Inventory Forecasting & Sales Analysis Project
Self-Initiated | Mar 2025

🌍 Project Overview

This project simulates a real-world inventory environment using synthetic retail sales data across 1,000 SKUs, 5 product categories, and 10 vendors over 52 weeks. The objective is to analyze sales performance, identify inventory challenges, and build foundational forecasting insights to support inventory planning and replenishment decisions.

📊 Dataset Summary

- **Data Source:** Simulated synthetic data
- **Structure:** 12 months of weekly SKU-level sales
- **Fields:** SKU_ID, Category, Vendor, Week, Store_Location, Units_Sold, Unit_Cost, Lead_Time_Days
- **Total Records:** 52,000

💡 Methodology & Analysis

1. Data Cleaning

- Removed missing `Units_Sold` and `Store_Location` values
- Standardized data types (e.g., categorical conversion, numeric fields)
- Created `Total_Cost` field to assist with cost-based analysis
- Exported cleaned dataset for use across tools (Excel, Power BI, SQL)

[7]:
```python
# Loading datasets
df = pd.read_csv(r'C:\Users\brivi\Downloads\My Portfolio\inventory-forecasting-project\data\unstructured_sku_sales_data.csv')

# Previewing top 5 rows
df.head()
```

[7]:

	SKU_ID	Category	Vendor	Week	Store_Location	Units_Sold	Unit_Cost	Lead_Time_Days
0	SKU_0359	Electronics	Vendor_1	42	Store_10	18.0	75.95	29
1	SKU_0062	Toys	Vendor_10	51	Store_13	16.0	14.50	7
2	SKU_0458	Home & Kitchen	Vendor_1	38	Store_15	5.0	109.42	22
3	SKU_0346	Clothing	Vendor_7	35	Store_12	25.0	98.35	7
4	SKU_0188	Electronics	Vendor_5	23	Store_9	25.0	188.68	16

•[9]:
```python
# Checking if there are missing values
df.isnull().sum()
```

[9]:
```
SKU_ID               0
Category             0
Vendor               0
Week                 0
Store_Location    1040
Units_Sold        1040
Unit_Cost            0
Lead_Time_Days       0
dtype: int64
```

[11]:
```python
# Both the Store_Location and Units_Sold have same number of missing values: 1040
# Since 1040 missing values are significant portion of the total number of data
# We'll investigate further by chyecking if all 1040 rows belong to the same week/SKU
# Or if they're concentrated in a few vendors or categories
df[df['Units_Sold'].isnull()]['Vendor'].value_counts()
```

[11]:
```
Vendor
Vendor_2     130
Vendor_3     114
Vendor_10    109
Vendor_1     108
Vendor_7     105
Vendor_9     105
Vendor_8     103
Vendor_5      99
Vendor_6      91
Vendor_4      76
Name: count, dtype: int64
```

[46]:
```python
df = df.dropna(subset=['Store_Location', 'Units_Sold'])
df.isnull().sum()
```

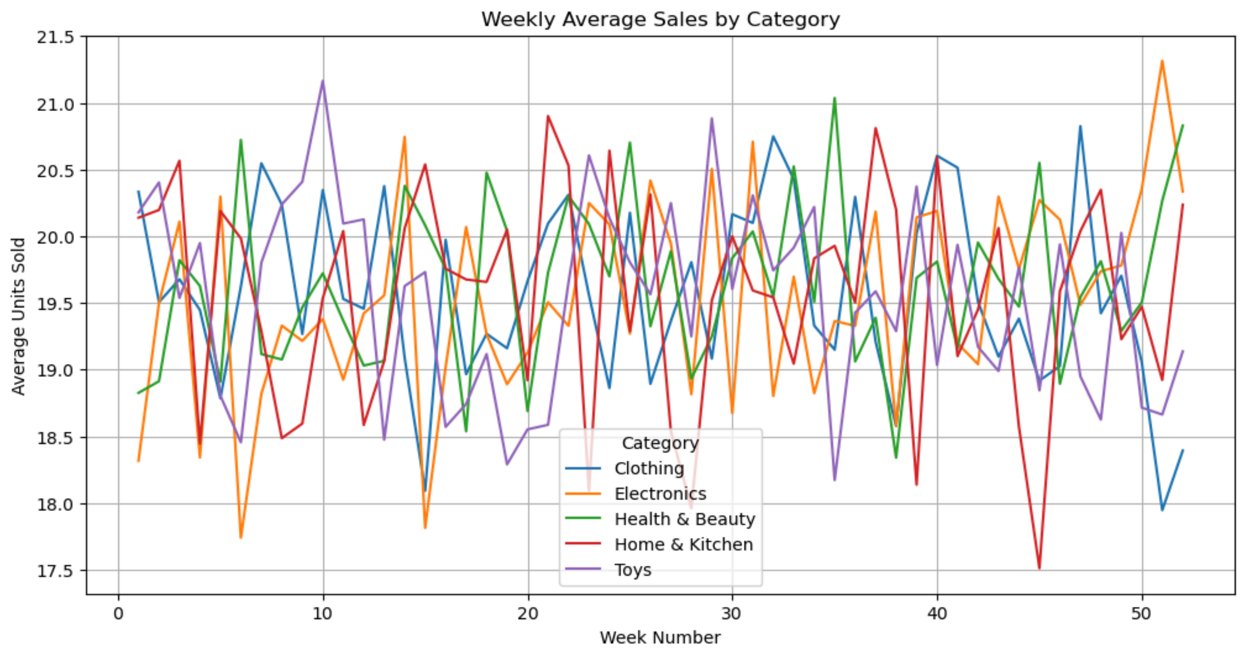
[46]:
```
SKU_ID            0
Category          0
Vendor            0
Week              0
Store_Location    0
Units_Sold        0
Unit_Cost         0
Lead_Time_Days    0
dtype: int64
```

2. Exploratory Data Analysis (Python)

- Performed using Jupyter Notebooks and Pandas
- Top SKUs identified: `SKU_0204`, `SKU_0593`, `SKU_0913`
- Vendor with highest volume: `Vendor_2` (109K+ units, 15-day lead time)
- Visualized stockouts (e.g., `SKU_0247` with 7 weeks of zero sales)
- Line, bar, and box plots created using Seaborn and Matplotlib

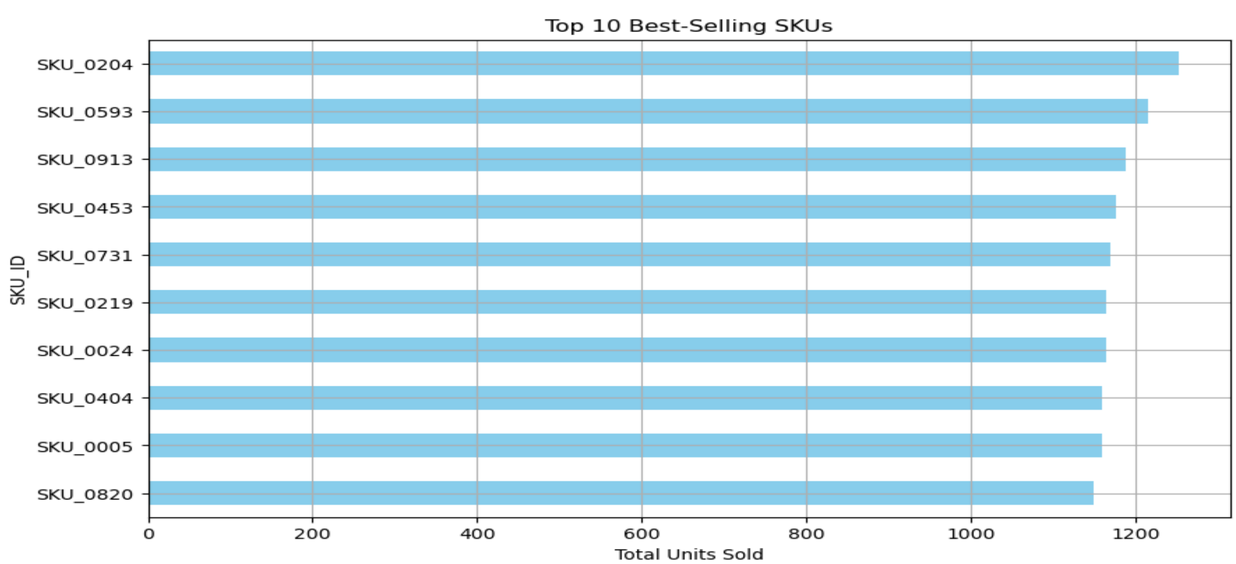
a. Weekly Average Sales by Category

Line plot analysis reveals consistent demand across all five categories with minor fluctuations, reflecting a fairly stable demand environment for weekly inventory planning.



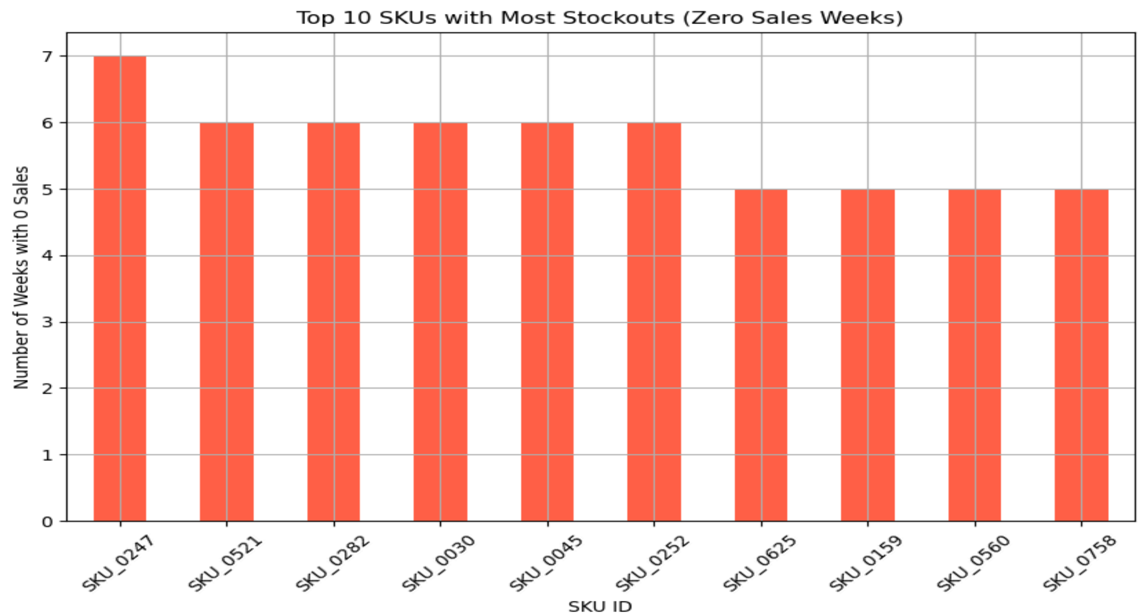
b. **Top 10 Best-Selling SKUs**

Horizontal bar chart shows that SKU_0204 led total annual sales, followed closely by other high-volume SKUs. These products should be prioritized in reorder strategies and monitored closely.



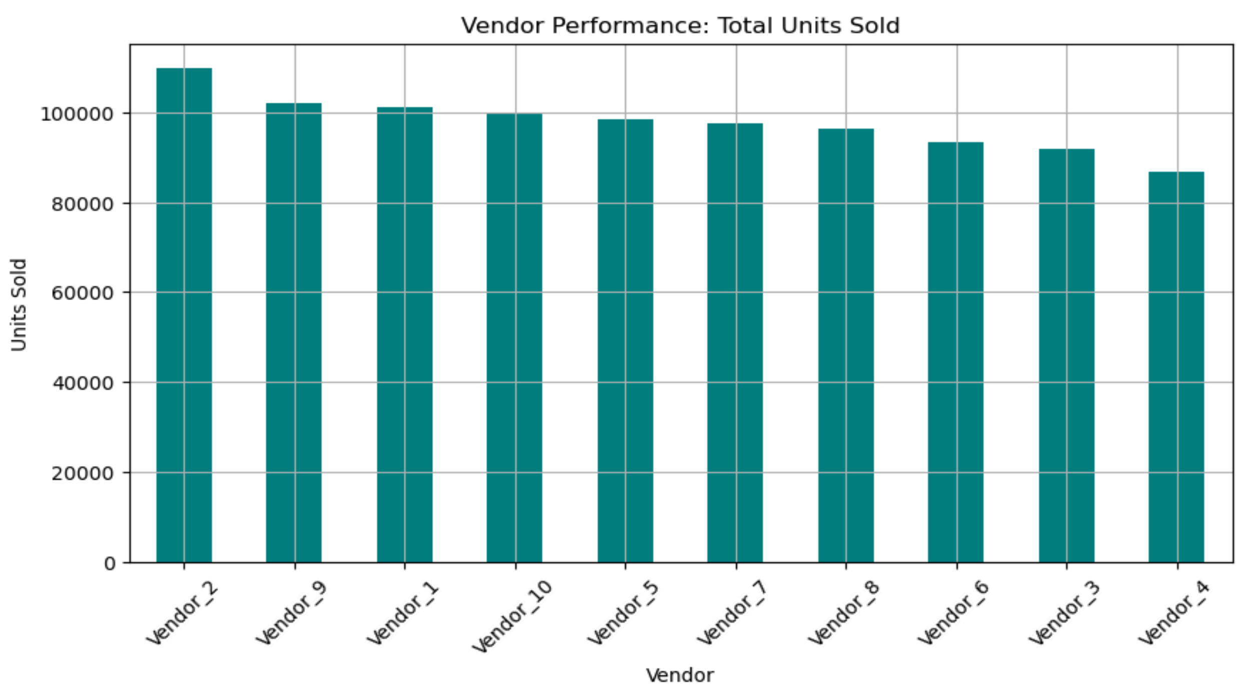
c. **Stockout Frequency by SKU**

Several SKUs, such as SKU_0247 and SKU_0521, had 5+ weeks of zero sales, suggesting stockout risks or poor demand. These require either better forecasting or strategic phasing out.

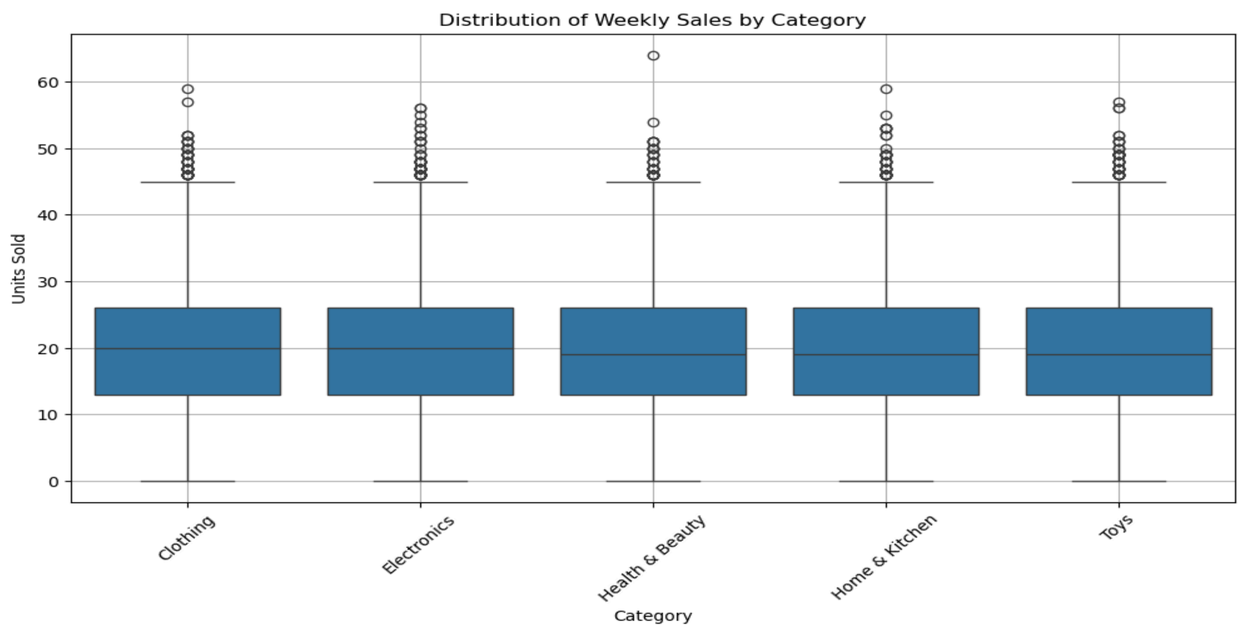


d. Vendor Performance Summary

Vendor_2 had the highest total units sold, followed by Vendor_9 and Vendor_1. Vendor performance evaluation helps prioritize partners for reliable replenishment and lead time planning.

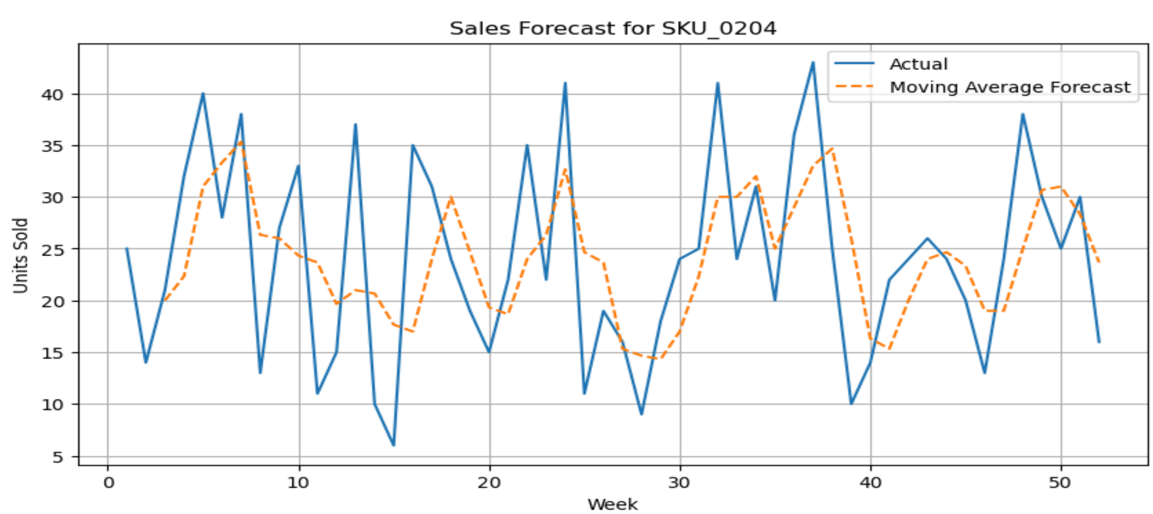


e. **Weekly Sales Variability by Category**
Box plot highlights that most categories have similar sales spread with noticeable outliers. Some categories like Health & Beauty showed higher variability, useful for safety stock considerations.



2. Forecasting (Python)

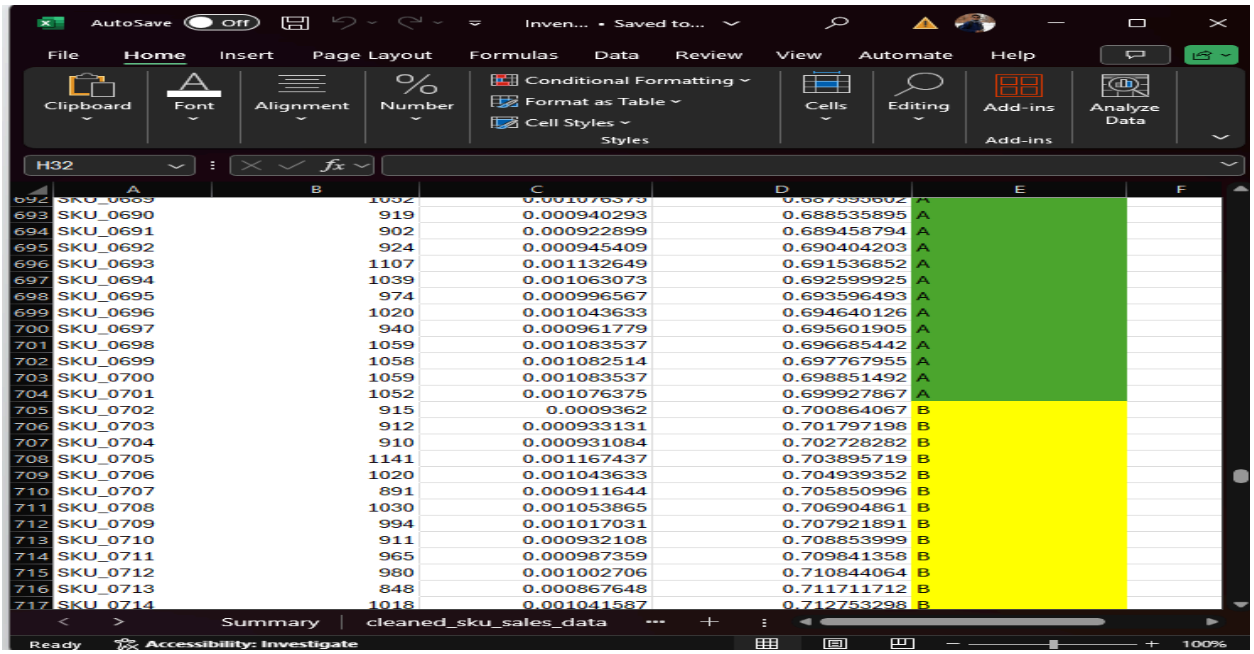
- Used a 3-week Moving Average for demand forecasting
- Focused on high-selling SKUs like `SKU_0204`
- Evaluated model performance: MAE = 6.74, RMSE = 8.04
- Insights support short-term planning for top SKUs.



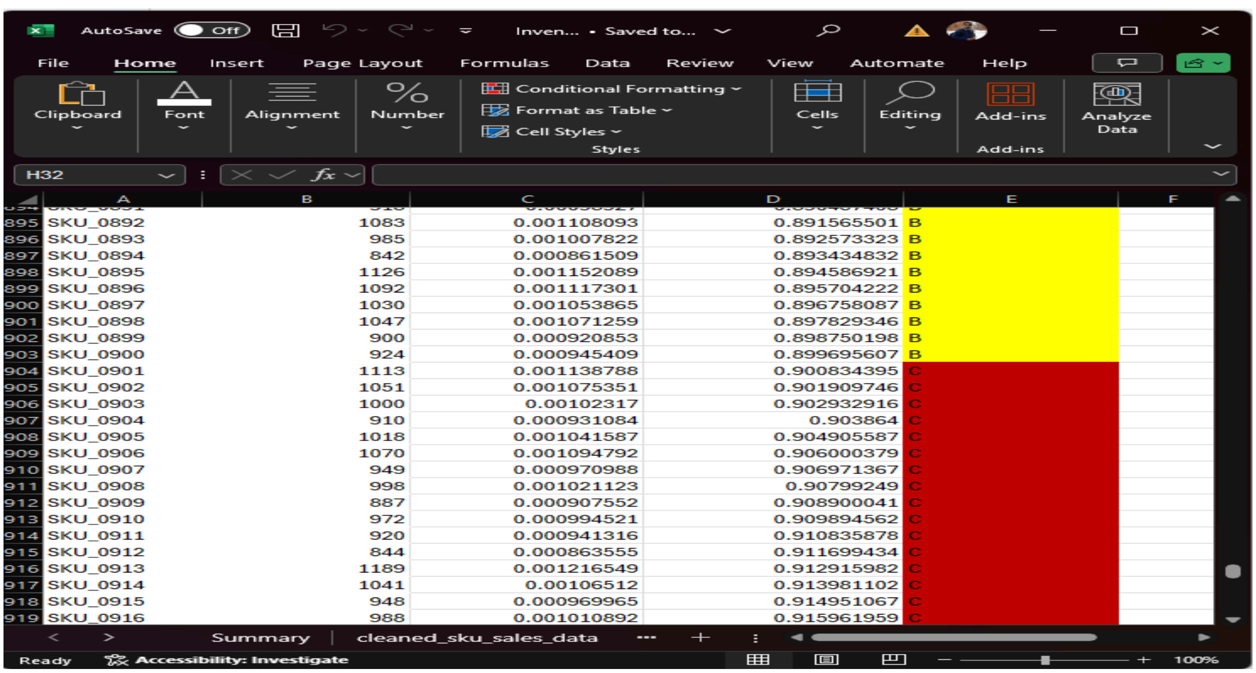
3. Excel Analysis

- **Tools used:** IF, VLOOKUP, XLOOKUP, Conditional Formatting.

- **ABC Analysis**: 70% of sales driven by ~700 SKUs



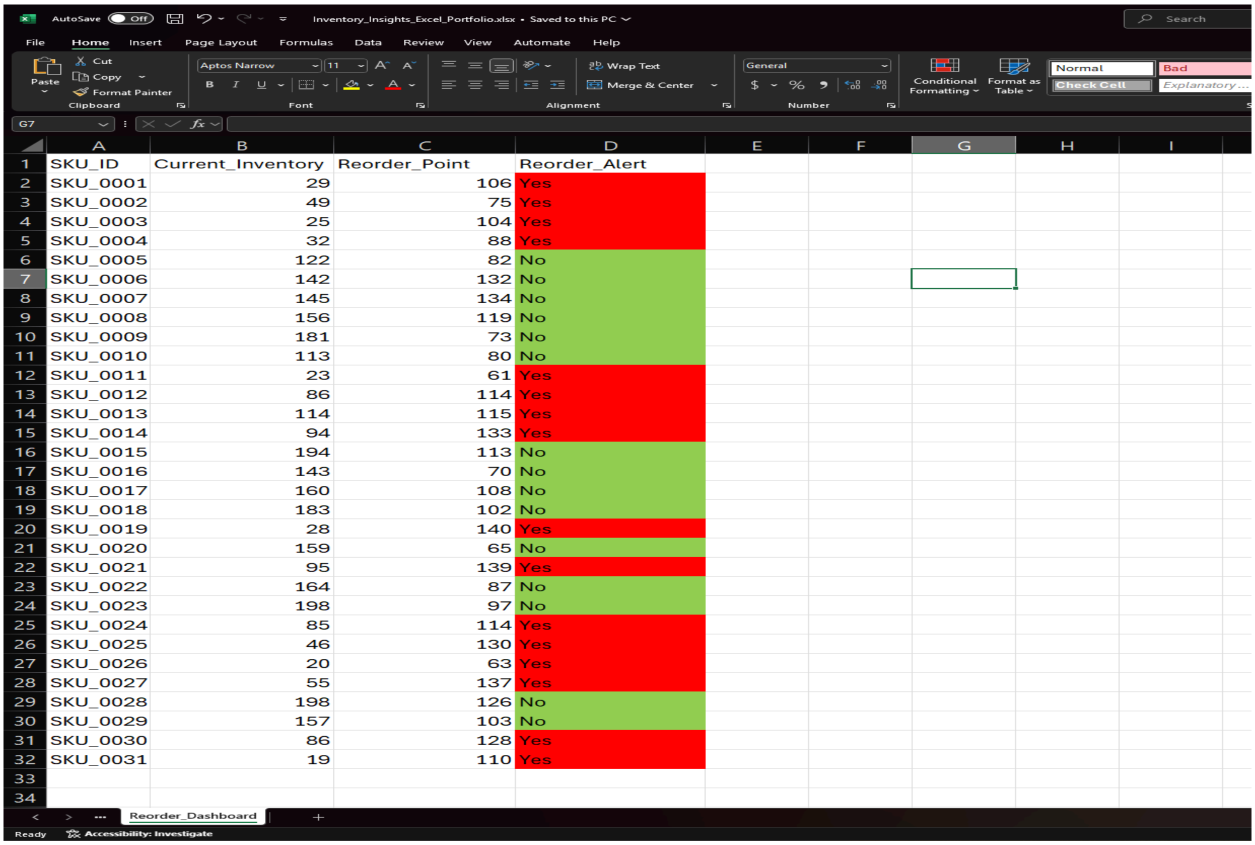
	A	B	C	D	E
693	SKU_0690	919	0.000940293	0.688535895	A
694	SKU_0691	902	0.000922899	0.689458794	A
695	SKU_0692	924	0.000945409	0.690404203	A
696	SKU_0693	1107	0.001132649	0.691536852	A
697	SKU_0694	1039	0.001063073	0.692599925	A
698	SKU_0695	974	0.000996567	0.693596493	A
699	SKU_0696	1020	0.001043633	0.694640126	A
700	SKU_0697	940	0.000961779	0.695601905	A
701	SKU_0698	1059	0.001083537	0.696685442	A
702	SKU_0699	1058	0.001082514	0.697767955	A
703	SKU_0700	1059	0.001083537	0.698851492	A
704	SKU_0701	1052	0.001076375	0.699927867	A
705	SKU_0702	915	0.0009362	0.700864067	B
706	SKU_0703	912	0.000933131	0.701797198	B
707	SKU_0704	910	0.000931084	0.702728282	B
708	SKU_0705	1141	0.001167437	0.703895719	B
709	SKU_0706	1020	0.001043633	0.704939352	B
710	SKU_0707	891	0.000911644	0.705850996	B
711	SKU_0708	1030	0.001053865	0.706904861	B
712	SKU_0709	994	0.001017031	0.707921891	B
713	SKU_0710	911	0.000932108	0.708853999	B
714	SKU_0711	965	0.000987359	0.709841358	B
715	SKU_0712	980	0.001002706	0.710844064	B
716	SKU_0713	848	0.000867648	0.711711712	B
717	SKU_0714	1018	0.001041587	0.712753298	B



	A	B	C	D	E
895	SKU_0892	1083	0.001108093	0.891565501	B
896	SKU_0893	985	0.001007822	0.892573323	B
897	SKU_0894	842	0.000861509	0.893434832	B
898	SKU_0895	1126	0.001152089	0.894586921	B
899	SKU_0896	1092	0.001117301	0.895704222	B
900	SKU_0897	1030	0.001053865	0.896758087	B
901	SKU_0898	1047	0.001071259	0.897829346	B
902	SKU_0899	900	0.000920853	0.898750198	B
903	SKU_0900	924	0.000945409	0.899695607	B
904	SKU_0901	1113	0.001138788	0.900834395	C
905	SKU_0902	1051	0.001075351	0.901909746	C
906	SKU_0903	1000	0.00102317	0.902932916	C
907	SKU_0904	910	0.000931084	0.903864	C
908	SKU_0905	1018	0.001041587	0.904905587	C
909	SKU_0906	1070	0.001094792	0.906000379	C
910	SKU_0907	949	0.000970988	0.906971367	C
911	SKU_0908	998	0.001021123	0.90799249	C
912	SKU_0909	887	0.000907552	0.908900041	C
913	SKU_0910	972	0.000994521	0.909894562	C
914	SKU_0911	920	0.000941316	0.910835878	C
915	SKU_0912	844	0.000863555	0.911699434	C
916	SKU_0913	1189	0.001216549	0.912915982	C
917	SKU_0914	1041	0.00106512	0.913981102	C
918	SKU_0915	948	0.000969965	0.914951067	C
919	SKU_0916	988	0.001010892	0.915961959	C

Insight: ABC classification helped prioritize inventory attention. Reorder alerts triggered when Current Inventory < Reorder Point.

- **Reorder Dashboard**: Flagged SKUs under simulated reorder points

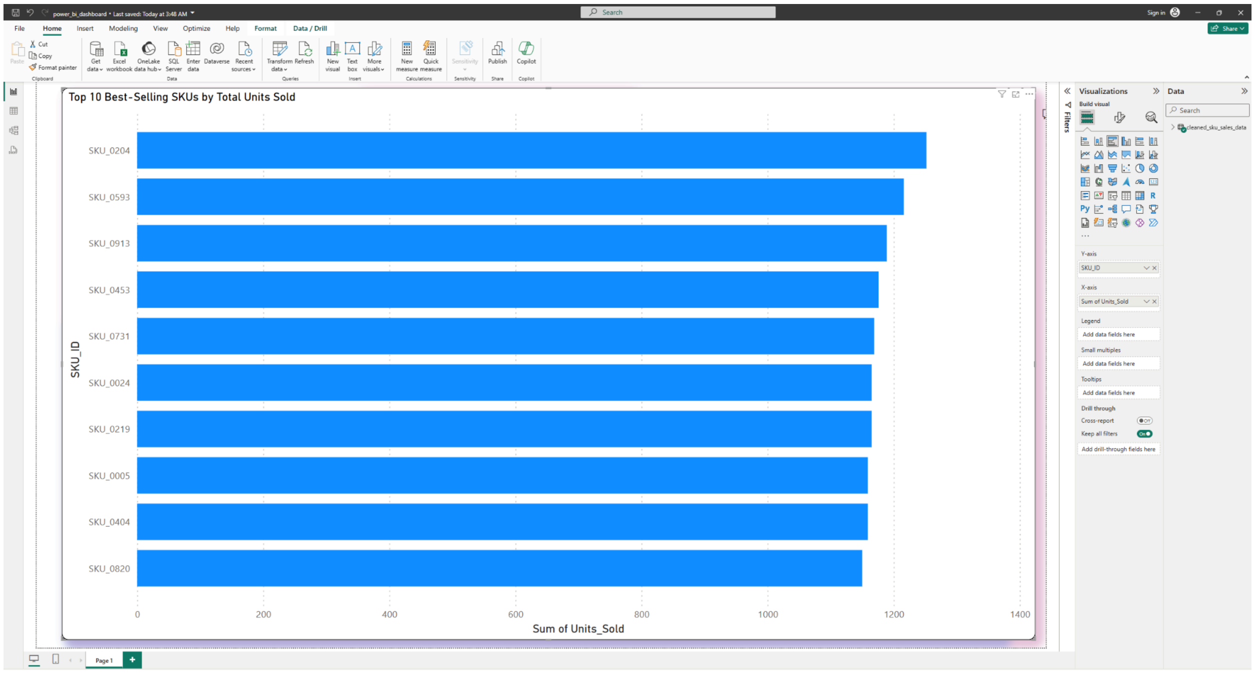


SKU_ID	Current_Inventory	Reorder_Point	Reorder_Alert
SKU_0001	29	106	Yes
SKU_0002	49	75	Yes
SKU_0003	25	104	Yes
SKU_0004	32	88	Yes
SKU_0005	122	82	No
SKU_0006	142	132	No
SKU_0007	145	134	No
SKU_0008	156	119	No
SKU_0009	181	73	No
SKU_0010	113	80	No
SKU_0011	23	61	Yes
SKU_0012	86	114	Yes
SKU_0013	114	115	Yes
SKU_0014	94	133	Yes
SKU_0015	194	113	No
SKU_0016	143	70	No
SKU_0017	160	108	No
SKU_0018	183	102	No
SKU_0019	28	140	Yes
SKU_0020	159	65	No
SKU_0021	95	139	Yes
SKU_0022	164	87	No
SKU_0023	198	97	No
SKU_0024	85	114	Yes
SKU_0025	46	130	Yes
SKU_0026	20	63	Yes
SKU_0027	55	137	Yes
SKU_0028	198	126	No
SKU_0029	157	103	No
SKU_0030	86	128	Yes
SKU_0031	19	110	Yes

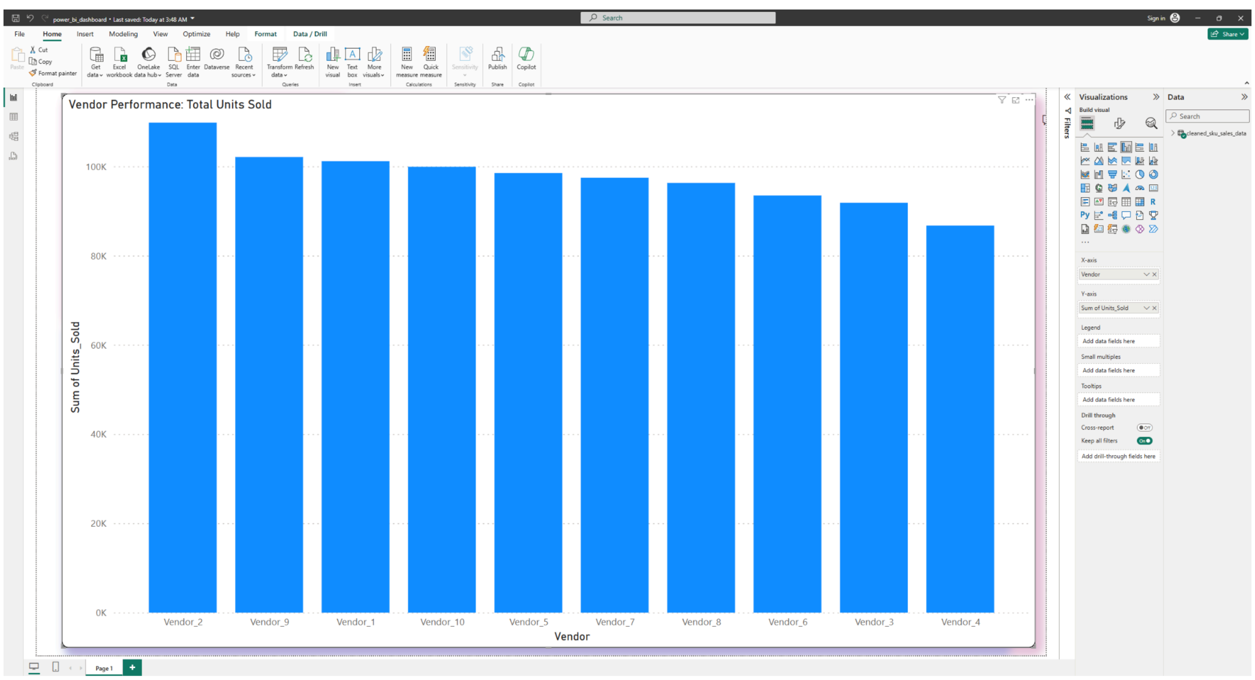
Insight: Dynamic dashboard highlighted SKUs at risk, improving visibility of potential stock gaps. If the **Current_Inventory** is less than **Reorder_Point,** it will trigger the Reorder_Alert by saying "Yes" or "No" and changing the color to Red/Green. This helps in determining when to order the finishing stock before it completely become out of stock and ends up hurting the business.

4. Power BI Dashboard

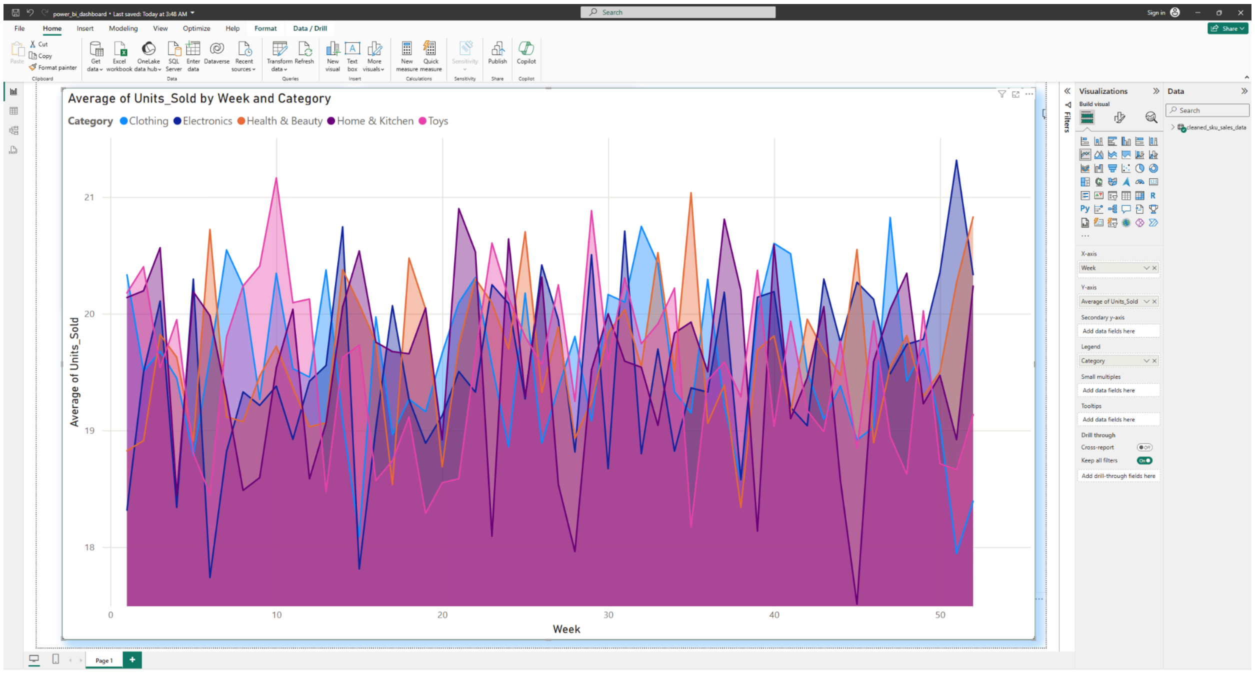
1. Created interactive charts on:
    a. **SKU-level sales**



b. Vendor performance



c. Stockout analysis

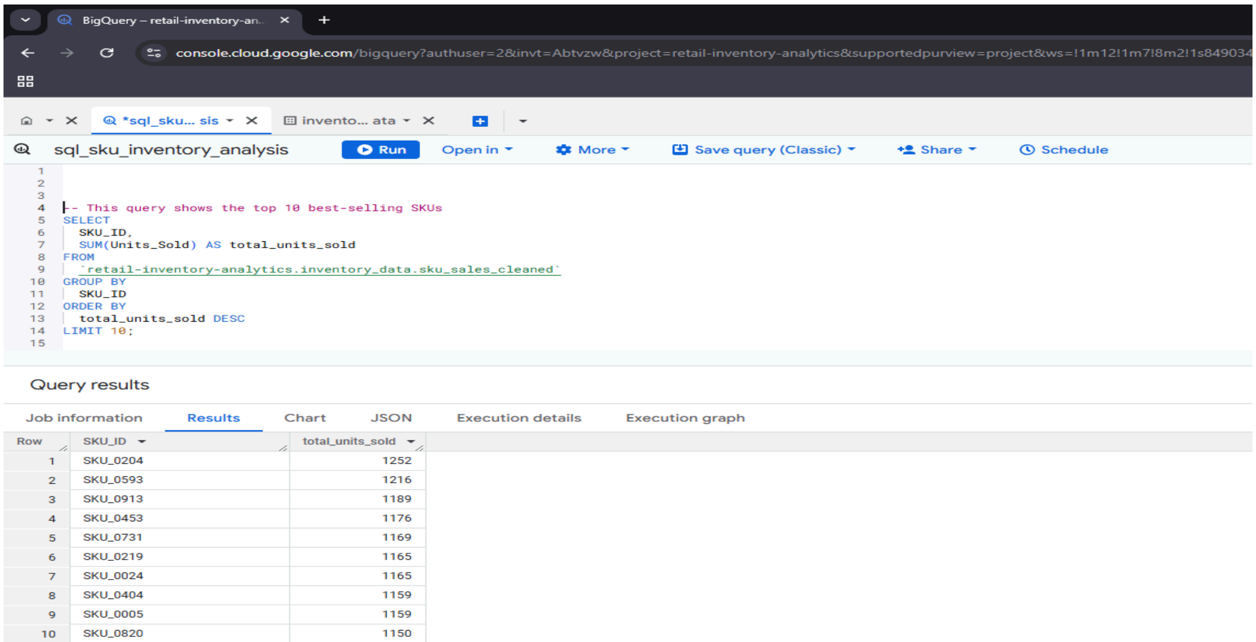


2. Used **slicers for filtering** by vendor, category, and week
3. Exported **dashboard visuals** to PDF

Insight: BI dashboard enabled category managers to filter performance by region, vendor, and SKU in real time.

5. SQL Analysis (BigQuery)

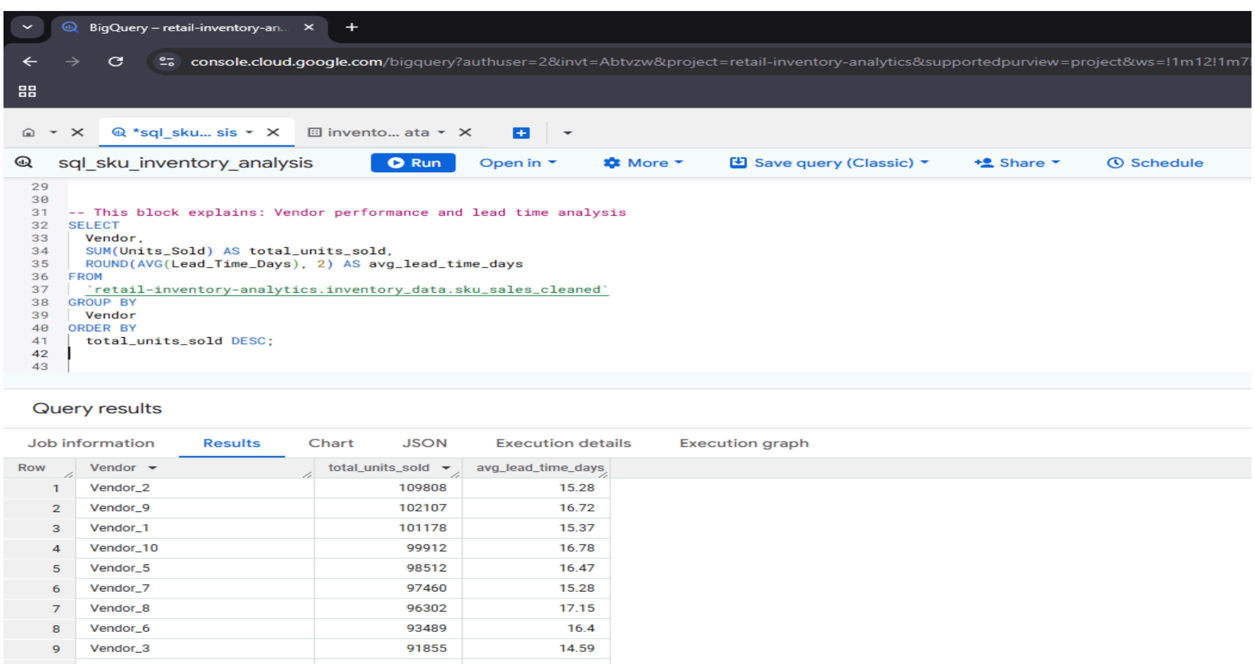
- Queried SKU sales and vendor performance in `sku_sales_cleaned`
- **Top SKUs**: Same as Python findings




```sql
-- This query shows the top 10 best-selling SKUs
SELECT
  SKU_ID,
  SUM(Units_Sold) AS total_units_sold
FROM
  `retail-inventory-analytics.inventory_data.sku_sales_cleaned`
GROUP BY
  SKU_ID
ORDER BY
  total_units_sold DESC
LIMIT 10;
```

Row	SKU_ID	total_units_sold
1	SKU_0204	1252
2	SKU_0593	1216
3	SKU_0913	1189
4	SKU_0453	1176
5	SKU_0731	1169
6	SKU_0219	1165
7	SKU_0024	1165
8	SKU_0404	1159
9	SKU_0005	1159
10	SKU_0820	1150

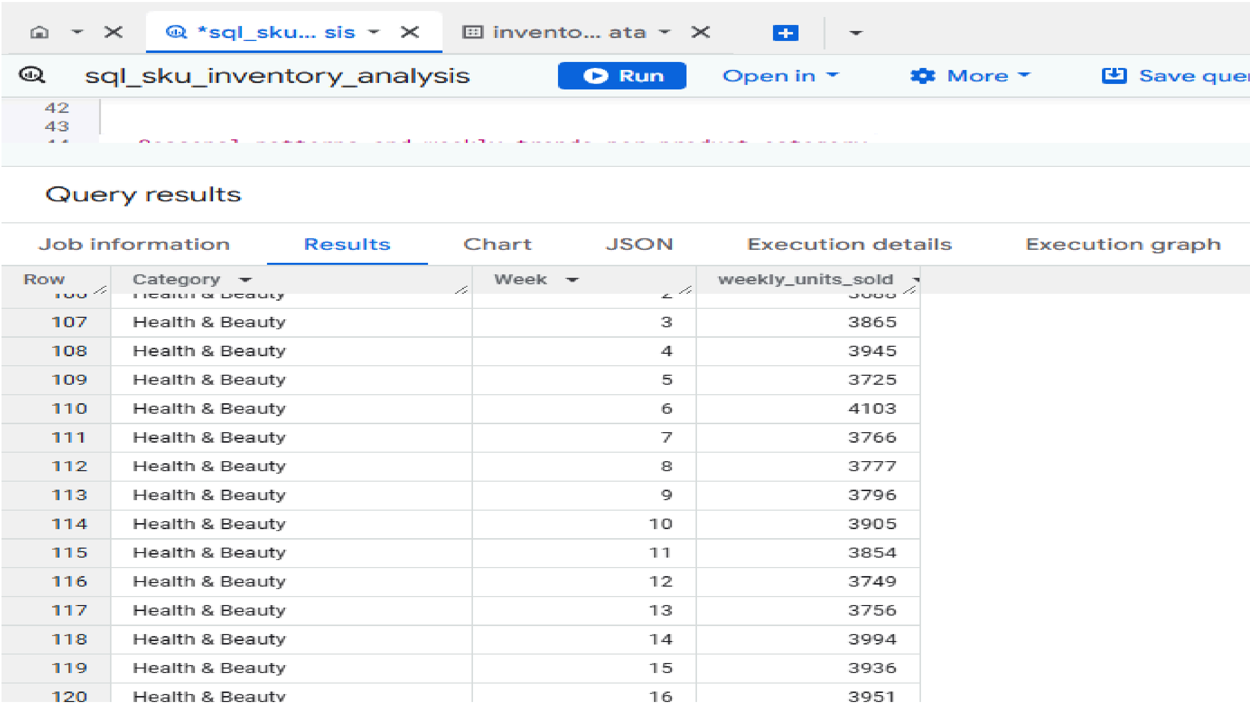
- **Vendors**: `Vendor_2`, `Vendor_9`, and `Vendor_1` led in volume



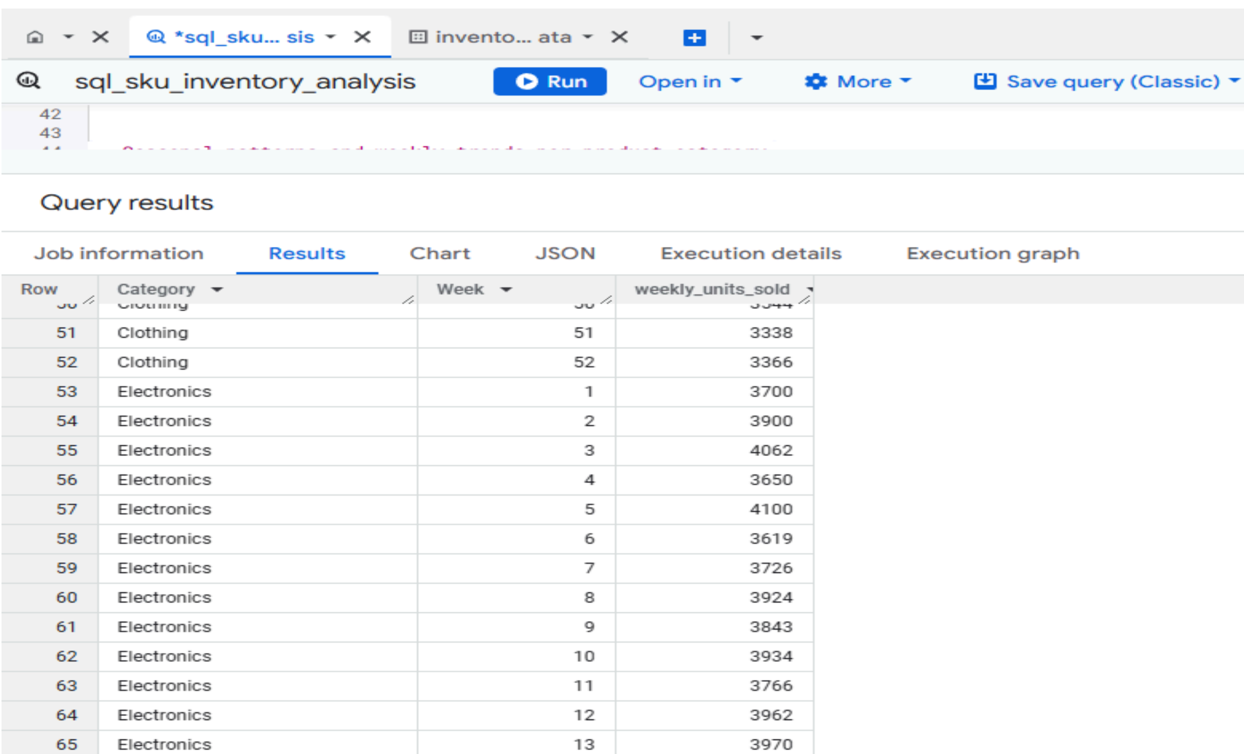

```sql
-- This block explains: Vendor performance and lead time analysis
SELECT
  Vendor,
  SUM(Units_Sold) AS total_units_sold,
  ROUND(AVG(Lead_Time_Days), 2) AS avg_lead_time_days
FROM
  `retail-inventory-analytics.inventory_data.sku_sales_cleaned`
GROUP BY
  Vendor
ORDER BY
  total_units_sold DESC;
```

Row	Vendor	total_units_sold	avg_lead_time_days
1	Vendor_2	109808	15.28
2	Vendor_9	102107	16.72
3	Vendor_1	101178	15.37
4	Vendor_10	99912	16.78
5	Vendor_5	98512	16.47
6	Vendor_7	97460	15.28
7	Vendor_8	96302	17.15
8	Vendor_6	93489	16.4
9	Vendor_3	91855	14.59
10	Vendor_4	86732	14.31

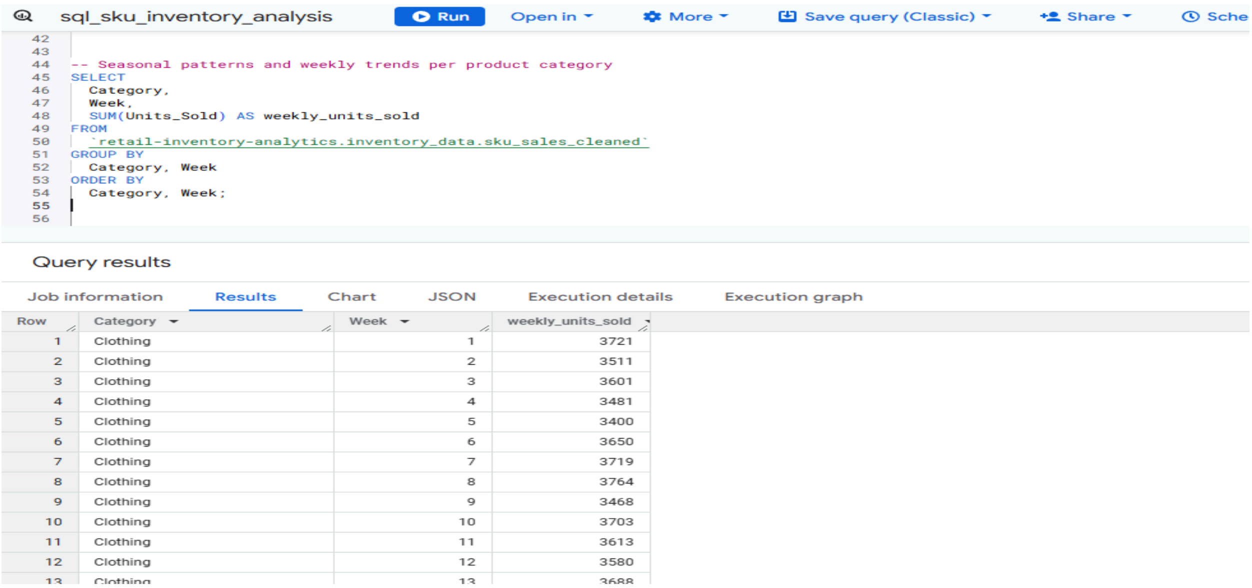
- **Stockouts**: SKUs like `SKU_0247` had 6+ zero-sales weeks



Row	Category	Week	weekly_units_sold
107	Health & Beauty	3	3865
108	Health & Beauty	4	3945
109	Health & Beauty	5	3725
110	Health & Beauty	6	4103
111	Health & Beauty	7	3766
112	Health & Beauty	8	3777
113	Health & Beauty	9	3796
114	Health & Beauty	10	3905
115	Health & Beauty	11	3854
116	Health & Beauty	12	3749
117	Health & Beauty	13	3756
118	Health & Beauty	14	3994
119	Health & Beauty	15	3936
120	Health & Beauty	16	3951



Row	Category	Week	weekly_units_sold
51	Clothing	51	3338
52	Clothing	52	3366
53	Electronics	1	3700
54	Electronics	2	3900
55	Electronics	3	4062
56	Electronics	4	3650
57	Electronics	5	4100
58	Electronics	6	3619
59	Electronics	7	3726
60	Electronics	8	3924
61	Electronics	9	3843
62	Electronics	10	3934
63	Electronics	11	3766
64	Electronics	12	3962
65	Electronics	13	3970



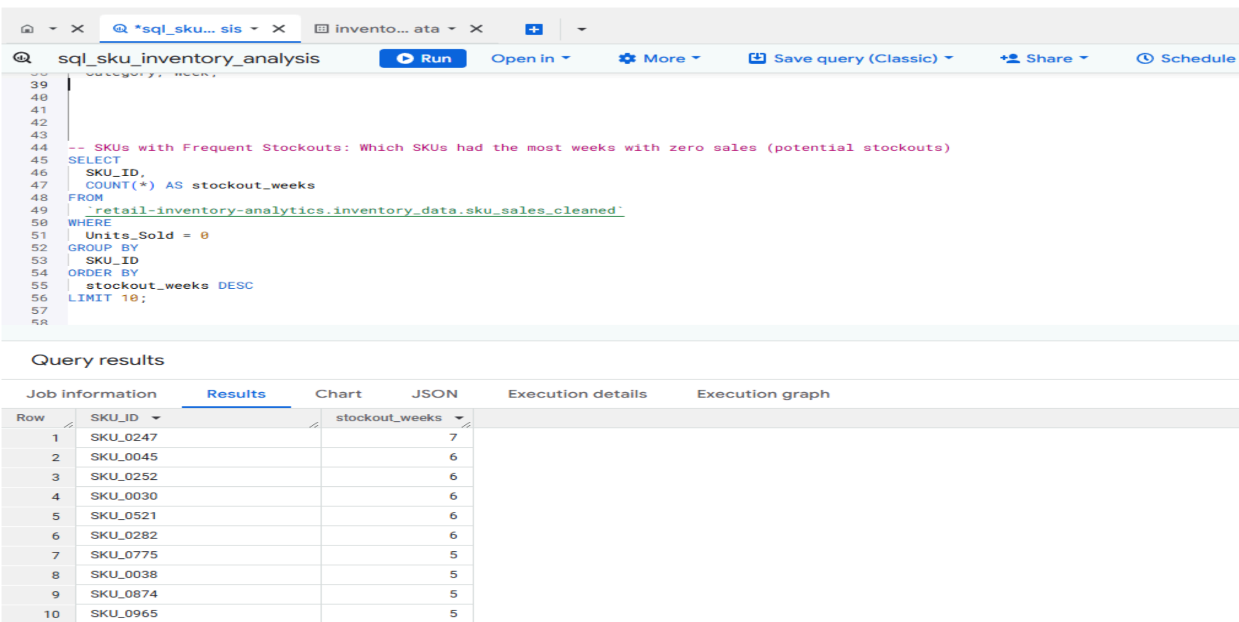
sql_sku_inventory_analysis

```sql
-- Seasonal patterns and weekly trends per product category
SELECT
  Category,
  Week,
  SUM(Units_Sold) AS weekly_units_sold
FROM
  `retail-inventory-analytics.inventory_data.sku_sales_cleaned`
GROUP BY
  Category, Week
ORDER BY
  Category, Week;
```

Query results

Row	Category	Week	weekly_units_sold
1	Clothing	1	3721
2	Clothing	2	3511
3	Clothing	3	3601
4	Clothing	4	3481
5	Clothing	5	3400
6	Clothing	6	3650
7	Clothing	7	3719
8	Clothing	8	3764
9	Clothing	9	3468
10	Clothing	10	3703
11	Clothing	11	3613
12	Clothing	12	3580
13	Clothing	13	3688

- **Weekly trends**: Category-level demand fluctuations visible



sql_sku_inventory_analysis

```sql
-- SKUs with Frequent Stockouts: Which SKUs had the most weeks with zero sales (potential stockouts)
SELECT
  SKU_ID,
  COUNT(*) AS stockout_weeks
FROM
  `retail-inventory-analytics.inventory_data.sku_sales_cleaned`
WHERE
  Units_Sold = 0
GROUP BY
  SKU_ID
ORDER BY
  stockout_weeks DESC
LIMIT 10;
```

Query results

Row	SKU_ID	stockout_weeks
1	SKU_0247	7
2	SKU_0045	6
3	SKU_0252	6
4	SKU_0030	6
5	SKU_0521	6
6	SKU_0282	6
7	SKU_0775	5
8	SKU_0038	5
9	SKU_0874	5
10	SKU_0965	5

Insight: SQL queries reinforced Python findings and added drill-downs by vendor and category, supporting data validation.

🎓 Skills Demonstrated

- Data Cleaning & Wrangling (**Python/Pandas**)
- Exploratory Analysis & Visualization (**Seaborn/Matplotlib**)
- Forecasting (Moving Average)
- **Excel** Modeling & Dashboarding (**ABC, ROP, Loss Estimation**)
- **Power BI** for BI storytelling & filtering
- SQL querying **(BigQuery)** for business insights

📊 Key Insights & Business Impact

- ~70% of total sales came from "A" category SKUs
- Vendor_2 is a high performer with optimal lead time (15.28 days)
- Stockouts identified on top SKUs — indicates room for alert systems
- 3-week moving average captured short-term demand trends reliably
- SQL & BI visuals supported executive-level reporting

Brijeshkumar Patel Aka Dadaga
bspwave5696@gmail.com
www.linkedin.com/in/brijeshkumarpatel5696