analysis-data-analytics-project-1

March 18, 2024

0.1 AMAZON SALES DATA ANALYSIS

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
pd.options.display.max_columns= None
pd.options.display.max_rows= None

np.set_printoptions(suppress=True)
```

[3]:
```
data= pd.read_csv(r'C:\Amisha\Unified Mentor\Amazon Sales Dataset\Amazon Sales␣
↪data.csv')
data.head()
```

[3]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243.50	951410.50
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376.44	248406.36
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158502.59	933903.84	224598.75
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014	8102					
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062					

```
3        9.33       6.92       75591.66    56065.84      19525.82
4      651.21     524.96     3296425.02  2657347.52     639077.50
```

[4]:
```
data.info()
```

```
<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
```

[5]:
```
data.describe(include=[np.number]).T
```

[5]:

	count	mean	std	min	25%	50%	75%	max
Order ID	100.0	5.550204e+08	2.606153e+08	1.146066e+08	3.389225e+08	5.577086e+08	7.907551e+08	9.940222e+08
Units Sold	100.0	5.128710e+03	2.794485e+03	1.240000e+02	2.836250e+03	5.382500e+03	7.369000e+03	9.925000e+03
Unit Price	100.0	2.767613e+02	2.355922e+02	9.330000e+00	8.173000e+01	1.798800e+02	4.372000e+02	6.682700e+02
Unit Cost	100.0	1.910480e+02	1.882082e+02	6.920000e+00	3.584000e+01	1.072750e+02	2.633300e+02	5.249600e+02
Total Revenue	100.0	1.373488e+06	1.460029e+06	4.870260e+03	2.687212e+05	7.523144e+05	2.212045e+06	5.997055e+06
Total Cost	100.0	9.318057e+05	1.083938e+06	3.612240e+03	1.688680e+05	3.635664e+05	1.613870e+06	4.509794e+06
Total Profit	100.0	4.416820e+05	4.385379e+05	1.258020e+03	1.214436e+05	2.907680e+05	6.358288e+05	1.719922e+06

[6]:
```
data.describe(include='object').T
```

[6]:

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

[7]:
```
data.dtypes
```

[7]:
```
Region             object
Country            object
Item Type          object
Sales Channel      object
Order Priority     object
Order Date         object
Order ID            int64
Ship Date          object
Units Sold          int64
Unit Price        float64
Unit Cost         float64
Total Revenue     float64
Total Cost        float64
Total Profit      float64
dtype: object
```

[8]:
```
# Changing the data type
data['Order Date'] = pd.to_datetime(data['Order Date'])
data['Ship Date'] = pd.to_datetime(data['Ship Date'])
```

[9]:
```
data.shape
```

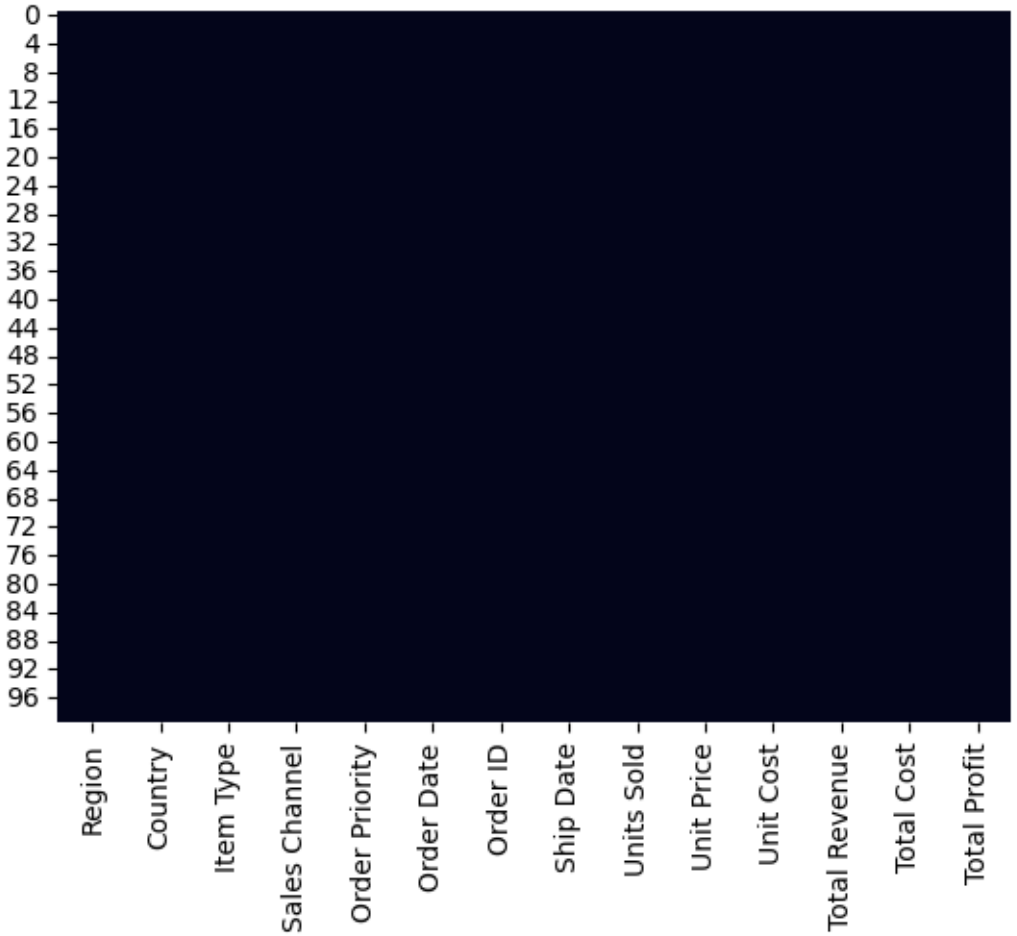
[9]:
```
(100, 14)
```

[10]:
```
#checking for the null values

Total = data.isnull().sum().sort_values(ascending=False)
Percent = (data.isnull().sum()*100)/(data.isnull().sum().count())

missing_values = pd.concat([Total, Percent], axis=1, keys=␣
 ↪['Total','Percentage'])
missing_values
```

[10]:

	Total	Percentage
Region	0	0.0
Country	0	0.0
Item Type	0	0.0
Sales Channel	0	0.0
Order Priority	0	0.0
Order Date	0	0.0
Order ID	0	0.0
Ship Date	0	0.0
Units Sold	0	0.0
Unit Price	0	0.0
Unit Cost	0	0.0
Total Revenue	0	0.0
Total Cost	0	0.0
Total Profit	0	0.0

[11]:
```
sns.heatmap(data.isnull(), cbar= False)
plt.show()
```



This shows that there are no null values in the given data.

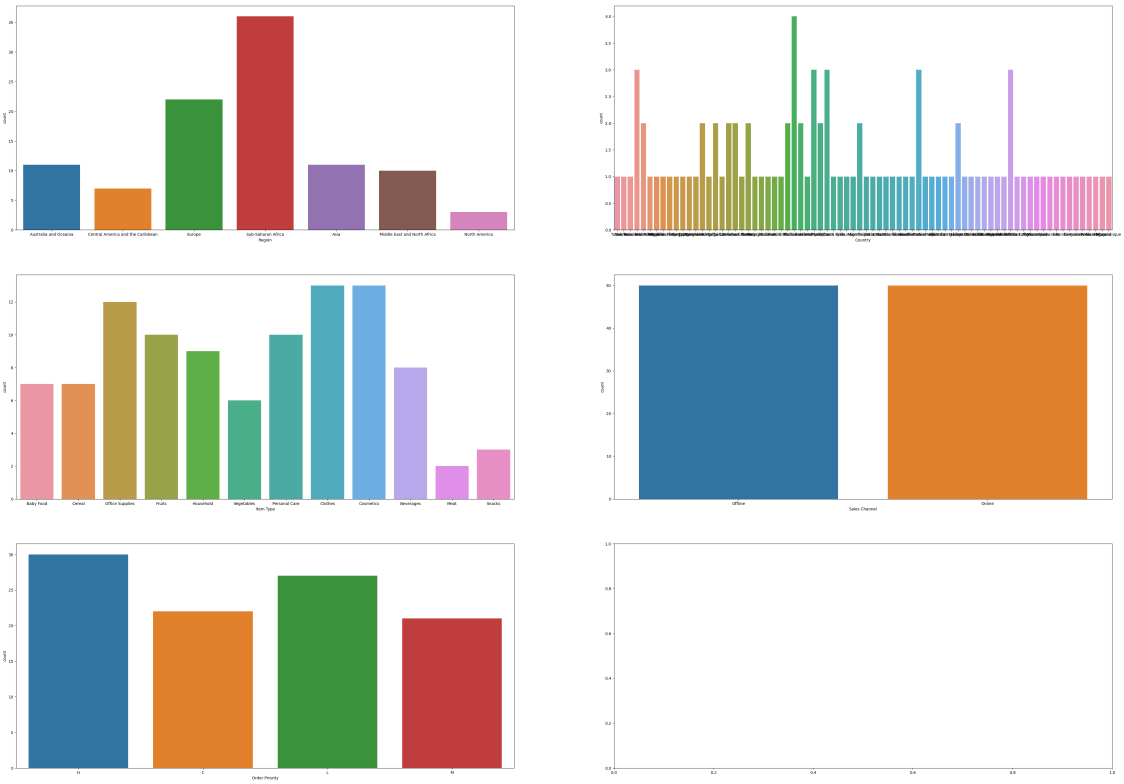
Let's analyse the data in it's rawest form before checking for the outliers and removing them.

[12]:
```
#plotting the histogram of numeric variables

numerical=['Units Sold','Unit Price','Unit Cost','Total Revenue','Total␣
 ↪Cost','Total Profit']

data[numerical].hist(xrot=45, color= 'c')
plt.tight_layout()
plt.show()
```

The histograms shows the type of distribution, the numerical variables have. Some of them are:

1. Total Revenue, Total Cost and Total Profit are rightly skewed.
2. Units Sold seems to have a uniform distribution.
3. Unit Price and Unit cost have non-symmetric and uneven distribution

[13]:
```
#plotting countplot of categorical variables

from pandas.api.types import is_string_dtype

categorical=[]

for column in data:
    if is_string_dtype(data[column]):
        categorical.append(column)

categorical
fig, ax = plt.subplots(nrows=3, ncols=2,figsize=[50,35])

for variable,subplot in zip(categorical, ax.flatten()):
    sns.countplot(data[variable], ax= subplot)

plt.show()
```



The countplot of variables shows the spread of data, the categorical variables have. Some of them are:

1. The maximum data is given of Sub Sharan Africa and the minimum data is of North America.
2. Clothing items data is the most while the least data is of meat.

3. There isn't much difference in the data in sales channel modes which is considered good for analysis of data.

[14]:
```
data['Year']= data['Order Date'].dt.year
data['Month']= data['Order Date'].dt.month
data['Year-Month']= data['Order Date'].dt.to_period('M')
```

[15]:
```
data['Shipping Days'] = (data['Ship Date'] - data['Order Date']).dt.days
data.head()
```

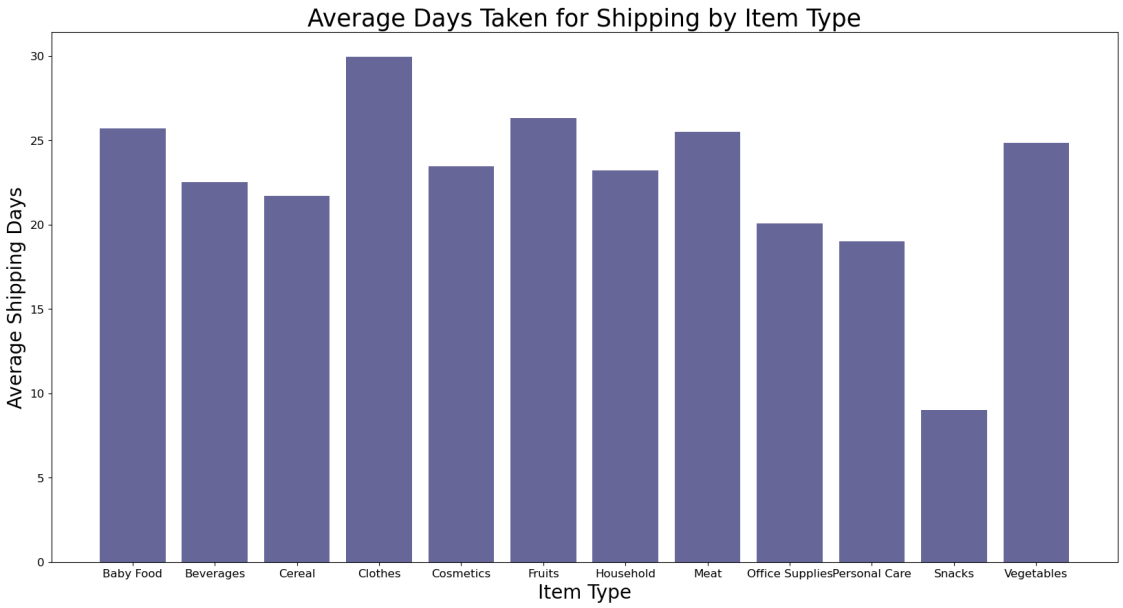
[15]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit	Year	Month	Year-Month	Shipping Days
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.00	1582243.50	951410.50	2010	5	2010-05	30
1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	576782.80	328376.44	248406.36	2012	8	2012-08	24
2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	2014-05-08	1779	651.21	524.96	1158502.59	933903.84	224598.75	2014	5	2014-05	6
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	2014-07-05	8102	9.33	6.92	75591.66	56065.84	19525.82	2014	6	2014-06	15
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-02-01	115456712	2013-02-06	5062	651.21	524.96	3296425.02	2657347.52	639077.50	2013	2	2013-02	5

[16]:
```
plt.figure(figsize=(20,10))

# Calculating the average days taken for shipping of each item type
avg_shipping_days = data.groupby('Item Type')['Shipping Days'].mean().
 ↪reset_index()

plt.figure(figsize=(20,10))
plt.bar(avg_shipping_days['Item Type'], avg_shipping_days['Shipping Days'],␣
 ↪color=(0.4, 0.4, 0.6))
```

```
plt.xlabel('Item Type', fontsize=20)
plt.ylabel('Average Shipping Days', fontsize=20)
plt.title('Average Days Taken for Shipping by Item Type', fontsize=25)
plt.xticks(fontsize='12')
plt.yticks(fontsize='12')
plt.show()
```

```
<Figure size 2000x1000 with 0 Axes>
```



Analysis of average days taken for shipping of each item type: 1. Clothing items took around 29 days on an average for shipping. It was the highest days taken among all.

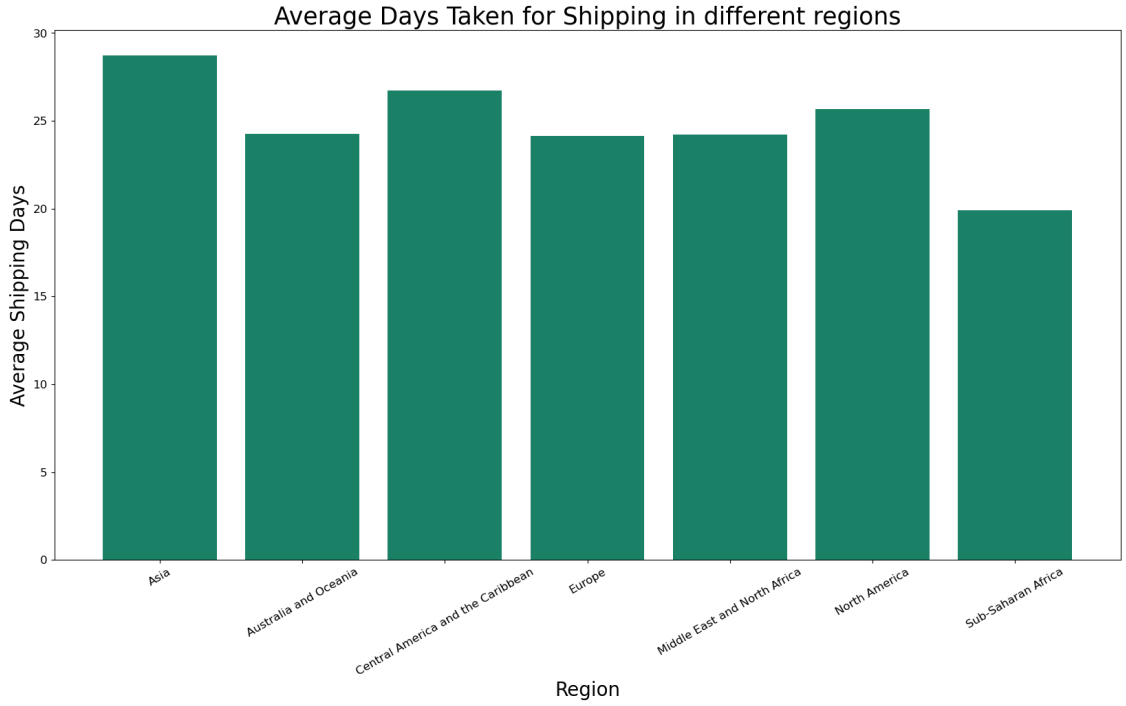
2. Snacks took only 8-9 days on an average to get shipped. It was the least days taken among all.

3. Other items took approximately between 20-27 days for shipping.

[17]:
```
plt.figure(figsize=(20,10))

# Calculating the average days taken for shipping in each region
avg_shipping_days = data.groupby('Region')['Shipping Days'].mean().reset_index()

plt.figure(figsize=(20,10))
plt.bar(avg_shipping_days['Region'], avg_shipping_days['Shipping Days'],␣
↪color=(0.1, 0.5, 0.4))
plt.xlabel('Region', fontsize=20)
plt.ylabel('Average Shipping Days', fontsize=20)
plt.title('Average Days Taken for Shipping in different regions', fontsize=25)
plt.xticks(fontsize='12', rotation=30)
```

```
plt.yticks(fontsize='12')
plt.show()
```

```
<Figure size 2000x1000 with 0 Axes>
```



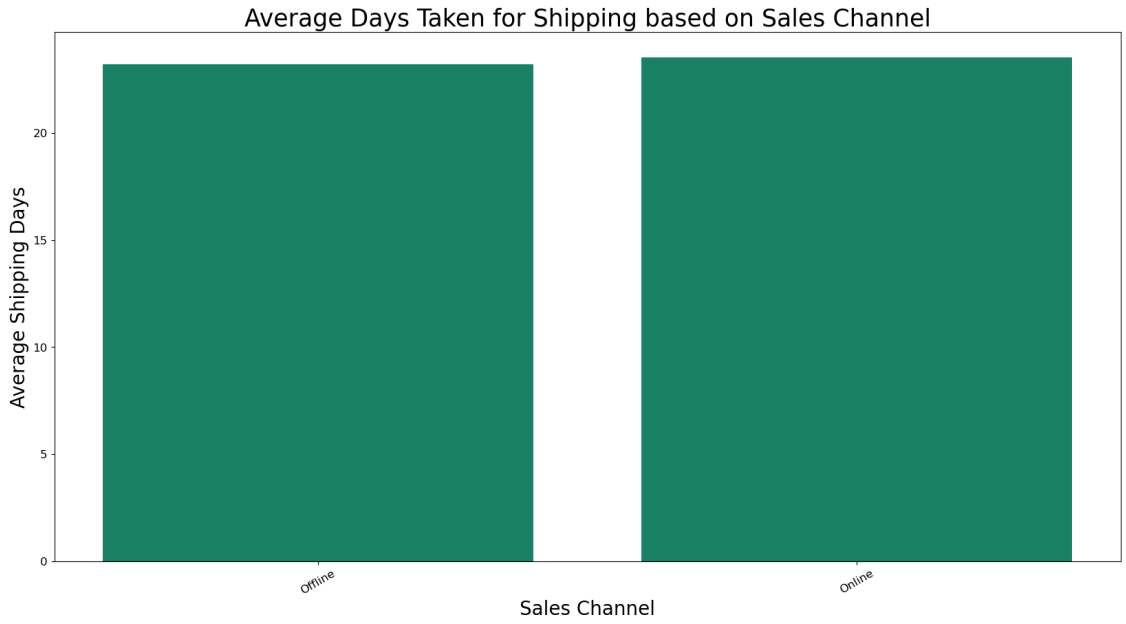
Analysis of average days taken for shipping in different regions: 1. There's no vast difference between each of the region. In all the regions, days taken for shipping on an average was between 20-30 days.

2. In Asia, the number was the highest(28-29 days) while in Sub-Saharan Africa, the number was the least(19-20 days).

[18]:
```
plt.figure(figsize=(20,10))

avg_shipping_days = data.groupby('Sales Channel')['Shipping Days'].mean().
 ↪reset_index()

plt.figure(figsize=(20,10))
plt.bar(avg_shipping_days['Sales Channel'], avg_shipping_days['Shipping Days'],␣
 ↪color=(0.1, 0.5, 0.4))
plt.xlabel('Sales Channel', fontsize=20)
plt.ylabel('Average Shipping Days', fontsize=20)
plt.title('Average Days Taken for Shipping based on Sales Channel', fontsize=25)
plt.xticks(fontsize='12', rotation=30)
plt.yticks(fontsize='12')
```

```
plt.show()
```

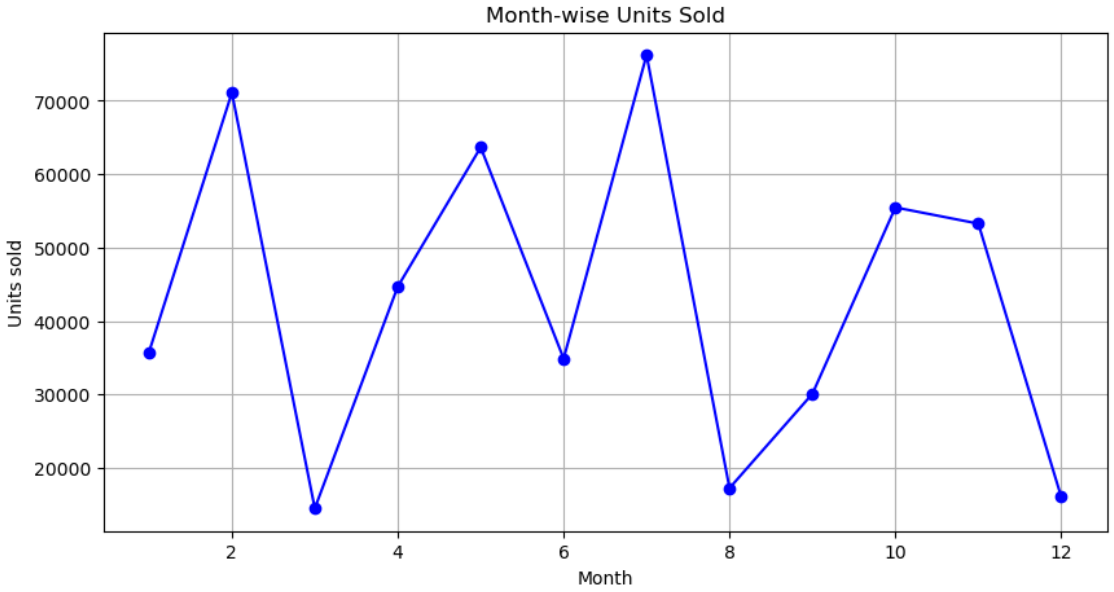
```
<Figure size 2000x1000 with 0 Axes>
```



Analysis of average days taken for shipping based on Sales Channel: There's no such impact on the shipping days based on the Sales Channel. The Online mode caused 1 or 2 days extra for shipping, not much.

0.2 UNITS SOLD ANALYSIS

[19]:
```
plt.figure(figsize=[10,5])

monthly_sales = data.groupby('Month')['Units Sold'].sum()
monthly_sales.plot( color='b', marker='o', grid=True)

plt.title('Month-wise Units Sold')
plt.xlabel('Month')
plt.ylabel('Units sold')
plt.show()
```



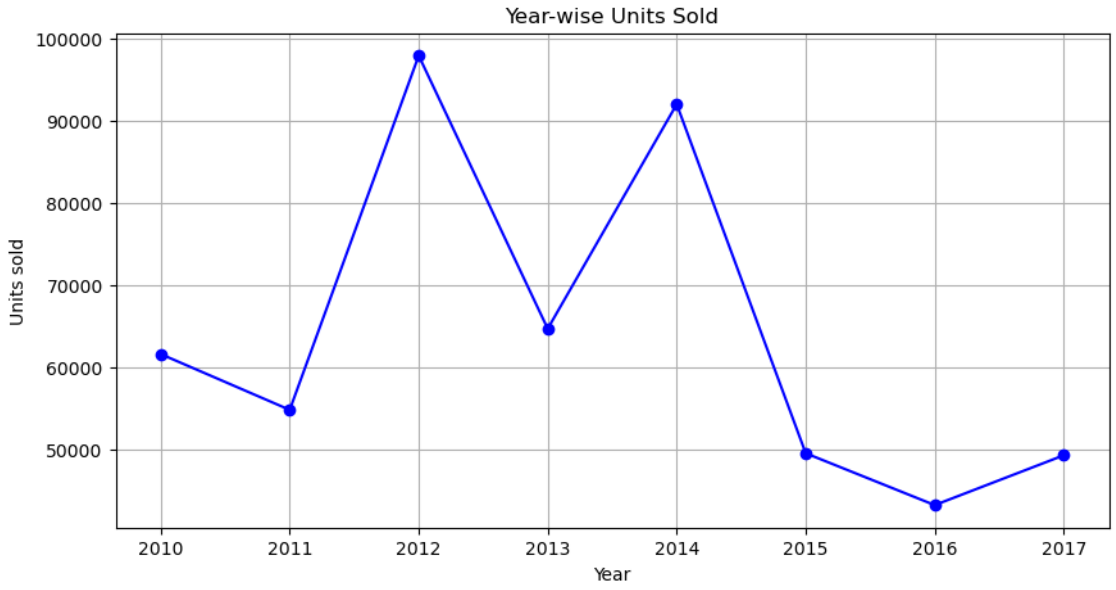
0.2.1 Month-wise analysis of UNITS SOLD:

1. Overall out of all months, the highest units were sold during July(more than 70000 units) followed by February while the least units were sold in month March(less than 10000 units) followed by December and August.

2. After drastic decline in March, units were sold at a good amount during March and May.

3. Mostly within 2 months, the graph was gradually increasing and decreasing.

[20]:
```
plt.figure(figsize=[10,5])

yearly_sales = data.groupby('Year')['Units Sold'].sum()
yearly_sales.plot(marker='o', color='b', grid=True)

plt.title('Year-wise Units Sold')
plt.xlabel('Year')
plt.ylabel('Units sold')
plt.show()
```



0.2.2 Year-wise analysis of UNITS SOLD:

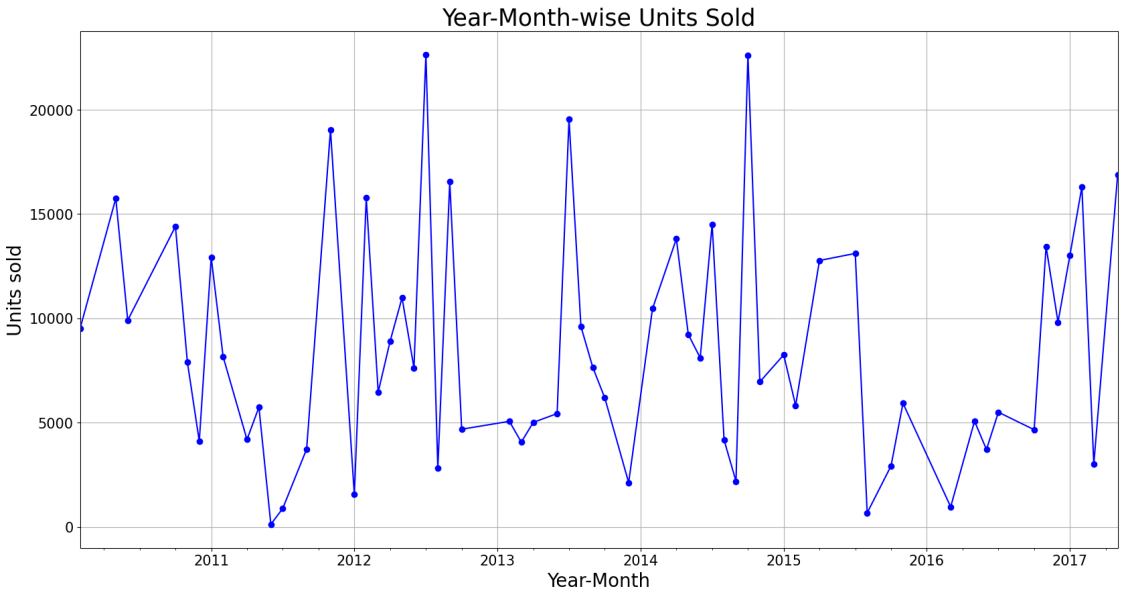
1. Out of all years, almost 100000 units were sold in 2012 followed by 90000-95000 units sold in 2014. These numbers are the highest among all.

2. The least number of units were even less than 45000 which were sold in 2016.

3. In 2015 and 2017, same number of units were sold which were almost 50000.

4. Most of the units were sold during 2011-2015.

[21]:
```
plt.figure(figsize=[20,10])

yearly_monthly = data.groupby('Year-Month')['Units Sold'].sum()
yearly_monthly.plot(marker='o', color='b',grid=True)

plt.title('Year-Month-wise Units Sold', fontsize=25)
plt.xlabel('Year-Month', fontsize=20)
plt.ylabel('Units sold', fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



0.2.3 Year-Month-wise analysis of UNITS SOLD:

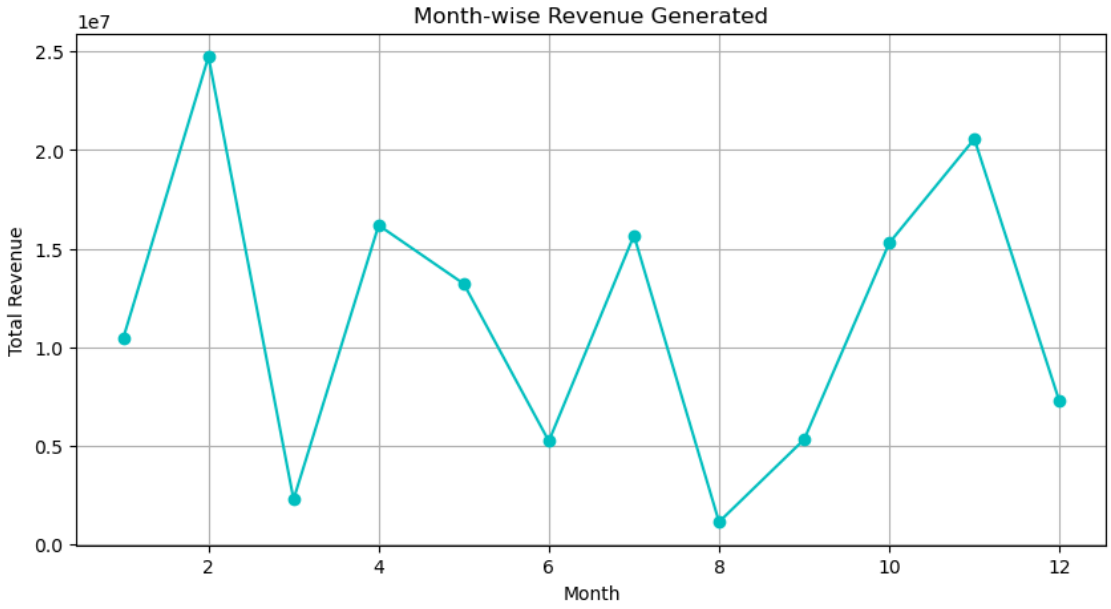
1. The highest number of units,i.e, more than 20000 units sold were during June 2012 and during September 2014 followed by the selling of almost 20000 units during October-November 2011 and June 2013.

2. Almost zero units were sold in the second quadrant of year 2011,i.e, during May 2011.

3. The graph can be divided into three part (January 2010 and May 2011), (May 2011 and July 2015) and (July 2015 and April 2017). It can be obsereved that in all three parts, there was a sudden rise and fall in number of units sold which was taking place.

0.3 TOTAL REVENUE ANALYSIS

[22]:
```
plt.figure(figsize=[10,5])

monthly_revenue = data.groupby('Month')['Total Revenue'].sum()
monthly_revenue.plot( color='c', marker='o', grid=True)

plt.title('Month-wise Revenue Generated')
plt.xlabel('Month')
plt.ylabel('Total Revenue')
plt.show()
```

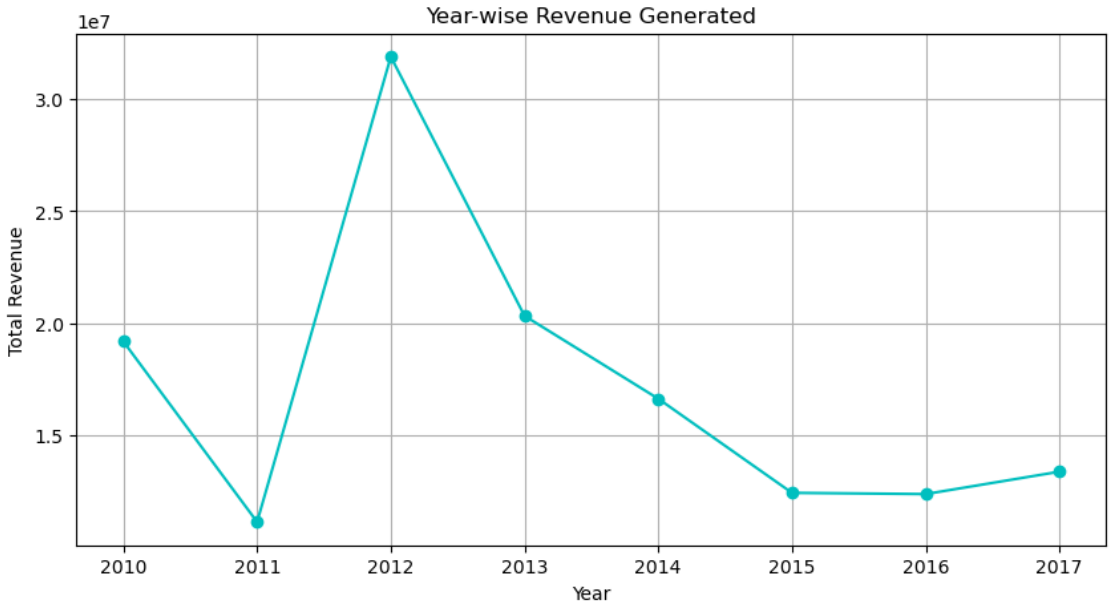


Month-wise Analysis of TOTAL REVENUE: 1. The highest revenue in total generated out of all months was in February followed by month November.

2. The least revenue was generated in August followed by March.

3. June and September have similar values for total revenue and here in both the value for revenue is very less as compared to others.

[23]:
```
plt.figure(figsize=[10,5])

yearly_revenue = data.groupby('Year')['Total Revenue'].sum()
yearly_revenue.plot(marker='o', color='c', grid=True)

plt.title('Year-wise Revenue Generated')
plt.xlabel('Year')
plt.ylabel('Total Revenue')
plt.show()
```



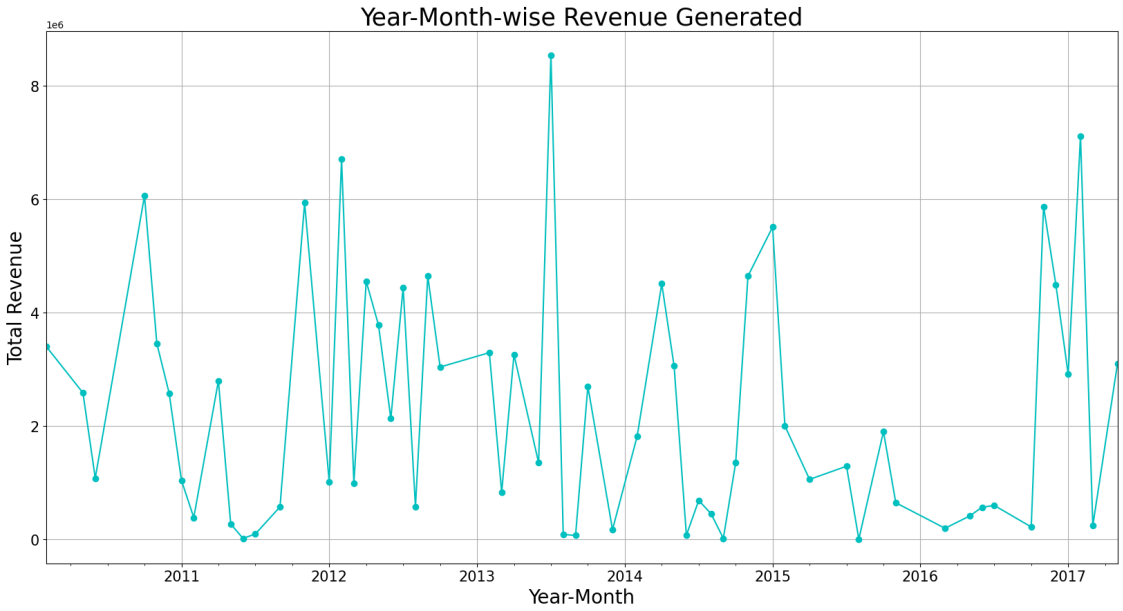
Year-wise Analysis of TOTAL REVENUE: 1. The total revenue was quite good in year 2010.

2. The maximum revenue was generated in 2012 after being the lowest in 2011.

3. After 2012, it gradually decreased till year 2015.

4. There was a slight improvement in 2017 after having same revenue value in 2016.

[24]:
```
plt.figure(figsize=[20,10])

yearly_monthly = data.groupby('Year-Month')['Total Revenue'].sum()
yearly_monthly.plot(marker='o', color='c',grid=True)

plt.title('Year-Month-wise Revenue Generated', fontsize=25)
plt.xlabel('Year-Month', fontsize=20)
plt.ylabel('Total Revenue', fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



Year-Month-wise Analysis of TOTAL REVENUE: 1. The highest revenue was generated during June 2013 while the lowest revenue(almost zero) was generated in many periods throughout all years, i.e, during May 2011, July-August 2013, in mid 2014(during May-August), and July 2015.

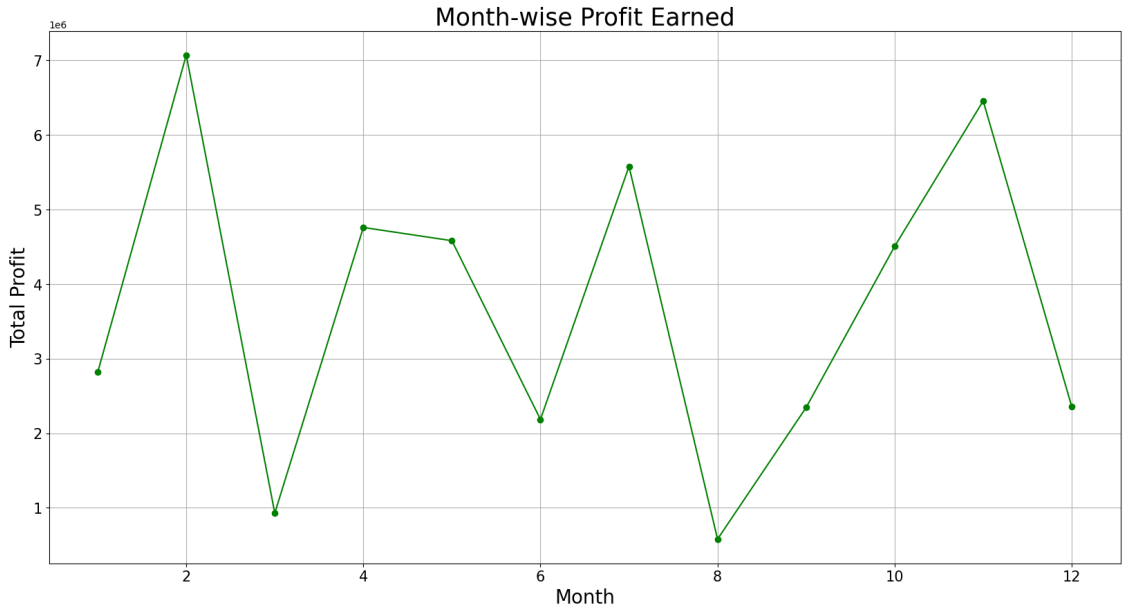
2. The graph is not regular and has numerous ups and downs.

3. From July 2015 to September 2016, this was the period where it added least to the total revenue overall.

4. Appriciable revenue generation is observed at certain points, i.e., during September 2010, October 2011, February 2012, October 2016 and February 2017.

0.4 TOTAL PROFIT ANALYSIS

[25]:
```
plt.figure(figsize=[20,10])

monthly = data.groupby('Month')['Total Profit'].sum()
monthly.plot(marker='o', color='g',grid=True)

plt.title('Month-wise Profit Earned', fontsize=25)
plt.xlabel('Month', fontsize=20)
plt.ylabel('Total Profit', fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```

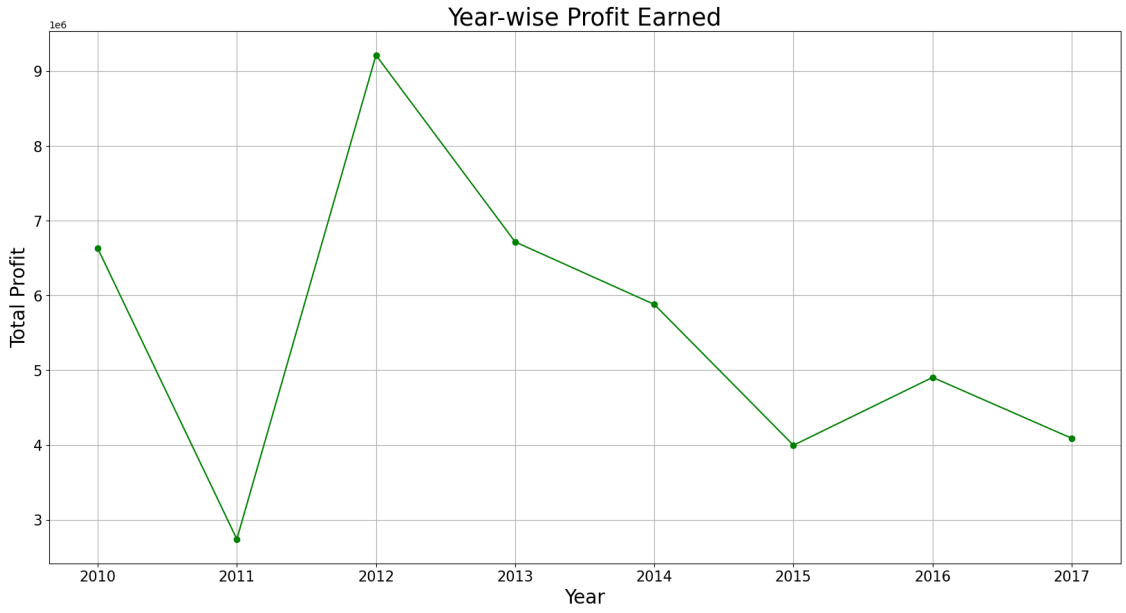


Month-wise Analysis of TOTAL PROFIT: 1. The graph shows the highest profit was earned in month February followed by November and July respectively.

2. The least profit was earned in month August. After that it gradually increaed for months September, October and November but again decreased for December.

3. There was insignificant difference in the values of total profit for months April and May.

[26]:
```
plt.figure(figsize=[20,10])

yearly = data.groupby('Year')['Total Profit'].sum()
yearly.plot(marker='o', color='g',grid=True)

plt.title('Year-wise Profit Earned', fontsize=25)
plt.xlabel('Year', fontsize=20)
plt.ylabel('Total Profit', fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```

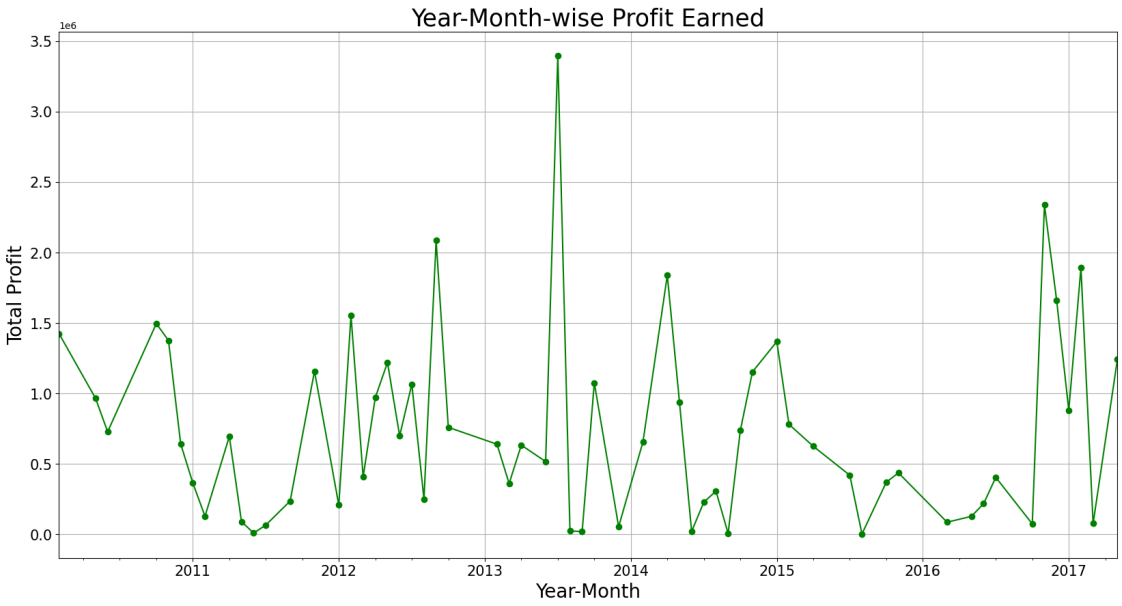


Year-wise Analysis of TOTAL PROFIT: 1. A good amount of profit was earned in 2010 but it decreased drastically in 2011 having the lowest value among all.

2. The highest value of profit is in 2012 but after that it again gradually decreased till year 2015.

3. There was a slight improvement in 2016 but it again got back to the value which was similar to that in 2015.

[27]:
```
plt.figure(figsize=[20,10])

yearly_monthly = data.groupby('Year-Month')['Total Profit'].sum()
yearly_monthly.plot(marker='o', color='g',grid=True)

plt.title('Year-Month-wise Profit Earned', fontsize=25)
plt.xlabel('Year-Month', fontsize=20)
plt.ylabel('Total Profit', fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```

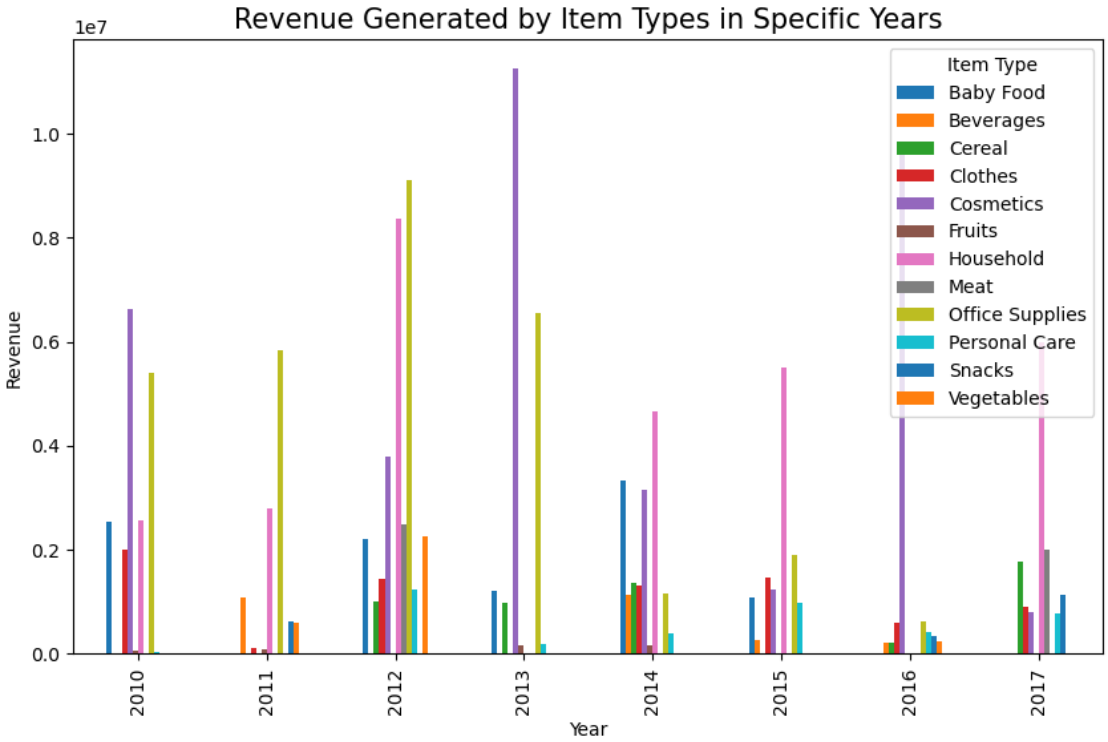


Year-Month-wise Analysis of TOTAL PROFIT: 1. The highest profit value is observed during June 2013.

2. The least profit value generated was zero which happened many times, specifically during May 2011, July-August 2013, May 2014, August 2014 and July 2015.

3. Appreciable values were observed during August 2012, February-March 2014, October 2016 and January 2017.

[28]:
```
revenue_data = data.groupby(['Item Type', 'Year'])['Total Revenue'].sum().
 ↪reset_index()

# Pivot the DataFrame for easy plotting
pivot_data = revenue_data.pivot(index='Year', columns='Item Type',␣
 ↪values='Total Revenue')

# Plottingb
pivot_data.plot(kind='bar', figsize=(10, 6))
plt.xlabel('Year', fontsize=10)
plt.ylabel('Revenue', fontsize=10)
plt.title('Revenue Generated by Item Types in Specific Years', fontsize=15)
plt.legend(title='Item Type')
plt.show()
```



0.4.1 Year-wise analysis (REVENUE GENERATED BY ITEM TYPES IN EVERY YEARS)

1. It can be seen that not all items were bought by the customers in specific years. Only few of each item type were bought in each year.

2. Overall, the highest revenue was generated by Cosmetics in year 2013 followed by Cosmetics only in year 2016.

3. The most common choice were Cosmetics, Office Supplies, Baby Food and Houshold items which generated more than half of the revenue overall.

4. Office supplies generated a significant and appreciable amount of revenue for four years, i.e., 2010-2013 but after 2013 there was a huge decline which continued every year till 2017.

5. Though household items generated significant revenue in each year but there was zero revenue generated by it in years 2013 and 2016.

6. Meat was sold only in years 2012 and 2017.

7. Vegetables were bought in years 2011, 2012 and 2016 which generated less revenue in comparison to all.

8. Snacks could able to contribute in total revenue in years 2011,2016 and 2017.

9. Personal care items generated very less revenue each year except for year 2011 where it was zero.

10. Fruits also generated very less revenue in years 2010, 2011, 2013 and 2014. In rest of the years it couldn't contribute anything to the revenue.

11. Out of the blue, Cosmetics produced zero revenue in year 2011. Rest of the years, it contributed significant values to the overall revenue.

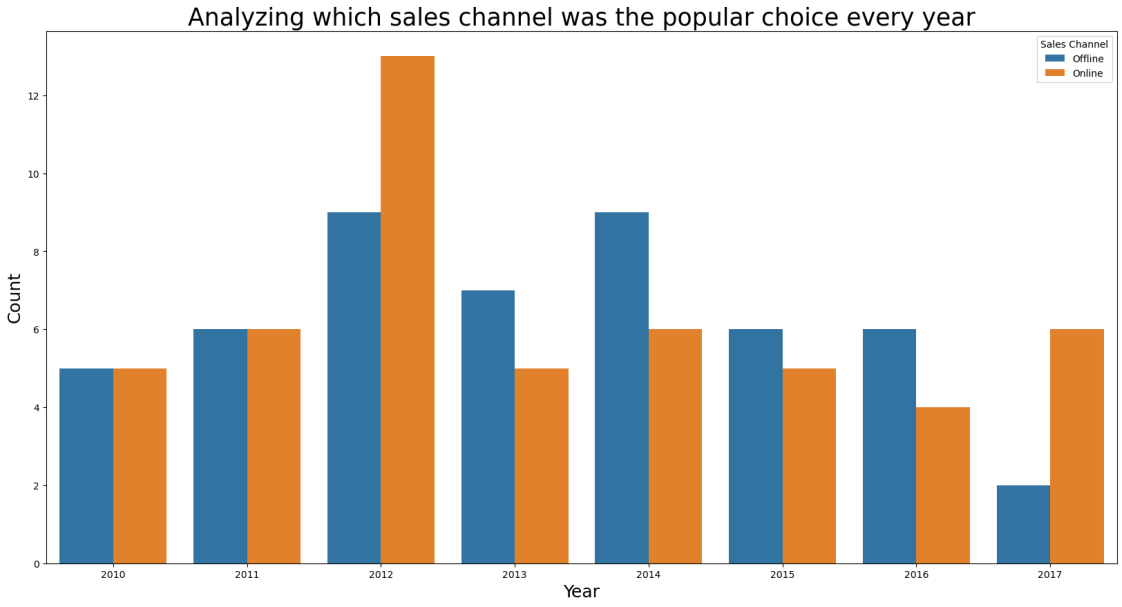
12. Clothes also produced zero revenue in year 2013. Rest of the years, it maintained similar values except for years 2011 and 2016 where it didn't do well.

13. Cereal produced zero revenue in years 2010, 2011 and 2015. Similar amount of revenue was generated in years 2012,2013,2014 and 2017. It didn't do well in year 2016.

14. Beverages generated zero revenue in years 2010, 2012, 2013 and 2017. Rest of the years, it generated very less revenue.

15. Baby food also generated zero revenue in years 2011, 2016 and 2017. Rest of the years, it did really well.

[29]:
```
plt.figure(figsize=(20,10))
sns.countplot(x='Year', hue='Sales Channel', data=data)

plt.title('Analyzing which sales channel was the popular choice every year',␣
 ↪fontsize=25)
plt.xlabel('Year', fontsize=18)
plt.ylabel('Count', fontsize=18)
plt.show()
```



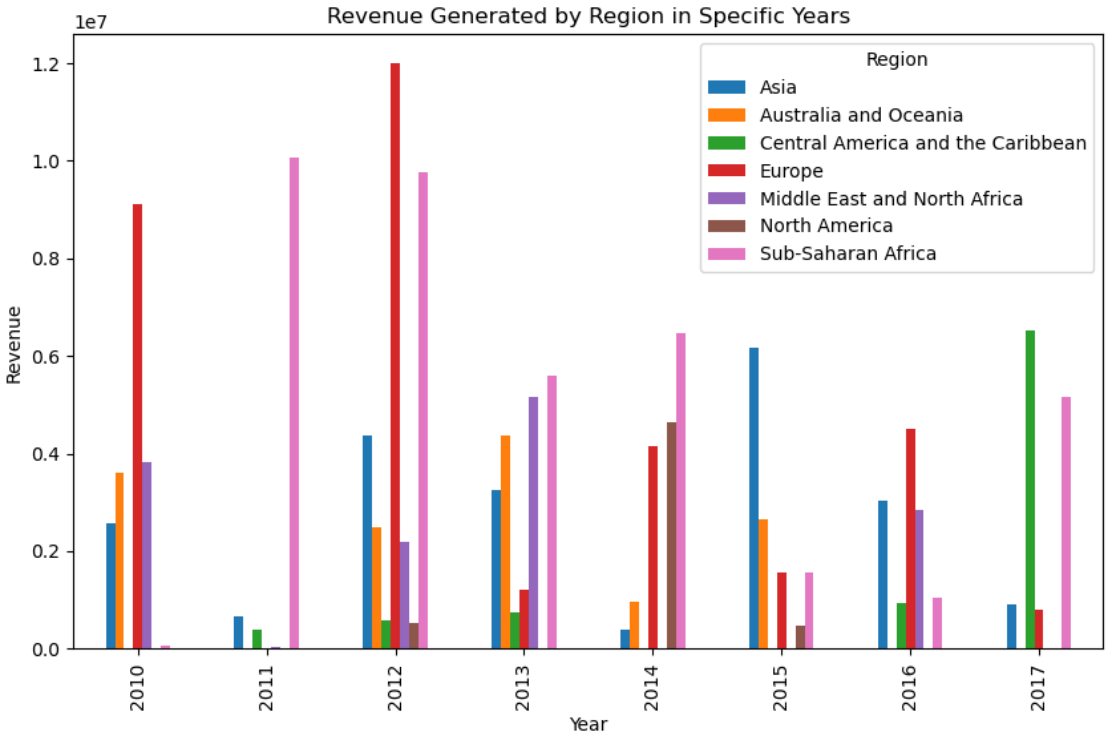
0.4.2 Year-wise analysis (SALES CHANNEL USED THROUGHOUT THE YEARS)

1. Overall, maximum times a sales channel used was Online mode, that too in year 2012 while the least number of times a sales channel used was Offline mode in year 2017.

2. Both the channels have been used significantly throughout the years in 2010-2017.

3. Offline mode was mostly or equally preferred in all years except in 2012 and 2017 where Online mode was used majorly.

[30]:
```
revenue_data = data.groupby(['Region', 'Year'])['Total Revenue'].sum().
↪reset_index()

pivot_data = revenue_data.pivot(index='Year', columns='Region', values='Total␣
↪Revenue')

pivot_data.plot(kind='bar', figsize=(10, 6))
plt.xlabel('Year')
plt.ylabel('Revenue')
plt.title('Revenue Generated by Region in Specific Years')
plt.legend(title='Region')
plt.show()
```



Region-wise Revenue generated throughout the years: 1. Sub-Saharan Africa generated significant large values of revenue in years 2011,2012,2013,2014 and 2017 while it was less in 2015-

2016. The worst was observed in 2010 where it was negligible. It was also able to generate the second highes value of revenue in 2011 followed by the third highest in 2012.

2. North America did well only in 2014 while very less in years 2012 and 2015. Rest of the years, it was either negligible or zero.

3. Middle East and North Africa region ccontributed with a nice value in 2010, 2012, 2013 and 2016. Rest of the years, it was either negligible or zero.

4. Europe was the one which generated the maximum revenue in year 2012 among all. The fourth highest revenue value was also earned in by this region in year 2010. Rest of the years the value was less except in 2011 where it was zero.

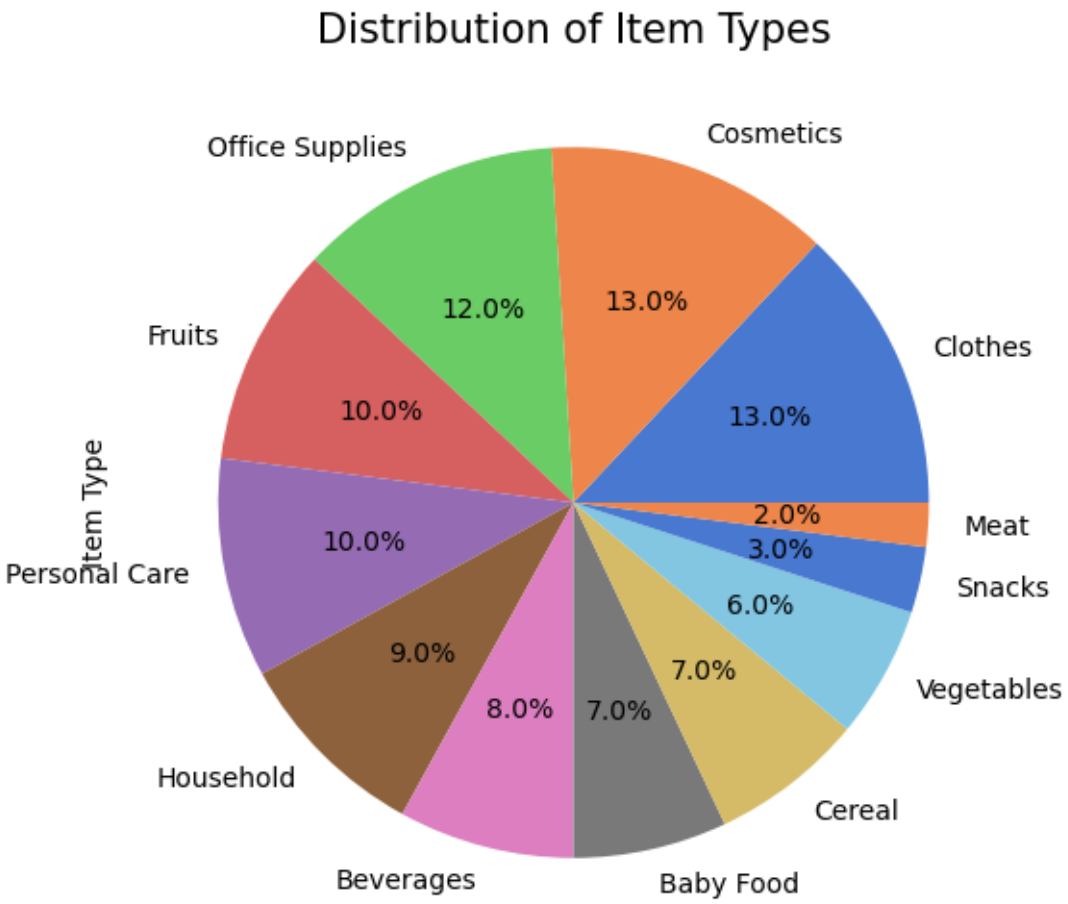
5. Central America and the Caribbean' s revenue value is either very less or zero throughout all years except for year 2017 where it did really well as compared to its history.

6. In Australia and Oceania, the total revenue value was good in 2010, 2012, 2013 and 2015. It was less in 2014 while the rest of the years, it was zero.

7. Asia was able to mark its presence in generating revenue every year. It was very less in 2011, 2014 and 2017.

8. Overall Europe and Sub-Saharan generated the highest revenue.

[31]:
```
plt.figure(figsize=[6,6])
data['Item Type'].value_counts().plot(kind='pie', colors= sns.
  ↪color_palette('muted'), autopct='%1.1f%%')
plt.title('Distribution of Item Types',fontsize=15)
plt.show()
```



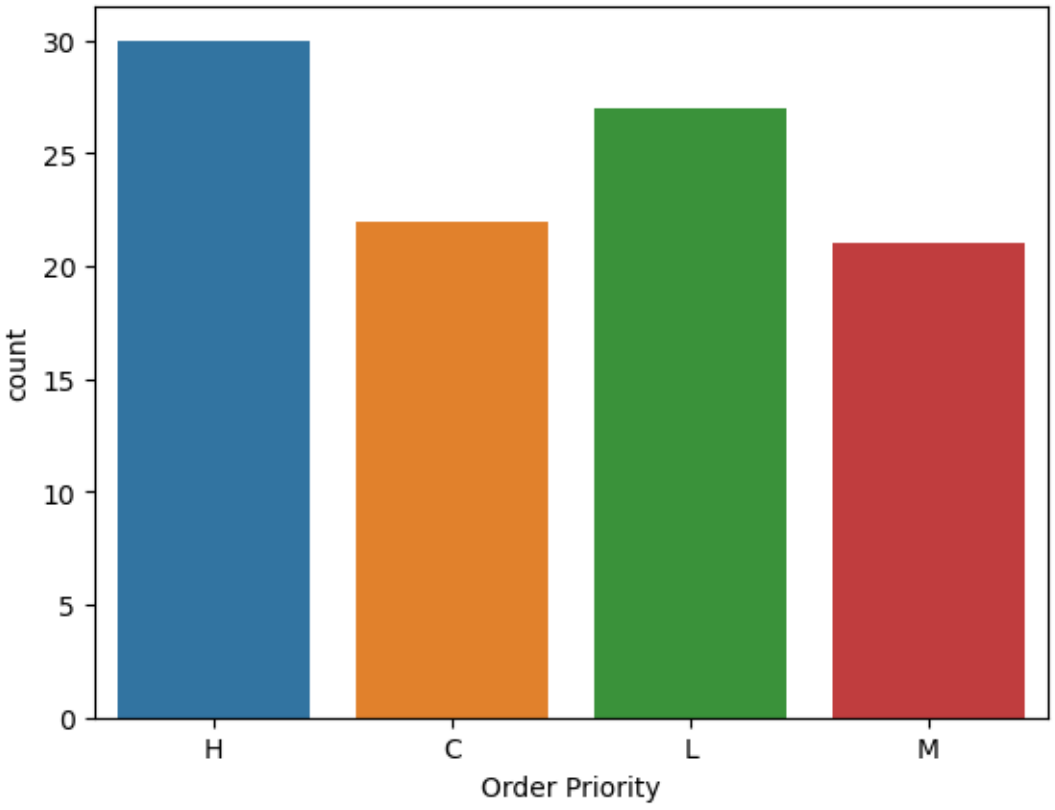
1. The piechart shows the distribution of type of items.

2. Among all the Item Types, other than Clothes and Cosmetics, most of the data is of the Fruits, Personal Care and household items while least is for meat section.

2. Rest of the items have considerable data.

[32]:
```
sns.countplot(data['Order Priority'])
```

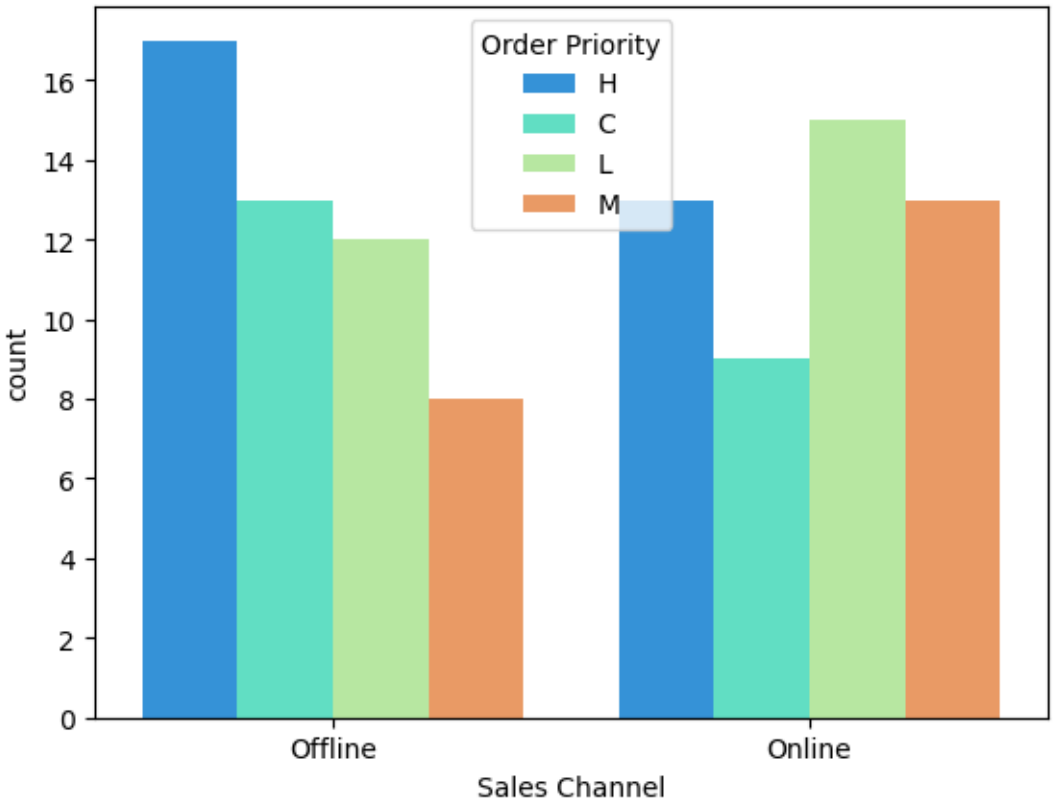
[32]:
```
<AxesSubplot:xlabel='Order Priority', ylabel='count'>
```



There are 4 order priority types: High, Medium, Low, and "C" where we have more data of High category.

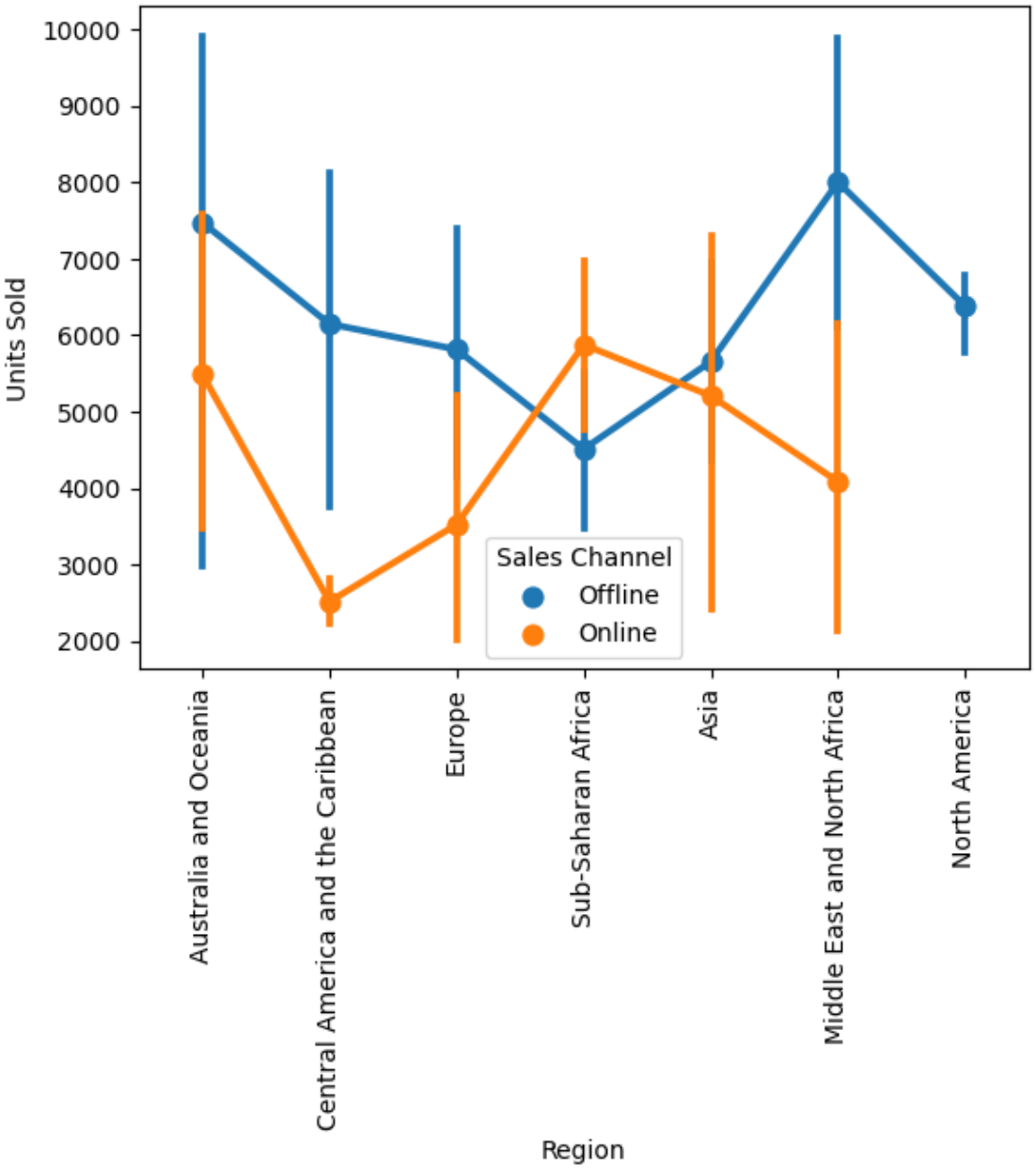
[33]:
```
sns.countplot(x='Sales Channel',hue='Order Priority', data=data,␣
↪palette='rainbow')
```

[33]:
```
<AxesSubplot:xlabel='Sales Channel', ylabel='count'>
```



The products of high order priority were purchased more in offline mode and that with medium order priority was given less preference in the same.

[34]:
```
sns.pointplot(x='Region',y='Units Sold', hue='Sales Channel', data=data)
plt.xticks(rotation=90)
plt.show()
```



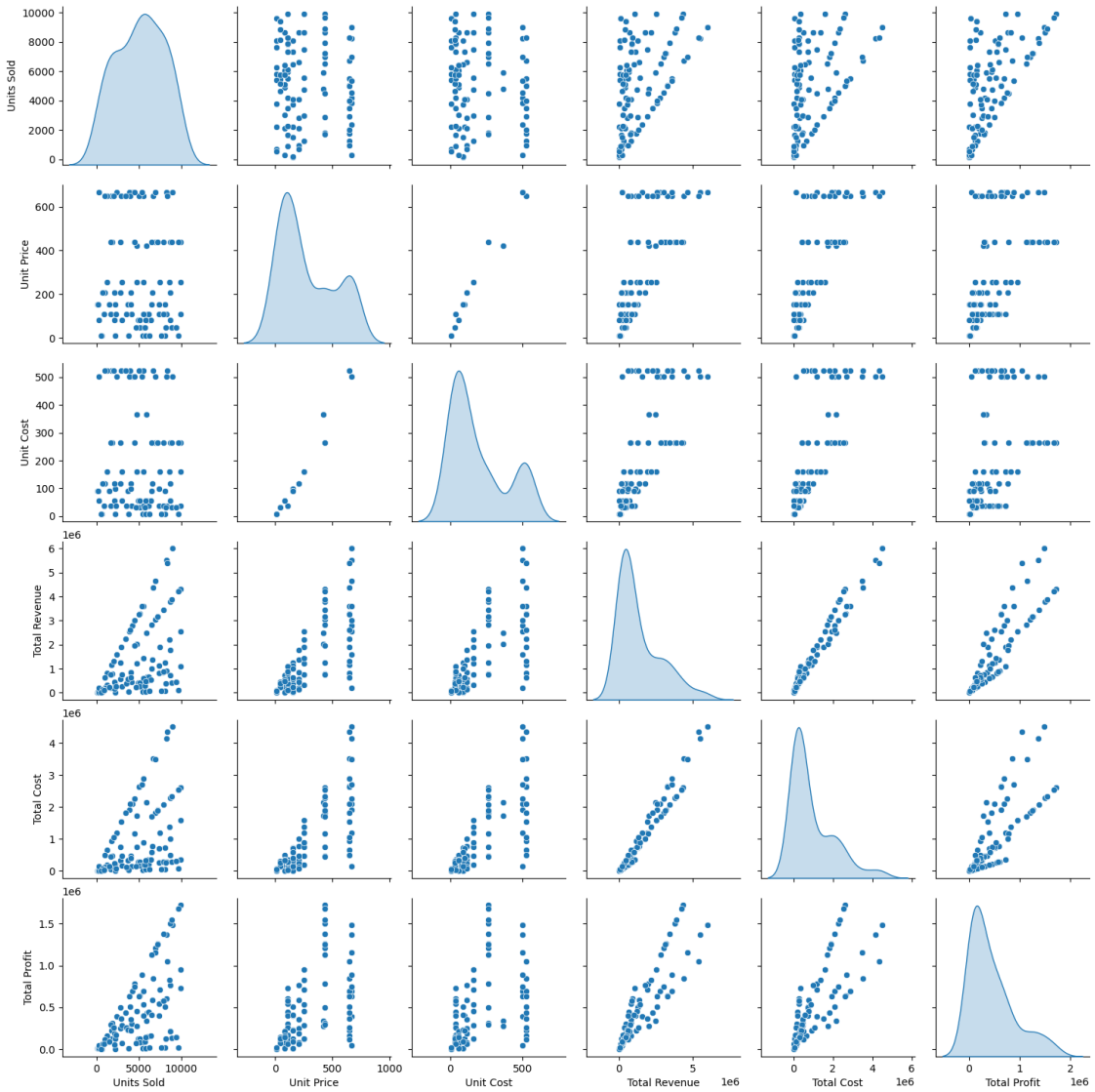
1. Australia and Oceania region have the maximum units wthin the range of 3000-10000 sold via both the sales channel modes.

2. Least units were sold in North America region only via offline mode. Customers didn't opt for online mode in this region.

3. Only Online mode of purchase was preferred the most in Asia in comparison to the offline mode.

[35]:
```
#understanding the distribution of single variables and the relationship␣
↪between two variables
```

```
sns.pairplot(data[numerical], kind= 'scatter', diag_kind= 'kde')
plt.show()
```



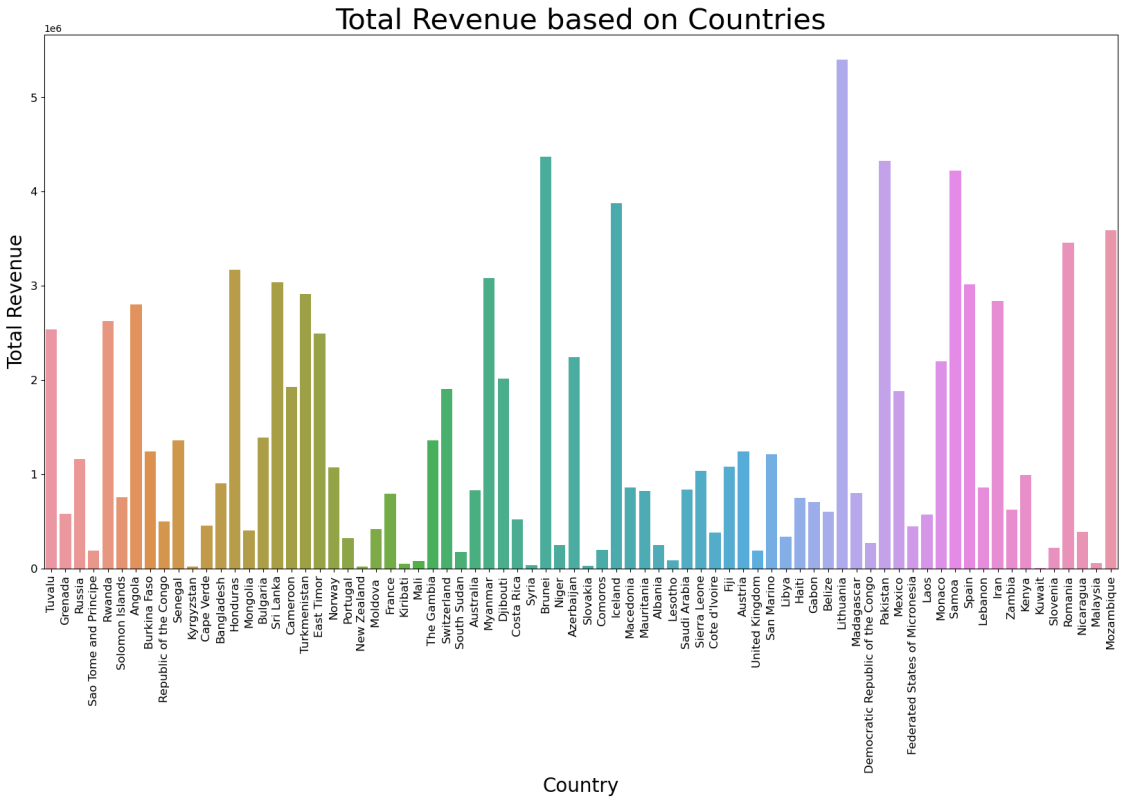
1. Total Revenue, Total Cost and Total Profit have rightly skewed distribution.

2. There is normal distribution obsereved in Units Sold variable while others have unsymmetric and almost rightly skewed distribution.

3. To understand the relationship between two variables, the scatter plots can be analysed.

[36]:
```
plt.figure(figsize=[20,10])

sns.barplot(x='Country',y='Total Revenue', data=data, ci=None)
```

```
plt.title('Total Revenue based on Countries', fontsize=30)
plt.xlabel('Country', fontsize=20)
plt.ylabel('Total Revenue', fontsize=20)
plt.xticks(fontsize=12, rotation=90)
plt.yticks(fontsize=12)
plt.show()
```

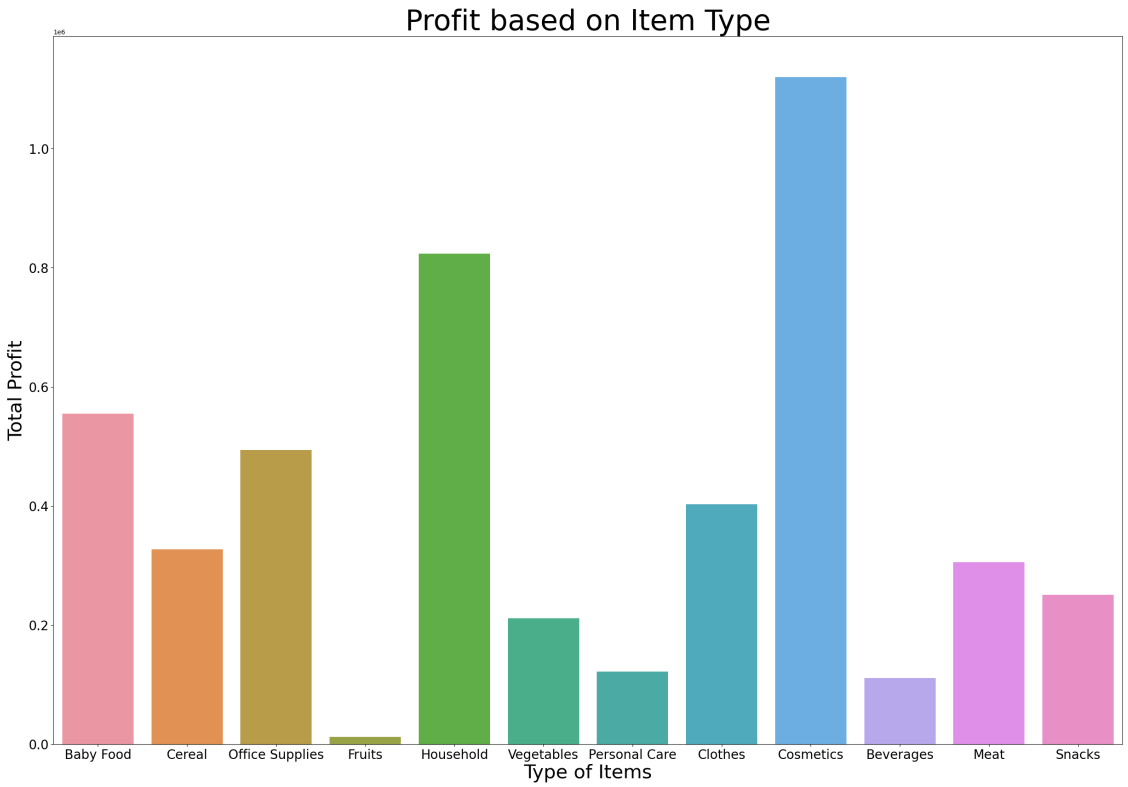


1. Maximum revenue was generated from Lithuania followed by Brunel, Pakistan, Samoa and Iceland respectively while negligible revenue from Kuwait.

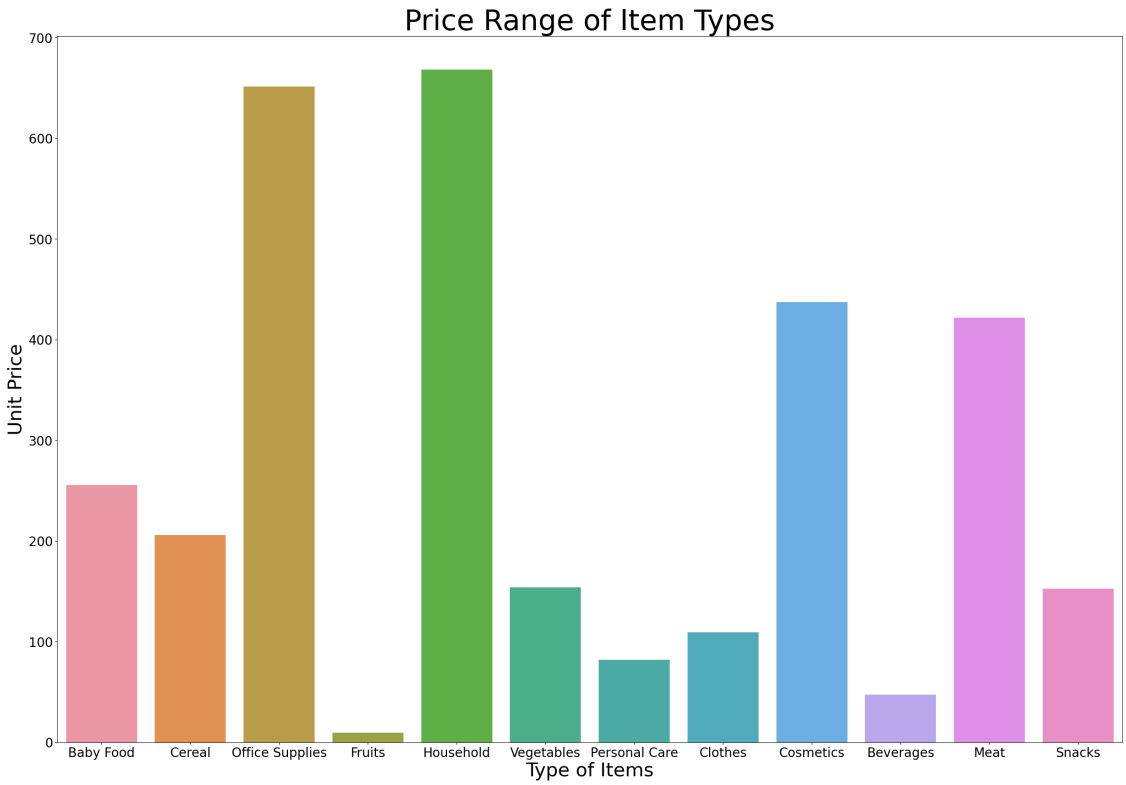
2. Minimal revenue(almost negligibile) was generated from Kyrgyzstan, New Zealand, Kiribati, Mali, Syria, Slovakia, Lesbotho and Malaysia as compared to others.

[37]:
```
plt.figure(figsize=[30,20])

sns.barplot(x=data['Item Type'], y=data['Total Profit'], ci=False)

plt.title('Profit based on Item Type', fontsize=45)
plt.xlabel('Type of Items', fontsize=30)
plt.ylabel('Total Profit', fontsize=30)
plt.xticks(fontsize=20)
```

```
plt.yticks(fontsize=20)
plt.show()
```



[38]:
```
plt.figure(figsize=[30,20])

sns.barplot(x=data['Item Type'], y=data['Unit Price'])

plt.title('Price Range of Item Types', fontsize=45)
plt.xlabel('Type of Items', fontsize=30)
plt.ylabel('Unit Price', fontsize=30)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.show()
```



1. Cosmetics having average unit price were able to generate the highest profit among all to Amazon.

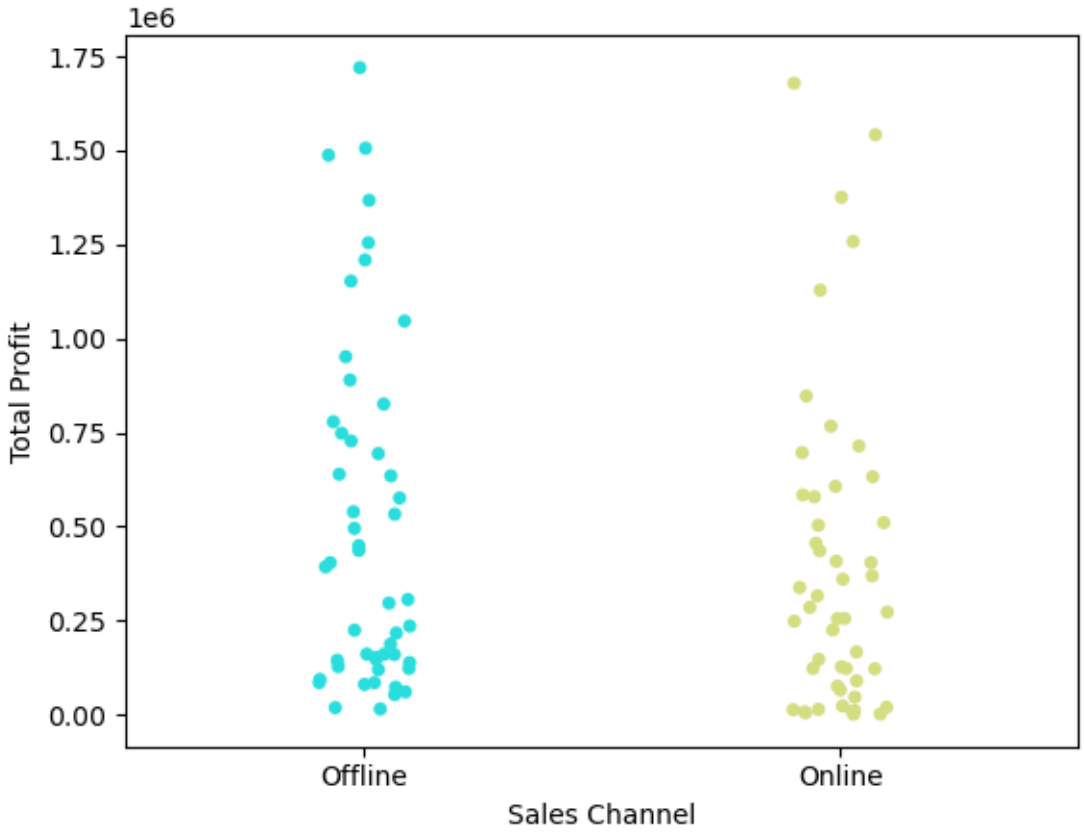
2. The Household items' unit prices are the highest (between 650-700). Having this, it was the second Item type having the most profit.

3. Fruits couldn't add a valuable amount to the profit though it's unit price is the least.

4. Office supplies having second highest unit price among all was able to generate near average value of profit.

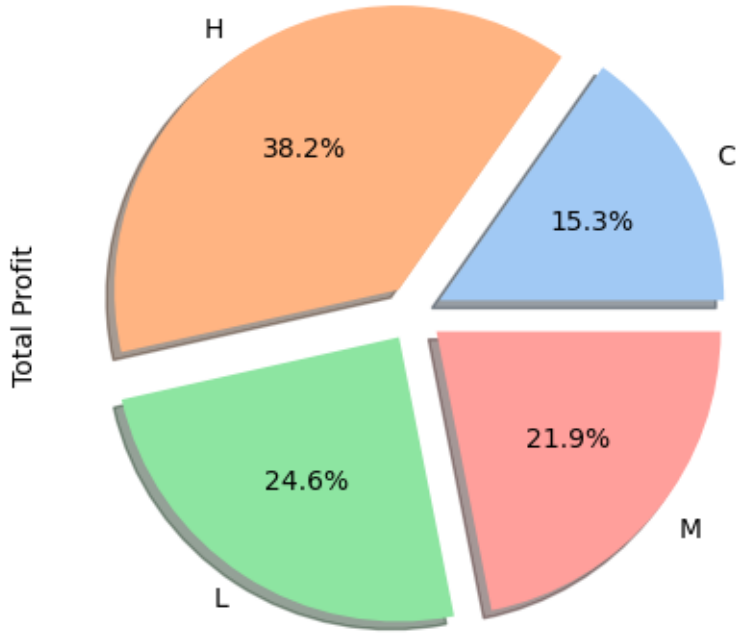
[39]:
```
sns.stripplot(x="Sales Channel", y="Total Profit", data=data, palette=␣
↪'rainbow')
```

[39]:
```
<AxesSubplot:xlabel='Sales Channel', ylabel='Total Profit'>
```



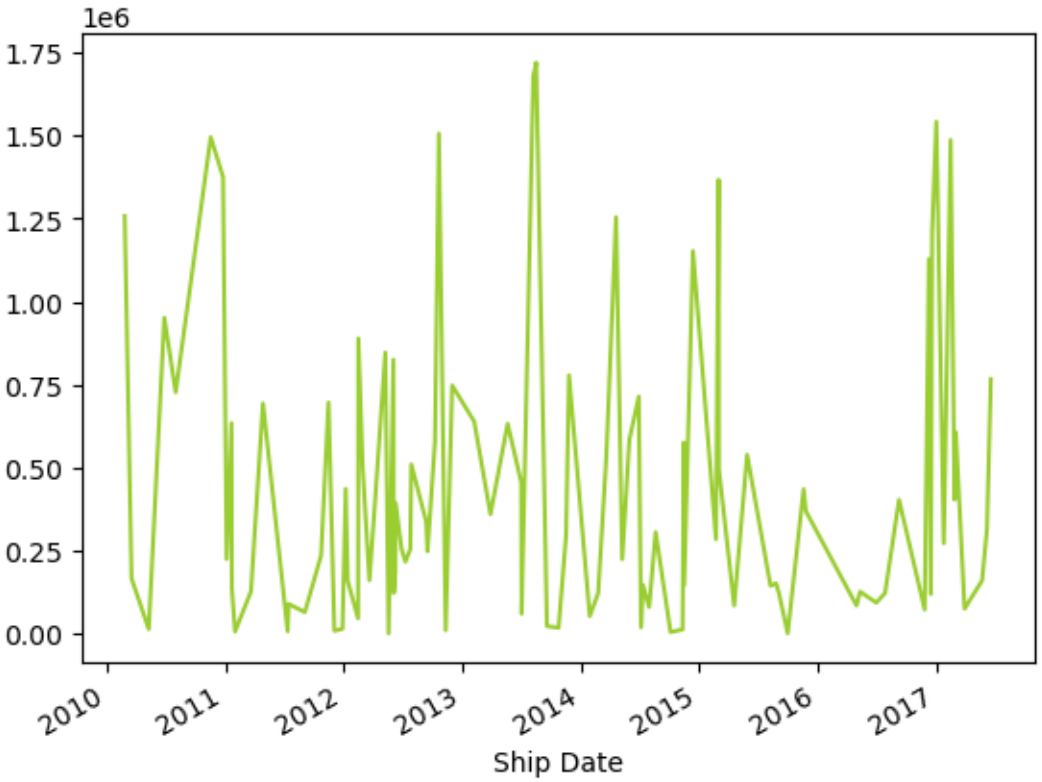
1. The highest profit was generated via offline mode.
2. Most profits were earned in offline mode and that too better than the online mode ones.

[40]:
```
explode=0.1,0.1,0.1,0.1

data.groupby('Order Priority')['Total Profit'].sum().
 ↪plot(kind='pie',autopct='%1.1f%%',explode=explode,
                                                    shadow=True, colors=␣
 ↪sns.color_palette('pastel'))
plt.show()
```



[41]:
```
data.groupby('Ship Date')['Total Profit'].sum().plot(kind='line',␣
↪color='yellowgreen')
```

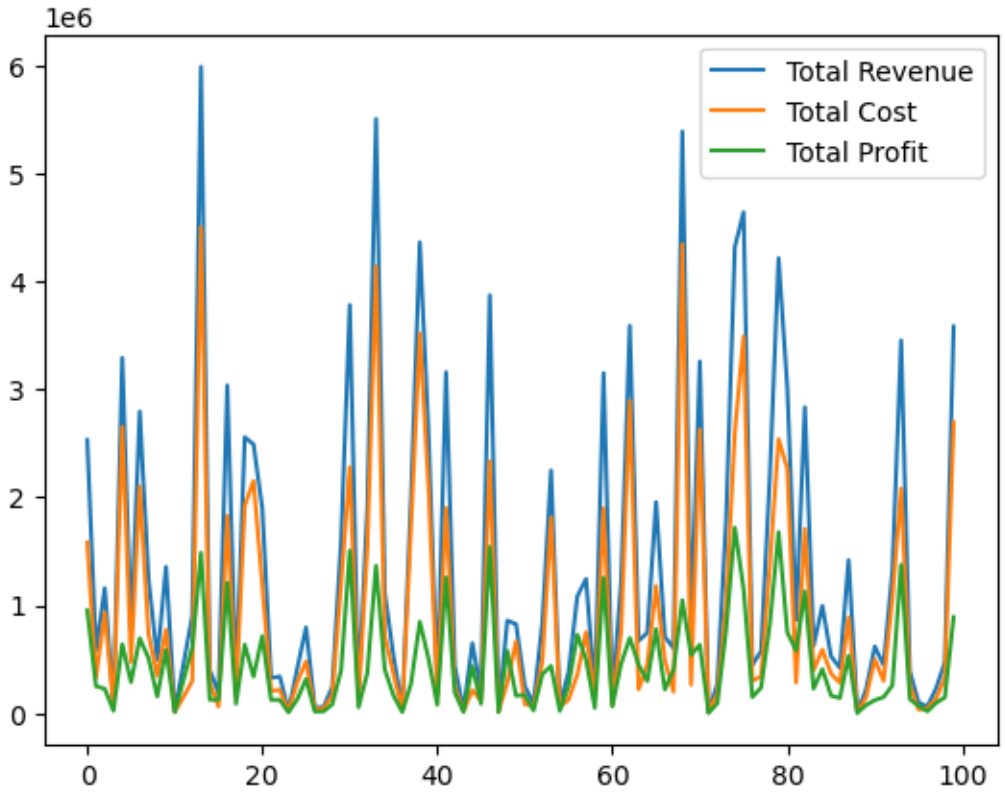
[41]: <AxesSubplot:xlabel='Ship Date'>



Maximum profit had been achieved by products having high order priority; between 2013 and 2014.

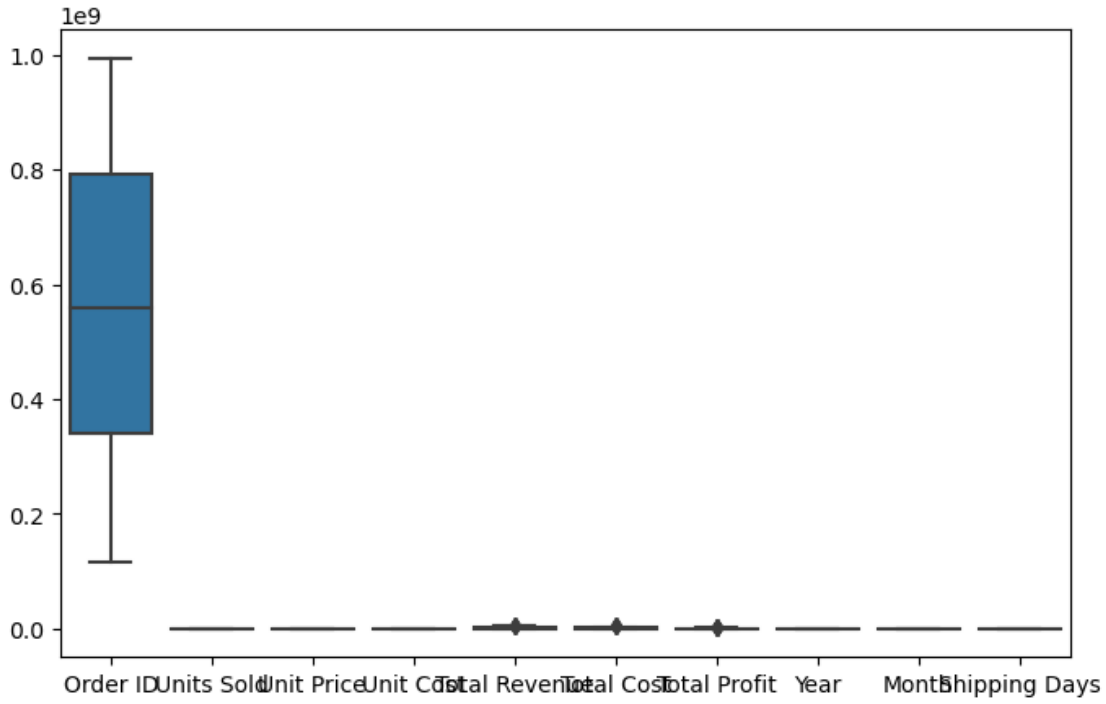
[42]:
```
cols=['Total Revenue','Total Cost','Total Profit']

plt.figure(figsize=(10,6))
data[cols].plot(kind='line')
plt.show()
```

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



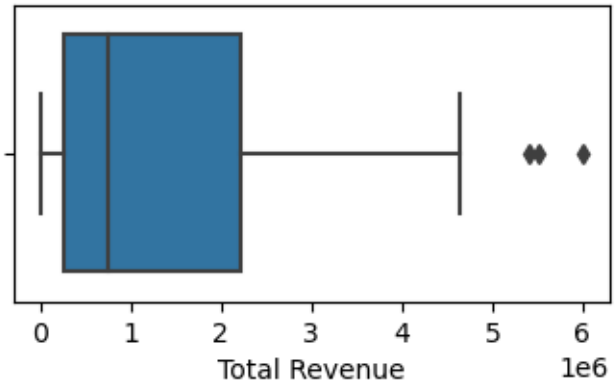
[43]:
```
plt.figure(figsize=[8,5])
sns.boxplot(data=data)
```

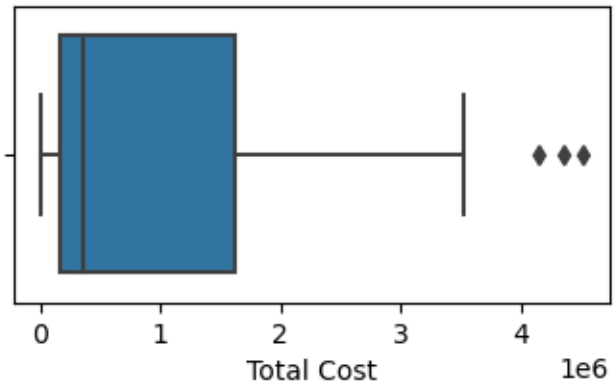
[43]: <AxesSubplot:>

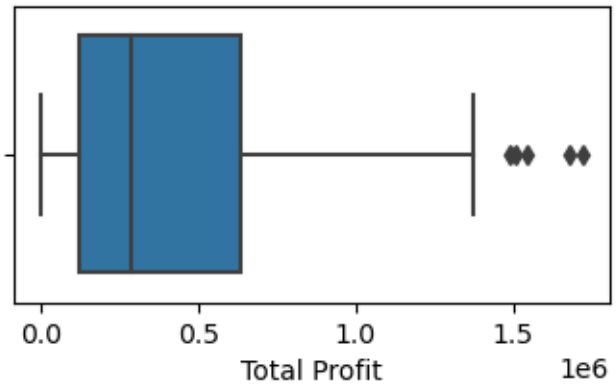


Total Revenue, Total Cost and Total Profit seems to have outliers. So let's dive into these.

[44]:
```
cols=['Total Revenue', 'Total Cost', 'Total Profit']

for col in cols:
    plt.figure(figsize=[4,2])
    sns.boxplot(data[col])
    plt.show()

plt.tight_layout()
```








```
<Figure size 640x480 with 0 Axes>
```

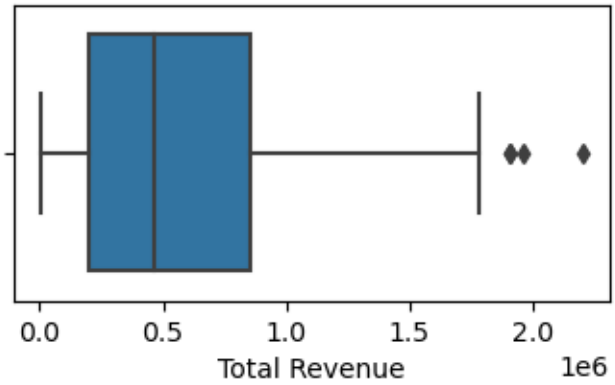
[50]:
```
Q1= data['Total Revenue'].quantile(0.25)
Q3= data['Total Revenue'].quantile(0.75)
IQR= Q3 - Q1

data= data[(data['Total Revenue']< Q3 + 1.5*IQR) & (data['Total Revenue'] > Q1␣
 ↪- 1.5*IQR) ]
```

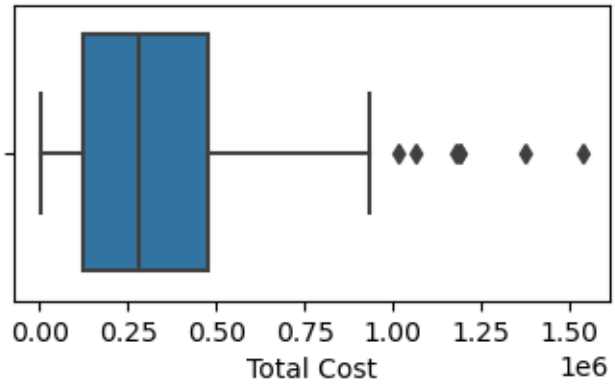
[51]:
```
Q1= data['Total Cost'].quantile(0.25)
Q3= data['Total Cost'].quantile(0.75)
IQR= Q3 - Q1

data= data[(data['Total Cost']< Q3 + 1.5*IQR) & (data['Total Cost'] > Q1 - 1.
 ↪5*IQR) ]
```

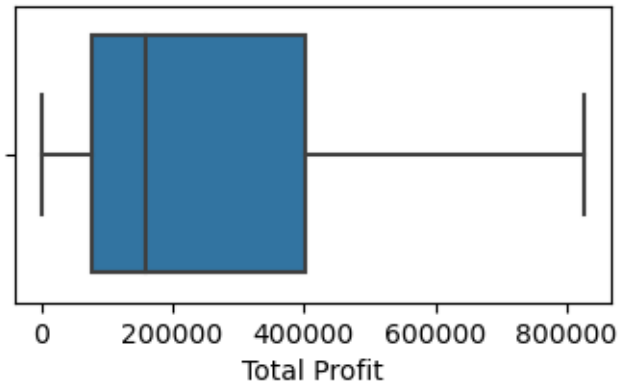
[52]:
```
Q1= data['Total Profit'].quantile(0.25)
Q3= data['Total Profit'].quantile(0.75)
```

```
IQR= Q3 - Q1

data= data[(data['Total Profit']< Q3 + 1.5*IQR) & (data['Total Profit'] > Q1 -␣
 ↪1.5*IQR) ]
```

[53]:
```
cols=['Total Revenue', 'Total Cost', 'Total Profit']

for col in cols:
    plt.figure(figsize=[4,2])
    sns.boxplot(data[col])
    plt.show()

plt.tight_layout()
```








```
<Figure size 640x480 with 0 Axes>
```

[54]:
```
data.shape
```

[54]:
```
(74, 18)
```

The boxplots shows that most of the outliers have been removed from the specifix 3 columns.

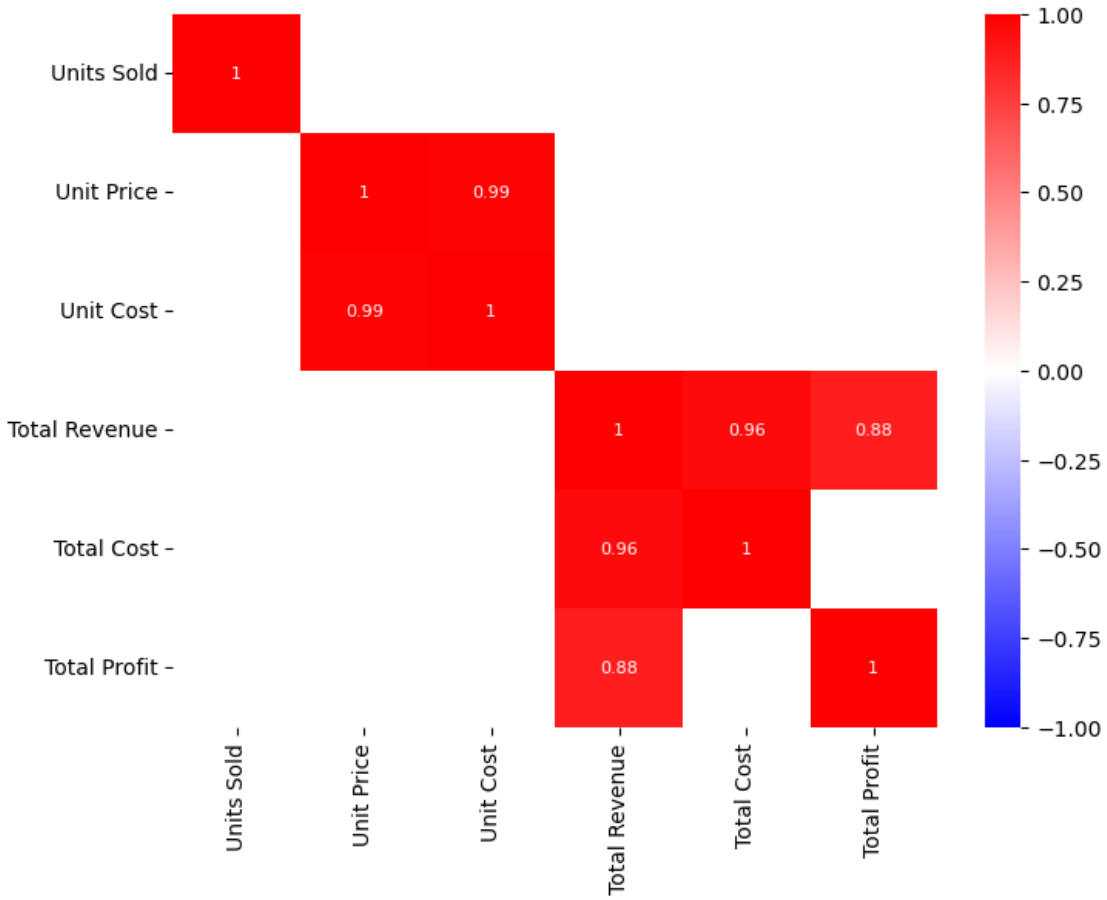
[55]:
```
# checking for correlation

corr= data[numerical].corr()
corr
```

[55]:
```
                Units Sold  Unit Price  Unit Cost  Total Revenue  Total Cost  \
Units Sold        1.000000   -0.415296  -0.408547       0.292825    0.168865
Unit Price       -0.415296    1.000000   0.985602       0.549331    0.655987
Unit Cost        -0.408547    0.985602   1.000000       0.494265    0.636038
Total Revenue     0.292825    0.549331   0.494265       1.000000    0.955131
Total Cost        0.168865    0.655987   0.636038       0.955131    1.000000
Total Profit      0.436250    0.269931   0.168804       0.877866    0.696633

                Total Profit
Units Sold          0.436250
Unit Price          0.269931
Unit Cost           0.168804
Total Revenue       0.877866
Total Cost          0.696633
Total Profit        1.000000
```

[56]:
```
plt.figure(figsize=[10,6])

sns.heatmap(corr, annot=True, vmax=1.0, vmin=-1.0, cmap='YlGnBu',␣
 ↪annot_kws={'size':8})
```

```
plt.show()
```

[57]:
```
plt.figure(figsize=[8,6])

sns.heatmap(corr[(corr>0.85) | (corr< -0.85)], annot=True, vmax=1.0, vmin=-1.0,␣
↪cmap='bwr', annot_kws={'size':8})
plt.show()
```



1. It can be observed that Unit Price and Unit Cost are highly correlated. Similary, Total Revenue and Total Cost are highly correlated.

2. Even Total Revenue and Total Profit are also highly correlated.

The monthly, yearly and monthly-yearly sales trend can be seen from above plottings and findings observed on the basis of each chart. For further analysis, tableau platform is used.

[ ]: