Estimate of days left of stock by Product for product supply (based on average recent sales)

The project aimed to estimate how long the current stock of each product will last (in days or months) across various warehouses. This calculation used several factors: monthly sales per product across different U.S. states, current stock quantities at each warehouse, incoming product quantities from purchase orders (POs), and quantities from pre-orders for each product. By considering the average monthly sales from the past few months, alongside the current stock, incoming POs, and pre-order quantities (which are on a waiting list due to stock shortages), customers could make more informed purchasing decisions. This was based on the estimated time until stock runs out for each product at each warehouse. The results were presented in an easy-to-read Google Sheet.

1	SKU-Product	Average Monthly sales (last 3 months)	Avg QTY sold per day	Stock	Preorder Quantity	QTY in open PO	Total stock + PO- Preorders	**Total days left of Stock**
2	Product1	120	4.00	90.00	13	118	195.00	48.75
3	Product2	104	3.47	20.00	15	146	151.00	43.55769231
4	Product3	88	2.93	49.00	13	130	166.00	56.59090909
5	Product4	160	5.33	3.00	7	130	126.00	23.625
6	Product5	184	6.13	2.00	11	140	131.00	21.35869565
7	Product6	96	3.20	7.00	10	84	81.00	25.3125
8	Product7	136	4.53	10.00	16	96	90.00	19.85294118
9	Product8	176	5.87	0.00	17	158	141.00	24.03409091
10	Product9	56	1.87	25.00	3	70	92.00	49.28571429
11	Product10	8	0.27	0.00	1	24	23.00	86.25
12	Product11	24	0.80	1.00	0	34	35.00	43.75
13	Product12	24	0.80	10.00	0	0	10.00	12.5
14	Product13	64	2.13	5.00	1	58	62.00	29.0625
15	Product14	8	0.27	0.00	2	34	32.00	120
16	Product15	24	0.80	6.00	1	14	19.00	23.75
17	Product16	64	2.13	4.00	3	52	53.00	24.84375
18	Product17	16	0.53	1.00	0	8	9.00	16.875
19	Product18	32	1.07	1.00	2	18	17.00	15.9375
20	Product19	8	0.27	0.00	1	12	11.00	41.25
21	Product20	8	0.27	0.00	1	16	15.00	56.25
22	Product21	32	1.07	0.00	3	26	23.00	21.5625
23	Product22	8	0.27	0.00	0	44	44.00	165
24	Product23	24	0.80	0.00	4	10	6.00	7.5
25	Product24	32	1.07	1.00	0	38	39.00	36.5625
27	Product25	16	0.53	2.00	1	24	25.00	46.875
28	Product26	16	0.53	0.00	1	0	-1.00	-1.875
30	Product27	0	0.00	2.00	0	16	18.00	0
31	Product28	24	0.80	0.00	2	10	8.00	10
32	Product29	8	0.27	0.00	1	4	3.00	11.25
33	Product30	8	0.27	0.00	1	8	7.00	26.25
34	Product31	16	0.53	0.00	2	14	12.00	22.5
36	Product32	40	1.33	0.00	3	14	11.00	8.25

Data Engineering:

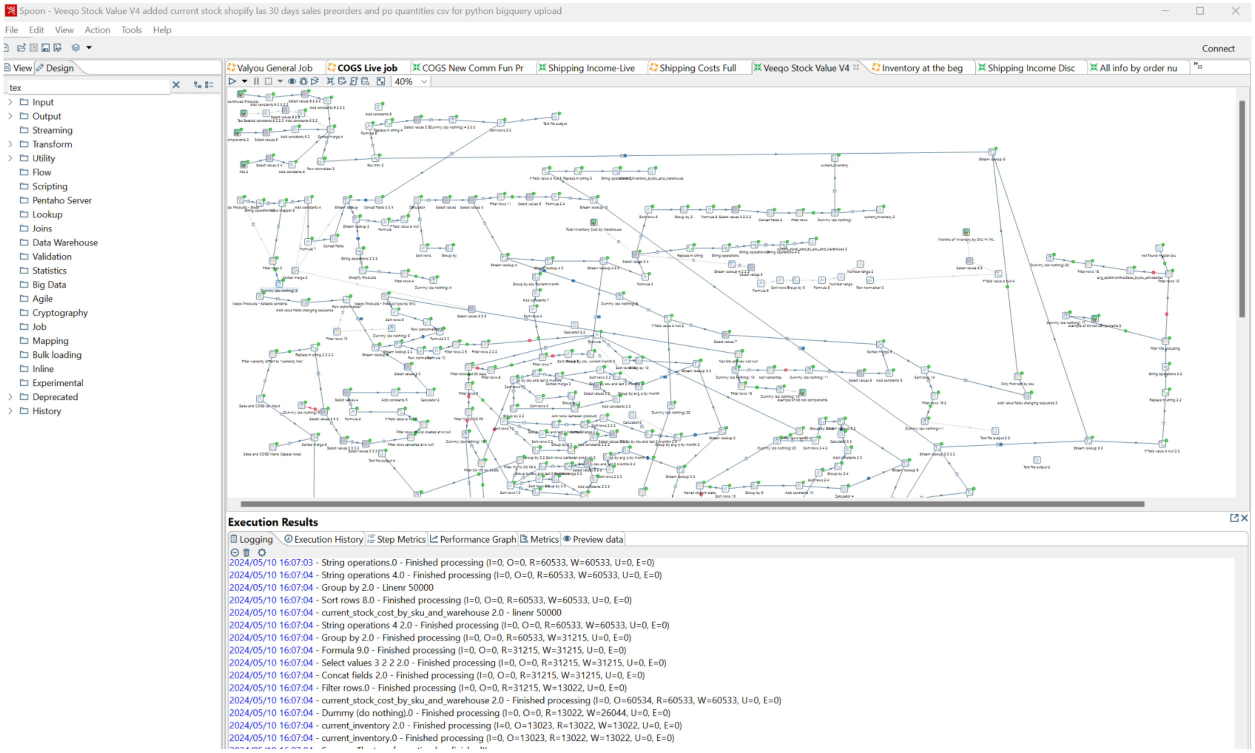
ETL Program with Pentaho Data Integration:

The resulting data set needed to display a row by each product in each warehouse with the current stock quantity, the average of products sold monthly in the last 3 months, the total pieces arriving with incoming POs, total quantity of pre orders (orders in waiting list because of lack of stock) and the estimate in days in stock left.

So with the desired resulting dataset in mind, the ETL program basically consisted in the integration of 4 data sources: sales data from Shopify, current stock (Veeqo inventory system), arriving stock from purchase orders (also from Veeqo) and preorders (MongoDB).

Separation of Kit products into Components for independent sales quantity calculation

Probably the biggest complexity of the Data Engineering process was to separate the quantity of sales by product, because in a very important proportion of cases, the products sold in Shopify as one product actually consisted in the sale of kits that grouped several products, each one with its corresponding SKU. So, the main task of the ETL program was to perform this separation of kits into components to calculate the actual sold quantities by each SKU.



Basic data preparation for PO quantity, Pre Order quantities and Stock quantities by product and warehouse.

The data preparation for the other 3 data sources was simple. The goal was to make it easy to upload the data into Big query to then be able to apply SQL language to make the merging or final integration of the data.

Since each datasource contained important information that could be used for independent analysis, each of the 4 data sources were stored in 4 different Big Query tables.

.

Uploading of Resulting Data to Google Big Query:

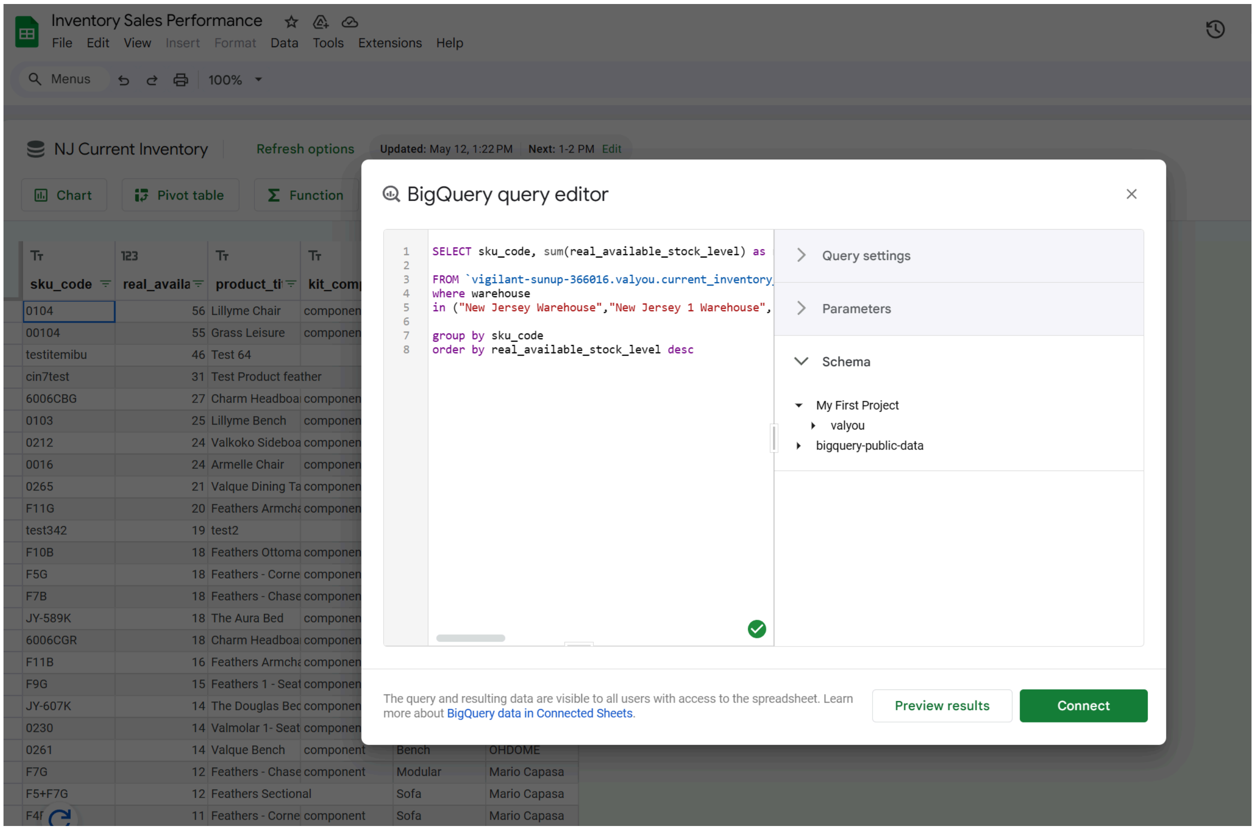
The resulting 4 CSVs file from the last step were then uploaded to 4 specific tables within a Google Big Query DataBase through the use of Python programming.

Bulk_load_python_fulfilled_lines_shipping_costs.py | Bulk_load_python_bigquery_last4monthssSales_bysku_allwebsites.py* | Bulk load inventory beggining c

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Jul 10 20:23:44 2023

@author: franc
"""

from pathlib import Path
import time
from google.cloud import bigquery
import os

def table_reference(project_id,dataset_id,table_id):
    dataset_ref=bigquery.DatasetReference(project_id, dataset_id)
    table_ref=bigquery.TableReference(dataset_ref,table_id)
    return table_ref

def upload_csv_sales(client, table_ref,csv_file):
    client.delete_table(table_ref, not_found_ok=True)
    load_job_configuration=bigquery.LoadJobConfig()
    load_job_configuration.schema=[
        bigquery.SchemaField('sku', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_1m_hi', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_1m_ml', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_quantity_by_month_1m_hi', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_quantity_by_month_1m_ml', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_1m__nj_ny_co_ph', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_quantity_by_month_1m_nj_ny_co_ph', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_1m_ca', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_quantity_by_month_1m_ca', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('product_title_msku', 'STRING', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_1m_ca_nc', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_quantity_by_month_1m_ca_nc', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_ml_3m', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_sold_by_month_3m_ml', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_hi_3m', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_sold_by_month_3m_hi', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_ny_3m', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_sold_by_month_3m_ny', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_ca_3m', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_sold_by_month_3m_ca', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('total_quantity_ca_nc_3m', 'NUMERIC', mode='NULLABLE'),
        bigquery.SchemaField('avg_sold_by_month_3m_ca_nc', 'NUMERIC', mode='NULLABLE'),
```

Connection of BigQuery tables with Google Sheets

The 4 resulting Qig Query tables were then connected to a Google Sheet through Google Sheet connect and SQL



Data Visualization:

Creation of Summary Google Sheet with estimate of days left in stock by SKU(product)

After the connection with Big Query and the uploading of data, I created a Data Sheet with the information needed. Final displaying of data consist in:

- The average quantity sales of monthly sales of the last 3 months
- Average quantity of of pieces sold per day
- Total remaining Stock
- Total pre-order quantity
- Quantity in open PO
- Total Stock + PO- Pre Orders
- Total days Left estimation (Total Stock including POs and Preorders / Avg quantity sold per day)

Inventory Manager for product Supplying

SKU-Product	Average Monthly sales (last 3 months)	Avg QTY sold per day	Stock	Preorder Quantity	QTY in open PO	Total stock + PO- Preorders	Total days left
Product1	120	4.00	90.00	13	118	195.00	48.75
Product2	104	3.47	20.00	15	146	151.00	43.55769231
Product3	88	2.93	49.00	13	130	166.00	56.59090909
Product4	160	5.33	3.00	7	130	126.00	23.625
Product5	184	6.13	2.00	11	140	131.00	21.35869565
Product6	96	3.20	7.00	10	84	81.00	25.3125
Product7	136	4.53	10.00	16	96	90.00	19.85294118
Product8	176	5.87	0.00	17	158	141.00	24.03409091
Product9	56	1.87	25.00	3	70	92.00	49.28571429
Product10	8	0.27	0.00	1	24	23.00	86.25
Product11	24	0.80	1.00	0	34	35.00	43.75
Product12	24	0.80	10.00	0	0	10.00	12.5
Product13	64	2.13	5.00	1	58	62.00	29.0625
Product14	8	0.27	0.00	2	34	32.00	120
Product15	24	0.80	6.00	1	14	19.00	23.75
Product16	64	2.13	4.00	3	52	53.00	24.84375
Product17	16	0.53	1.00	0	8	9.00	16.875
Product18	32	1.07	1.00	2	18	17.00	15.9375
Product19	8	0.27	0.00	1	12	11.00	41.25
Product20	8	0.27	0.00	1	16	15.00	56.25
Product21	32	1.07	0.00	3	26	23.00	21.5625
Product22	8	0.27	0.00	0	44	44.00	165
Product23	24	0.80	0.00	4	10	6.00	7.5
Product24	32	1.07	1.00	0	38	39.00	36.5625
Product25	16	0.53	2.00	1	24	25.00	46.875
Product26	16	0.53	0.00	1	0	-1.00	-1.875
Product27	0	0.00	2.00	0	16	18.00	0
Product28	24	0.80	0.00	2	10	8.00	10
Product29	8	0.27	0.00	1	4	3.00	11.25
Product30	8	0.27	0.00	1	8	7.00	26.25

Impact for the company

As a direct result of the use of this sheet, the company was able to better handle the inventory, reducing the frequency of products out of stock that in the past affected the company sales or the customer experience. Before this information was available it was really difficult for the company to make purchasing decisions based on actual information. Very frequently they would unexpectedly run out of stock of several different products which resulted in order delivery problems, order cancellations, or loss of sales due to lack of stock.

In the last months they have significantly reduced their order cancellation rates, preorder rates and customer claims related to late deliveries.