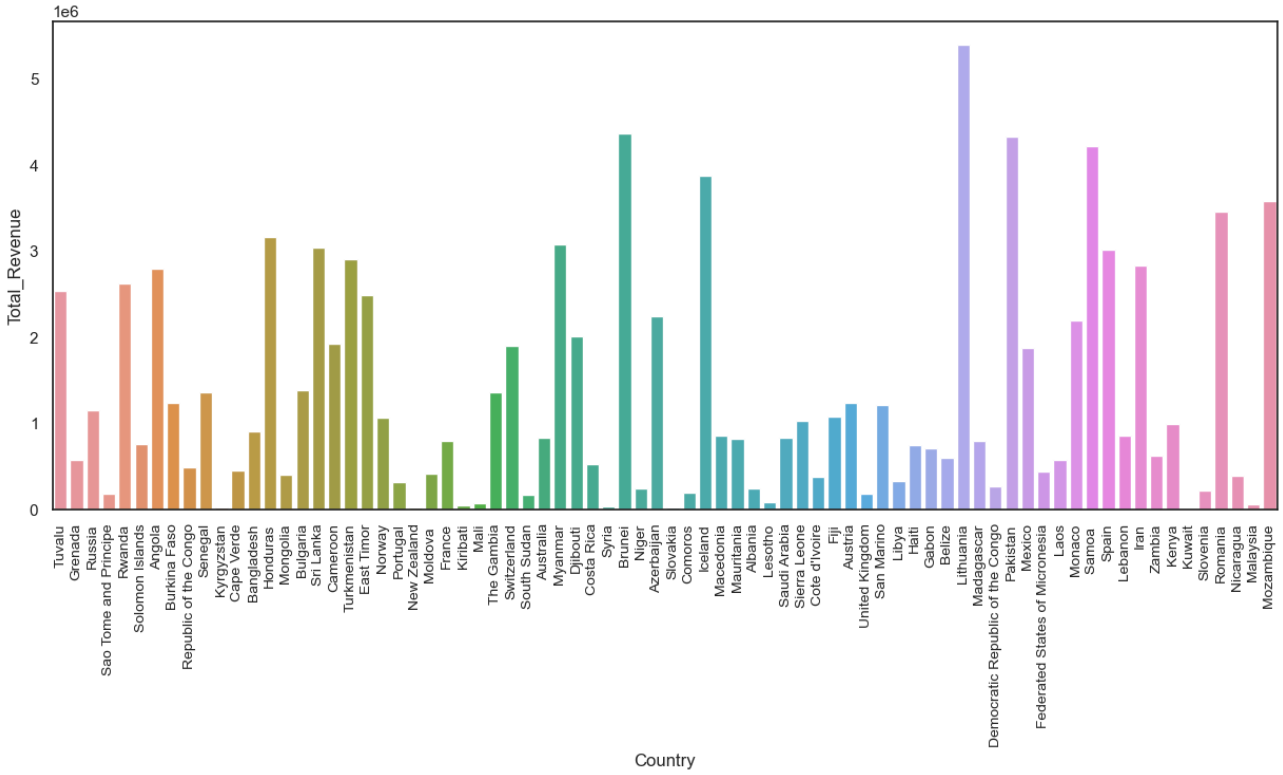


In [128]:
```
# Plot line graph of Total Profit and Order Year
df.groupby('Order_Year')['Total_Profit'].mean().plot()
plt.xlabel('Order Year')
plt.ylabel('Total Profit')
plt.title('Profit per year')
```

Out[128]:
```
Text(0.5, 1.0, 'Profit per year')
```

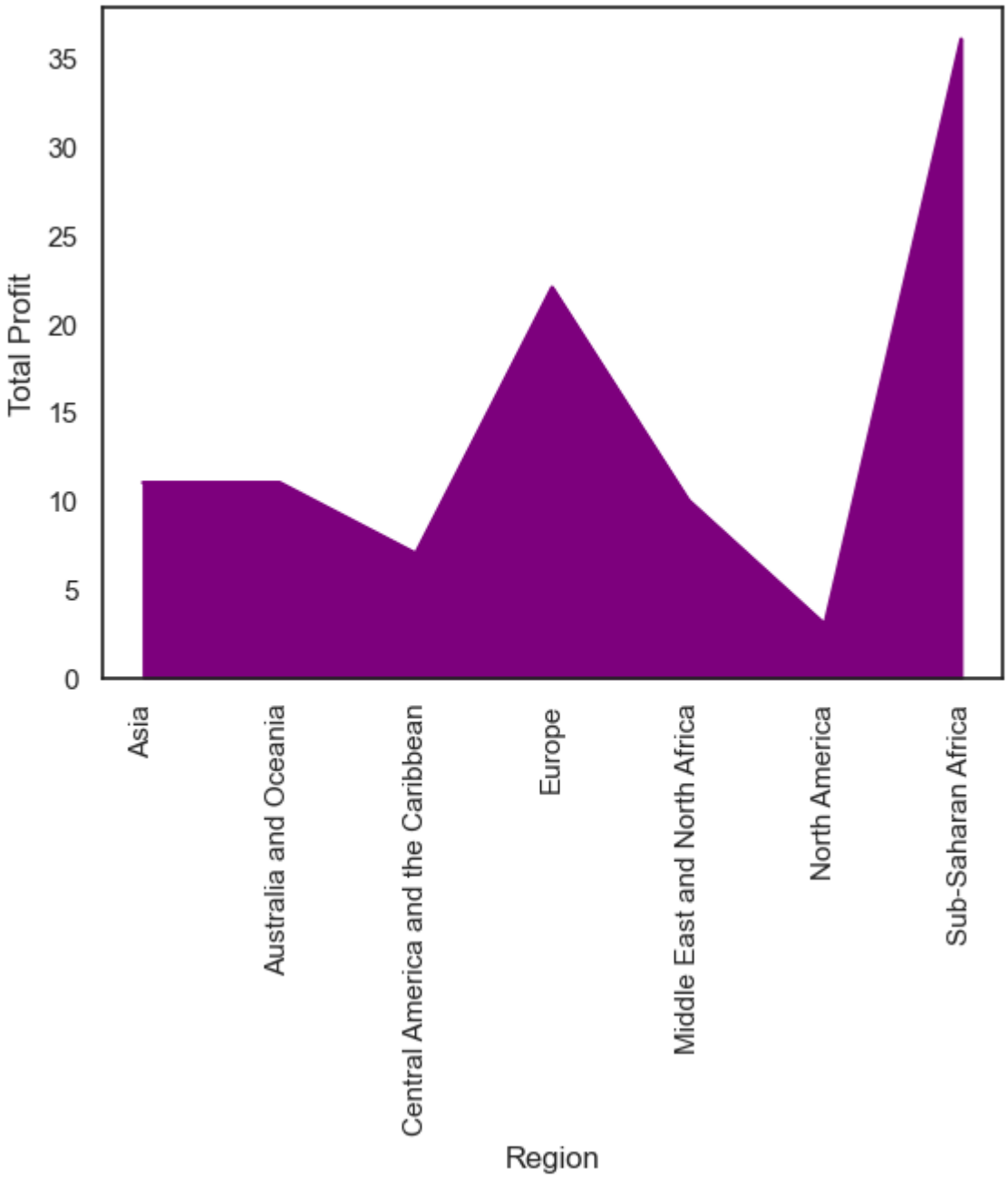


In [129]:
```
plt.figure(figsize=(15,6))
sns.barplot(x='Country', y='Total_Revenue', data=df, ci=None)
plt.xticks(rotation=90)
plt.tick_params(axis='x', which='major', labelsize=10)

# Lithuania is the country where maximum revenue has been generated followed by Brune
```



In [130]:
```
df.groupby('Region')['Total_Profit'].count().plot(kind='area',color=['purple','brown'
plt.xticks(rotation=90)
plt.ylabel('Total Profit')

# Maximum profit has been generated in the Sub-Saharan African region while minimum p
```

Out[130]: Text(0, 0.5, 'Total Profit')

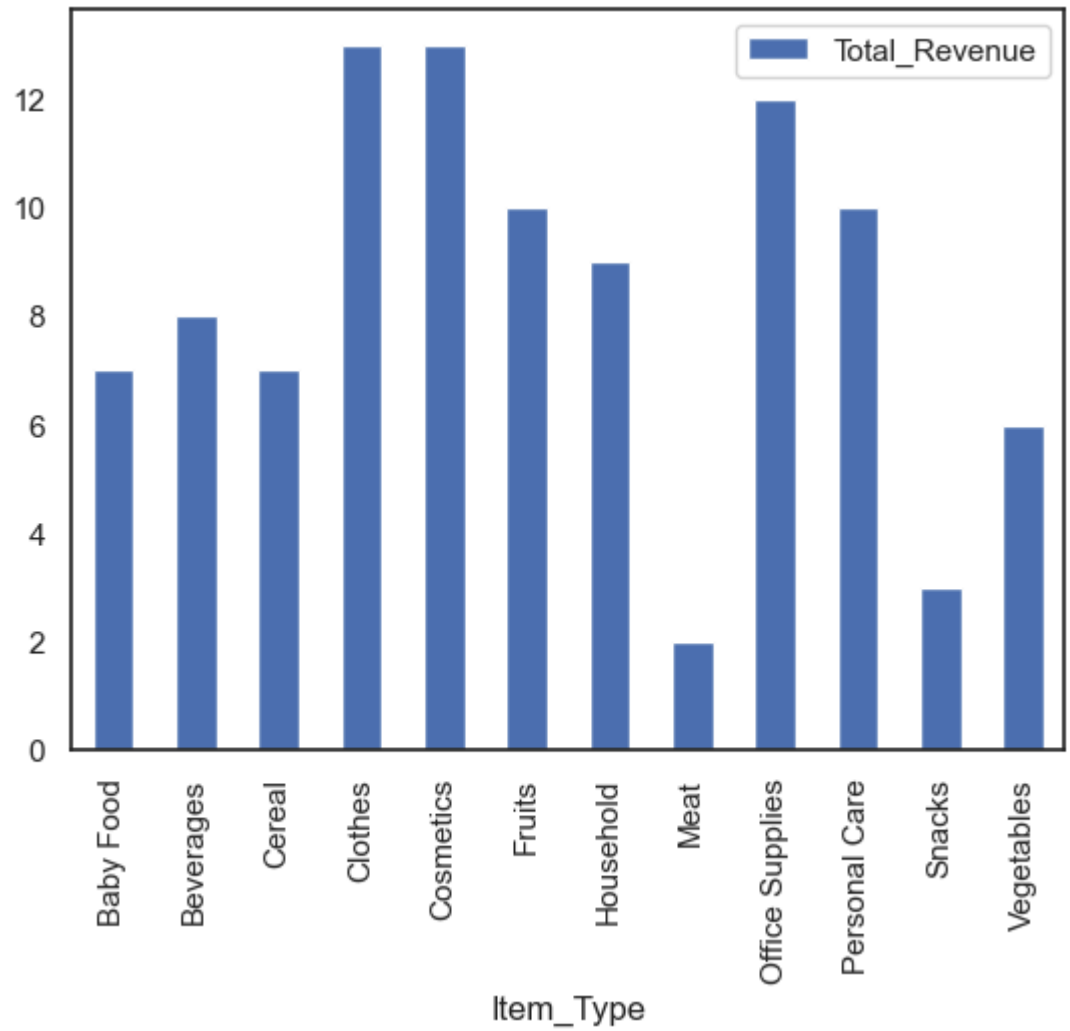


In [131]:
```
# Calculating the total revenue for each group with respect to Item Type and then sor
revenue_by_category = df.groupby('Item_Type')['Total_Revenue'].sum().sort_values(asce
revenue_by_category
```

Out[131]:
```
Item_Type
Cosmetics          36601509.60
Office Supplies    30585380.07
Household          29889712.29
Baby Food          10350327.60
Clothes             7787292.80
Cereal              5322898.90
Meat                4503675.75
Personal Care       3980904.84
Vegetables          3089057.06
Beverages           2690794.60
Snacks              2080733.46
Fruits               466481.34
Name: Total_Revenue, dtype: float64
```

In [132]:
```
pd.pivot_table(df,values='Total_Revenue',index='Item_Type',aggfunc='count').plot(kinc

# Maximum revenue has been generated from the items 'Clothes' and 'Cosmetics' closely
```

Out[132]:
```
<AxesSubplot:xlabel='Item_Type'>
```

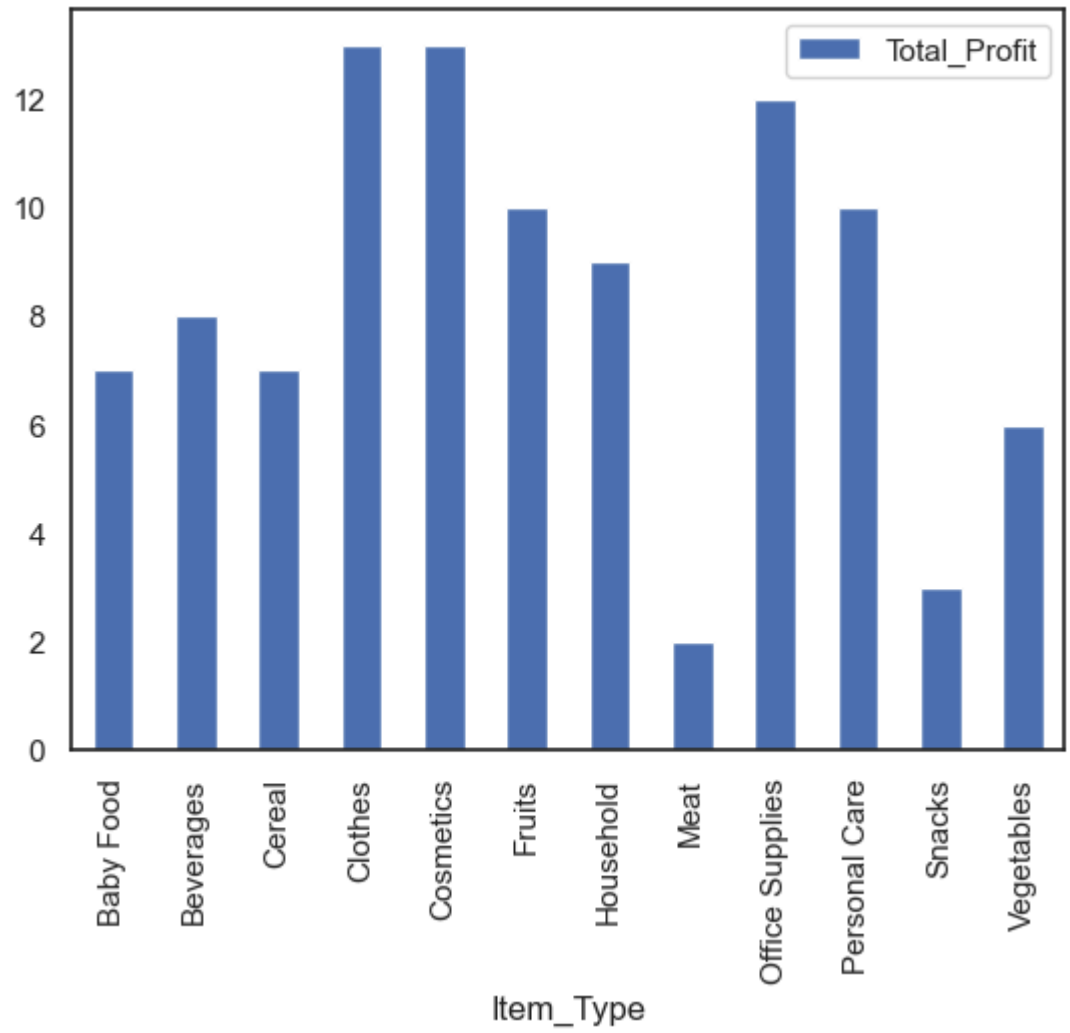


In [133]:
```
# Calculating the total profit for each group with respect to Item Type and then sort
profit_by_category = df.groupby('Item_Type')['Total_Profit'].sum().sort_values(ascend
profit_by_category
```

Out[133]:
```
Item_Type
Cosmetics          14556048.66
Household           7412605.71
Office Supplies     5929583.75
Clothes             5233334.40
Baby Food           3886643.70
Cereal              2292443.43
Vegetables          1265819.63
Personal Care       1220622.48
Beverages            888047.28
Snacks               751944.18
Meat                 610610.00
Fruits               120495.18
Name: Total_Profit, dtype: float64
```

In [134]:
```
pd.pivot_table(df,values='Total_Profit',index='Item_Type',aggfunc='count').plot(kind=

# Maximum profit has been generated from the items 'Clothes' and 'Cosmetics' closely
```

Out[134]:
```
<AxesSubplot:xlabel='Item_Type'>
```

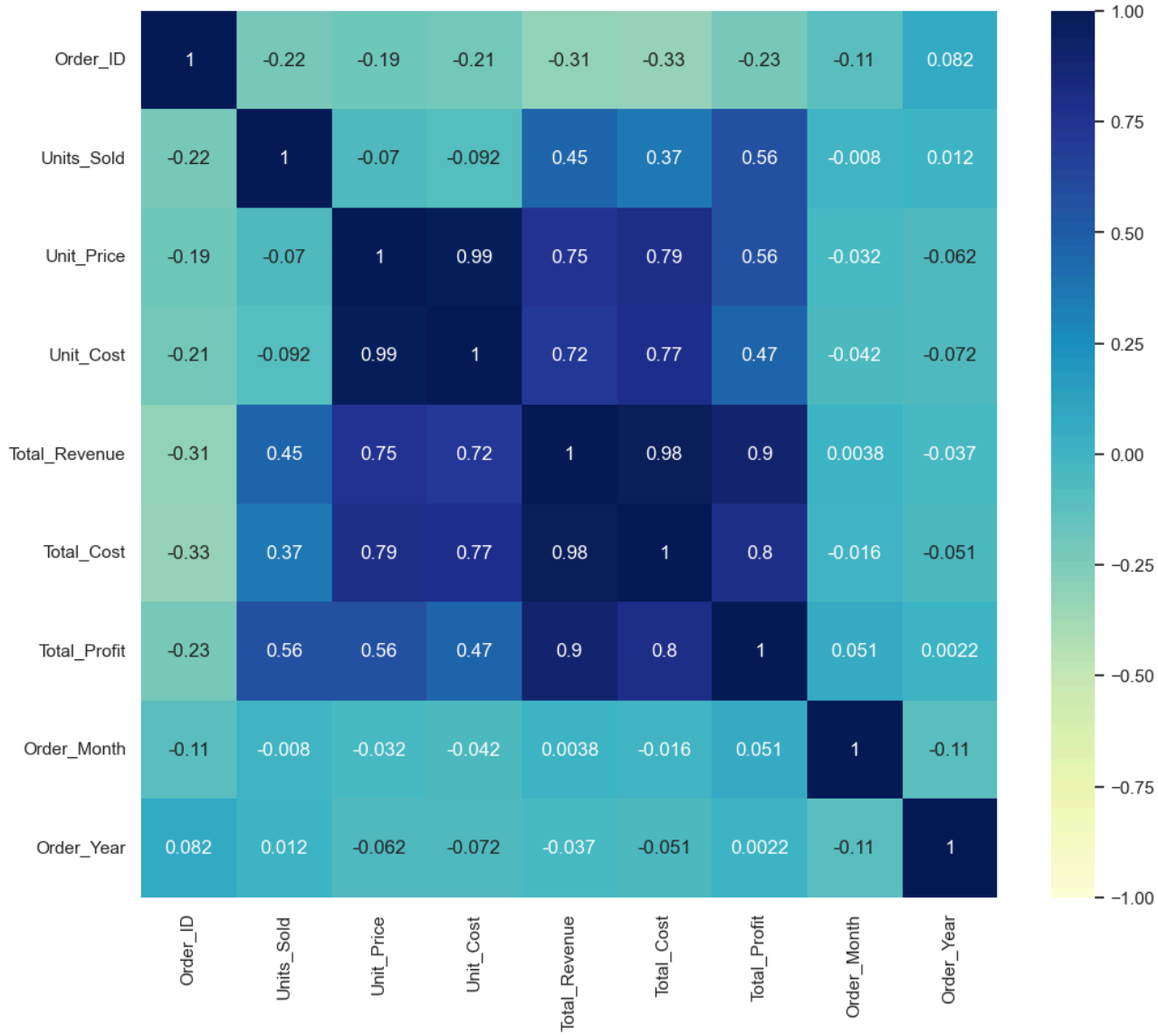


In [135]:
```
# Calculating correlation of 'Total Revenue', 'Total Cost' and 'Total Profit' columns
print(df[['Total_Revenue', 'Total_Cost', 'Total_Profit']].corr())
```

```
               Total_Revenue  Total_Cost  Total_Profit
Total_Revenue       1.000000    0.983928      0.897327
Total_Cost          0.983928    1.000000      0.804091
Total_Profit        0.897327    0.804091      1.000000
```

In [136]:
```
# Checking the correlation
plt.figure(figsize=(12,10))
sns.heatmap(df.corr(method='pearson'), annot=True, vmin=-1, vmax=1, cmap='YlGnBu')



# From the above heatmap, we can infer that Total Cost is strongly related to Unit Pr
# Units Sold and {Unit Price and Unit Cost} are completely independent. Number of uni
# Unit Cost, Unit Price and Total Cost are almost completely independent of Total Rev
```

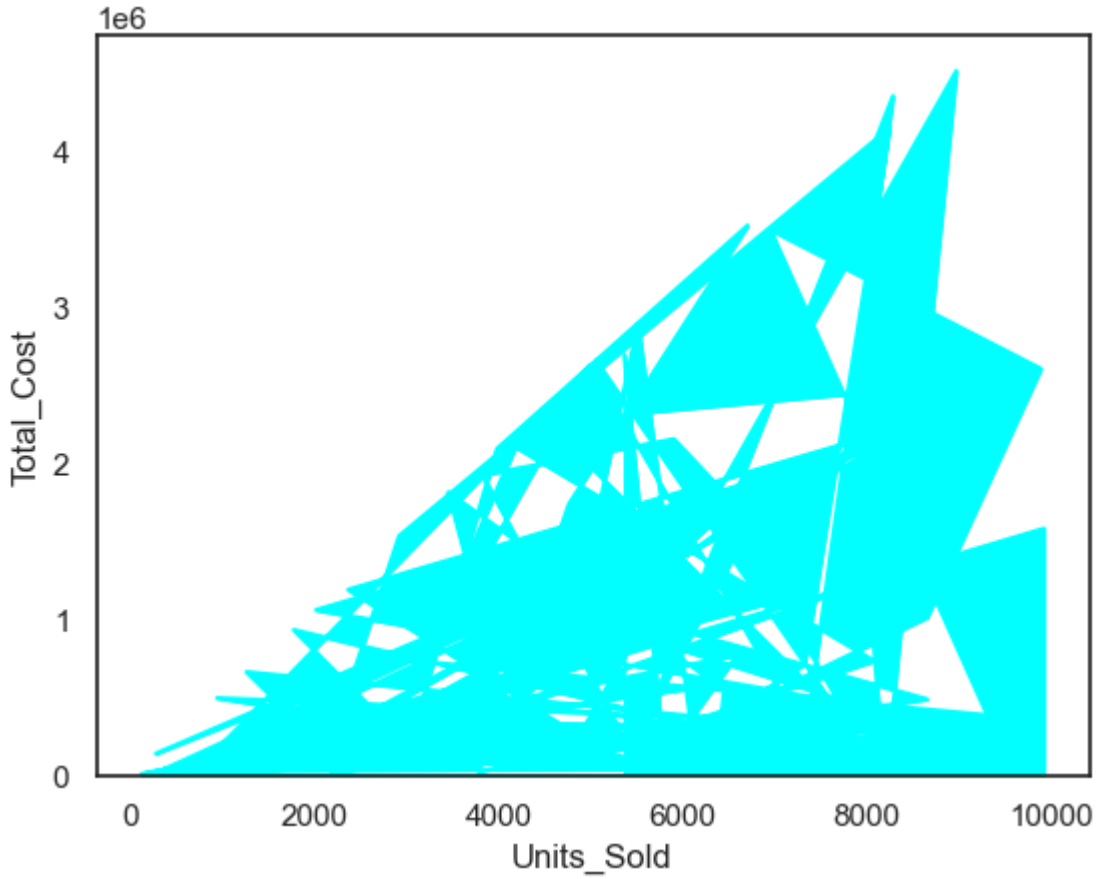
Out[136]: <AxesSubplot:>



Unit_Sold and its Total_Cost

In [137]:
```
df.plot.area(x='Units_Sold',y='Total_Cost',color='aqua',legend=None)
plt.ylabel('Total_Cost')


# Maximum cost has been generated when 8000-9000 units were sold.
```

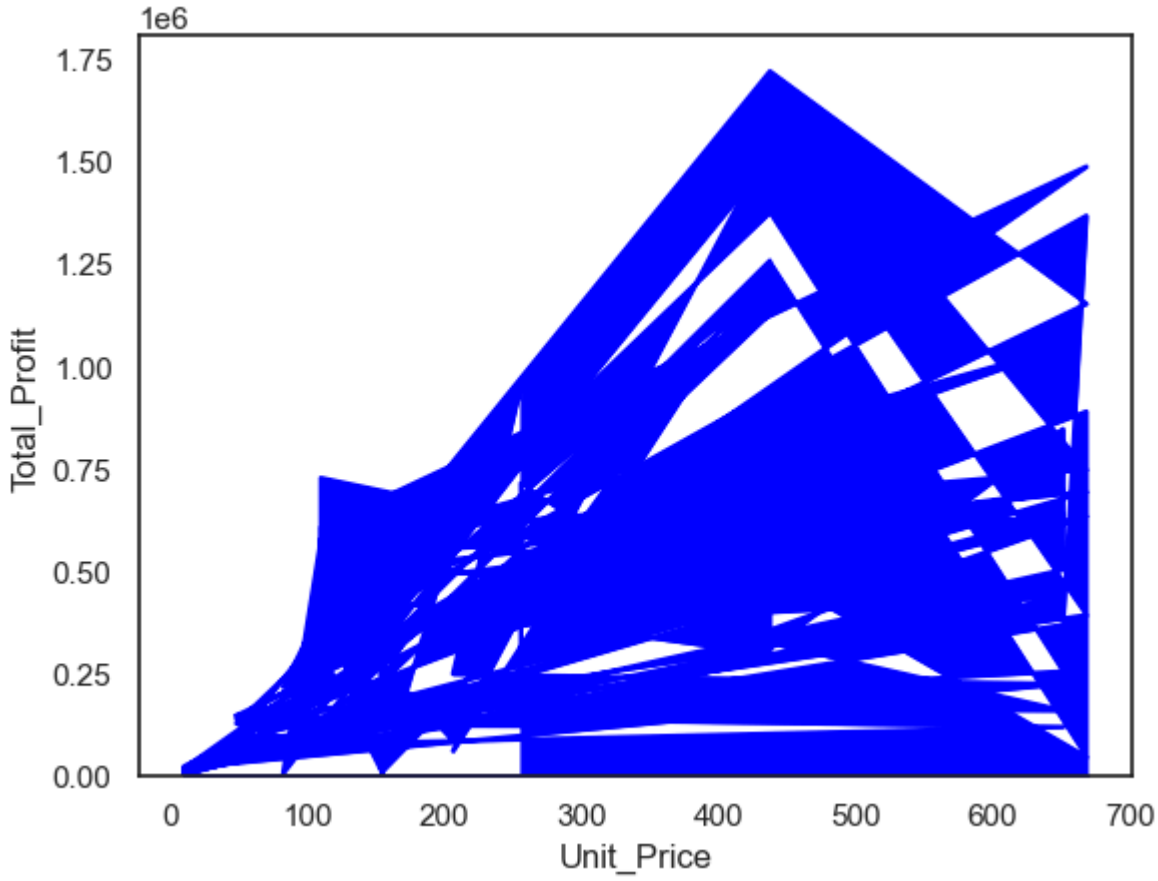
Out[137]:
```
Text(0, 0.5, 'Total_Cost')
```



Unit_Price and its Total Profit

In [138]:
```
area_plot = df.plot.area(x='Unit_Price',y='Total_Profit',color='blue',stacked=True,le
plt.ylabel('Total_Profit')

# Maximum profit has been generated in the unit price range of ₹400-₹500.
```

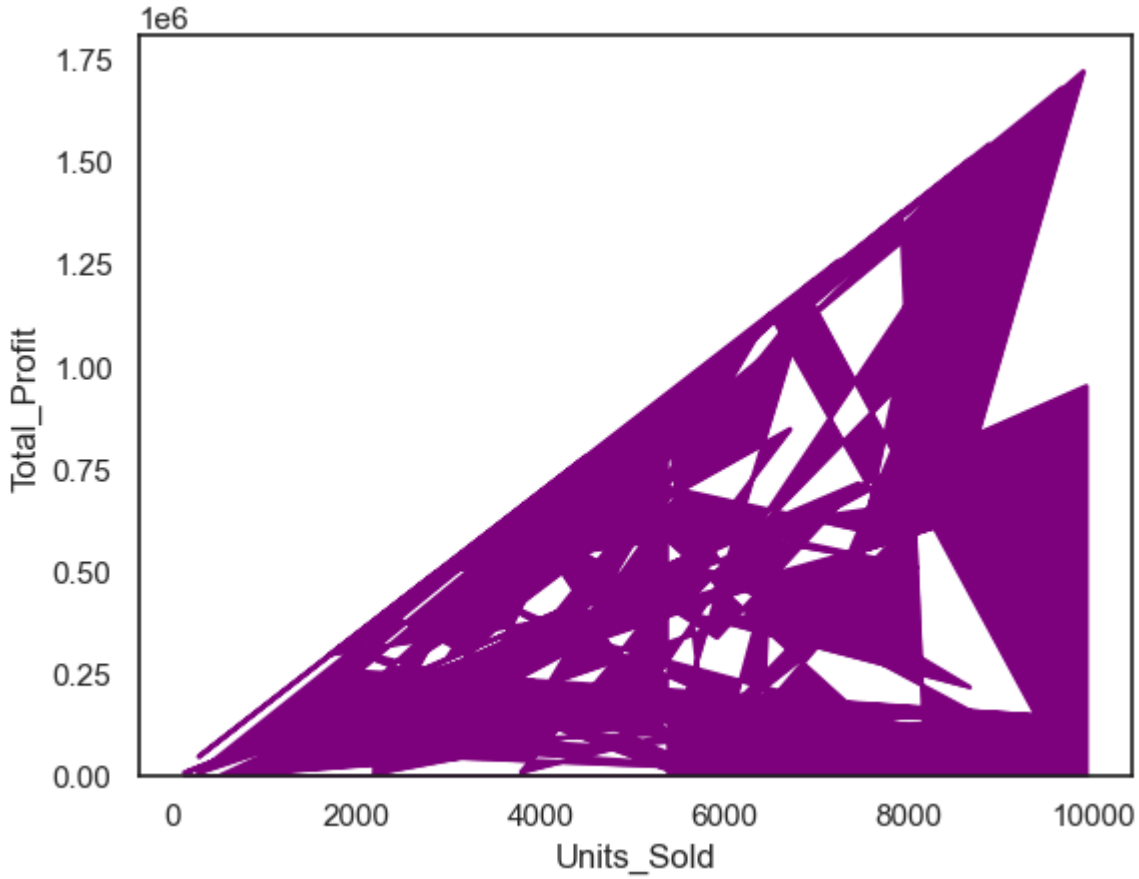
Out[138]: Text(0, 0.5, 'Total_Profit')



Total_Profit & Units_Sold

In [139]:
```
df.plot.area(x='Units_Sold',y='Total_Profit',color='purple',legend=None)
plt.ylabel('Total_Profit')


# Maximum profit has been generated when the number of units sold were between 8000 a
```

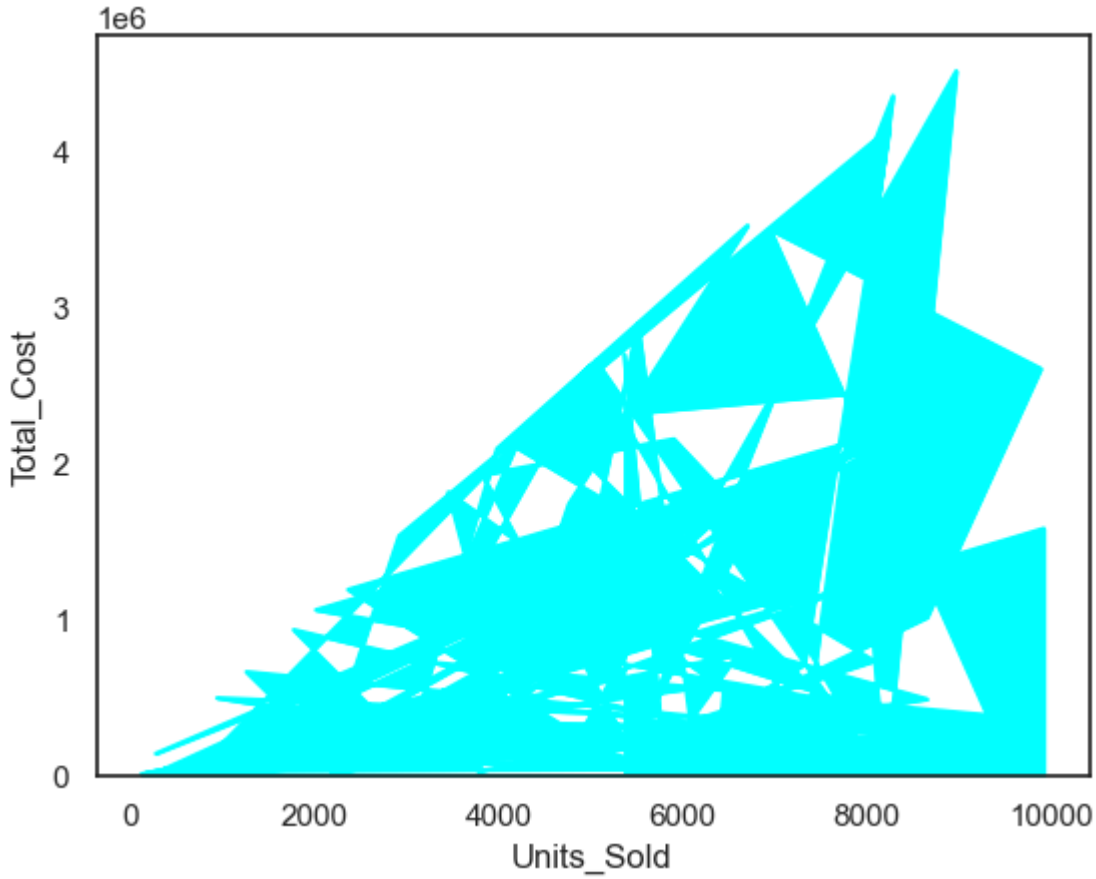
Out[139]: Text(0, 0.5, 'Total_Profit')



Total_Cost

In [140]:
```
df.plot.area(x='Units_Sold',y='Total_Cost',color='aqua',legend=None)
plt.ylabel('Total_Cost')


# Maximum cost has been generated when 8000-9000 units were sold.
```

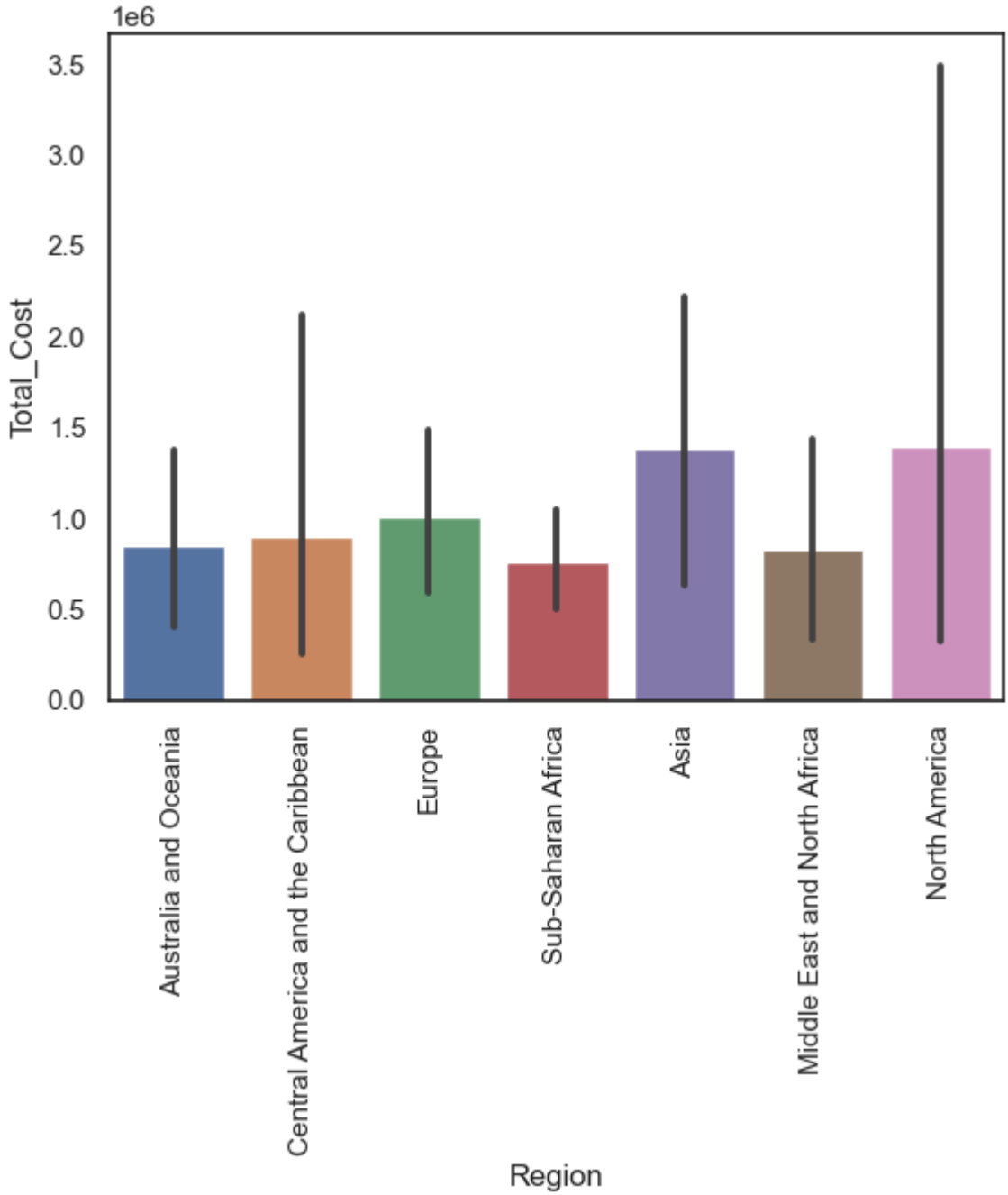
Out[140]:
```
Text(0, 0.5, 'Total_Cost')
```



Total_Cost as per Regions

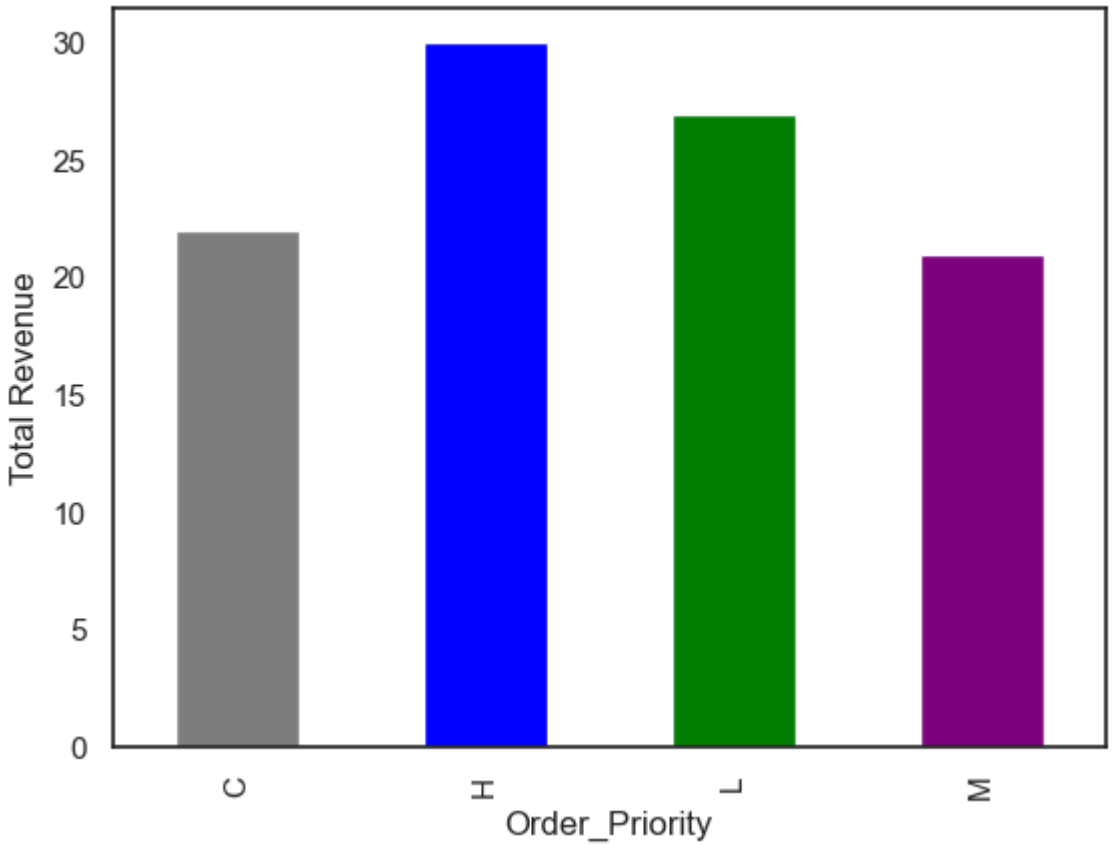
In [141]:
```
sns.barplot(x='Region',y='Total_Cost',data=df)
plt.xticks(rotation=90)

# Cost of items is maximum in Asia and North America, and minimum in Sub-Saharan Afri
```

Out[141]:
```
(array([0, 1, 2, 3, 4, 5, 6]),
 [Text(0, 0, 'Australia and Oceania'),
  Text(1, 0, 'Central America and the Caribbean'),
  Text(2, 0, 'Europe'),
  Text(3, 0, 'Sub-Saharan Africa'),
  Text(4, 0, 'Asia'),
  Text(5, 0, 'Middle East and North Africa'),
  Text(6, 0, 'North America')])
```



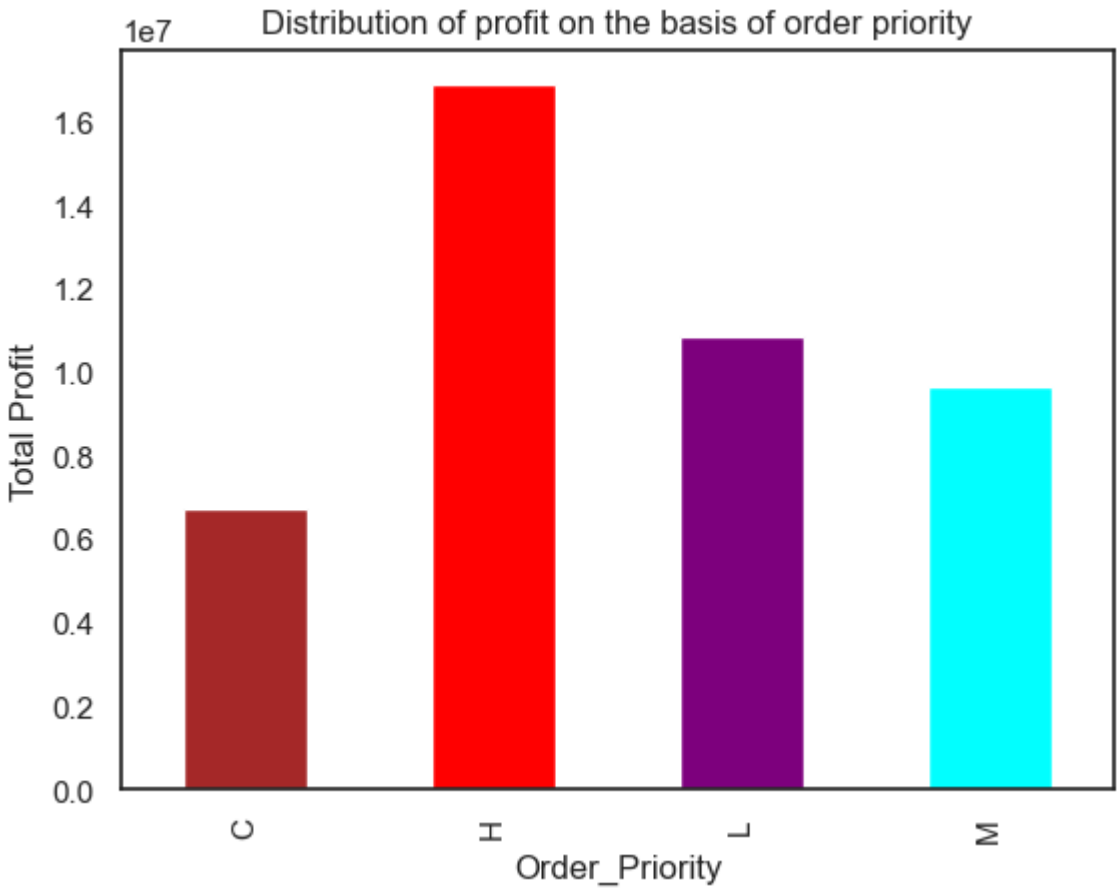
In [142]:
```
df.groupby('Order_Priority')['Total_Revenue'].count().plot(kind='bar',color=['grey','
plt.ylabel('Total Revenue')

# Maximum number of revenues has been generated by the products having order priority
#     minimum revenues has been generated by 'M' priority products.
```

Out[142]:
```
Text(0, 0.5, 'Total Revenue')
```



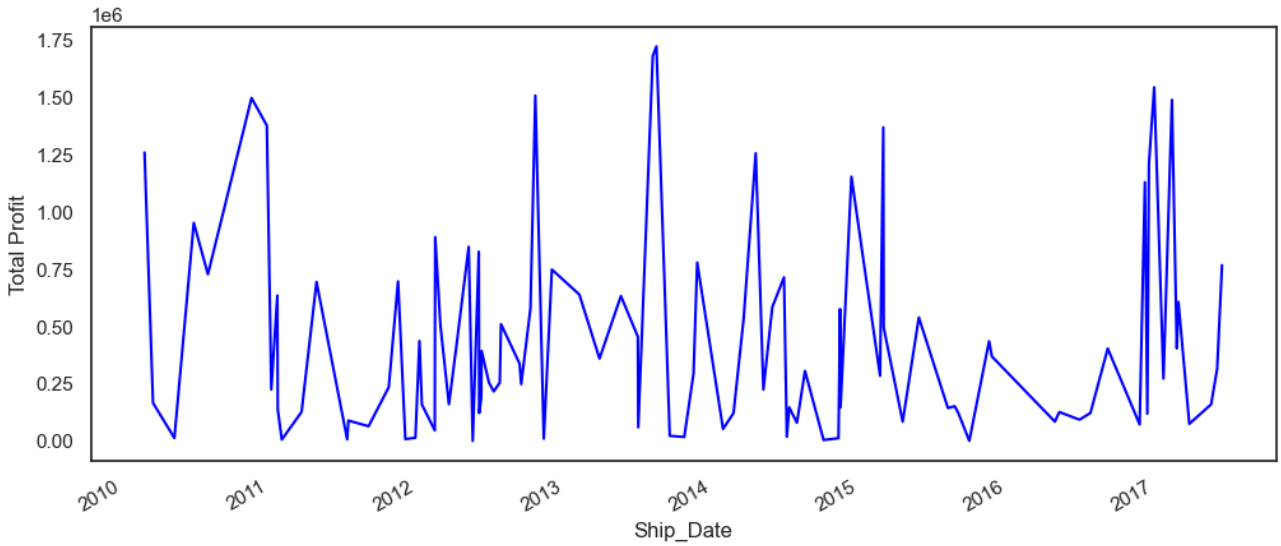
In [143]:
```
df.groupby('Order_Priority')['Total_Profit'].sum().plot(kind='bar',color=['brown','re
plt.ylabel('Total Profit')
plt.title('Distribution of profit on the basis of order priority')

# Maximum profit has been generated by products having order priority 'H' while
#   minimum profit has been obtained in case of 'C' priority product orders.
```

Out[143]: Text(0.5, 1.0, 'Distribution of profit on the basis of order priority')



In [144]:
```
plt.figure(figsize=(12,5))
df.groupby('Ship_Date')['Total_Profit'].sum().plot(kind='line',color='blue',sort_colu
plt.ylabel('Total Profit')

# Maximum profit has been achieved during the year 2014.
```

Out[144]: Text(0, 0.5, 'Total Profit')



In [145]:
```
# Label Encoding of Item Type, Sales Channel and Order Priority for model training
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df["Item_Type"] = le.fit_transform(df["Item_Type"])
df["Sales_Channel"] = le.fit_transform(df["Sales_Channel"])
df["Order_Priority"] = le.fit_transform(df["Order_Priority"])
```

In [146]:
```
df.head(4)
```

Out[146]:

	Region	Country	Item_Type	Sales_Channel	Order_Priority	Order_ID	Ship_Date	Units_Sold	Unit
0	Australia and Oceania	Tuvalu	0	0	1	669165933	2010-06-27	9925	
1	Central America and the Caribbean	Grenada	2	1	0	963881480	2012-09-15	2804	
2	Europe	Russia	8	0	2	341417157	2014-05-08	1779	
3	Sub-Saharan Africa	Sao Tome and Principe	5	1	0	514321792	2014-07-05	8102	

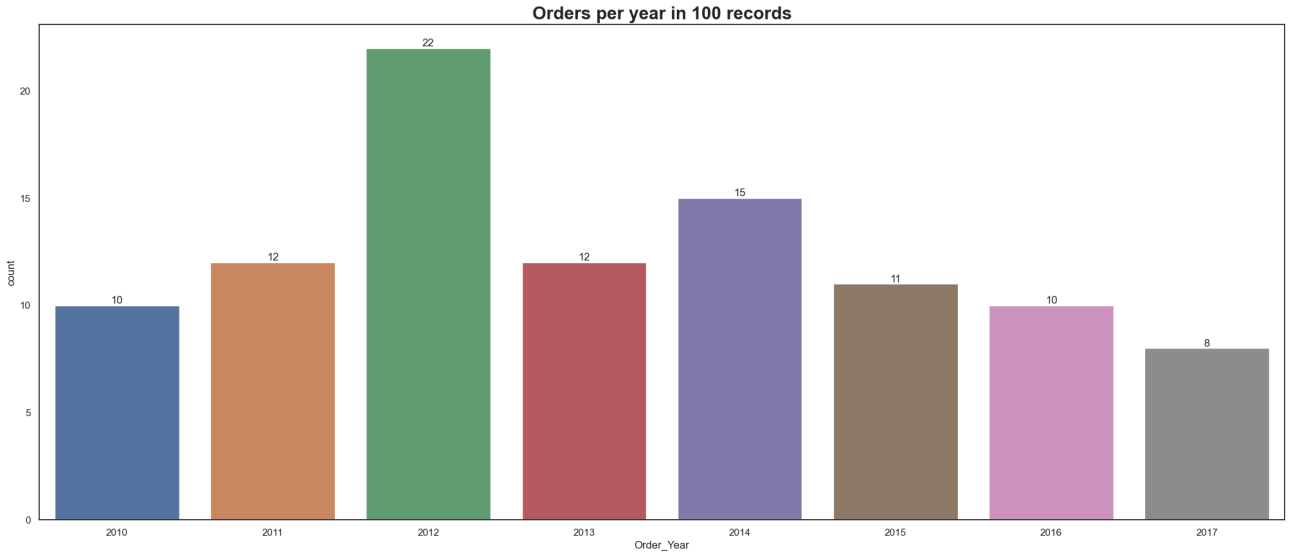
In [147]:
```
# Drop columns Region, Country, Order Date MonthYear, Order ID and Ship Date
#df = df.drop("Region", axis=1)
#df = df.drop("Country", axis=1)
#df = df.drop("Order_Date_MonthYear", axis=1)
#df = df.drop("Order_ID", axis=1)
#df = df.drop("Ship_Date", axis=1)
```

In [148]:
```
df.head()
```

Out[148]:

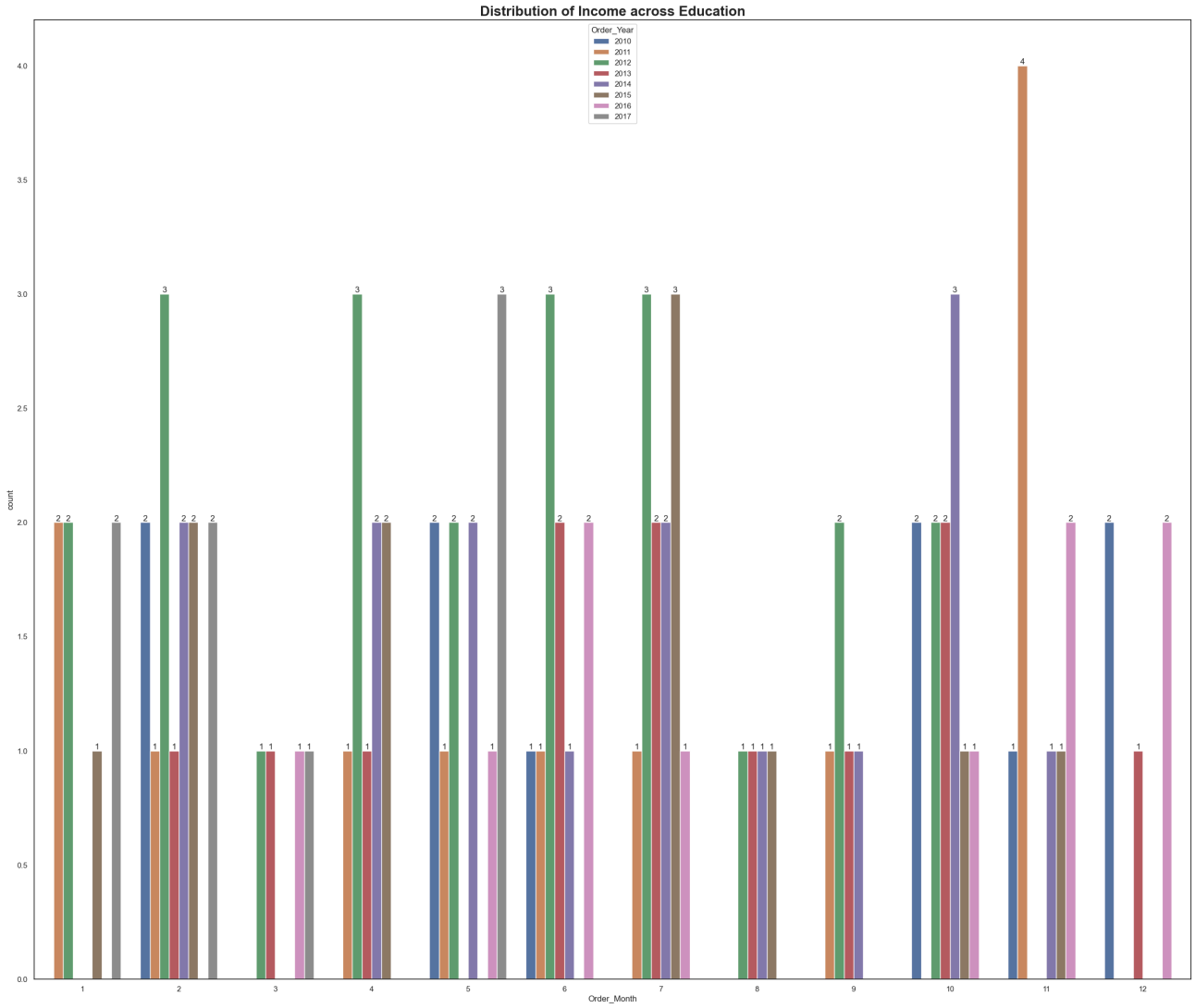
	Region	Country	Item_Type	Sales_Channel	Order_Priority	Order_ID	Ship_Date	Units_Sold	Unit
0	Australia and Oceania	Tuvalu	0	0	1	669165933	2010-06-27	9925	
1	Central America and the Caribbean	Grenada	2	1	0	963881480	2012-09-15	2804	
2	Europe	Russia	8	0	2	341417157	2014-05-08	1779	
3	Sub-Saharan Africa	Sao Tome and Principe	5	1	0	514321792	2014-07-05	8102	
4	Sub-Saharan Africa	Rwanda	8	0	2	115456712	2013-02-06	5062	

In [149]:
```
# Creating a countplot for 'Order_Year'
plt.figure(figsize=(25,10))
graph=sns.countplot(x="Order_Year",data=df)
for i in graph.containers:
    graph.bar_label(i)
    plt.title('Orders per year in 100 records', fontdict={'fontsize': 20, 'fontweight
```

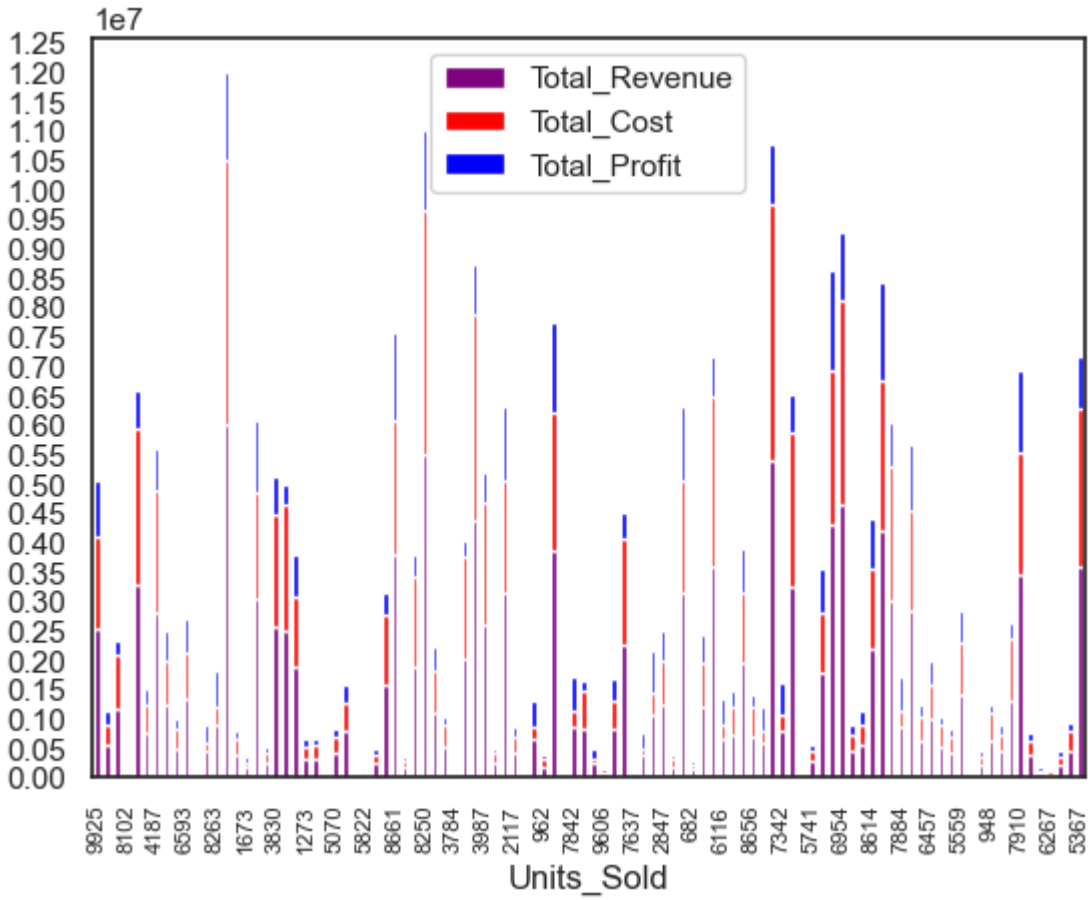


In [150]:
```
plt.figure(figsize=(30,25))
graph=sns.countplot(data=df, x='Order_Month', hue='Order_Year')
for i in graph.containers:
    graph.bar_label(i)
plt.title('Distribution of Income across Education', fontdict={'fontsize': 20, 'fontw
```

Out[150]:
```
Text(0.5, 1.0, 'Distribution of Income across Education')
```

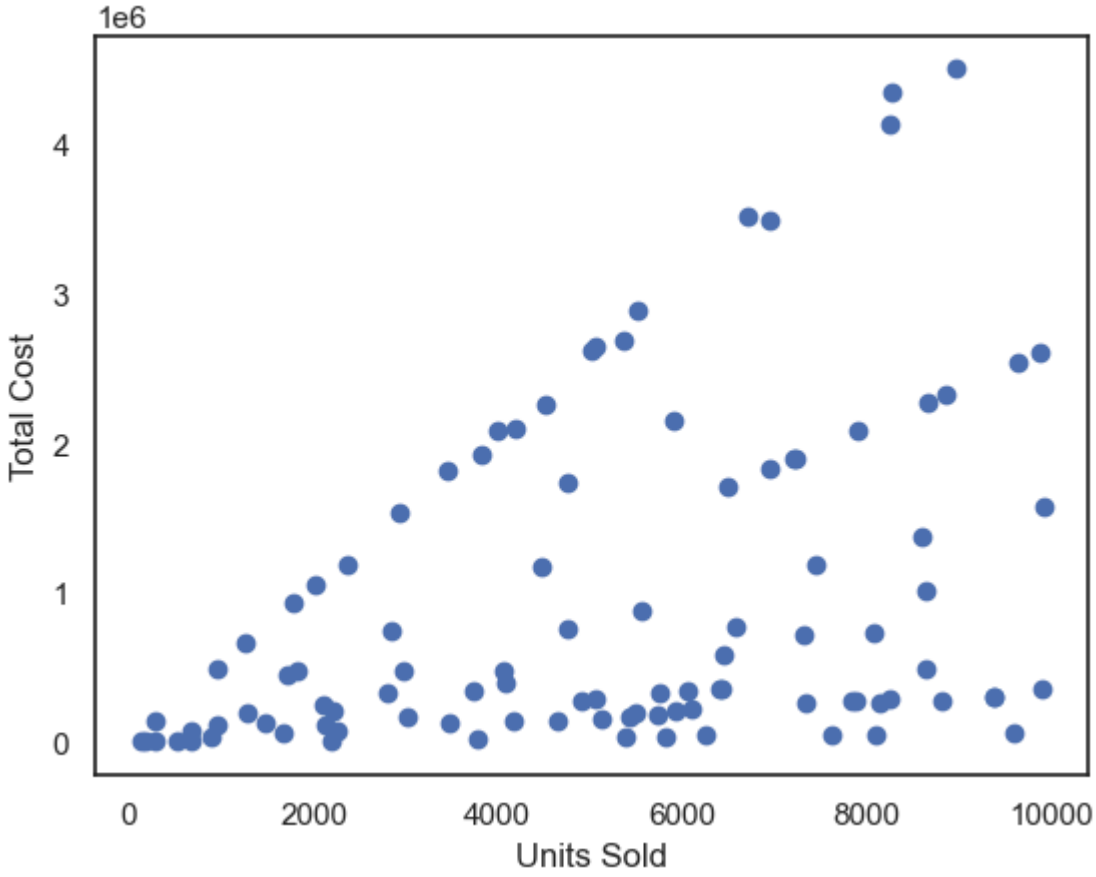


In [151]:
```
bar_plot = df.plot.bar(x='Units_Sold',y=['Total_Revenue','Total_Cost','Total_Profit'])
plt.xticks(rotation=90)
plt.locator_params(nbins=40)
plt.tick_params(axis='x', labelsize=8)
```



In [152]:
```
plt.scatter(df['Units_Sold'],df['Total_Cost'])
plt.xlabel('Units Sold')
plt.ylabel('Total Cost')

# More the number of units sold of a product, more will be the total cost associated
```

Out[152]:
```
Text(0, 0.5, 'Total Cost')
```



In [ ]: