Bicycle Sales Analysis

CONTENT

- 1)Introduction to Data in the columns.
- 2)Null Value and Reduction.
- 3)Statistical Information
- 4)Statistical Figures
- Mean, Median and Mode.
- 5)Category Counting
- 6)Pivot Tables
- 7)Cross Tabs
- 8)Strip Plot
- 9)Line Plot
- 10)Categorical Plot[Catplot]
- 11)Heat Map
- 12)Box Plot
- 13)KDE Plot
- 14)Pie Plot
- 15)Product Analysis According to thier Category
- 16)Conclusion

In [1]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

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

In [3]:
```
cb=pd.read_csv('bike data.csv')
cb
```

Out[3]:

	Date	year	Customer ID	Customer Age	Age Group	Customer Gender	Country	State	Produc Categor
0	11/26/2013	2013	11019	19	Youth (<25)	M	Canada	British Columbia	Accessorie
1	11/26/2015	2015	11019	19	Youth (<25)	M	Canada	British Columbia	Accessorie
2	3/23/2014	2014	11039	49	Adults (35-64)	M	Australia	New South Wales	Accessorie
3	3/23/2016	2016	11039	49	Adults (35-64)	M	Australia	New South Wales	Accessorie
4	5/15/2014	2014	11046	47	Adults (35-64)	F	Australia	New South Wales	Accessorie
...	...	...	...	...	...	...	...	...	.
113031	4/12/2016	2016	29443	41	Adults (35-64)	M	United Kingdom	England	Clothin
113032	4/2/2014	2014	29462	18	Youth (<25)	M	Australia	Queensland	Clothin
113033	4/2/2016	2016	29462	18	Youth (<25)	M	Australia	Queensland	Clothin
113034	3/4/2014	2014	29472	37	Adults (35-64)	F	France	Seine (Paris)	Clothin
113035	3/4/2016	2016	29472	37	Adults (35-64)	F	France	Seine (Paris)	Clothin

113036 rows × 18 columns

1)Introduction to the Dataset

In [4]:

```
cb.columns
```

Out[4]:

```
Index(['Date', 'year', 'Customer ID', 'Customer Age', 'Age Group',
       'Customer Gender', 'Country', 'State', 'Product Category',
       'Sub Category', 'Product', 'Frame Size', 'Order Quantity', 'Unit Cost',
       'Unit Price', 'Cost', 'Revenue', 'Profit'],
      dtype='object')
```

In [5]:

```
cb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113036 entries, 0 to 113035
Data columns (total 18 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Date              113036 non-null  object 
 1   year              113036 non-null  int64  
 2   Customer ID       113036 non-null  int64  
 3   Customer Age      113036 non-null  int64  
 4   Age Group         113036 non-null  object 
 5   Customer Gender   113036 non-null  object 
 6   Country           113036 non-null  object 
 7   State             113036 non-null  object 
 8   Product Category  113036 non-null  object 
 9   Sub Category      113036 non-null  object 
 10  Product           113036 non-null  object 
 11  Frame Size        25982 non-null   float64
 12  Order Quantity    113036 non-null  int64  
 13  Unit Cost         113036 non-null  int64  
 14  Unit Price        113036 non-null  int64  
 15  Cost              113036 non-null  int64  
 16  Revenue           113036 non-null  int64  
 17  Profit            113036 non-null  int64  
dtypes: float64(1), int64(9), object(8)
memory usage: 15.5+ MB
```

In [6]:

```
cb.describe()
```

Out[6]:

	year	Customer ID	Customer Age	Frame Size	Order Quantity	Unit Cost	
count	113036.000000	113036.000000	113036.000000	25982.000000	113036.000000	113036.000000	1
mean	2014.401739	19227.874341	35.919212	47.313063	11.901660	267.296366	
std	1.272510	5307.581302	11.021936	6.860797	9.561857	549.835483	
min	2011.000000	11000.000000	17.000000	38.000000	1.000000	1.000000	
25%	2013.000000	14611.000000	28.000000	42.000000	2.000000	2.000000	
50%	2014.000000	18664.000000	35.000000	46.000000	10.000000	9.000000	
75%	2016.000000	23475.000000	43.000000	52.000000	20.000000	42.000000	
max	2016.000000	29483.000000	87.000000	62.000000	32.000000	2171.000000	

```
In [7]: cb.nunique()
```

```
Out[7]: Date                1884
        year                   6
        Customer ID         8232
        Customer Age          70
        Age Group              4
        Customer Gender        2
        Country                6
        State                 53
        Product Category       3
        Sub Category          17
        Product              130
        Frame Size            13
        Order Quantity        32
        Unit Cost             34
        Unit Price            36
        Cost                 360
        Revenue             1876
        Profit              1256
        dtype: int64
```

2)Null Values

```
In [8]: cb.isnull().sum().sum()
```

```
Out[8]: 87054
```

```
In [9]: cb.drop(['Frame Size'],axis=1,inplace=True)
```

```
In [10]: cb.isnull().sum().sum()
```

```
Out[10]: 0
```

- Before Dimensionality Reduction we have around 87 thousand null values in the dataset.
- After Drop the Frame Size column we reduce all the null values because, it contains almost all null values in the dataset.

3)Statistical Information

Correlation

- It is relation between Two group or Two column or Feature.
- Relation Between +1 to -1.

In [11]:
```
cb.corr()
```

Out[11]:

	year	Customer ID	Customer Age	Order Quantity	Unit Cost	Unit Price	Cost	Revenue	
year	1.000000	0.000160	0.040994	0.123169	-0.217575	-0.213673	-0.215604	-0.208673	-
Customer ID	0.000160	1.000000	-0.036474	0.004824	-0.040788	-0.051058	-0.043865	-0.053147	-
Customer Age	0.040994	-0.036474	1.000000	0.026887	-0.021374	-0.020262	-0.016013	-0.009326	
Order Quantity	0.123169	0.004824	0.026887	1.000000	-0.515835	-0.515925	-0.340382	-0.312895	-
Unit Cost	-0.217575	-0.040788	-0.021374	-0.515835	1.000000	0.997894	0.829869	0.817865	
Unit Price	-0.213673	-0.051058	-0.020262	-0.515925	0.997894	1.000000	0.826301	0.818522	
Cost	-0.215604	-0.043865	-0.016013	-0.340382	0.829869	0.826301	1.000000	0.988758	
Revenue	-0.208673	-0.053147	-0.009326	-0.312895	0.817865	0.818522	0.988758	1.000000	
Profit	-0.181525	-0.067768	0.004319	-0.238863	0.741020	0.749870	0.902233	0.956572	

Covariance

- Covariance provides the a measure of strength of correlation between two variable or more set of variables.
- The mean of each variable is used as reference and relative positions of observations compared to mean is important.

In [12]:
```
cb.cov()
```

Out[12]:

	year	Customer ID	Customer Age	Order Quantity	Unit Cost	Unit Price
year	1.619283	1.082623e+00	0.574961	1.498666	-152.230727	-250.711973
Customer ID	1.082623	2.817042e+07	-2133.737829	244.820349	-119031.365245	-249874.260049
Customer Age	0.574961	-2.133738e+03	121.483065	2.833591	-129.533079	-205.920399
Order Quantity	1.498666	2.448203e+02	2.833591	91.429105	-2711.975668	-4548.759185
Unit Cost	-152.230727	-1.190314e+05	-129.533079	-2711.975668	302319.058484	505919.547067
Unit Price	-250.711973	-2.498743e+05	-205.920399	-4548.759185	505919.547067	850215.333511
Cost	-242.769932	-2.060127e+05	-156.177323	-2879.955796	403756.821761	674187.002018
Revenue	-347.614619	-3.692699e+05	-134.568318	-3916.624648	588688.371911	988020.078177
Profit	-104.844687	-1.632572e+05	21.609004	-1036.668852	184931.550150	313833.076159

Standard Deviation

- A quantity expressing by how much the members of a group differ from the mean value for the group.
- The standard deviation is used in conjunction with the mean to summarise continuous data, not categorical data.

```
In [13]: cb.std()
```

```
Out[13]: year                  1.272510
         Customer ID        5307.581302
         Customer Age         11.021936
         Order Quantity        9.561857
         Unit Cost           549.835483
         Unit Price          922.071219
         Cost                884.866118
         Revenue            1309.094674
         Profit              453.887443
         dtype: float64
```

4)Statistical Figures

- A statistic or sample statistic is any quantity computed from values in a sample which is considered for a statistical purpose.
- Statistical purposes include estimating a population parameter,

```
In [14]: cb['Order Quantity'].count()
```

```
Out[14]: 113036
```

- Count: The quantity of the Order placed for the bikes.

```
In [15]: cb['Order Quantity'].mean()
```

```
Out[15]: 11.901659648253654
```

- Mean: Average of the order quantity of the bike.

```
In [16]: cb['Order Quantity'].max()
```

```
Out[16]: 32
```

- Max: Maximum Order quantity for the bike.

```
In [17]: cb['Order Quantity'].min()
```

```
Out[17]: 1
```

- Min: Minimum Order quantity for the bike.

```
In [18]: cb['Order Quantity'].median()
```

```
Out[18]: 10.0
```

- Median: The middle observation of order quantity in the dataset.

```
In [19]: cb['Order Quantity'].mode()
```

```
Out[19]: 0    1
         Name: Order Quantity, dtype: int64
```

- Mode: The Most Frequent quantity occuring in order for the bike.

```
In [20]: cb['Order Quantity'].std()
```

```
Out[20]: 9.56185675955091
```

- Standard Deviation: It substract each value in the data from mean and divide it by overall deviation.

5)Count Number Of Categories

```
In [21]: cb['Product'].value_counts()
```

```
Out[21]: Water Bottle - 30 oz.      10794
         Patch Kit/8 Patches        10416
         Mountain Tire Tube          6816
         AWC Logo Cap                4358
         Sport-100 Helmet, Red       4220
                                    ...
         Mountain-100 Silver, 48       58
         Mountain-100 Silver, 42       54
         Mountain-500 Silver, 48       40
         Road-650 Red, 52              32
         Mountain-500 Black, 52        22
         Name: Product, Length: 130, dtype: int64
```

```
In [22]: cb['Product Category'].value_counts()
```

```
Out[22]: Accessories    70120
         Bikes          25982
         Clothing       16934
         Name: Product Category, dtype: int64
```

In [23]:

```
cb['Sub Category'].value_counts()
```

Out[23]:

```
Tires and Tubes      33870
Bottles and Cages    15876
Road Bikes           13430
Helmets              12158
Mountain Bikes        8854
Jerseys               6010
Caps                  4358
Fenders               4032
Touring Bikes         3698
Gloves                2686
Cleaners              1802
Shorts                1794
Hydration Packs       1334
Socks                 1122
Vests                  964
Bike Racks             592
Bike Stands            456
Name: Sub Category, dtype: int64
```

6)Pivot Tables

In [24]:

```
PV1=pd.pivot_table(cb, values='Profit', index=['year'],
                    columns=['Customer Gender'], aggfunc=np.sum)
PV1
```

Out[24]:

Customer Gender	F	M
year		
2011	1475580	1405721
2012	1512853	1439140
2013	2981277	2977931
2014	2716334	3147753
2015	3767734	3760829
2016	3280090	3755858

- Shows the Genderwise sales profit in dataset.

```
In [25]: cb_Category_profit=pd.pivot_table(data=cb[['Product Category','Sub Category','Ord
                                          index=['Product Category','Sub Category'],
                                          values='Order Quantity',
                                          aggfunc='max')
         cb_Category_profit
```

Out[25]:

		Order Quantity
Product Category	**Sub Category**	
Accessories	**Bike Racks**	30
	Bike Stands	12
	Bottles and Cages	32
	Cleaners	32
	Fenders	32
	Helmets	32
	Hydration Packs	32
	Tires and Tubes	32
Bikes	**Mountain Bikes**	4
	Road Bikes	4
	Touring Bikes	3
Clothing	**Caps**	32
	Gloves	32
	Jerseys	32
	Shorts	32
	Socks	32
	Vests	32

- It Shows the category wise products order quantity in the categorical manner.

7)Crosstab

- Table showing the relationship between two or more variables. Where the table only shows the relationship between two categorical variables, a crosstab is also known as a contingency table.
- It helps us to make informed decisions regarding your research by identifying patterns, trends, and the correlation between your study parameters.

In [26]:
```
pd.crosstab(cb["Product Category"],cb["Country"])
```

Out[26]:

Country Product Category	Australia	Canada	France	Germany	United Kingdom	United States
Accessories	13498	10232	6688	6546	8114	25042
Bikes	7064	1548	2770	3048	3524	8028
Clothing	3374	2398	1540	1504	1982	6136

Data Visualisation

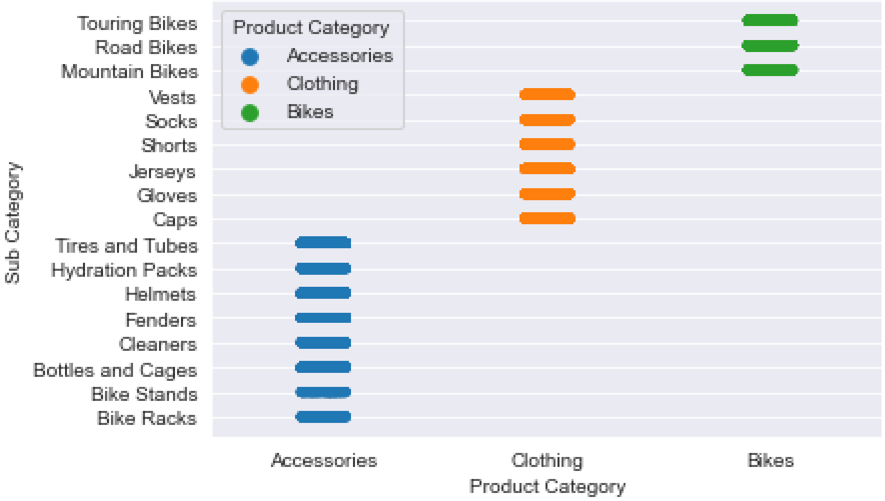
8) Strip Plot

- A strip plot is very simple to understand. It is basically a scatter plot that differentiates different categories.

Plot Info:

- Plot Shows the categorical wise product differentiation in the stripplot plot and shows the distribution of orders.
- In the Dataset Accessories has the higest sub category range in the products.

In [27]:
```
sns.set_style('darkgrid')
sns.stripplot(x='Product Category',y='Sub Category',hue='Product Category',data=
plt.show()
```



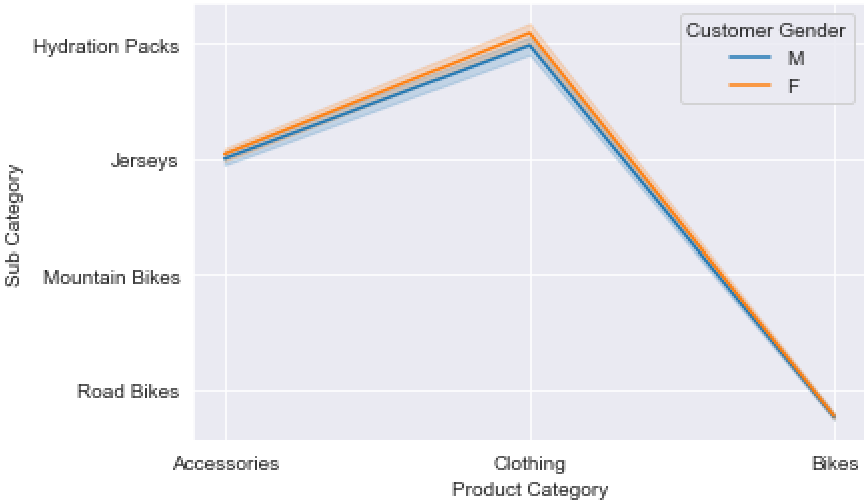
9)Lineplot

- Line Plots depict the relationship between continuous as well as categorical values in a continuous data point format.

Plot Info:

- Plot shows the product category and the gender wise scope of products which show the clothing Category has the highest order in the category for both Male and Female.

```
In [28]: sns.lineplot(x='Product Category',y='Sub Category',hue='Customer Gender',data=cb)
         plt.show()
```



10)Categorical plot

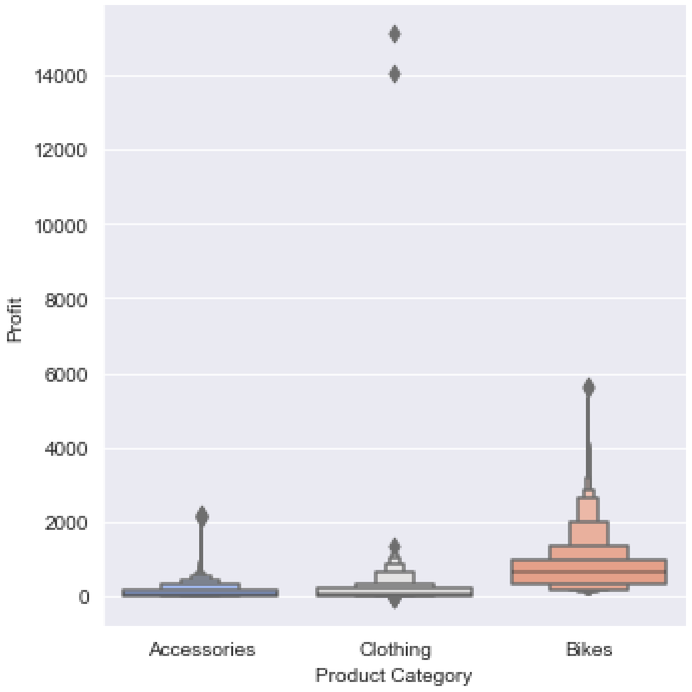
- It is the factor plot() produce a stripplot as default plot type.

Plot Info:

- In the plot it shows the product categorical wise profit in the catplot and there is two outlier occurs in the given dataset.
- By the obervation in the plot we see the Profit of the Bikes in the Products is highest.

In [29]:

```
sns.catplot(x='Product Category',y="Profit",kind='boxen',palette='coolwarm',data=
plt.show()
```



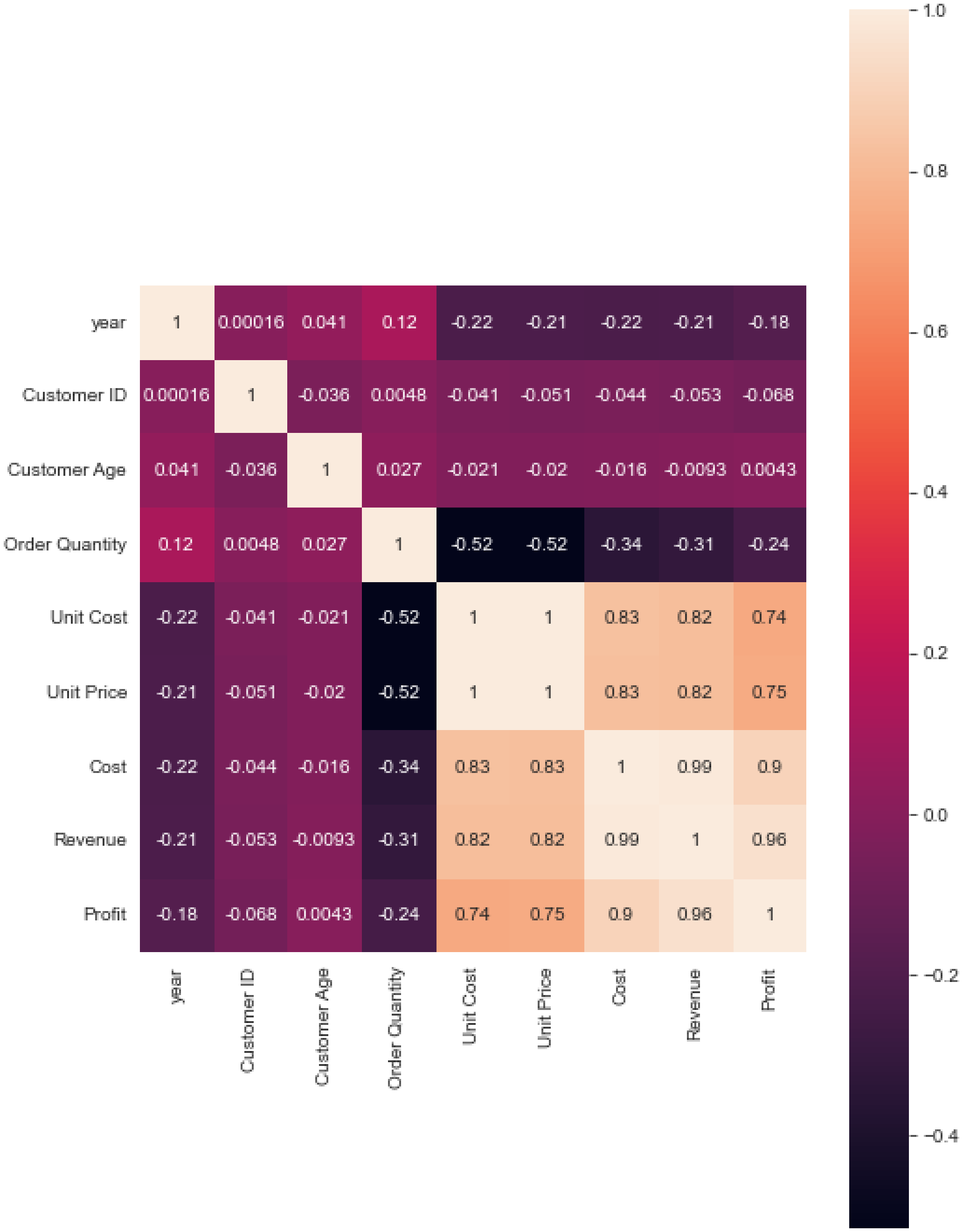
11)Heat Map

- Heat Maps is to better visualize the volume of locations/events within a dataset and assist in directing viewers towards areas on data visualizations that matter most.

Plot Info:

- In the plot it shows the correlation between the all attributes in the dataset.
- By the observation we see the relation between the order quantity and cost of the product is very strong.

In [30]:
```
cb_corr=cb.corr()
plt.figure(figsize=(8,12))
sns.heatmap(cb_corr,annot=True,square=True)
plt.show()
```



	year	Customer ID	Customer Age	Order Quantity	Unit Cost	Unit Price	Cost	Revenue	Profit
year	1	0.00016	0.041	0.12	-0.22	-0.21	-0.22	-0.21	-0.18
Customer ID	0.00016	1	-0.036	0.0048	-0.041	-0.051	-0.044	-0.053	-0.068
Customer Age	0.041	-0.036	1	0.027	-0.021	-0.02	-0.016	-0.0093	0.0043
Order Quantity	0.12	0.0048	0.027	1	-0.52	-0.52	-0.34	-0.31	-0.24
Unit Cost	-0.22	-0.041	-0.021	-0.52	1	1	0.83	0.82	0.74
Unit Price	-0.21	-0.051	-0.02	-0.52	1	1	0.83	0.82	0.75
Cost	-0.22	-0.044	-0.016	-0.34	0.83	0.83	1	0.99	0.9
Revenue	-0.21	-0.053	-0.0093	-0.31	0.82	0.82	0.99	1	0.96
Profit	-0.18	-0.068	0.0043	-0.24	0.74	0.75	0.9	0.96	1

12)Barplot(Seaborn library)

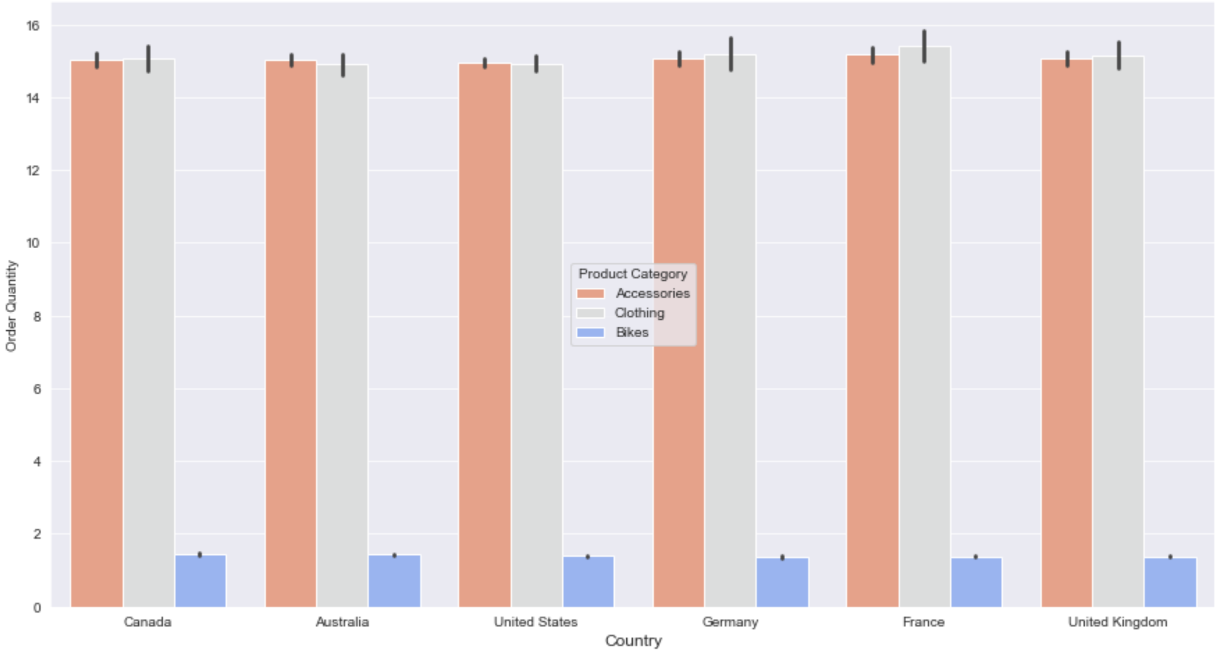
- A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent

Plot info

- In the given plot we shows the Product Category according to the Order Quantity in the Various countries.
- In the plot Accessories has the highest Order in all the country.

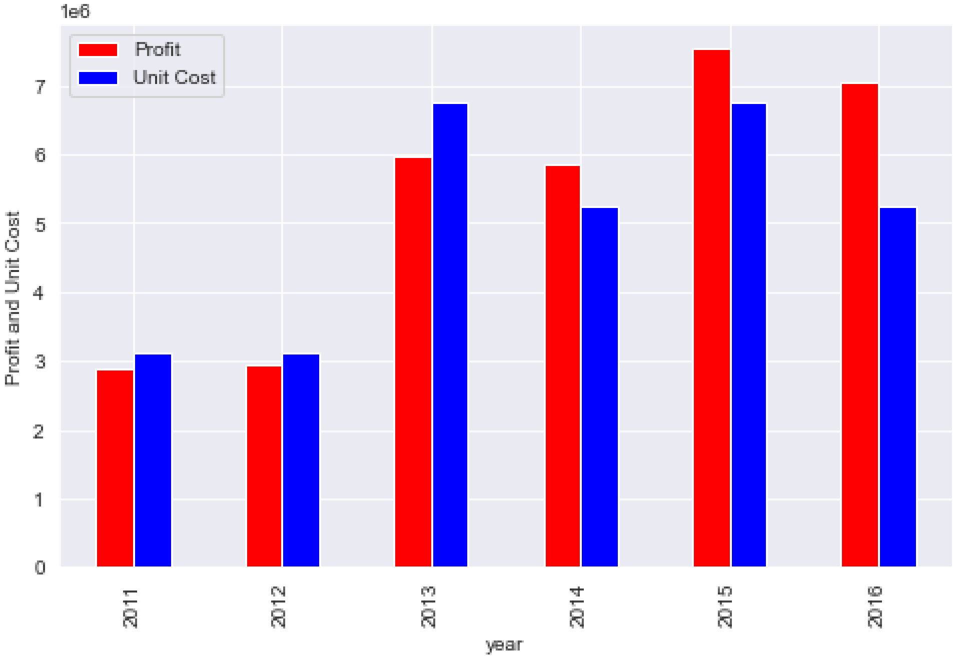
In [31]:

```
plt.figure(figsize=(15,8))
sns.barplot(x='Country',y='Order Quantity',hue='Product Category',data=cb,palette
plt.xlabel('Country',fontsize=12)
plt.show()
```



In [32]:

```
cb.groupby('year')[['Profit','Unit Cost']].sum().plot.bar(color=['red','blue'],fi
plt.ylabel('Profit and Unit Cost')
plt.show()
```



- In the Comparison for Cost price and Profit, We observe the Profit is fluctuating year by year.

13)KDE Plot

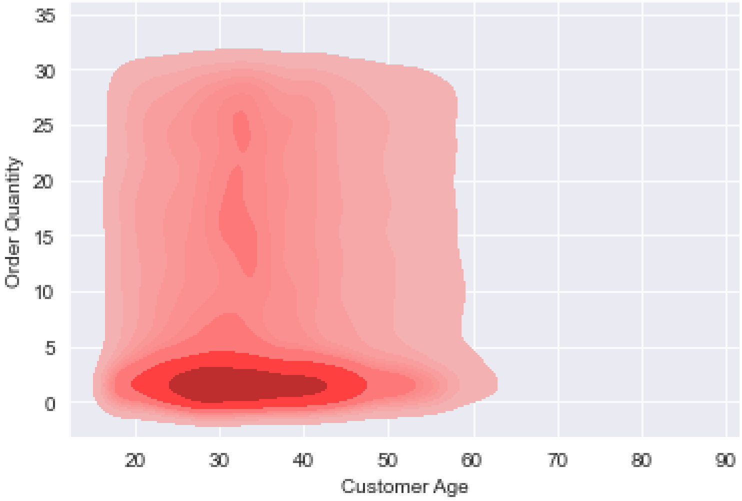
- Kdeplot is a Kernel Distribution Estimation Plot which depicts the probability density function of the continuous or non-parametric data variables

Plot Info:

- 1) The kdeplot between the Customer Age and Order Quantity which shows the which customer according to thier age and the how much they order in the product category.
- In the plot we see the age of 20 to 45,order quantity is around 5 is popular.

In [33]:

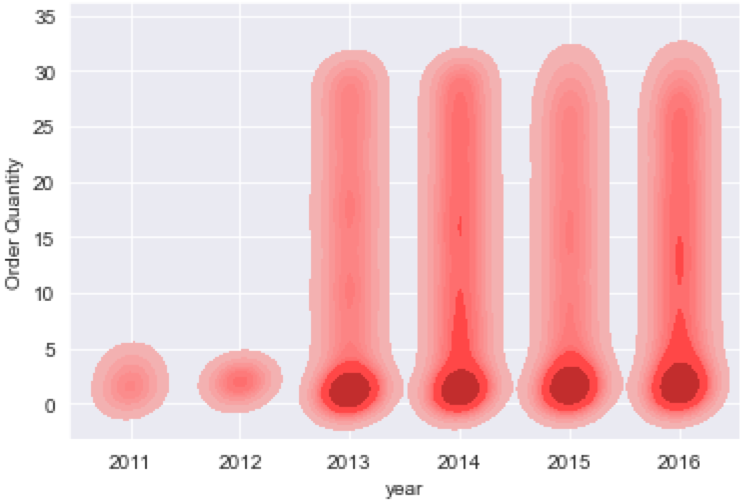
```
sns.kdeplot(cb['Customer Age'],cb['Order Quantity'],color='Red',shade=True)
plt.show()
```



- 2)In this kdeplot we check the year wise order quantity of product.
- In 2016 the order quantity is highest compare to the all order.

In [34]:

```
sns.kdeplot(cb['year'],cb['Order Quantity'],color='Red',shade=True)
plt.show()
```

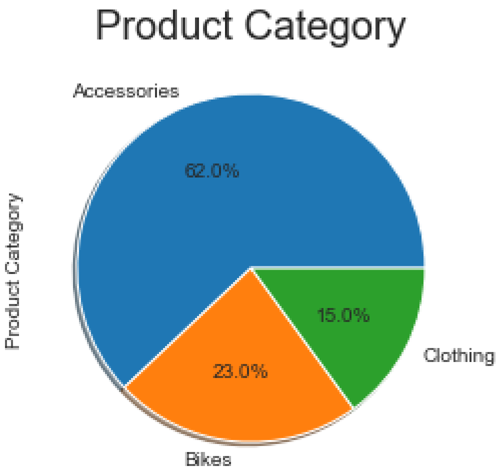


14) Pie Plot

- Pie chart show the distribution of the product category.
- It shows the Accessories has the highest portion acquired.

In [35]:

```
plt.title('Product Category',fontsize=20)
cb['Product Category'].value_counts().plot.pie(autopct='%1.1f%%',shadow=True)
plt.show()
```



In [36]:

```
cb.columns
```

Out[36]:

```
Index(['Date', 'year', 'Customer ID', 'Customer Age', 'Age Group',
       'Customer Gender', 'Country', 'State', 'Product Category',
       'Sub Category', 'Product', 'Order Quantity', 'Unit Cost', 'Unit Price',
       'Cost', 'Revenue', 'Profit'],
      dtype='object')
```

15)Product Analysis According to thier Category.

1)Bikes

In [37]:

```

a_cb = cb.loc[cb['Country']=='Country']
a_bike= cb.loc[cb['Product Category']=='Bikes']
a_subbike = pd.pivot_table(a_bike, values='Order Quantity', index=['year'],
                    columns=['Sub Category'], aggfunc=np.sum)
a_subbike
```

Out[37]:

Sub Category	Mountain Bikes	Road Bikes	Touring Bikes
year			
2011	1245.0	4015.0	NaN
2012	1230.0	4124.0	NaN
2013	2088.0	2797.0	825.0
2014	1724.0	1856.0	1024.0
2015	3124.0	4202.0	1230.0
2016	2581.0	2777.0	1569.0

2)Accessories

In [38]:

```
a_cb = cb.loc[cb['Country']=='Country']
a_acc= cb.loc[cb['Product Category']=='Accessories']
a_subAcc = pd.pivot_table(a_acc, values='Order Quantity', index=['year'],
                    columns=['Sub Category'], aggfunc=np.sum)
a_subAcc
```

Out[38]:

Sub Category	Bike Racks	Bike Stands	Bottles and Cages	Cleaners	Fenders	Helmets	Hydration Packs	Tires and Tubes
year								
2013	1044	670	50633	5941	14164	40592	4781	113317
2014	1388	582	71916	8041	17348	51597	5314	147452
2015	1000	622	49350	5766	13802	39498	4647	110155
2016	1309	538	69827	7826	16824	49919	5172	143127

3)Clothing

In [39]:

```
a_cb = cb.loc[cb['Country']=='Country']
a_clothing= cb.loc[cb['Product Category']=='Clothing']
a_subclothing = pd.pivot_table(a_clothing, values='Order Quantity', index=['year'
                    columns=['Sub Category'], aggfunc=np.sum)
a_subclothing
```

Out[39]:

Sub Category	**Caps**	**Gloves**	**Jerseys**	**Shorts**	**Socks**	**Vests**
year						
2013	14337	8797	20520	7000	3846	3435
2014	19800	11499	24167	6784	5132	3961
2015	13916	8499	19909	6786	3729	3282
2016	19215	11183	23529	6598	4971	3848

16)Conclusion

Products :

- Sales decline from 2015 to 2016
- Since the rise of accessories in 2013 there is an inverse relationship with bikes
- The decline in Sales in 2016 was due to lower demand for bikes especially mountain bikes and road bikes,There's a big drop in the U.S. and Australia
- Most of them are female, specifically in the 25-to 34-year-old age group

Valuations :

- Total Revenue of All Product Category is 85 Million.
- United State Highest Revenue of 28 Million.
- In Category Bike has the highest Revenue of 62 Million.
- Count of Coustomer is 8000.
- Total Transaction in number is 113K.

In [ ]: