**Data Cleaning**

**Importing Data**

* Imported CSV tables “**fact\_sales**”, “**dim\_producsts**” and “**dim\_customers**” to Power Bi.
* Changed the name of the tables to “**Sales**”, “**Products**” and “**Customers**” respectively.
* Corrected the automatic relationships setup by Power Bi so that the “**Products**” table is related to the “**Sales**” table on a one to many relationship using “**Stock Code**” column.
* Created a measure “**Total Rows**” for the “**Sales**” table and visualized it as a card.
* Created a measure “**Count of Invoice No**” to count the total number of invoices.

**Data Clean Up**  
Business Users have informed us that all valid transactions have an “**Invoice No**”. So we filtered all “null” values from the sales table in Power Query.

The “**Order State**” column in the “**Customers**” table needs cleaning since there are different entries for the same states. So we’ll and a new table for the states and connect it to the customer table so that we can have more accurate data.

* We uploaded the “**state\_region\_mapping**” CSV table to the file.
* Edited it in Power Query to use first row as headers.
* Renamed the table to “**State Mapping**”.
* Categorized the “State” column to “**State or Province**”.
* Created a many to many relationship between the “**State Mapping**” and “**Customers**” table.

**Customer Metrics**

Now that we have cleaned the dataset, we’ll create a few customer metrics like to understand how many customers the company has and their overall business.

* Created a measure “**Number of Customers**” that counts the unique number of customers from the sales table.
* Created a measure “**Customer LTV (avg)**” to see the average customer lifetime by dividing the sum of sales by the total number of customers

**Quantities**

We want to have a look at the total sales by average quantity. Using Power Query we replaced all “Indoor Pet Camera (Wi-Fi)” with “Indoor Pet Camera” so that it’s not represented as 2 entities.

Still in Power Query, we duplicated the sales table and name the new table “Invoice Totals”. I grouped the “Quantity” and “Sales” column by the “Invoice No” column to and named the new columns “Total Quantity” and “Total Sales” respectively. We did this to see quantity and sales per invoice.

**Market Basket Analysis**

We’ll have to duplicate the sales table and create a new relationship to understand which products are purchased more often in combination with other specific products they select.

* We duplicated the “Sales” table and named it “Market Basket”
* We connected the 2 table with a “many to many” relationship

**Shipping Costs (What-if)**

We added the “**Shipping\_Cost\_1000\_miles**” from the “**Products**” table to the “**Sales**” table using the DAX measure “**RELATED()**” and we renamed it to “**Shipping Cost**”.

To calculate this shipping cost we created a new measure in the “**Sales**” table and called it “**Shipping (Baseline)**”. This measure checks every row of the sales table, if the quantity is 1, then the original shipping cost is returned, but if the quantity is more than 1, we add the base shipping cost of the first item to the discounted shipping costs of the rest of the items (70% of the base shipping cost). For example we have **5** items with a base shipping cost of **$10**. It’s calculate this way: **$10** (*base shipping cost*) + (**4** (*quantity of discounted shipping cost items*) \* **$7**) = **$38** (*total shipping*).

The slicer we added is basically a parameter that allows you to control the quantity of items so you can see the shipping costs.

Added a new measure “**Blended Shipping Cost Factor**” to the “**What-if Quantity**” table. This measure consists of nested IF statements that calculated the discounted shipping costs based on the table below:

|  |  |
| --- | --- |
| What-if quantity value (less than or equal) | Cost Multiplier (output) |
| 1 | 1 |
| 2 | 0.8 |
| 4 | 0.6 |
| 7 | 0.5 |
| 9 | 0.4 |
| else | 0.3 |

This measure is created to be dynamic and allow users to try different values for effective shipping rates rather than using a fixed value of 0.7.

We created a new measure “**Shipping (What-if)**” which is basically the same as the measure “**Shipping (Baseline)**”we created before, only difference here is that we exchanged the fixed value 0.7 with “**Blended Shipping Cost Factor**” which is a dynamic value.

Created another measure “**Shipping (Difference)**” that calculated the difference between “**Shipping (Baseline)**” and “**Shipping (What-if)**”.

**Shipping Metrics**

We need to find the running total for the baseline shipping. To do this we created a measure “**Baseline Running Total**” by ensuring that all values are selected, and each new total is summed for a date before the max “**Transaction Date**”.

We added the same measure for the what-if shipping and difference shipping. We added all of these measures to the area chart.

**Revenue and Profits**

We Added a new column cost of good sold “**COGS**” to the “Sales” table. We did that by multiplying “**Landed Cost**” with “**Quantity**”.

Created another column “**Profit (Baseline)**” that subtracts the “**COGS**” and “**Shipping (Baseline)**” from the “**Sales**”.

Defined a new measure “**Profit %**” that divides the sum of baseline profit by the sum of sales.