Methodology Document

Problem Statement

OList is an e-commerce company that has faced some losses recently and they want to manage their inventory very well so as to reduce any unnecessary costs that they might be bearing.

As part of managing the inventory cost ,Olist wants to identify top products that contribute to the revenue and also use market basket analysis to analyze the purchase behavior of individual customers to estimate with relative certainty, what items are more likely to be purchased individually or in combination with some other products.

Objective

The company wants to understand the following factors.

- The top selling and least selling product categories by quantity and revenue.
- The combinations of product categories frequently ordered together.
- Action points to reduce the inventory cost.

Background

- Olist, an E-commerce company faced losses recently due to high inventory costs.
- To improve the profits and reduce losses, the company wants to optimize the inventory based on customer purchase behavior.
- Company aims to get rid of the least selling products from inventory without any impact on the business.

EDA

Understanding the dataset

The dataset consists of information related to the order details. The data is spread in 5 excel sheets in the following categories.

- Orders – Contains orders details like order id, Order time stamp etc.
- Order Items – Contains specific details like product ID,Seller ID, Price etc.
- Customers – Contains customer id, City, State etc.
- Payments – Payment details like mode of Payment, Payment Installment etc.
- Products – Details like Product Category name, product specifications etc.

Since from the ERD document it is clear that the datasets can be merged together, the datasets are merged to form a single dataset for the ease of analysis.

However, the Customers and Payments datasets are not included while merging.

The merged dataset consists of 112650 rows and 17 columns.

Image 1:

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

```
(112650, 17)
```

Image 2:

```
# checking the info of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 112650 entries, 0 to 112649
Data columns (total 17 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   order_id                       112650 non-null  object
 1   customer_id                    112650 non-null  object
 2   order_status                   112650 non-null  object
 3   order_purchase_timestamp       112650 non-null  datetime64[ns]
 4   order_approved_at              112635 non-null  datetime64[ns]
 5   order_delivered_timestamp      110196 non-null  datetime64[ns]
 6   order_estimated_delivery_date  112650 non-null  datetime64[ns]
 7   order_item_id                  112650 non-null  int64
 8   product_id                     112650 non-null  object
 9   seller_id                      112650 non-null  object
 10  price                          112650 non-null  float64
 11  shipping_charges               112650 non-null  float64
 12  product_category_name          112257 non-null  object
 13  product_weight_g               112632 non-null  float64
 14  product_length_cm              112632 non-null  float64
 15  product_height_cm              112632 non-null  float64
 16  product_width_cm               112632 non-null  float64
dtypes: datetime64[ns](4), float64(6), int64(1), object(6)
memory usage: 15.5+ MB
```

Image 3:

```
#Checking the summary of the dataset
df.describe()
```

	order_item_id	price	shipping_charges	product_weight_g	product_length_cm	product_height_cm	product_width_cm
count	112650.000000	112650.000000	112650.000000	112632.000000	112632.000000	112632.000000	112632.000000
mean	1.197834	120.653739	19.990320	2093.672047	30.153669	16.593766	22.996546
std	0.705124	183.633928	15.806405	3751.596884	16.153449	13.443483	11.707268
min	1.000000	0.850000	0.000000	0.000000	7.000000	2.000000	6.000000
25%	1.000000	39.900000	13.080000	300.000000	18.000000	8.000000	15.000000
50%	1.000000	74.990000	16.260000	700.000000	25.000000	13.000000	20.000000
75%	1.000000	134.900000	21.150000	1800.000000	38.000000	20.000000	30.000000
max	21.000000	6735.000000	409.680000	40425.000000	105.000000	105.000000	118.000000

Data Cleaning

1. There are no incorrect datatypes in the dataset.

2. There are missing values in the following columns
   - order_approved_at – 15
   - order_delivered_timestamp - 2454
   - product_category_name – 393
   - product_weight_g - 18
   - product_length_cm - 18
   - product_height_cm - 18
   - product_width_cm – 18

3. Imputing the missing values

   - The missing values in order_approved_at , order_delivered_timestamp were not imputed since imputing them with mean or median would create incorrect data.
   - The missing values in the Product_category_name was imputed with the mode of the column.
   - The missing values in the'product_weight_g','product_length_cm',‘product_height_cm,'product_width_cm' is less than 0.01 %. Hence imputing them using the mean value.

4. Checking for outliers

Price

Image 4:

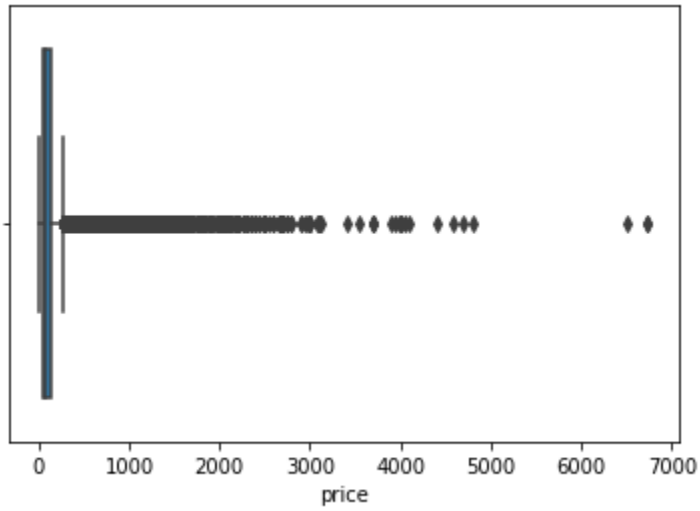


Image 5:

```
1 df[df.price>3500]
```

seller_id	price	shipping_charges	product_category_name	product_weight_g	product_length_cm	product_height_cm	product_width_cm
b029fc6e495ca4f08a35d51e53a5	4059.00	104.51	toys	8000.0	55.0	25.0	45.0
b029fc6e495ca4f08a35d51e53a5	3690.00	136.80	toys	8000.0	55.0	25.0	45.0
8f15b1dded4d213a468ba4eb391	6499.00	227.66	toys	7400.0	47.0	25.0	25.0
7c56835dd8e2e72f91f809cd4092	3999.00	195.76	small_appliances	20500.0	39.0	39.0	58.0
98ac2a96d1931b6bd30aa21f61d	4399.87	113.45	toys	3550.0	71.0	34.0	22.0
cab38528488b65a9a9c603ff024a	4099.99	75.27	toys	3050.0	34.0	10.0	22.0
98c7a498169dc7bce44e6bb6277	6735.00	194.31	housewares	30000.0	60.0	61.0	33.0

Looking at the distribution of Price, we can see that the values are gradually increasing until 3500.While looking at the dataset for Price above 3500, most of the products are heavier. So heavier products can have higher prices. Hence these outliers are left untreated.

Shipping Charges

Image 6:

```
1 sns.boxplot(df.shipping_charges)
2 plt.show()
```

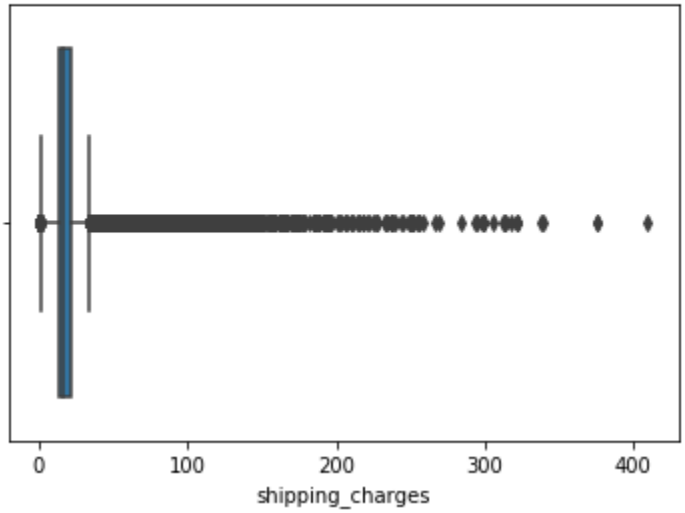


Image 7:

```
1 df[df.shipping_charges>300]
```

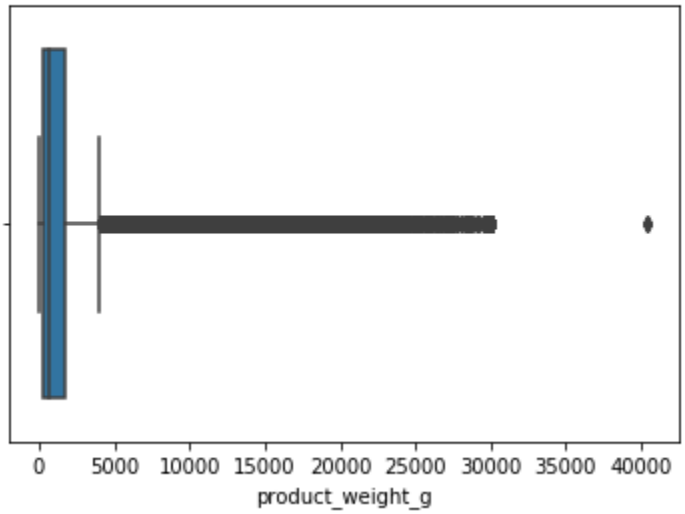
seller_id	price	shipping_charges	product_category_name	product_weight_g	product_length_cm	product_height_cm	product_width_cm
'01186712e53a39c564617	1050.00	321.88	toys	30000.0	55.0	75.0	61.0
'01186712e53a39c564617	1050.00	322.10	toys	30000.0	55.0	75.0	61.0
'01186712e53a39c564617	1050.00	338.30	toys	30000.0	55.0	75.0	61.0
491e472dffd59a3e82b2a3	2338.08	375.28	toys	30000.0	75.0	58.0	65.0
491e472dffd59a3e82b2a3	2338.08	375.28	toys	30000.0	75.0	58.0	65.0
'01186712e53a39c564617	990.00	314.02	toys	30000.0	55.0	75.0	61.0
0d900012f2e8e6e30d06d7	1045.00	314.40	construction_tools_construction	30000.0	100.0	50.0	50.0
0d900012f2e8e6e30d06d7	990.00	321.46	construction_tools_construction	30000.0	100.0	50.0	50.0
916fed0986310385ec9009	1149.00	339.59	toys	25250.0	68.0	66.0	66.0
'01186712e53a39c564617	760.00	312.41	toys	30000.0	55.0	75.0	61.0

The items with higher shipping charges are heavier items too. Hence these values seems to be correct data and are left untreated.

product_weight_g

Image 8:

```
1 sns.boxplot(df.product_weight_g)
2 plt.show()
```

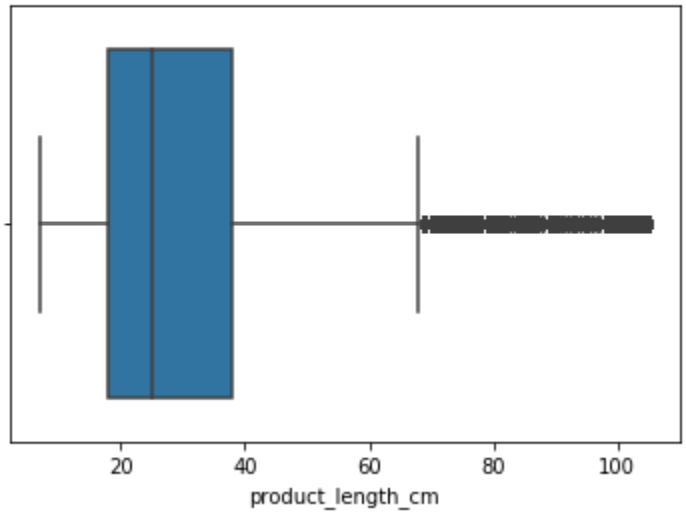


The datapoints are all continuously increasing up until 30000.Since there is only one datapoint above that, it can be considered as an outlier and can be removed.

product_length_cm

Image 9:

```
1 sns.boxplot(df.product_length_cm)
2 plt.show()
```

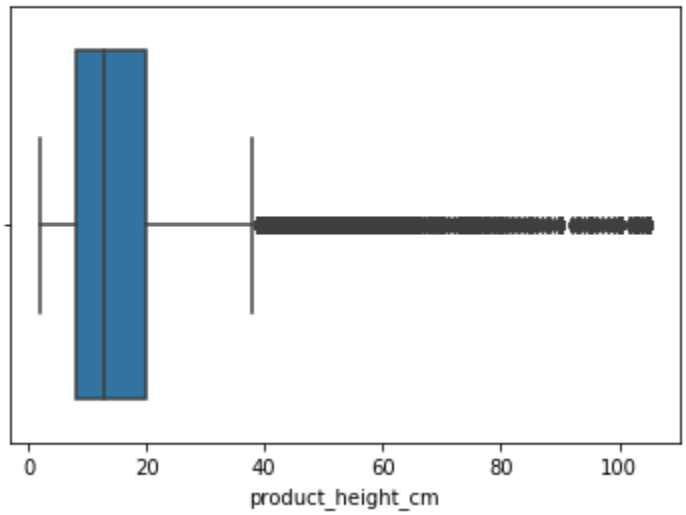


The values seem to be gradually increasing and hence the outlier values does not seem to be incorrect and hence left untreated.

product_height_cm

Image 10:

```
1 sns.boxplot(df.product_height_cm)
2 plt.show()
```



The values seem to be gradually increasing and hence outliers are not treated

product_width_cm

Image 11:

```
1 sns.boxplot(df.product_width_cm)
2 plt.show()
```

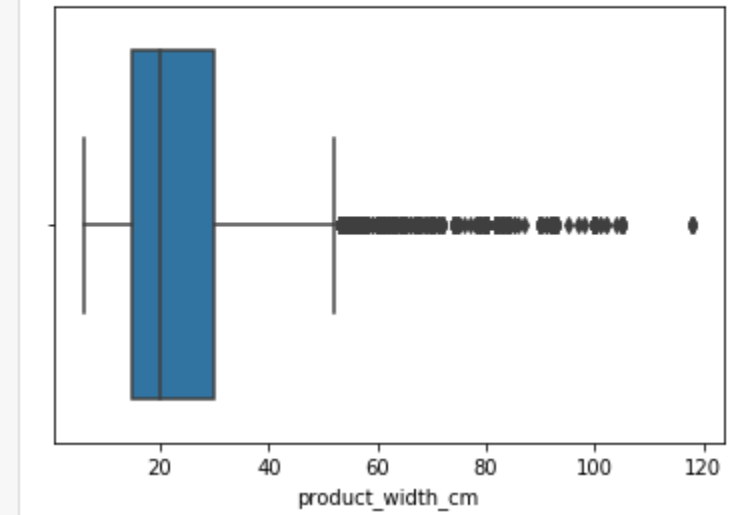


Image 12:

```
1 df[df.product_width_cm>110]
```

seller_id	price	shipping_charges	product_category_name	product_weight_g	product_length_cm	product_height_cm	product_width_cm
a3ca9315b744ce9f8e9374361493884	85.5	9.91	toys	1050.0	23.0	93.0	118.0
a3ca9315b744ce9f8e9374361493884	68.9	11.87	toys	1050.0	23.0	93.0	118.0
a3ca9315b744ce9f8e9374361493884	68.9	12.82	toys	1050.0	23.0	93.0	118.0
a3ca9315b744ce9f8e9374361493884	68.9	12.82	toys	1050.0	23.0	93.0	118.0
a3ca9315b744ce9f8e9374361493884	85.5	13.62	toys	1050.0	23.0	93.0	118.0
a3ca9315b744ce9f8e9374361493884	68.9	15.69	toys	1050.0	23.0	93.0	118.0
a3ca9315b744ce9f8e9374361493884	68.9	17.49	toys	1050.0	23.0	93.0	118.0
a3ca9315b744ce9f8e9374361493884	85.5	17.85	toys	1050.0	23.0	93.0	118.0

The values seem to be incorrect hence left untreated.

Checking the Shape of dataset after data cleaning.

Image 13:

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

```
(109803, 17)
```

Image 14:

```
(109803/112650)*100
```

```
97.47270306258322
```

Inference: 97% of data is left after data cleaning.

Checking for duplicate rows

Image 15:

```
duplicate
```

order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_timestamp	order_estimated_delivery_date	order_item_id	p

Inference: There are no duplicate rows in the database.

Dropping unwanted rows from the dataset

Since the analysis needs to be done on the items where order status is delivered, the remaining rows can be dropped.

With this EDA of the dataset is done. The final cleaned dataset is saved as an excel file.

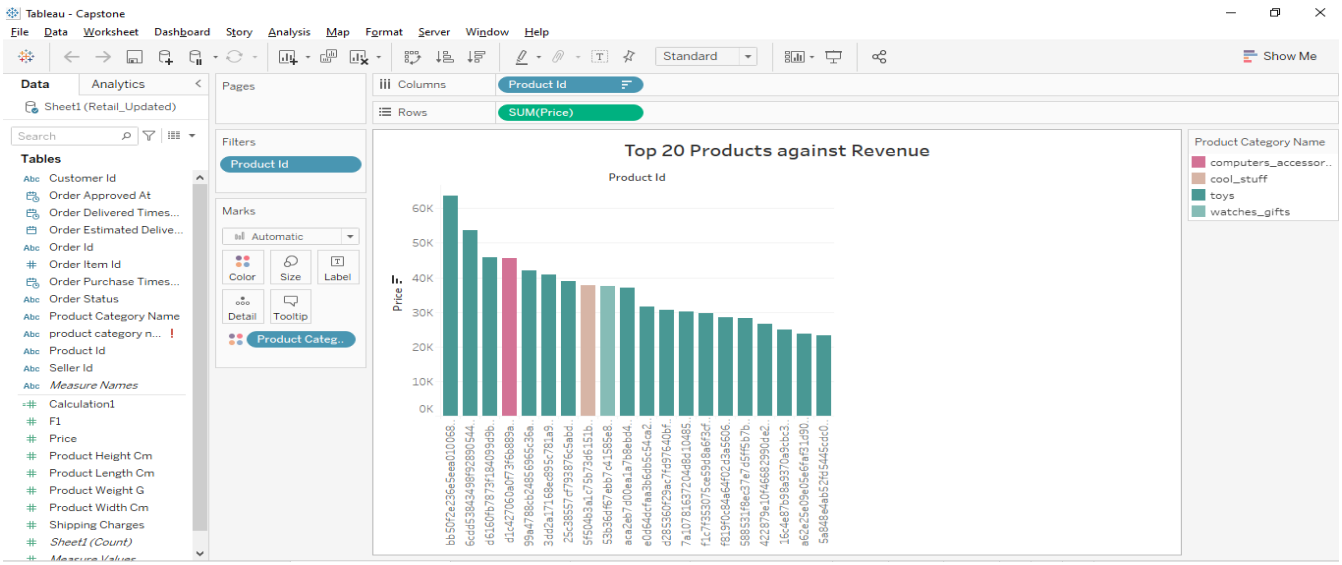
The further analysis and visualizations are done using Tableau.

Data Analysis and Interpretations

Top 20 ordered products by quantity

- Plotted a bar chart between the Product ID against the SUM of Price.
- The top 20 products were filtered with the field SUM of Price.
- The Product Category Name was used to differentiate the top 20 products of the categories based on revenue.

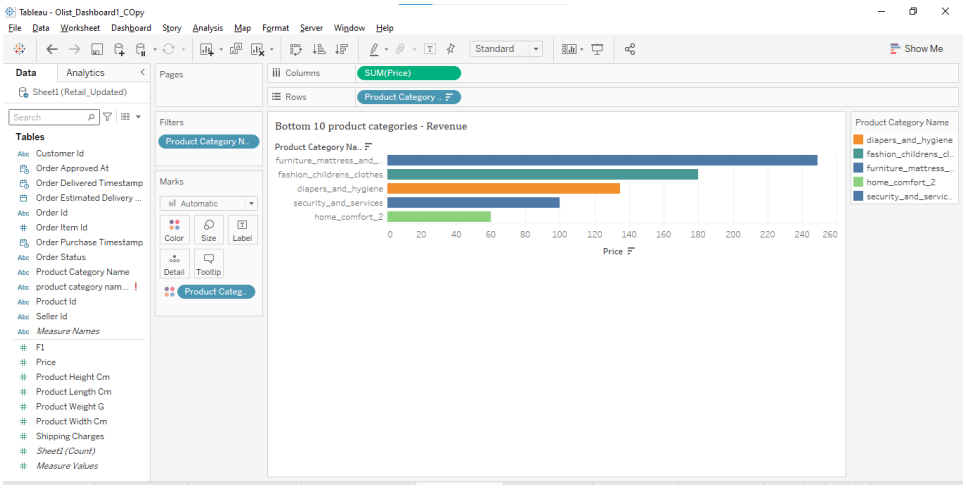
Image 16:



Bottom 5 Product categories based on Revenue

- Plotted a bar chart between the Product Category Name against the SUM of Price.
- The Bottom 5 products were filtered with the field SUM of Price.
- The Product Category Name was used to differentiate the Bottom 5 products of the categories based on revenue.

Image 17:



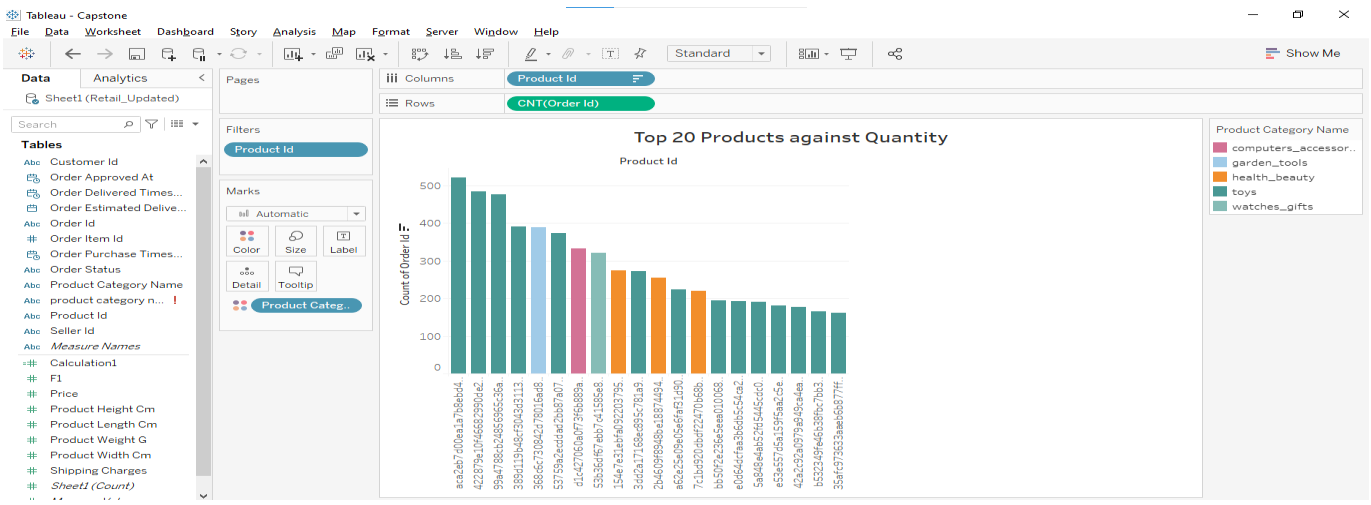
Inference:

- Among the top 20 products based on Revenue, 17 of them belong to the ‘Toys’ category.
- The other categories are ‘Computer accessories', 'Cool_Stuff’ and ‘Watches_gifts’.
- ‘Home_comfort_2’,’Security_and_services’, diapers_and_hygeine’ ,fashion_childrens_clothes’,’furniture_mattress_and_upholestry’ are the bottom 5 product categories based on revenue.

Top 20 products by revenue

- Plotted a bar chart between the Product ID against the distinct count of Order ID.
- The top 20 products were filtered with the distinct count of Order ID.
- The Product Category Name was used to differentiate the top 20 products of the categories based on quantity of order.

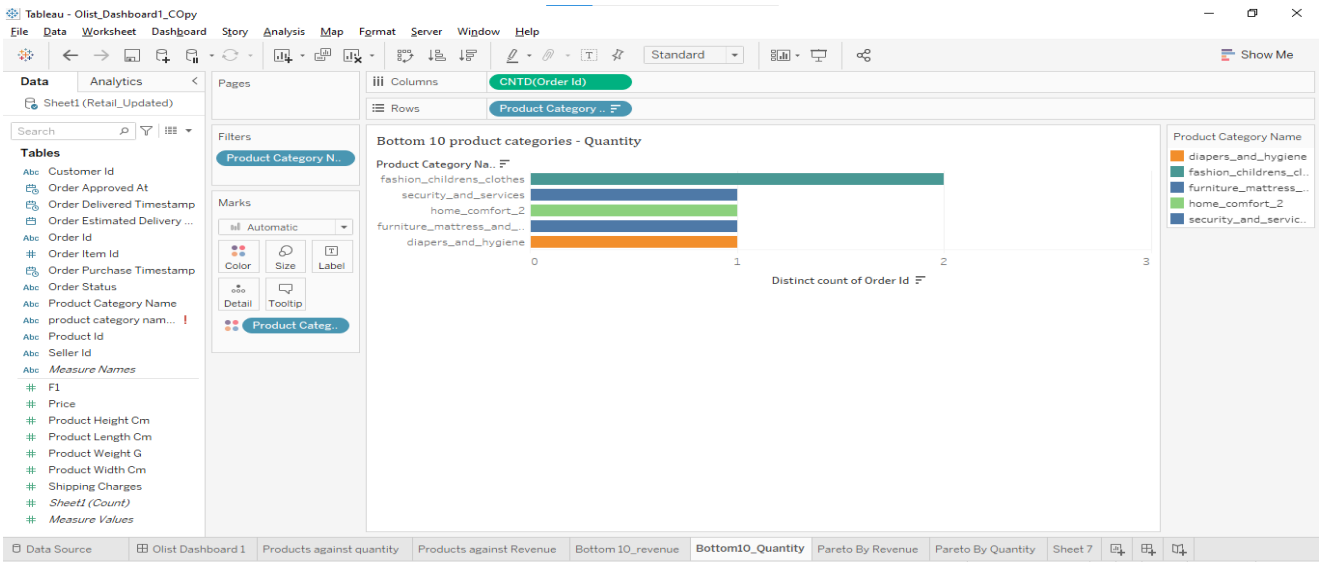
Image 18:



Bottom 5 Product categories based on Quantity

Image 19:

- Plotted a bar chart between the Product Category Name against the Distinct Count of Order ID.
- The Bottom 5 products were filtered with the field Distinct Count Of Order ID.
- The Product Category Name was used to differentiate the Bottom 5 products of the categories based on quantity.



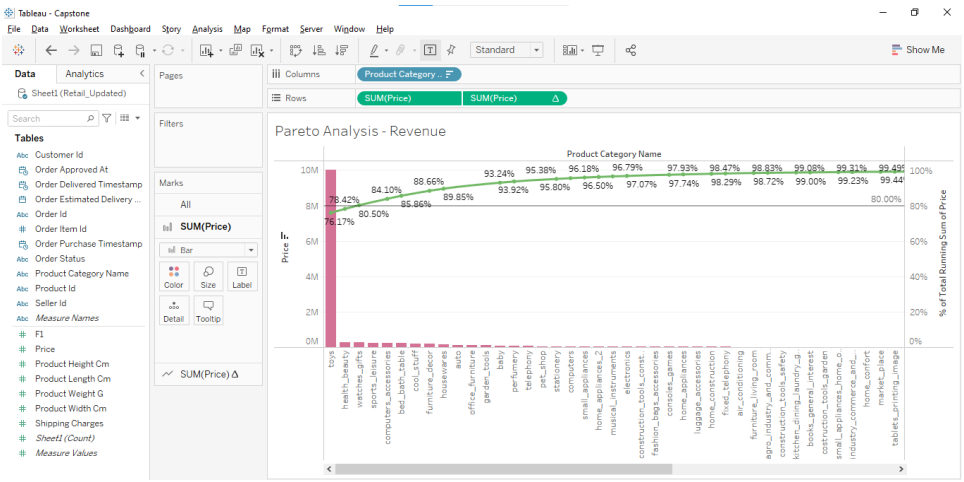
Inference:

- Among the top 20 products based on Quantity, 14 of them belong to the 'Toys' category.
- The other categories are 'Computer accessories' ,'garden_tools','health_beauty' and 'Watches_gifts'.
- 'Home_comfort_2','Security_and_services', diapers_and_hygeine', fashion_childrens_clothes','furniture_mattress_and_upholestry' are the bottom 5 product categories based on revenue.

Pareto Analysis – Revenue

- Plotted a bar chart against Product Category and Sum of Price.
- Plotted a Running total of Sum of Price on Dual Axis.
- Percentage Running total was added as a secondary Calculation.
- A reference line was added at 80% to identify the categories which have a major contribution towards the Revenue.

Image 20:



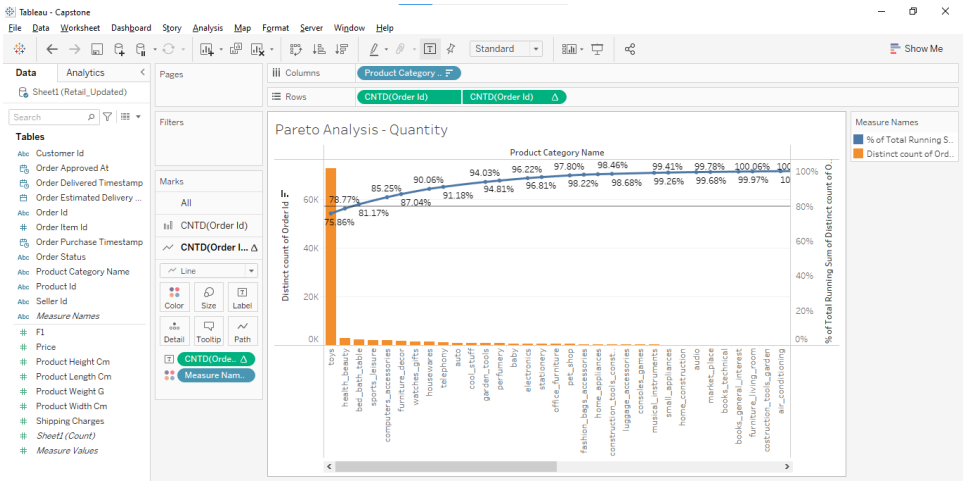
Inference:

- 76 % of the Revenue is coming from 'Toys' Product categories.
- 80% of the Revenue is coming from 'Toys' and 'Health_beauty' product categories.

Pareto Analysis – Quantity

- Plotted a bar chart against Product Category and Distinct count of Order ID.
- Plotted a Running total of Sum of Distinct count of Order ID on Dual Axis.
- Percentage Running total was added as a secondary Calculation.
- A reference line was added at 80% to identify the categories which ordered the most.

Image 21:



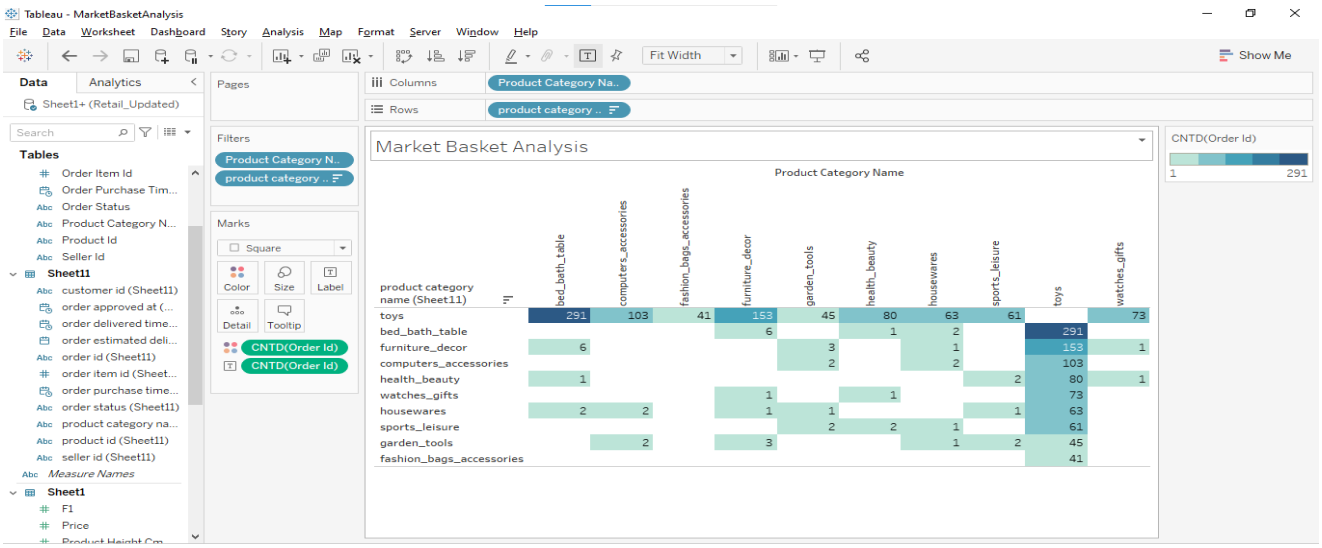
Inference:

- 76 % of the Orders belong to 'Toys' Product categories.
- 80% of the Orders belong to 'Toys' and 'Health_beauty' product categories.

Market Basket Analysis

- Joined the tables using self-join on Order ID.
- Plotted 'Product Category Name' against 'Product Category Name'
- Distinct count of Order ID was added to Label.
- Filter has been applied to view the top product combinations.

Image 22:



Inference:

- 'Toys' and 'Bed_bath_Table' combination is the most frequently ordered combination.
- 'Toys' and 'Furniture_décor' is the second most ordered combination.
- It is followed by 'Toys' and 'Computer Accessories'.

Key Findings

Top Selling Product Categories :

- Toys
- Computer_Accessories
- Watches_gifts
- Health_beauty

Most Preferred product Combinations:

- Toys and Bed_bath_Table
- Toys and Furniture_décor
- Toys and Computer Accessories

Least Selling Product Categories:

- Home_comfort_2
- Security_and_services
- Diapers_and_hygeine
- Fashion_childrens_clothes
- Furniture_mattress_and_upholestry

Recommendations:

- The company should focus more on the top selling product categories.
- The most preferred product combinations need to be stocked up and listed among suggestions.
- The company can either have less stock or can even completely remove the least selling product categories to save on inventory cost.

Submitted By

Sumitha T