



RevoGrocers Sales Performance Analysis



Husnaa Yumn Sinaga

Business Overview & Disclaimer

Company Overview: RevoGrocers

- **RevoGrocers** is a simulated grocery retail enterprise that operates across various locations, offering a wide selection of grocery products to consumers.
- The company focuses on **enhancing sales strategies, elevating customer experience, and maximizing revenue** through the use of data-driven insights.

Disclaimer

- This analysis utilizes a publicly available dataset from Kaggle:
 [Kaggle Grocery Sales Dataset](#)
- RevoGrocers is a fictional entity created solely for the purpose of analytical exploration.
- Any insights or recommendations are based on this dataset and do not reflect real-world data.

Dataset Overview

Key Tables:

- sales (transactional data)
- products (product data)
- categories (product categories)
- customers (unique customer details)

Unused Tables:

- cities
- countries
- employees

Relevant Fields:

- | | |
|------------|----------------|
| • Quantity | • CategoryName |
| • Price | • CustomerID |
| • Discount | • CategoryID |

Project Goal

- Identify top-performing product categories
- Analyze factors driving revenue (units sold vs. customer count)
- Evaluate pricing strategies and their impact on sales

Grocery Sales Database

Simulated grocery sales data from 2018-01-01 to 2018-05-09.

Methodology

▼	 grocery_dataset		
	 categories		
	 cities		
	 countries		
	 customers		
	 employees		
	 products		
	 sales		

► Approach Taken

- Mini exploratory data analysis of grocery dataset
- SQL queries to sort and filter data as needed
- Analysis query results into five key business insights
- Key insights documentation and reporting

► Tools Used

BigQuery to run SQL queries and analysis of grocery_dataset (structured format)

Mini Exploratory Data Analysis

Step 1: Row Counting for Total Transactions and Total Products

```
SELECT
    COUNT(*) AS TotalTransactions
FROM
    `fsda-sql-01.grocery_dataset.sales` AS s
JOIN
    `fsda-sql-01.grocery_dataset.products` AS p
    ON s.ProductID = p.ProductID
JOIN
    `fsda-sql-01.grocery_dataset.customers` AS cu
    ON s.CustomerID = cu.CustomerID;

SELECT
    COUNT(*) AS TotalProducts
FROM
    `fsda-sql-01.grocery_dataset.products` AS p
JOIN
    `fsda-sql-01.grocery_dataset.categories` AS c
    ON p.CategoryID = c.CategoryID;
```

Results:

Row	TotalTransactions
1	6758125

Results:

Row	TotalProducts
1	451

Step 2: Check Important Value from Sales Table

- Quantity column all values > 0
- Price column all values > 0
- Discount column all values > 0 and < 1

► Step 2 Query:

```
-- Now we move on to check sales table (crucial step)

-- Check if quantity below zero
SELECT
    *
FROM
    `fsda-sql-01.grocery_dataset.sales`
WHERE
    Quantity <= 0;

-- Check if any discount values are an anomaly (less than 0 or more than 1)
SELECT
    *
FROM
    `fsda-sql-01.grocery_dataset.sales`
WHERE
    Discount < 0 OR Discount > 1;
```

Step 3: Checking Product Table

- Checking important value:
 - Price column all values > 0

```
-- Check for any product price (if any is below 0 or negative value)
SELECT
    ProductID,
    ProductName,
    Price
FROM
    `fsda-sql-01.grocery_dataset.products`
WHERE
    Price <= 0;
```

BQ 1 - Identify the product category that generates the highest revenue after discount.

► Query:

```
SELECT
    c.CategoryName,
    SUM(s.Quantity * p.Price * (1 - s.Discount)) AS RevenueAfterDiscount,
    ((SUM(s.Quantity * p.Price)) / (SUM(Discount))) AS RevenuePerDiscount
FROM
    `fsda-sql-01.grocery_dataset.sales` AS s
JOIN
    `fsda-sql-01.grocery_dataset.products` AS p
    ON s.ProductID = p.ProductID
JOIN
    `fsda-sql-01.grocery_dataset.categories` AS c
    ON p.CategoryID = c.CategoryID
GROUP BY
    c.CategoryName
ORDER BY
    RevenueAfterDiscount DESC;
```

► Results:

Row	CategoryName	RevenueAfterDis...	RevenuePerDisco...
1	Confections	556930717.3468...	22496.21012195...
2	Meat	492888844.6946...	22630.53671973...
3	Poultry	440025564.9656...	21476.60997889...
4	Cereals	427393431.9101...	21867.59142410...
5	Snails	372084885.2075...	23114.52785585...
6	Beverages	366515024.0050...	22089.57458062...
7	Produce	362861133.5171...	19871.33814906...
8	Dairy	354358156.7688...	23344.64699082...
9	Seafood	330527987.4676...	21174.68072429...
10	Grain	323879126.6500...	26676.20651331...
11	Shell fish	299598294.0299...	19225.65540450...

Insights:

- Top 3 categories with **highest Revenue**: Confections, Meat, Poultry.
 - The “cash cows” categories which generate the highest amount of revenue.
- Top 3 categories with **highest Revenue/Discount**: Grain, Dairy, Snails.
 - Categories with highest revenue/discount /= highest revenue count indicates “underdogs” categories with the potential to generate more revenue with strategic discounts.
- **Healthy** categories with **good Revenue** and **decent Discount**: Cereals, Beverages, Seafood.
 - The discount strategies for these categories should kept as they are.
- Categories with **poor overall performance**: Produce and Shell fish.
 - These categories should be looked upon further, there may be some issues with pricing and discount calculation.

BQ 2 - Assess the relation between revenue after discount and total units sold for each product category.

Query:

```
SELECT
    c.CategoryName,
    SUM(s.Quantity * p.Price * (1 - s.Discount)) AS RevenueAfterDiscount,
    SUM(s.Quantity) AS TotalUnitsSold,
    (SUM(s.Quantity * p.Price * (1 - s.Discount)) / SUM(s.Quantity)) AS RevenuePerUnitSold
FROM
    `fsda-sql-01.grocery_dataset.sales` AS s
JOIN
    `fsda-sql-01.grocery_dataset.products` AS p
    ON s.ProductID = p.ProductID
JOIN
    `fsda-sql-01.grocery_dataset.categories` AS c
    ON p.CategoryID = c.CategoryID
GROUP BY
    c.CategoryName
ORDER BY
    RevenueAfterDiscount DESC;
```

Results

Row	CategoryName	RevenueAfterDis...	TotalUnitsSold	RevenuePerUnitS...
1	Confections	556930717.3468...	11078474	50.2714288400...
2	Meat	492888844.6946...	9719292	50.71242274588...
3	Poultry	440025564.9656...	9159847	48.03852782318...
4	Cereals	427393431.9101...	8735296	48.92718368217...
5	Snails	372084885.2075...	7199409	51.68269856700...
6	Beverages	366515024.0050...	7393693	49.57130678877...
7	Produce	362861133.5171...	8174673	44.38845853738...
8	Dairy	354358156.7688...	6815143	51.99570379798...
9	Seafood	330527987.4676...	6996152	47.24425476571...
10	Grain	323879126.6500...	5433152	59.61164470458...
11	Shell fish	299598294.0299...	6983457	42.90114395061...

Insights:

- Top 3 categories with highest **Revenue per Unit Sold**: Grain, Dairy, Snails.
 - These categories are great starting points to know which products generate the most revenue for each unit sold.
- Top 3 categories with lowest **Revenue per Unit Sold**: Shell fish, Seafood, Cereals, Poultry.
 - These categories should be looked upon further to evaluate effective pricing strategies in order to maximize revenue gain.
- Relationship between **Revenue per Unit Sold** and **Total Units Sold** is not proportional.
 - Categories where R/US is high but TUS is low: (Grain, Dairy, & Snails) indicate higher-end categories
 - Categories with low R/US but high TUS (Poultry & Cereals) indicate common goods with usually higher daily demands.

BQ 3 - Assess the relation between revenue after discount and the number of unique customers for each product category.

► Query:

```
SELECT
    c.CategoryName,
    SUM(s.Quantity * p.Price * (1 - s.Discount)) AS RevenueAfterDiscount,
    COUNT(DISTINCT s.CustomerID) AS UniqueCustomers,
    (SUM(s.Quantity * p.Price * (1 - s.Discount)) / COUNT(DISTINCT s.CustomerID)) AS RevenuePerAverageCustomer
FROM
    `fsda-sql-01.grocery_dataset.sales` AS s
JOIN
    `fsda-sql-01.grocery_dataset.products` AS p
    ON s.ProductID = p.ProductID
JOIN
    `fsda-sql-01.grocery_dataset.categories` AS c
    ON p.CategoryID = c.CategoryID
GROUP BY
    c.CategoryName
ORDER BY
    RevenueAfterDiscount DESC;
```

► Results

Row	CategoryName	RevenueAfterDis...	UniqueCustomers	RevenuePerAvera...
1	Confections	556930717.3468...	98743	5640.204544594...
2	Meat	492888844.6946...	98701	4993.757354988...
3	Poultry	440025564.9656...	98679	4459.161168694...
4	Cereals	427393431.9101...	98651	4332.378099666...
5	Snails	372084885.2075...	98376	3782.272965027...
6	Beverages	366515024.0050...	98424	3723.837925760...
7	Produce	362861133.5171...	98601	3680.095876483...
8	Dairy	354358156.7688...	98308	3604.570907442...
9	Seafood	330527987.4676...	98334	3361.278779137...
10	Grain	323879126.6500...	97335	3327.468296604...
11	Shell fish	299598294.0299...	98338	3046.617726921...

Insights:

- Top 3 categories with highest **Revenue per Average Customer**: Confections, Meat, Poultry.
 - Constant count of unique customers but different numbers of revenue per average customer.
- Top 3 categories with lowest **Revenue per Average Customer**: Shell fish, Grain, Seafood.
 - For these categories unique customers are still high, but revenue gain is low.
- Relationship between **Number of Unique Customers** and **Revenue After Discount** is not proportional, meaning another factor may be affecting total Revenue made (i.e: customers making repeat purchase of certain categories).

BQ 4 - Calculate the average price per unit for each product category in the catalog.

► Query:

```
SELECT
    c.CategoryName,
    AVG(p.Price) AS AveragePricePerUnit
FROM
    mydatabase.main.products AS p
JOIN
    mydatabase.main.categories AS c
    ON p.CategoryID = c.CategoryID
GROUP BY
    c.CategoryName
ORDER BY
    AveragePricePerUnit DESC;
```

► Results

Row	CategoryName	AveragePricePer...
1	Grain	61.40394642857...
2	Dairy	53.55683142857...
3	Snails	53.19933243243...
4	Meat	52.27465199999...
5	Confections	51.84993157894...
6	Beverages	51.04368421052...
7	Cereals	50.41638888888...
8	Poultry	49.46088297872...
9	Seafood	48.66665555555...
10	Produce	45.79683333333...
11	Shell fish	44.27278333333...

Insights:

- Top 3 categories with highest **Average Price per Unit**: Grain, Dairy, Snails.
 - Higher pricing points towards more luxury items.
- Top 3 categories with lowest **Average Price per Unit**: Shell fish, Produce.
 - Lower pricing points towards low value items.

BQ 5 - Evaluate the relation between the average price per unit and the number of buyers (unique customers) per category.

Query:

```
SELECT
    c.CategoryName,
    AVG(p.Price) AS AveragePricePerUnit,
    COUNT(DISTINCT s.CustomerID) AS UniqueCustomers
FROM
    `fsda-sql-01.grocery_dataset.sales` AS s
JOIN
    `fsda-sql-01.grocery_dataset.products` AS p
    ON s.ProductID = p.ProductID
JOIN
    `fsda-sql-01.grocery_dataset.categories` AS c
    ON p.CategoryID = c.CategoryID
GROUP BY
    c.CategoryName
ORDER BY
    AveragePricePerUnit DESC;
```

Results

Row	CategoryName	AveragePricePer...	UniqueCustomers
1	Grain	61.43254359308...	97335
2	Dairy	53.61147548203...	98308
3	Snails	53.28068015957...	98376
4	Meat	52.31211525017...	98701
5	Confections	51.81190322273...	98743
6	Beverages	51.12591015803...	98424
7	Cereals	50.43802735917...	98651
8	Poultry	49.50921884342...	98679
9	Seafood	48.67796017344...	98334
10	Produce	45.78994061087...	98601
11	Shell fish	44.23266616321...	98338

Insights:

- Top 3 categories with highest **Unique Customers**: Confections, Meat, Produce.
 - Low relation between average price per unit and number of unique customers.
- Top 3 categories with lowest **Unique Customers**: Grain, Seafood, Shell fish.
 - Anomaly spotted, where average price for Grain products are high but unique customers are low. While both average price and unique customers count for both Seafood and Shell fish are similarly low. This also signals low relation.

BQ 6 - Which categories contribute the most to overall revenue after discount (percentage-wise)?

Query:

```
WITH CategoryRevenue AS (
    SELECT
        c.CategoryName,
        SUM(s.Quantity * p.Price * (1 - s.Discount)) AS RevenueAfterDiscount
    FROM
        `fsda-sql-01.grocery_dataset.sales` AS s
    JOIN
        `fsda-sql-01.grocery_dataset.products` AS p
        ON s.ProductID = p.ProductID
    JOIN
        `fsda-sql-01.grocery_dataset.categories` AS c
        ON p.CategoryID = c.CategoryID
    GROUP BY
        c.CategoryName
)
SELECT
    CategoryName,
    RevenueAfterDiscount,
    (RevenueAfterDiscount * 100.0 /
        (SELECT
            SUM(RevenueAfterDiscount)
        FROM
            CategoryRevenue)) AS PercentageOfTotalRevenue
FROM
    CategoryRevenue
ORDER BY
    PercentageOfTotalRevenue DESC;
```

Results

Row	CategoryName	RevenueAfterDis...	PercentageOfTot...
1	Confections	556930717.3468...	12.87087097897...
2	Meat	492888844.6946...	11.39084006222...
3	Poultry	440025564.9656...	10.16915048446...
4	Cereals	427393431.9101...	9.877217305557...
5	Snails	372084885.2075...	8.599016720690...
6	Beverages	366515024.0050...	8.470295207087...
7	Produce	362861133.5171...	8.385852471975...
8	Dairy	354358156.7688...	8.189345593731...
9	Seafood	330527987.4676...	7.638621733599...
10	Grain	323879126.6500...	7.484964147340...
11	Shell fish	299598294.0299...	6.923825294372...

Insights:

- Top 3 categories with highest **Percentage of Total Revenue**: Confections, Meat, and Poultry.
 - Categories that represent healthy numbers of revenue and customer demand, should always be in stock.
- Top 3 categories with lowest **Percentage of Total Revenue**: Shell fish, Grain, Seafood.
 - These categories should be evaluated due to poor margin value difference.

BQ 7 - Which product categories have the highest repeat purchase rate?

Query:

```

WITH CustomerCategoryPurchases AS (
    SELECT
        s.CustomerID,
        p.CategoryID,
        COUNT(DISTINCT s.SalesID) AS PurchaseCount
    FROM
        `fsda-sql-01.grocery_dataset.sales` AS s
    JOIN
        `fsda-sql-01.grocery_dataset.products` AS p
        ON s.ProductID = p.ProductID
    GROUP BY
        s.CustomerID,
        p.CategoryID
),
RepeatCustomersPerCategory AS (
    SELECT
        CategoryID,
        COUNT(DISTINCT CustomerID) AS RepeatCustomerCount
    FROM
        CustomerCategoryPurchases
    WHERE
        PurchaseCount > 1
    GROUP BY
        CategoryID
),
TotalCustomersPerCategory AS (
    SELECT
        p.CategoryID,
        COUNT(DISTINCT s.CustomerID) AS TotalCustomerCount
    FROM
        `fsda-sql-01.grocery_dataset.sales` AS s
    JOIN
        `fsda-sql-01.grocery_dataset.products` AS p
        ON s.ProductID = p.ProductID
    GROUP BY
        p.CategoryID
)
SELECT
    c.CategoryName,
    COALESCE(rc.RepeatCustomerCount, 0) AS RepeatCustomers,
    tc.TotalCustomerCount,
    (
        COALESCE(rc.RepeatCustomerCount, 0) * 100.0 / tc.TotalCustomerCount
    ) AS RepeatPurchaseRate
FROM
    `fsda-sql-01.grocery_dataset.categories` AS c
LEFT JOIN
    RepeatCustomersPerCategory AS rc
    ON c.CategoryID = rc.CategoryID
JOIN
    TotalCustomersPerCategory AS tc
    ON c.CategoryID = tc.CategoryID
ORDER BY
    RepeatPurchaseRate DESC;

```

Results

Row	CategoryName	RepeatCustomers	TotalCustomerCo...	RepeatPurchase...
1	Confections	98598	98743	99.85315414763...
2	Meat	98318	98701	99.61195935198...
3	Poultry	98122	98679	99.43554353003...
4	Cereals	97867	98651	99.20527921663...
5	Produce	97550	98601	98.93408788957...
6	Beverages	96679	98424	98.22705844103...
7	Snails	96324	98376	97.91412539643...
8	Seafood	96138	98334	97.76679480139...
9	Shell fish	96054	98338	97.67739836075...
10	Dairy	95677	98308	97.32371729665...
11	Grain	91184	97335	93.68058766117...

Insights:

- Top 3 categories with highest **Repeat Purchase Order**: Confections, Meat, Poultry.
 - Shows categories with high product demand and/or customer satisfaction.
- Top 3 categories with lowest **Revenue per Unit Sold**: Shell fish, Dairy, Grain.
 - Shows categories with low product demand and/or customer satisfaction.
- The higher customers make repeat orders of a product, the higher their satisfaction towards the product's cost-to-value is.

BQ 8 - Overall Findings Summary (OBIPR Framework)

► Summarized Finding

Category	Revenue	Revenue/ Discount	Units Sold	Avg Price/Unit	Verdict
Confections	High	Medium	High	Medium	High value product
Meat	High	Medium	High	Medium	High value product
Poultry	High	Medium	High	Medium	High value product
Cereals	Medium	Medium	Medium	Medium	Average value product
Produce	Low	Low	Medium	Low	Low value product
Beverages	Medium	Medium	Medium	Medium	Average value product
Snails	Medium	High	Medium	High	High value product
Seafood	Low	Medium	Low	Low	Low value product
Shell fish	Low	Low	Low	Low	Low value product
Dairy	Medium	High	Low	High	Discount and pricing should be evaluated (potentially medium volume of discount)
Grain	Low	High	Low	High	Discount and pricing should be evaluated (potentially high volume of discount)

► Insights

1. **Confections, Meat, and Poultry** categories should be utilized as ‘cash cows’ to push revenue gain.
2. **Snails** category is unique in a way that it is a luxury category (high price) but still demand when discounted.
3. **Dairy and Grain** categories being priorities of evaluation, where pushing discounts could potentially lead to higher sales depending on numbers crunched (high revenue/discount, high price point, but low sales).
4. Strategy optimization for **Cereals and Beverages** categories could be addressed after urgent priorities have been properly resolved.
5. **Produce, Seafood, and Shell fish** categories should be in top priority of evaluation, potential to be cut off due to insufficient pricings, minimal margins, or low demands.

BQ 9 - Find the cumulative amount of transaction of the top user (user with highest transaction value).

Query:

```
1 WITH Top1Customer AS (
2     SELECT
3         s.CustomerID,
4         SUM(s.Quantity * p.Price * (1 - s.Discount)) AS TotalSpent,
5         RANK() OVER (ORDER BY SUM(s.Quantity * p.Price * (1 - s.Discount)) DESC) AS Ranking
6     FROM
7         `fsda-sql-01.grocery_dataset.sales` AS s
8     JOIN
9         `fsda-sql-01.grocery_dataset.products` AS p
10    ON s.ProductID = p.ProductID
11    JOIN
12        `fsda-sql-01.grocery_dataset.customers` AS cu
13    ON s.CustomerID = cu.CustomerID
14    GROUP BY
15        s.CustomerID
16 ),
17 revenue AS (
18     SELECT
19         s.CustomerID,
20         s.SalesDate,
21         SUM(s.Quantity * p.Price * (1 - s.Discount)) AS Revenue
22     FROM
23         `fsda-sql-01.grocery_dataset.sales` AS s
24     JOIN
25         `fsda-sql-01.grocery_dataset.products` AS p
26     ON s.ProductID = p.ProductID
27     JOIN
28         `fsda-sql-01.grocery_dataset.customers` AS cu
29     ON s.CustomerID = cu.CustomerID
30     JOIN Top1Customer AS t
31     ON s.CustomerID = t.CustomerID
32     WHERE Ranking = 1
33     GROUP BY s.CustomerID, s.SalesDate
34 )
35 SELECT
36     CustomerID,
37     SalesDate,
38     ROUND(Revenue, 2) AS Revenue,
39     ROUND(SUM(Revenue) OVER (ORDER BY SalesDate), 2) AS CumulativeRevenue
40 FROM
41     revenue
42 ORDER BY
43     SalesDate;
```

Results

O	123 CustomerID	⌚ SalesDate	123 Revenue	123 CumulativeRevenue
1	94,800	2018-01-01 02:19:11.270	2,366.35	2,366.35
2	94,800	2018-01-04 03:16:26.030	2,251.31	4,617.66
3	94,800	2018-01-04 15:06:45.760	342.45	4,960.11
4	94,800	2018-01-05 01:40:34.040	377.36	5,337.47
5	94,800	2018-01-07 17:01:02.900	1,166.35	6,503.82
6	94,800	2018-01-08 15:14:29.770	2,245.47	8,749.3
7	94,800	2018-01-08 17:54:00.690	1,501.1	10,250.4
8	94,800	2018-01-09 18:38:29.950	1,493.92	11,744.32
9	94,800	2018-01-10 17:58:44.360	1,111.8	12,856.12
10	94,800	2018-01-11 05:17:41.040	2,313.9	15,170.03
11	94,800	2018-01-11 06:41:40.790	1,524.93	16,694.96
12	94,800	2018-01-17 09:32:05.180	1,038.87	17,733.84
13	94,800	2018-01-19 03:01:19.370	525.13	18,258.97
14	94,800	2018-01-19 08:52:07.000	1,113.68	19,372.65
15	94,800	2018-01-19 15:36:47.010	843.23	20,215.88
16	94,800	2018-01-20 06:53:53.360	585.39	20,801.27
17	94,800	2018-01-24 09:20:35.320	1,303.66	22,104.93
18	94,800	2018-01-25 12:51:51.010	482.42	22,587.35
19	94,800	2018-01-26 23:02:16.870	2,313.29	24,900.64
20	94,800	2018-01-27 22:55:58.600	687.68	25,588.32

Insights:

- The top customer by the ID of 94800 spent \$130,024 on groceries shopping at RevoGrocers in total.
- They made a total of 102 transactions with an average of \$1,265 spent during each transaction.

Overall Recap & Key Findings

► Category Performance

Distinct tiering of product categories based on revenue, sales volume, and unique customers.

- **The "Cash Cows"** (Confections, Meat, Poultry): they generate the highest revenue, sell the most units, with repeat purchase rates over 99%.
- **The "High-Potential Underdogs"** (Grain, Dairy, Snails): present high average prices per unit and good discounts, yet also has low sales volume and fewer unique buyers. This indicates potential in product values.
- **The "Underperformers"** (Produce, Seafood, Shellfish): consistently rank at the bottom for revenue, margin contribution, and customer retention. Their poor performance reflects the need for immediate attention.

► Customer Loyalty

The most successful categories are driven by:

- an extremely high rate of repeat business, proving that customer satisfaction and retention are the primary drivers of revenue.