

1. Project Overview

Built an interactive analytics stack for a grocery floral department to monitor **Units, Sales, Cost, Shrink, Waste %**, promo uplift, and **holiday windows** (Valentine's, Mother's Day, Thanksgiving, Christmas).

Outputs include a Python EDA notebook, a set of reusable SQL views, and a **Power BI dashboard** with date/holiday, category, and SKU slicers.

2. Dataset Summary

- **Grain:** daily by SKU
- **Span:** Jan-2024 → Dec-2025 (two retail years)
- **Rows / Cols:** ~3.6K–4K rows per SKU (daily) • ~15–20 fields
- **Key fields:**
 - date, year_week, sku, category
 - price, cost
 - on_hand_beg, delivery_qty, sales_qty, markdown_qty, waste_qty
 - promo_flag, holiday_flag, is_holiday_window_v2
 - weather_index (context signal)
- **Derived tables:** DimDate (marked as date table), DimSKU (category, lifecycle), and a **baseline units (6w)** measure for demand vs. holiday spikes.

3) Exploratory Data Analysis (Python)

Environment: pandas, numpy, matplotlib; CSV → parquet; reproducible notebook.

Steps

- **Load & health checks:** `df.info()`, null scan, duplicate scan; typed dates; verified positive quantities.

```
[3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1826 entries, 0 to 1825
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   date        1826 non-null    object  
 1   sku         1821 non-null    object  
 2   category    1821 non-null    object  
 3   price       1826 non-null    float64
 4   cost        1819 non-null    float64
 5   on_hand_beg 1826 non-null    int64  
 6   delivery_qty 1826 non-null    int64  
 7   sales_qty   1826 non-null    int64  
 8   markdown_qty 1826 non-null    int64  
 9   waste_qty   1826 non-null    int64  
 10  promo_flag  1826 non-null    int64  
 11  holiday_flag 1826 non-null    int64  
 12  weather_index 1826 non-null    float64
dtypes: float64(3), int64(7), object(3)
memory usage: 185.6+ KB
```

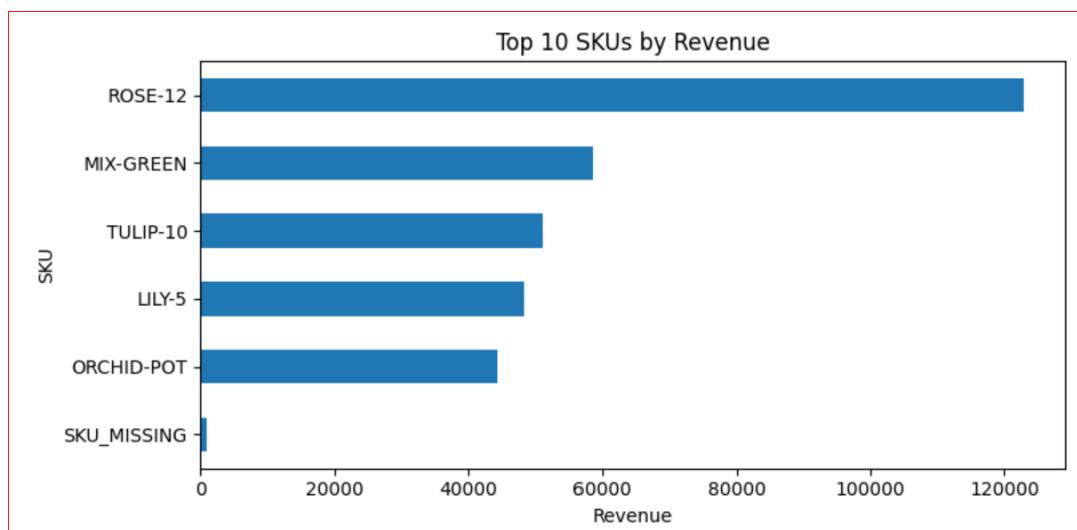
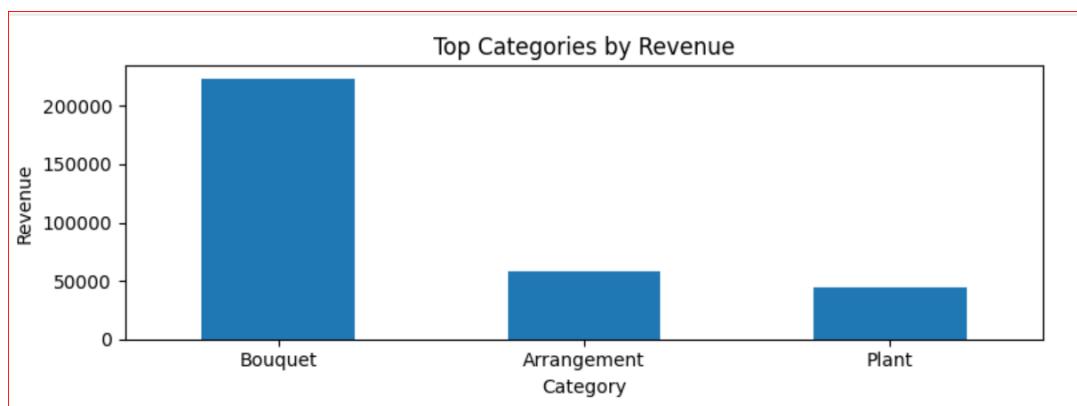
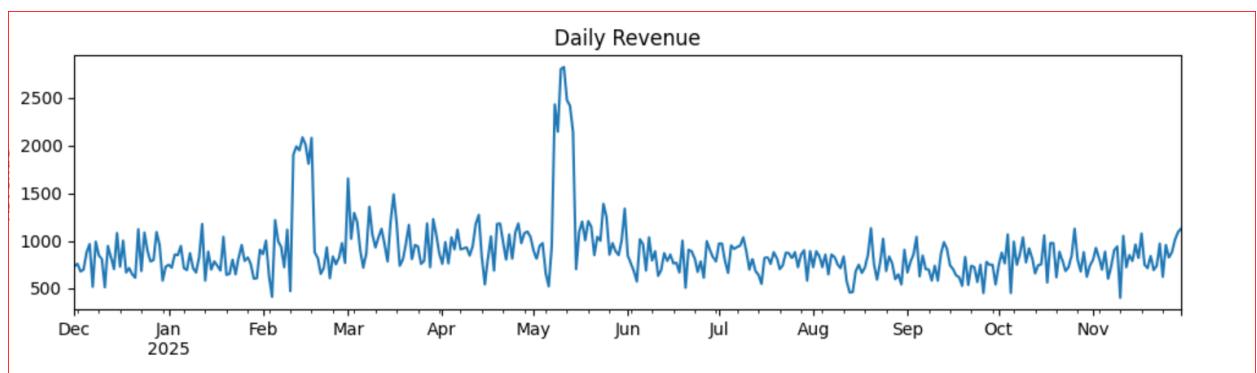
- **Summary statistics using .describe()**

```
[4]: # Summary statistics using .describe()
df.describe(include='all')

[4]:   date      sku   category      price      cost  on_hand_beg  delivery_qty  sales_qty  markdown_qty  waste_qty  promo_flag  holiday_flag  weather_in
      count    1826    1821     1821  1826.000000  1819.000000  1826.000000  1826.000000  1826.000000  1826.000000  1826.000000  1826.000000  1826.000000
      unique    365       5        3      NaN      NaN
      top  2025-02-25  TULIP-10  Bouquet      NaN      NaN
      freq      6      365     1093      NaN      NaN
      mean     NaN     NaN     NaN  24.973357  10.333260  5.596386  10.508215  7.634721  0.903067  1.939211  0.077218  0.037788  0.999
      std      NaN     NaN     NaN   7.083121   3.071661   5.010211   9.036012   5.300988   1.945820   2.292364   0.269054   0.193586   0.058
      min      NaN     NaN     NaN   9.600000   6.200000   0.000000   0.000000  -16.000000   0.000000  -3.000000   0.000000   0.000000  -1.000000   0.900
      25%     NaN     NaN     NaN  19.990000   8.000000   3.000000   5.000000   4.000000   0.000000   0.000000   0.000000   2.000000   0.000000   0.000000   0.948
      50%     NaN     NaN     NaN  24.990000  10.500000   5.000000   9.000000   6.000000   0.000000   0.000000   0.000000   2.000000   0.000000   0.000000   0.999
      75%     NaN     NaN     NaN  29.990000  12.000000  7.000000  14.000000  10.000000  1.000000  3.000000  0.000000   0.000000   0.000000   1.049
      max      NaN     NaN     NaN  34.990000  15.000000  47.000000  105.000000  41.000000  17.000000  21.000000  2.000000  1.000000  1.100
```

- **Cleaning:** clipped negative/erroneous quantities, harmonized SKUs/categories, forward-filled missing price/cost within SKU.
- **Feature engineering:**
 - $waste_rate = waste_qty / (sales_qty + waste_qty)$
 - $gm\$ = (price - cost) * sales_qty, gm\% = gm\$ / (price * sales_qty)$

- `is_promo` as union of file flag + holiday window
- `year_week`, `week_iso`, `is_weekend`
- **Quik EDA:**

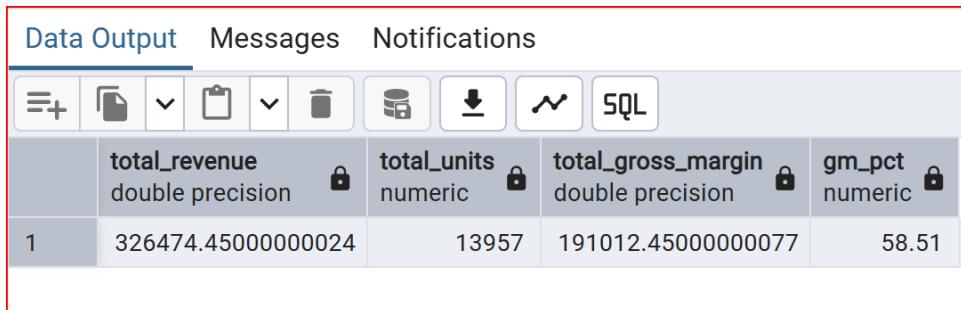


- **EDA highlights:**
 - **Holiday uplift:** Units spike 5–10× in holiday windows; pricing lifts 2–2.5× on Valentine’s/Mother’s Day; waste% dips slightly post-holiday as inventory is cleared.
 - **Category mix:** Bouquets dominate sales and share of waste \$; Plants carry higher unit margin but lower turnover.
 - **Price-volume curve:** elastic outside holidays; largely **inelastic** during event weeks.

4. Data Analysis using SQL

We performed structured analysis in PostgreSQL to answer key business questions.

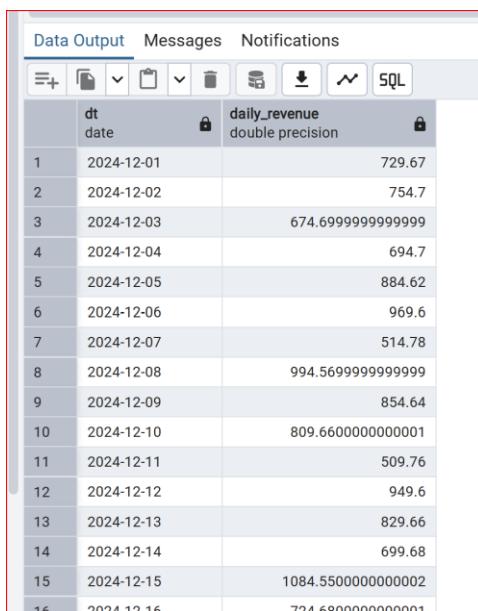
1) Total revenue, total units, and total gross margin



A screenshot of a PostgreSQL client interface showing a single row of results. The top navigation bar includes 'Data Output' (which is selected), 'Messages', and 'Notifications'. Below the bar are standard database management icons: a plus sign, a file icon, a dropdown arrow, a clipboard icon, another dropdown arrow, a trash bin, a cylinder icon, a download arrow, a line graph icon, and a 'SQL' button. The main data area is a table with four columns and one row of data. The columns are labeled 'total_revenue', 'total_units', 'total_gross_margin', and 'gm_pct'. The data row shows values: 326474.45000000024, 13957, 191012.45000000077, and 58.51 respectively. Each column has a 'double precision' type and a lock icon.

	total_revenue double precision	total_units numeric	total_gross_margin double precision	gm_pct numeric
1	326474.45000000024	13957	191012.45000000077	58.51

2) Daily revenue trend over time



A screenshot of a PostgreSQL client interface showing a daily revenue trend over 16 days. The top navigation bar includes 'Data Output' (selected), 'Messages', and 'Notifications'. Below the bar are standard database management icons. The main data area is a table with two columns and 16 rows of data. The columns are labeled 'dt' (date) and 'daily_revenue' (double precision). The data shows a general upward trend from day 1 to day 16. Each row has a lock icon.

	dt date	daily_revenue double precision
1	2024-12-01	729.67
2	2024-12-02	754.7
3	2024-12-03	674.699999999999
4	2024-12-04	694.7
5	2024-12-05	884.62
6	2024-12-06	969.6
7	2024-12-07	514.78
8	2024-12-08	994.569999999999
9	2024-12-09	854.64
10	2024-12-10	809.660000000001
11	2024-12-11	509.76
12	2024-12-12	949.6
13	2024-12-13	829.66
14	2024-12-14	699.68
15	2024-12-15	1084.550000000002
16	2024-12-16	724.680000000001

3) Categories ranked by total revenue

The screenshot shows a data output interface with a red border. At the top, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with various icons: a plus sign, a file, a dropdown, a clipboard, another dropdown, a trash can, a database icon, a download arrow, a line graph, and an SQL icon. The main area displays a table with three columns: 'category' (text), 'revenue' (double precision), and a primary key column. The data is as follows:

	category	revenue
1	Bouquet	223432.01000000112
2	Arrangeme...	58573.26000000004
3	Plant	44469.18000000004

4) Top 10 SKUs by revenue

The screenshot shows a data output interface with a red border. At the top, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with various icons: a plus sign, a file, a dropdown, a clipboard, another dropdown, a trash can, a database icon, a download arrow, a line graph, and an SQL icon. The main area displays a table with three columns: 'sku' (text), 'revenue' (double precision), and a primary key column. The data is as follows:

	sku	revenue
1	ROSE-12	122950.79999999997
2	MIX-GREEN	58433.30000000004
3	TULIP-10	51385.72000000003
4	LILY-5	48315.829999999965
5	ORCHID-P...	44439.19000000004

5) Gross-margin % by category

The screenshot shows a data output interface with a red border. At the top, there are tabs for 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with various icons: a plus sign, a file, a dropdown, a clipboard, another dropdown, a trash can, a database icon, a download arrow, a line graph, and an SQL icon. The main area displays a table with five columns: 'category' (text), 'gross_margin' (double precision), 'revenue' (double precision), and 'gm_pct' (numeric). The data is as follows:

	category	gross_margin	revenue	gm_pct
1	Plant	26661.18000000001	44469.18000000004	59.95
2	Bouquet	130888.0100000001	223432.01000000112	58.58
3	Arrangeme...	33463.26	58573.26000000004	57.13

6) Avg daily units on promo vs not

Data Output Messages Notifications

A screenshot of a data output interface. The top navigation bar includes 'Data Output', 'Messages', and 'Notifications'. Below the bar is a toolbar with various icons. The main area displays a table with two rows of data. The columns are labeled 'promo_flag' (datatype: bigint) and 'avg_daily_units' (datatype: numeric). Row 1 shows a value of 0 for promo_flag and 34.39 for avg_daily_units. Row 2 shows a value of 1 for promo_flag and 11.80 for avg_daily_units.

	promo_flag bigint	avg_daily_units numeric
1	0	34.39
2	1	11.80

7) Avg daily revenue on holiday vs not

Data Output Messages Notifications

A screenshot of a data output interface. The top navigation bar includes 'Data Output', 'Messages', and 'Notifications'. Below the bar is a toolbar with various icons. The main area displays a table with two rows of data. The columns are labeled 'holiday_flag' (datatype: bigint) and 'avg_daily_revenue' (datatype: numeric). Row 1 shows a value of 0 for holiday_flag and 841.69 for avg_daily_revenue. Row 2 shows a value of 1 for holiday_flag and 2217.26 for avg_daily_revenue.

	holiday_flag bigint	avg_daily_revenue numeric
1	0	841.69
2	1	2217.26

8) SKUs with highest markdown rate

Data Output Messages Notifications

A screenshot of a data output interface. The top navigation bar includes 'Data Output', 'Messages', and 'Notifications'. Below the bar is a toolbar with various icons. The main area displays a table with six rows of data. The columns are labeled 'sku' (datatype: text), 'markdown_qty' (datatype: numeric), 'sales_qty' (datatype: numeric), and 'markdown_rate_pct' (datatype: numeric). The data shows the following information:

	sku text	markdown_qty numeric	sales_qty numeric	markdown_rate_pct numeric
1	LILY-5	340	2417	14.07
2	ORCHID-POT	200	1483	13.49
3	MIX-GREEN	196	1670	11.74
4	ROSE-12	540	4920	10.98
5	TULIP-10	369	3428	10.76
6	SKU_MIS... SKU_MIS...	4	39	10.26

9) Categories with highest waste rate

The screenshot shows a data output interface with a toolbar at the top containing icons for file operations, a refresh button, a download button, and a SQL button. The main area displays a table with four columns: category (text), waste_qty (numeric), sales_qty (numeric), and waste_rate_pct (numeric). The data shows three categories: Bouquet, Arrangeme..., and Plant, with their respective waste rates.

	category text	waste_qty numeric	sales_qty numeric	waste_rate_pct numeric
1	Bouquet	3209	10799	29.72
2	Arrangeme...	335	1674	20.01
3	Plant	0	1484	0.00

10) Under-stocked days per SKU

The screenshot shows a data output interface with a toolbar at the top containing icons for file operations, a refresh button, a download button, and a SQL button. The main area displays a table with two columns: sku (text) and understock_days (bigint). The data shows six SKUs, all of which have 0 understock days.

	sku text	understock_days bigint
1	LILY-5	0
2	MIX-GREEN	0
3	ORCHID-POT	0
4	ROSE-12	0
5	SKU_MISSI...	0
6	TULIP-10	0

5. Dashboard in Power BI

Platform: views written to support the BI model and ad-hoc analysis.

1. Representative queries

1) Weekly sales & waste trend

```
SELECT year_week, SUM(price*sales_qty) AS sales$,
       SUM(waste_qty*cost) AS waste$,
       SUM(waste_qty)::numeric / NULLIF(SUM(sales_qty+waste_qty),0) AS waste_pct
```

```
FROM DailySKU JOIN DimDate USING(date)
GROUP BY year_week ORDER BY year_week;
```

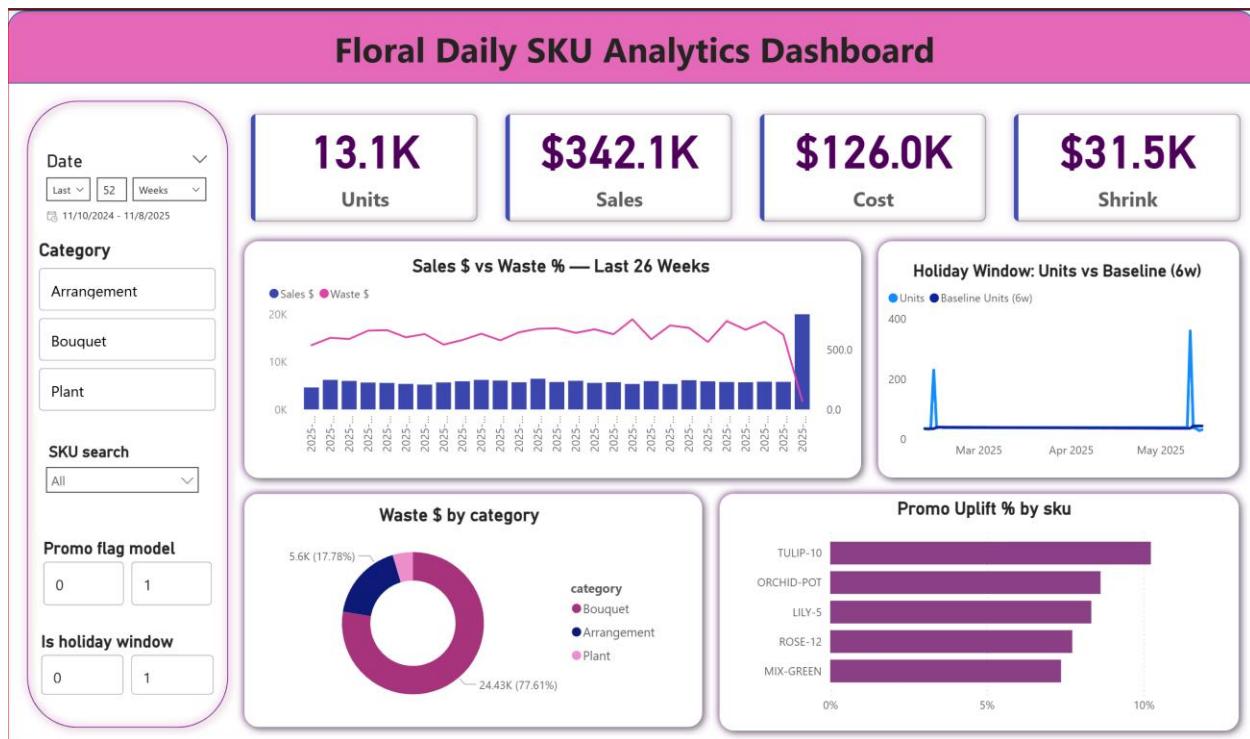
2) Waste \$ by category (YTD)

```
SELECT category, SUM(waste_qty*cost) AS waste$
FROM DailySKU JOIN DimDate USING(date)
WHERE date >= DATE_TRUNC('year', CURRENT_DATE)
GROUP BY category ORDER BY waste$ DESC;
```

3) Top SKUs by uplift in holiday weeks

```
SELECT sku, AVG(promo_uplift_pct) AS avg_uplift
FROM UpliftByWeek -- view from query #2
GROUP BY sku ORDER BY avg_uplift DESC;
```

2. Floral Daily SKU Analytics Dashboard



6. Business Recommendations & Actions

- Plan early for holidays: raise orders and price 2–2.5× in Valentine's/Mother's Day windows; hold a post-event markdown rule to protect margin.
- Focus waste control on Bouquets: largest share of Waste \$—pilot tighter delivery cadence and end-of-day markdowns.
- Promo targeting: keep offers on top uplift SKUs (e.g., TULIP-10, ORCHID-POT) and avoid deep discounts where demand is already inelastic.
- Run-the-business views: Weekly ops should use *Sales vs Waste%* and *Holiday Window* charts plus the SKU search slicer for quick interventions.