In [2]:
```python
# import the libraries we required
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.simplefilter('ignore')
```

In [3]:
```python
#  load the dataset
df=pd.read_csv('UM/Amazon Sales Data.csv')
df
```

Out[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	
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	9
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	2
2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1158502.59	933903.84	2
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	7/5/2014	8102	9.33	6.92	75591.66	56065.84	
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.96	3296425.02	2657347.52	6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	9/3/2011	888	109.28	35.84	97040.64	31825.92	
96	Asia	Malaysia	Fruits	Offline	L	11/11/2011	810711038	12/28/2011	6267	9.33	6.92	58471.11	43367.64	
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	6/1/2016	728815257	6/29/2016	1485	154.06	90.93	228779.10	135031.05	
98	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	8/8/2015	5767	81.73	56.67	471336.91	326815.89	1
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2/10/2012	665095412	2/15/2012	5367	668.27	502.54	3586605.09	2697132.18	8

100 rows × 14 columns

In [4]:
```python
# check the brief info of the dataset
df.head()
```

Out[4]:

	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	9.33	6.92	75591.66	56065.84	19525.82
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.96	3296425.02	2657347.52	639077.50

In [5]:
```python
# ckeck the bottom 5 records
df.tail()
```

Out[5]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	
95	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	9/3/2011	888	109.28	35.84	97040.64	31825.92	6
96	Asia	Malaysia	Fruits	Offline	L	11/11/2011	810711038	12/28/2011	6267	9.33	6.92	58471.11	43367.64	1
97	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	6/1/2016	728815257	6/29/2016	1485	154.06	90.93	228779.10	135031.05	9
98	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	8/8/2015	5767	81.73	56.67	471336.91	326815.89	14
99	Sub-Saharan Africa	Mozambique	Household	Offline	L	2/10/2012	665095412	2/15/2012	5367	668.27	502.54	3586605.09	2697132.18	88

In [6]:

```
# check the basic info of the dataset
df.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
```

In [7]:

```
# dividing the features based on their datatypes
continuous_features=[]
categorical_features=[]
continuous_or_discrete_count=[]
for i in df.columns:
    if df[i].dtypes=='float64':
        continuous_features.append(i)
    elif df[i].dtypes=='object':
        categorical_features.append(i)
    else:
        continuous_or_discrete_count.append(i)
print('continuous_features:',continuous_features)
print('categorical_features:',categorical_features)
print('continuous_or_discrete:',continuous_or_discrete_count)
```

```
continuous_features: ['Unit Price', 'Unit Cost', 'Total Revenue', 'Total Cost', 'Total Profit']
categorical_features: ['Region', 'Country', 'Item Type', 'Sales Channel', 'Order Priority', 'Order Date', 'Ship
Date']
continuous_or_discrete: ['Order ID', 'Units Sold']
```

In [8]:

```
# shape of the dataset
df.shape
```

Out[8]:

```
(100, 14)
```

In [9]:

```
# index of the dataset
df.index
```

Out[9]:

```
RangeIndex(start=0, stop=100, step=1)
```

In [10]:

```
# check the columns of the dataset
df.columns
```

Out[10]:

```
Index(['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'],
      dtype='object')
```

In [11]:

```
# check the sales channel unique values
df['Sales Channel'].unique()
```

Out[11]:

```
array(['Offline', 'Online'], dtype=object)
```

In [12]:

```
#  sales channel value counts
```

```
df['Sales Channel'].value_counts()
```

Out[12]:

```
Offline    50
Online     50
Name: Sales Channel, dtype: int64
```

In [12]:

```
# check the duplicated record
df.duplicated().sum()
```

Out[12]:

```
0
```

There is no duplicated records

In [13]:

```
# check the null values
df.isnull().sum()
```

Out[13]:

```
Region            0
Country           0
Item Type         0
Sales Channel     0
Order Priority    0
Order Date        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
dtype: int64
```

There is no null values

In [26]:

```
# Total profit wise top 5 countries
df.sort_values(by='Total Profit',ascending=False,ignore_index=True).head()
```

Out[26]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Tota
0	Middle East and North Africa	Pakistan	Cosmetics	Offline	L	7/5/2013	231145322	8/16/2013	9892	437.20	263.33	4324782.40	2604860.36	171
1	Australia and Oceania	Samoa	Cosmetics	Online	H	7/20/2013	670854651	8/7/2013	9654	437.20	263.33	4220728.80	2542187.82	167
2	Europe	Iceland	Cosmetics	Online	C	12/31/2016	331438481	12/31/2016	8867	437.20	263.33	3876652.40	2334947.11	154
3	Europe	Switzerland	Cosmetics	Offline	M	9/17/2012	249693334	10/20/2012	8661	437.20	263.33	3786589.20	2280701.13	150
4	Central America and the Caribbean	Honduras	Household	Offline	H	2/8/2017	522840487	2/13/2017	8974	668.27	502.54	5997054.98	4509793.96	148

In [36]:

```
# Profit wise regions in descending order
df.groupby('Region')['Total Profit'].sum().sort_values(ascending=False)
```

Out[36]:

```
Region
Sub-Saharan Africa                   12183211.40
Europe                               11082938.63
Asia                                  6113845.87
Middle East and North Africa          5761191.86
Australia and Oceania                 4722160.03
Central America and the Caribbean     2846907.85
North America                         1457942.76
Name: Total Profit, dtype: float64
```

In [37]:

```
# region wise total revenue by using group by
df.groupby('Region')['Total Revenue'].sum().sort_values(ascending=False)
```

Out[37]:

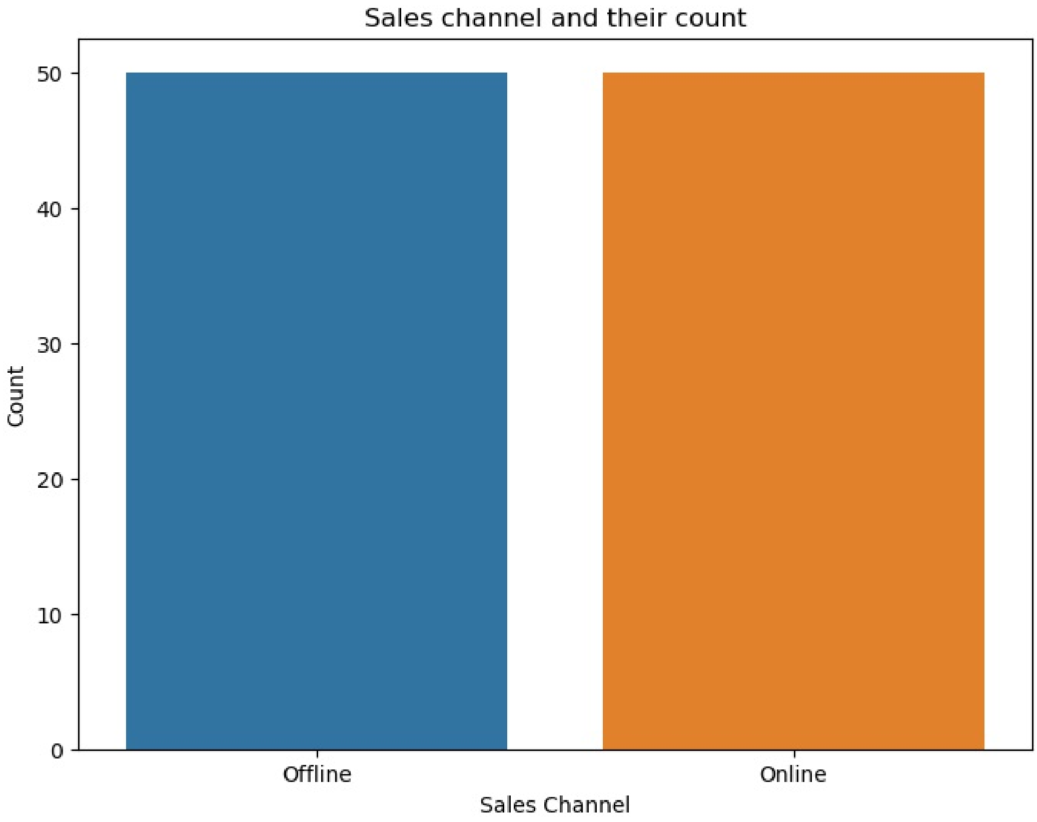
```
Region
Sub-Saharan Africa                   39672031.43
Europe                               33368932.11
Asia                                 21347091.02
Australia and Oceania                14094265.13
Middle East and North Africa         14052706.58
Central America and the Caribbean     9170385.49
North America                         5643356.55
Name: Total Revenue, dtype: float64
```

In [39]:

```
# region wise value counts
df['Region'].value_counts()
```

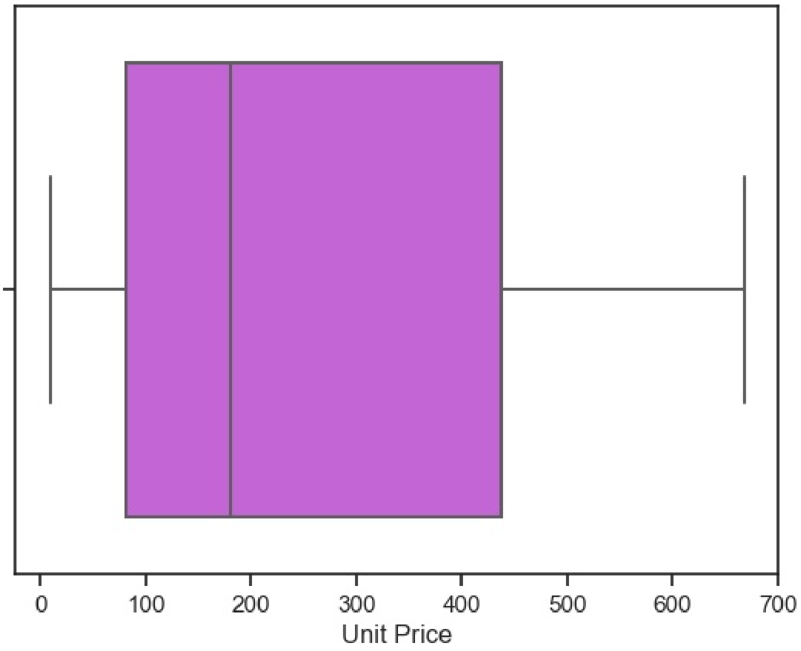
Out[39]:
```
Sub-Saharan Africa                   36
Europe                               22
Australia and Oceania                11
Asia                                 11
Middle East and North Africa         10
Central America and the Caribbean     7
North America                         3
Name: Region, dtype: int64
```

In [27]:
```
# countplot for the sales channel feature
plt.figure(figsize=(8,6))
sns.countplot(x='Sales Channel',data=df,edgecolor='linen',alpha=0.7,)
plt.title('Sales channel and their count')
plt.xlabel('Sales Channel')
plt.ylabel('Count')
plt.show()
```

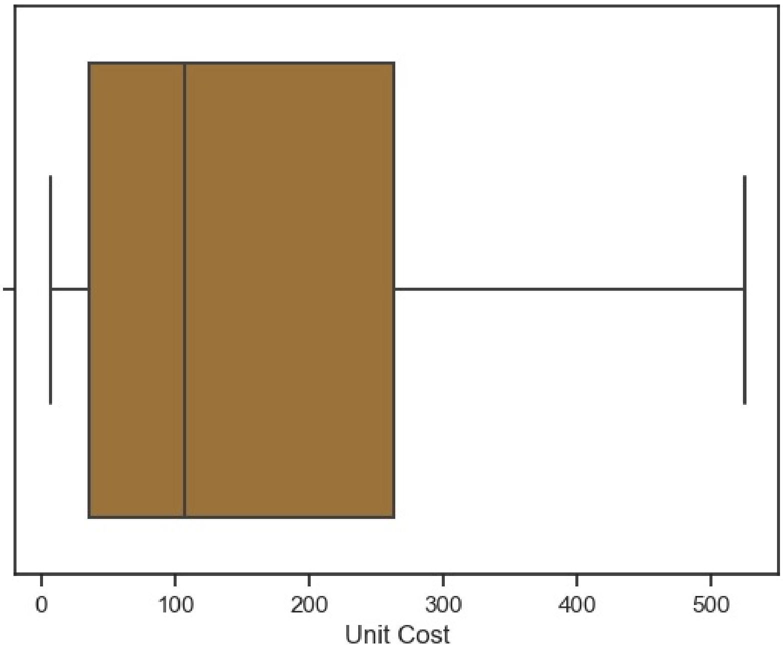


In [54]:
```
# check the outliers are present in the dataset by using boxplot
sns.set_theme(style="ticks")
for i in continuous_features:
    print(f'\t\t----- Boxplot of {i} -----')
    sns.boxplot(x=df[i],color=np.random.rand(4,))
    plt.show()
```

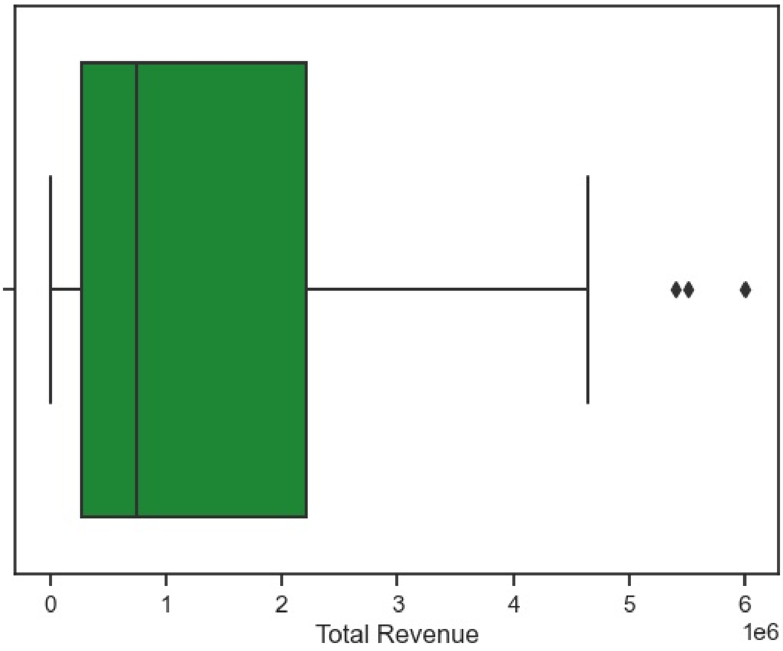
```
		----- Boxplot of Unit Price -----
```



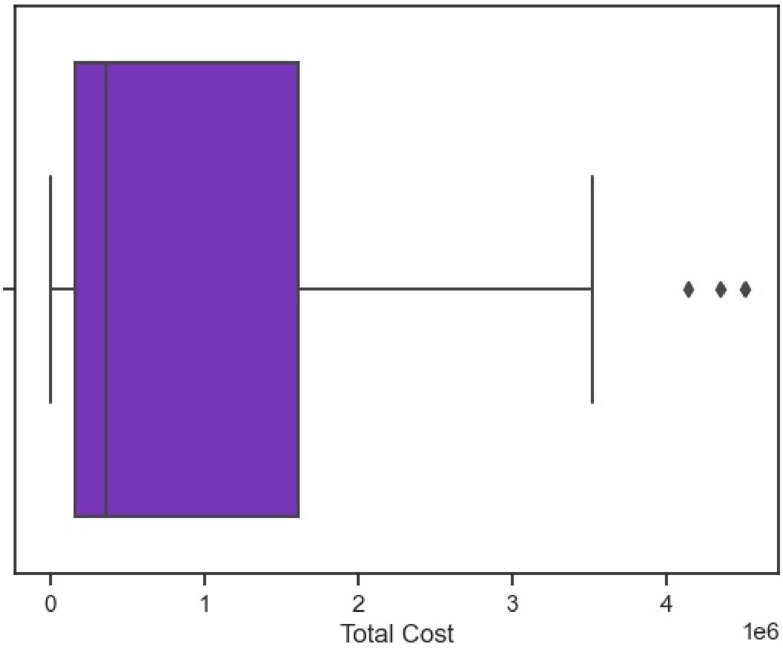
```
		----- Boxplot of Unit Cost -----
```



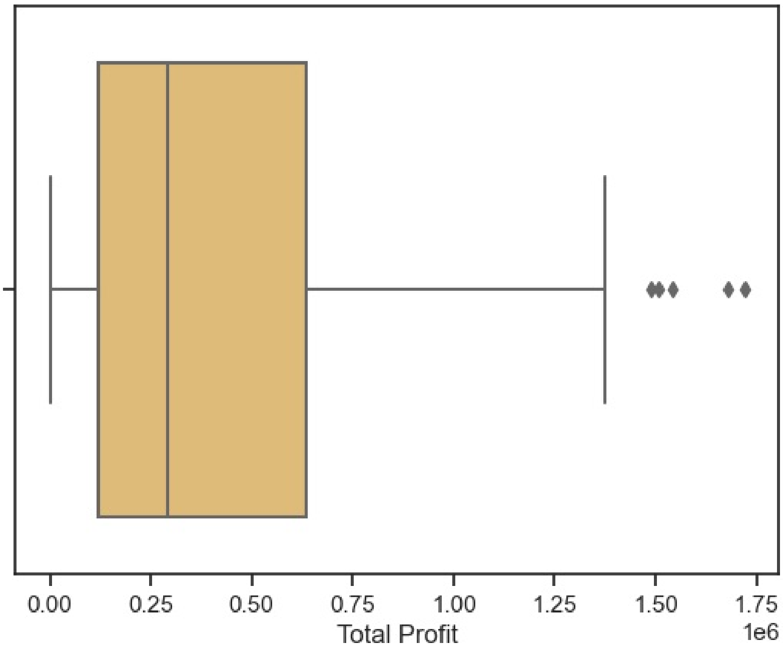
```
----- Boxplot of Total Revenue -----
```



```
----- Boxplot of Total Cost -----
```



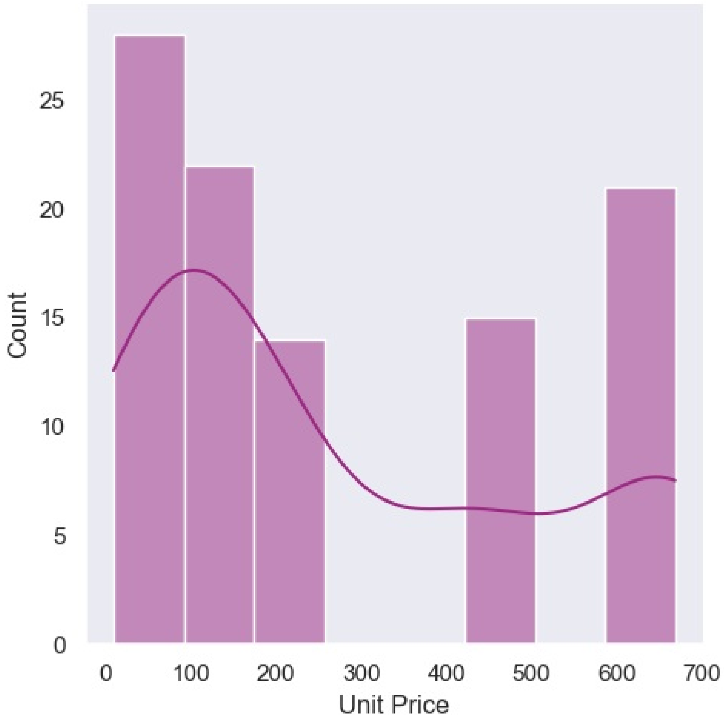
```
----- Boxplot of Total Profit -----
```

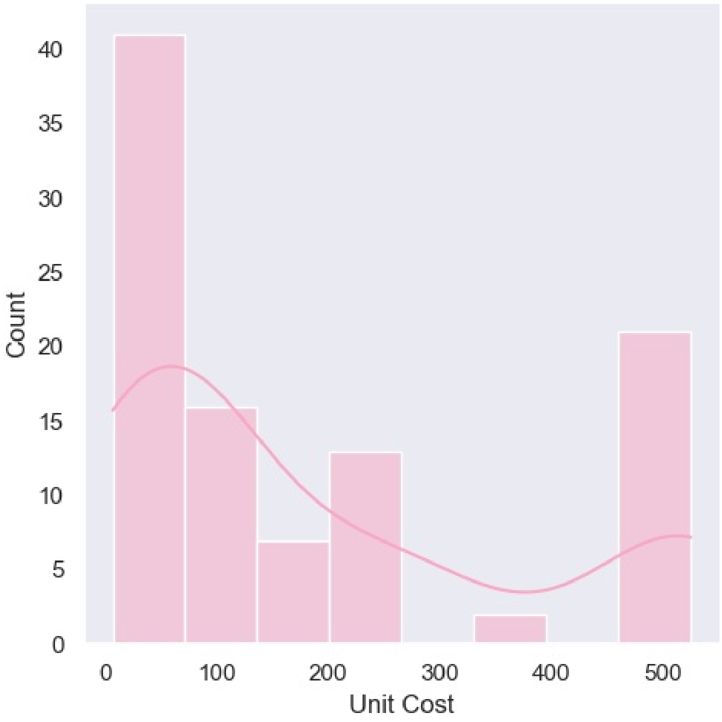


- Based on the boxplot there is an outliers

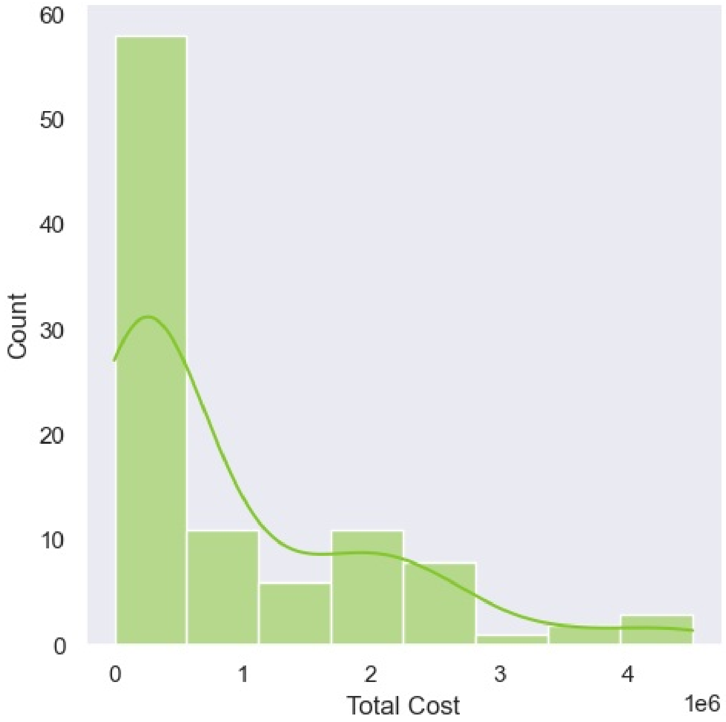
In [63]:

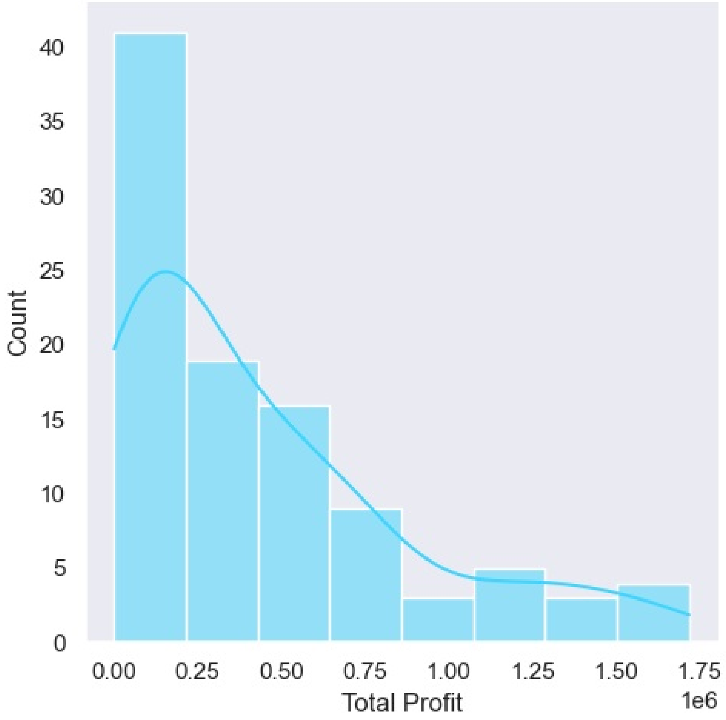
```
# check the distribution of a dataset
sns.set_theme(style='dark')
for i in continuous_features:
    sns.displot(x=df[i],kde=True,color=np.random.rand(3,))
    plt.show()
```




Count
0
5
10
15
20
25
30
35
40
0
100
200
300
400
500
Unit Cost

Count
0
10
20
30
40
50
0
1
2
3
4
5
6
1e6
Total Revenue





- Based on the above charts its a right skewed distribution