In [47]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.offline as py
import plotly.graph_objs as go
import plotly.tools as tls
from plotly.subplots import make_subplots
import plotly.figure_factory as ff
```

In [48]:
```
df =pd.read_csv('100 Sales Records.csv')
df.head()
```

Out[48]:

	Unnamed: 0	Region	Country	Item_Type	Sales_Channel	Order_Priority	Order_Date	Order_ID	Ship_Date	Units_Sold	Unit_Price	Unit_Cost	Total_Reven
0	0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	2533654.
1	1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	576782.
2	2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	2014-05-08	1779	651.21	524.96	1158502.
		Sub-	Sao						2014-07				

3	3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	2014-07-05	8102	9.33	6.92	75591.
4	4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-02-01	115456712	2013-02-06	5062	651.21	524.96	3296425.

In [49]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Unnamed: 0      100 non-null    int64  
 1   Region          100 non-null    object 
 2   Country         100 non-null    object 
 3   Item_Type       100 non-null    object 
 4   Sales_Channel   100 non-null    object 
 5   Order_Priority  100 non-null    object 
 6   Order_Date      100 non-null    object 
 7   Order_ID        100 non-null    int64  
 8   Ship_Date       100 non-null    object 
 9   Units_Sold      100 non-null    int64  
 10  Unit_Price      100 non-null    float64
 11  Unit_Cost       100 non-null    float64
 12  Total_Revenue   100 non-null    float64
```

```
 13  Total_Cost      100 non-null    float64
 14  Total_Profit    100 non-null    float64
dtypes: float64(5), int64(3), object(7)
memory usage: 11.8+ KB
```

In [50]:
```
total_Profit=df.Total_Profit.sum()
print('Total Profit:',total_Profit)
total_Revenue=df.Total_Revenue.sum()
print('Total Revenue:',total_Revenue)
```

```
Total Profit: 44168198.39999999
Total Revenue: 137348768.31
```

In [51]:
```
df['Order_Date']=pd.to_datetime(df['Order_Date'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Unnamed: 0      100 non-null    int64
 1   Region          100 non-null    object
 2   Country         100 non-null    object
 3   Item_Type       100 non-null    object
 4   Sales_Channel   100 non-null    object
 5   Order_Priority  100 non-null    object
 6   Order_Date      100 non-null    datetime64[ns]
```

```
 7   Order_ID        100 non-null    int64
 8   Ship_Date       100 non-null    object
 9   Units_Sold      100 non-null    int64
 10  Unit_Price      100 non-null    float64
 11  Unit_Cost       100 non-null    float64
 12  Total_Revenue   100 non-null    float64
 13  Total_Cost      100 non-null    float64
 14  Total_Profit    100 non-null    float64
dtypes: datetime64[ns](1), float64(5), int64(3), object(6)
memory usage: 11.8+ KB
```

In [52]:
```
df['Year']=df['Order_Date'].dt.year
df['Month']=df['Order_Date'].dt.month
df['day']=df['Order_Date'].dt.day
df
```

Out[52]:

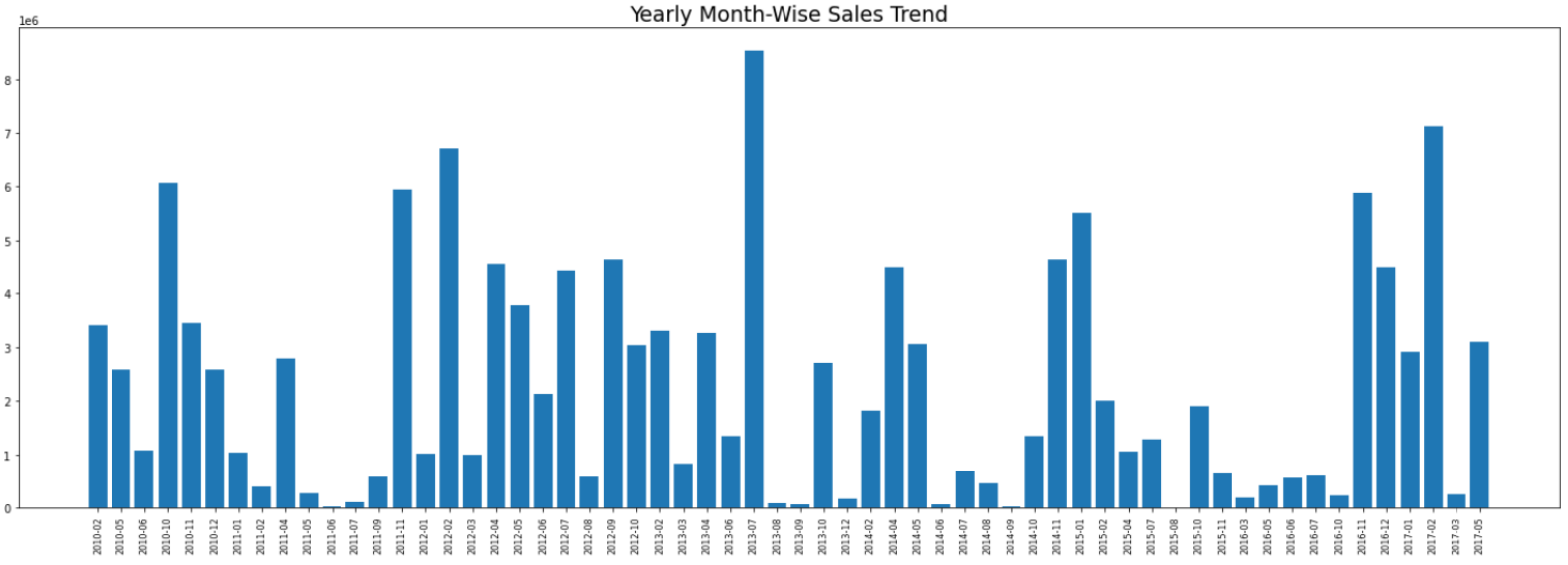
	Unnamed: 0	Region	Country	Item_Type	Sales_Channel	Order_Priority	Order_Date	Order_ID	Ship_Date	Units_Sold	Unit_Price	Unit_Cost	Total_R
0	0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	2010-05-28	669165933	2010-06-27	9925	255.28	159.42	253
1	1	Central America and the Caribbean	Grenada	Cereal	Online	C	2012-08-22	963881480	2012-09-15	2804	205.70	117.11	57
2	2	Europe	Russia	Office Supplies	Offline	L	2014-05-02	341417157	2014-05-08	1779	651.21	524.96	115

3	3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	2014-06-20	514321792	2014-07-05	8102	9.33	6.92	7
4	4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2013-02-01	115456712	2013-02-06	5062	651.21	524.96	329
...	...	...	...	...	...	...	...	...	...	...	...	...	
95	95	Sub-Saharan Africa	Mali	Clothes	Online	M	2011-07-26	512878119	2011-09-03	888	109.28	35.84	9
96	96	Asia	Malaysia	Fruits	Offline	L	2011-11-11	810711038	2011-12-28	6267	9.33	6.92	5
97	97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	2016-06-01	728815257	2016-06-29	1485	154.06	90.93	22
98	98	North America	Mexico	Personal Care	Offline	M	2015-07-30	559427106	2015-08-08	5767	81.73	56.67	47
99	99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2012-02-10	665095412	2012-02-15	5367	668.27	502.54	358

100 rows × 18 columns

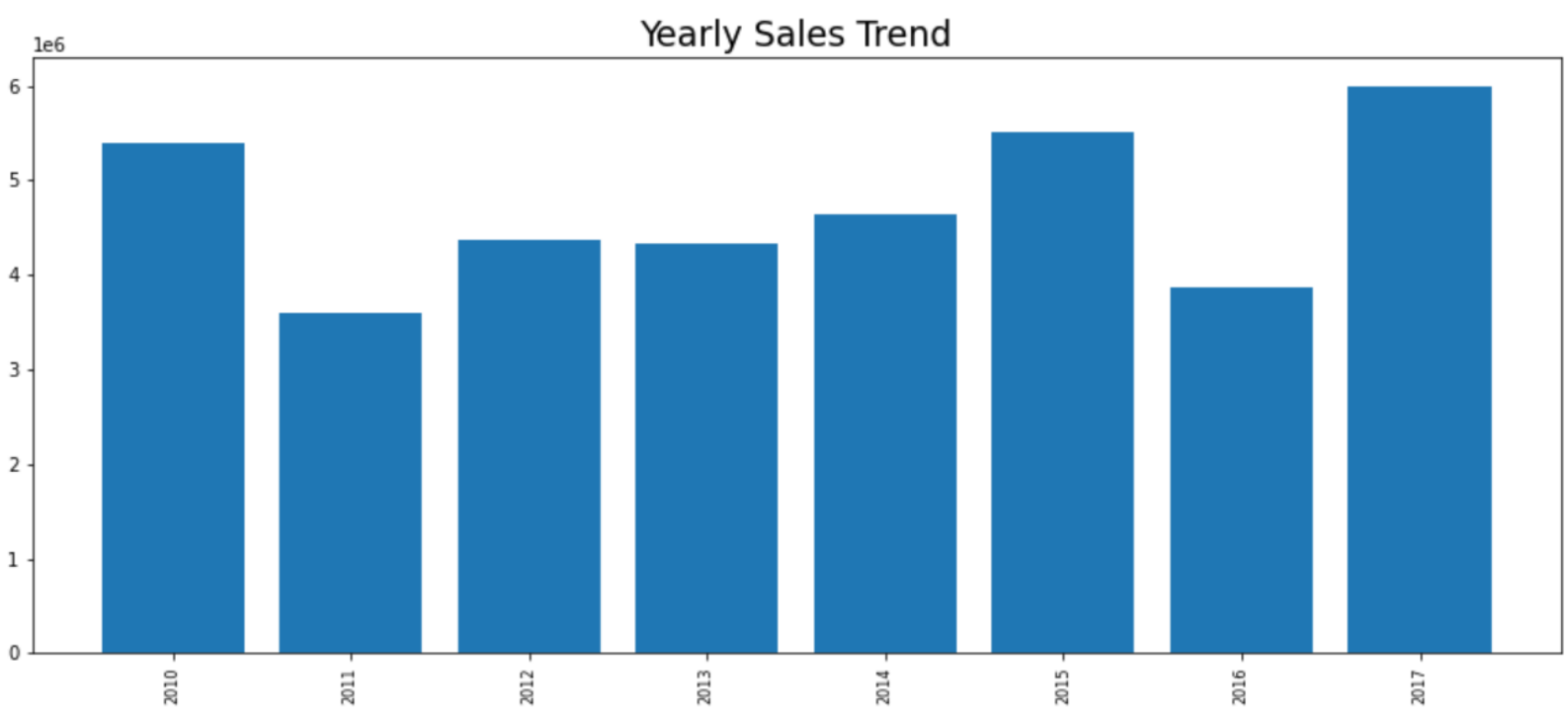
Yearly Month-Wise Sales Trend

In [53]:
```
df['month_year'] = df['Order_Date'].apply(lambda x: x.strftime('%Y-%m'))
month_year_ = df.groupby('month_year').sum()['Total_Revenue'].reset_index()
plt.figure(figsize=(25,8))
plt.title('Yearly Month-Wise Sales Trend',size=19)
plt.bar(month_year_['month_year'], month_year_['Total_Revenue'])
plt.xticks(rotation= 'vertical',size=8)
plt.show()
```



Yearly Sales Trend

In [54]:
```
plt.figure(figsize=(15,6))
plt.title('Yearly Sales Trend',size=19)
plt.bar(df['Year'], df['Total_Revenue'])
plt.xticks(rotation= 'vertical',size=8)
plt.show()
```


Yearly Sales Trend
1e6
6
5
4
3
2
1
0
2010
2011
2012
2013
2014
2015
2016
2017

Monthly Sales Trend

plt.figure(figsize=(15,6)) plt.title('Monthly Sales Trend',size=19) plt.bar(df['Month'], df['Total_Revenue']) plt.xticks(rotation= 'vertical',size=8) plt.show()

Find key metrics and factors and show the meaningful relationships between attributes.

In [55]:
```
#Grouping Item Column
Product_sale= pd.DataFrame(df.groupby('Item_Type').sum()['Total_Revenue'])
#Sorting Item_sales Column
Product_sale=Product_sale.sort_values('Total_Revenue',ascending=False)

#Top 10 Items By sales
Product_sale[:10]
```

Out[55]:

Item_Type	Total_Revenue
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

Top 10 selling products by sales amount by Descending:

1.Cosmetics

2.Office Supplies

3.Household

4.Baby Food

5.Clothes

6.Cereal

7.Meat

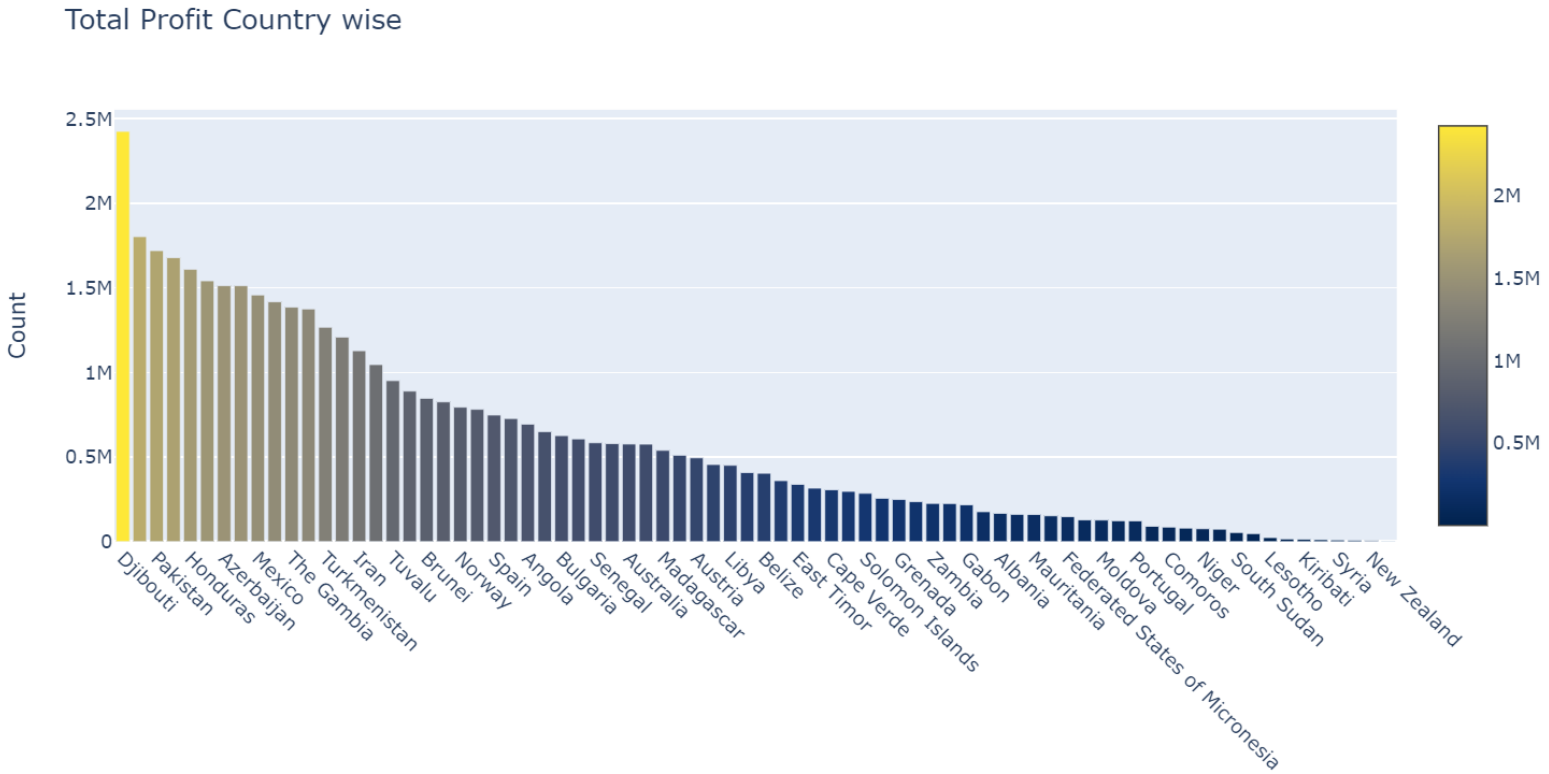
8.Personal Care

9.Vegetables

10.Beverages

Total Profit Country wise

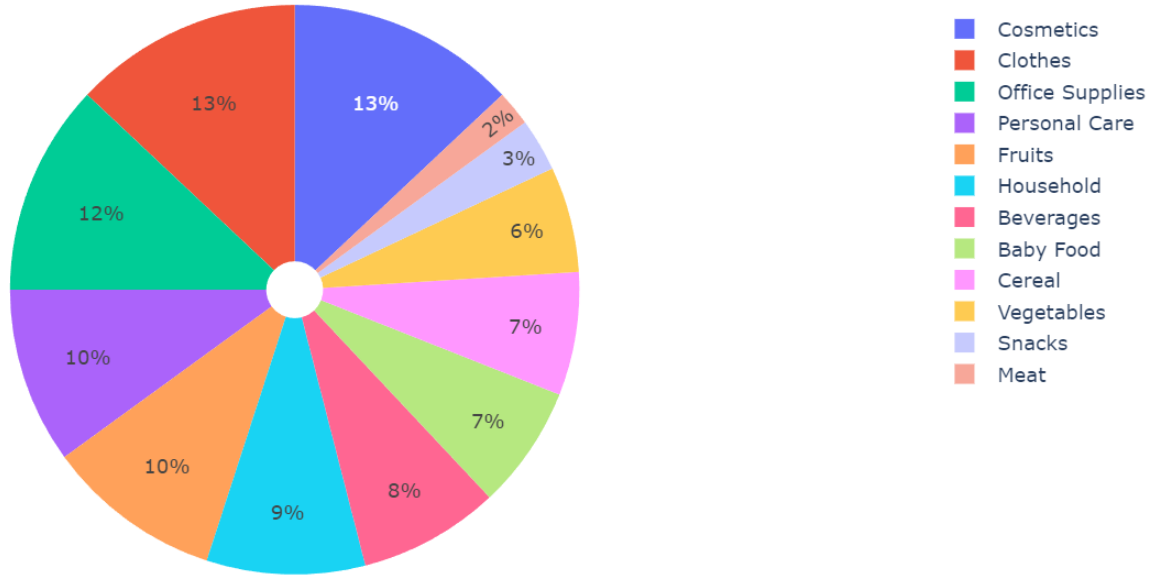
In [56]:
```
#TotalPrice = data.groupby(['Country'])['Pack Price'].sum().nlargest(15)
F=df.groupby('Country').Total_Profit.sum().to_frame().sort_values(by='Total_Profit', ascending=False)
trace=go.Bar(x=F.index, y=F.Total_Profit, marker=dict(color = F.Total_Profit, colorscale='Cividis',showscale=True))
ddata=[trace]
layout = go.Layout(xaxis=dict(tickangle=45),title='Total Profit Country wise', yaxis = dict(title = 'Count'))
fig = go.Figure(data=ddata, layout=layout)
py.iplot(fig)
```


Total Profit Country wise
Count
2.5M
2M
1.5M
1M
0.5M
0
Djibouti
Pakistan
Honduras
Azerbaijan
Mexico
The Gambia
Turkmenistan
Iran
Tuvalu
Brunei
Norway
Spain
Angola
Bulgaria
Senegal
Australia
Madagascar
Austria
Libya
Belize
East Timor
Cape Verde
Solomon Islands
Grenada
Zambia
Gabon
Albania
Mauritania
Federated States of Micronesia
Moldova
Portugal
Comoros
Niger
South Sudan
Lesotho
Kiribati
Syria
New Zealand
2M
1.5M
1M
0.5M

Distribution of Item Type

In [11]:
```
F=df['Item_Type'].value_counts().sort_values(ascending=False)
label=F.index
size=F.values
trace =go.Pie(labels=label, values=size, marker=dict(), hole=.1)
data_trace = [trace]
layout = go.Layout(title='Distribution of Item Type')
fig=go.Figure(data=data_trace,layout=layout)
py.iplot(fig)
```

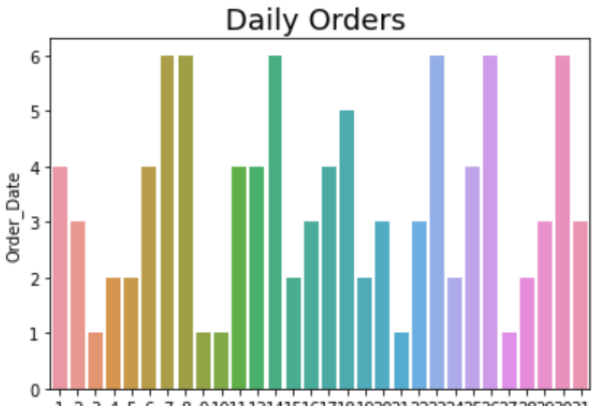
Distribution of Item Type



Daily Orders

In [12]:
```
Day = df['Order_Date'].dt.day.value_counts()
Day = Day.sort_index()
fig= plt.subplots(1)
sns.barplot(Day.index,Day).set_title("Daily Orders ", fontsize=18)
plt.show()
```

```
c:\Users\Pranesh Dorage\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and p
assing other arguments without an explicit keyword will result in an error or misinterpretation.
```



Monthly Orders

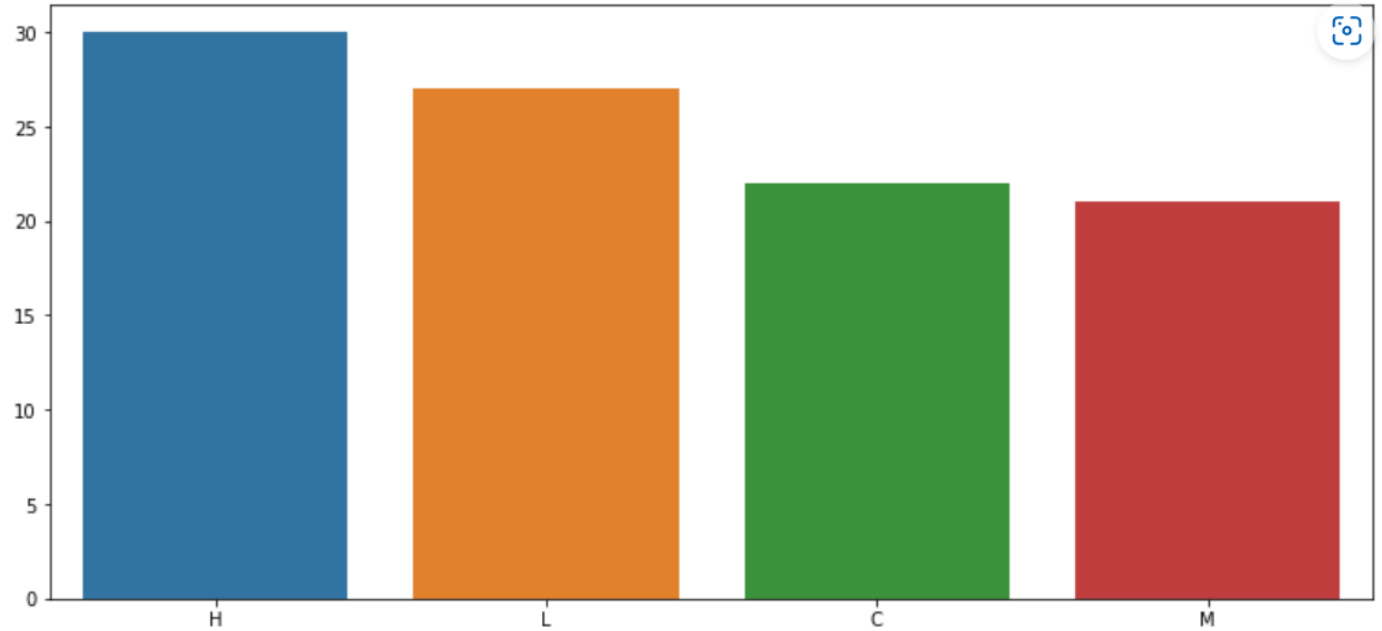
In [13]:
```
Month= df['Order_Date'].dt.month.value_counts()
Month= Month.sort_index()
fig = plt.subplots(1)
sns.barplot(Month.index,Month.values).set_title("Monthly Orders ", fontsize=18)
plt.show()
```

```
c:\Users\Pranesh Dorage\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and p
assing other arguments without an explicit keyword will result in an error or misinterpretation.
```

Priority Of Orders

In [14]:
```
F = df.Order_Priority.value_counts().sort_values(ascending=False)

plt.figure(figsize=(13,6))
sns.barplot(x=F.index,
            y=F.values)
```

Out[14]: <AxesSubplot:>



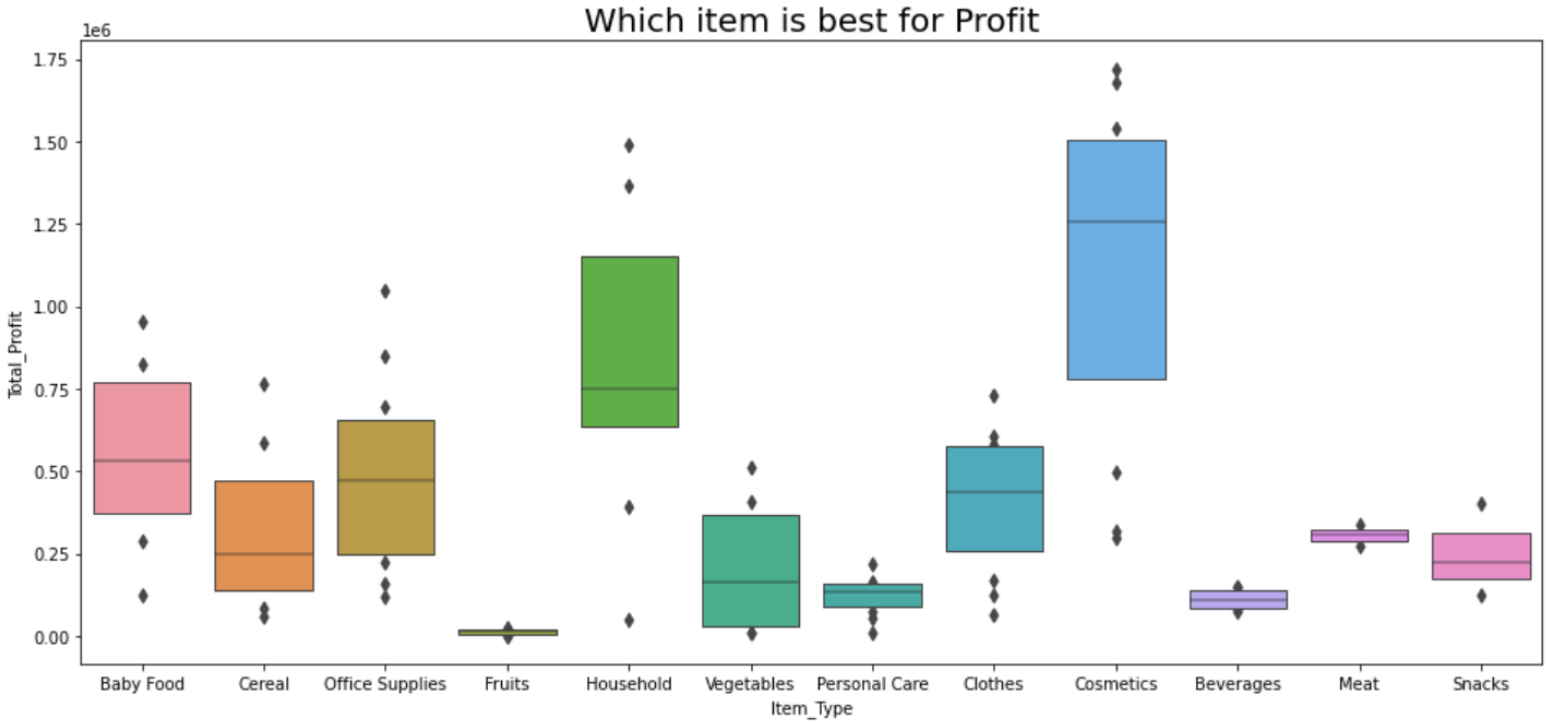
Which item is best for Profit?

Which item is best for Profit?

In [15]:
```
plt.figure(figsize=(16,7))
sns.boxenplot(x=df['Item_Type'], y=df['Total_Profit'])
plt.title("Which item is best for Profit",size=20)
```

Out[15]:
```
Text(0.5, 1.0, 'Which item is best for Profit')
```

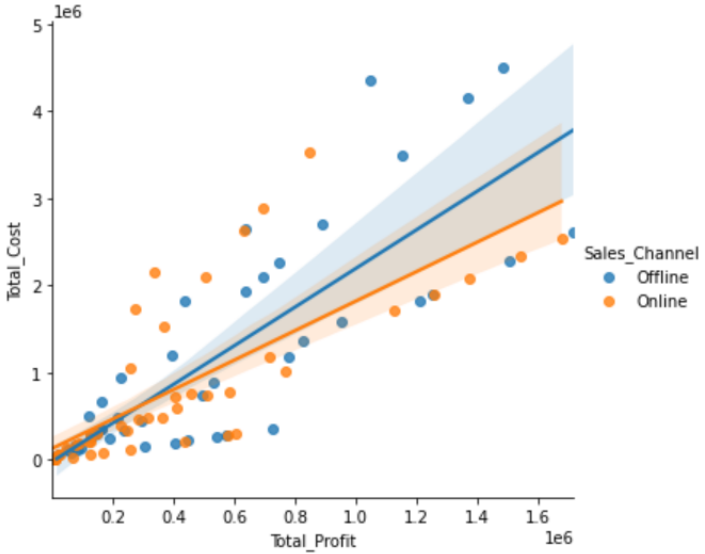
Out[15]: Text(0.5, 1.0, 'Which item is best for Profit')



Total Profit From Channel [Online or Offline]

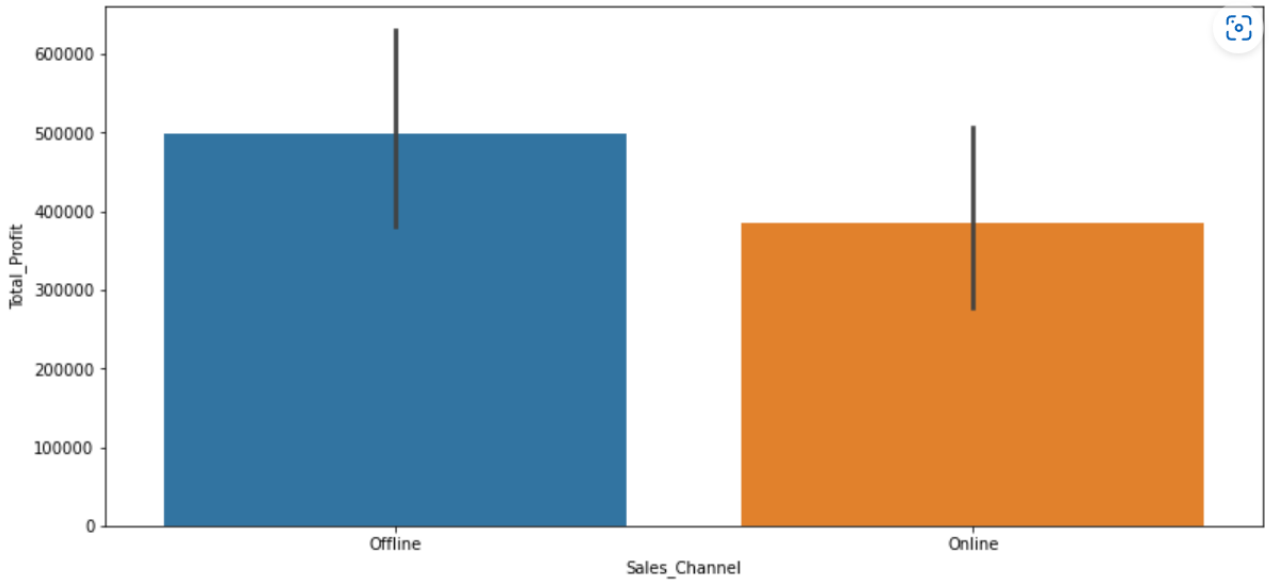
In [16]:
```
sns.lmplot(x="Total_Profit", y="Total_Cost", hue="Sales_Channel", data=df)
```

Out[16]:
```
<seaborn.axisgrid.FacetGrid at 0x1a772012a60>
```



In [17]:
```
plt.figure(figsize=(13,6))
sns.barplot(x=df['Sales_Channel'],
            y=df['Total_Profit'])
```

Out[17]: <AxesSubplot:xlabel='Sales_Channel', ylabel='Total_Profit'>

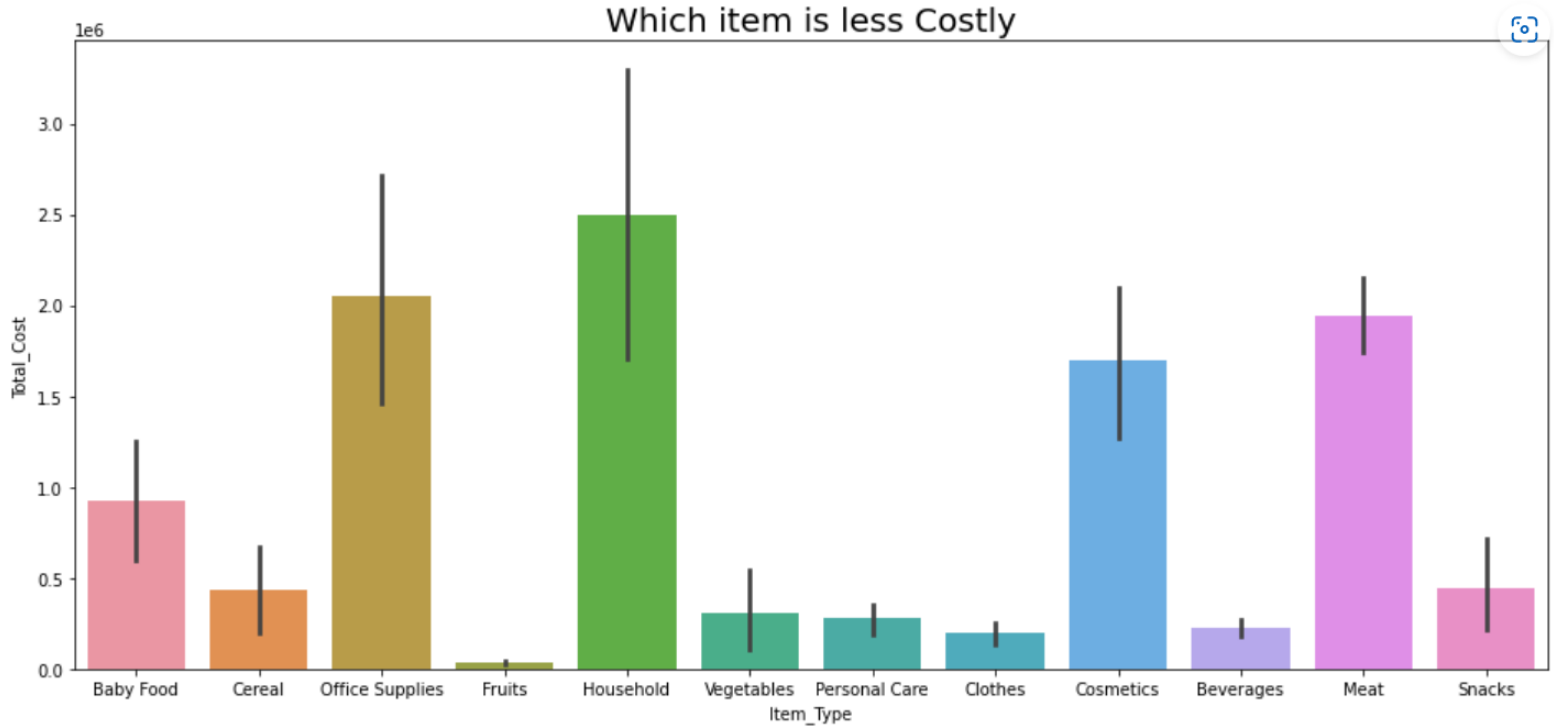


Which item is less Costly?

In [18]:
```
plt.figure(figsize=(16,7))
sns.barplot(x=df['Item_Type'], y=df['Total_Cost'])
plt.title("Which item is less Costly",size=20)
```

Out[18]:
```
Text(0.5, 1.0, 'Which item is less Costly')
```

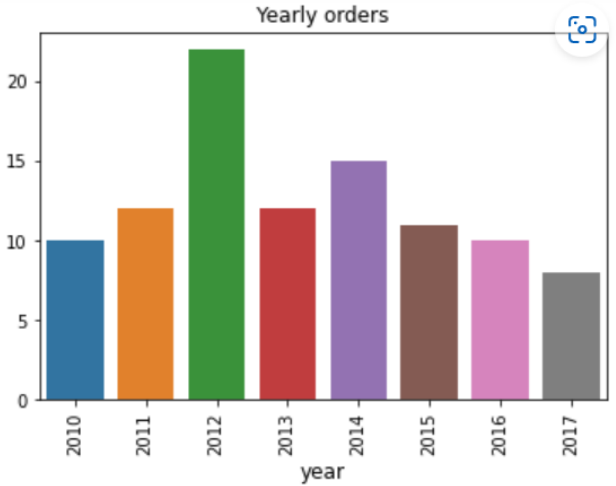
Out[18]: Text(0.5, 1.0, 'Which item is less Costly')



Yearly Orders

In [19]:
```
year = df['Order_Date'].dt.year.value_counts()
year= year.sort_index()
sns.barplot(year.index, year.values)
plt.xticks(rotation='vertical')
plt.xlabel('year', fontsize=12)
plt.ylabel('', fontsize=12)
plt.title("Yearly orders")
plt.show()
```

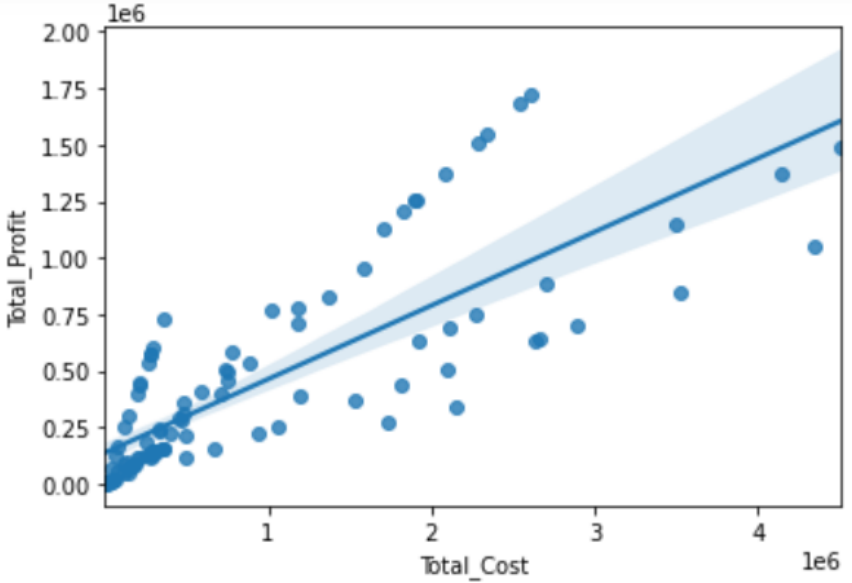
```
c:\Users\Pranesh Dorage\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and p
assing other arguments without an explicit keyword will result in an error or misinterpretation.
```



Total Profit

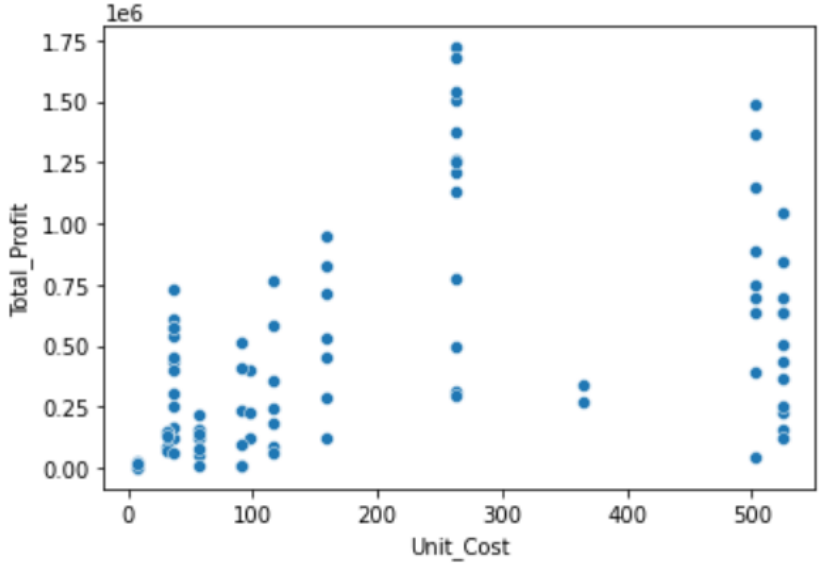
In [20]:
```
sns.regplot(x=df['Total_Cost'],y=df['Total_Profit'])
```

Out[20]:
```
<AxesSubplot:xlabel='Total_Cost', ylabel='Total_Profit'>
```



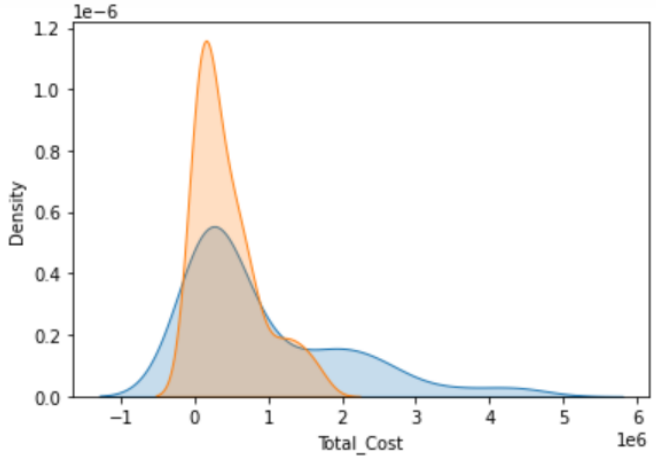
In [21]:
```
sns.scatterplot(x=df['Unit_Cost'],y=df['Total_Profit'])
```

Out[21]:
```
<AxesSubplot:xlabel='Unit_Cost', ylabel='Total_Profit'>
```



In [22]:
```
sns.kdeplot(data=df['Total_Cost'], label="DATA Cost", shade=True)
sns.kdeplot(data=df['Total_Profit'], label="DATA Profit", shade=True)
```

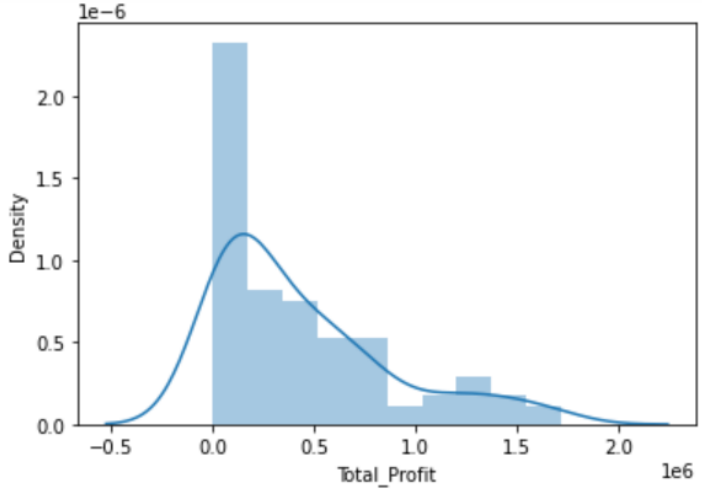
Out[22]: <AxesSubplot:xlabel='Total_Cost', ylabel='Density'>



In [23]:
```
sns.distplot(df['Total_Profit'],bins=10)
```

```
c:\Users\Pranesh Dorage\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
```

Out[23]: <AxesSubplot:xlabel='Total_Profit', ylabel='Density'>



Total Profit: 44168198.39999999

Total Revenue: 137348768.31