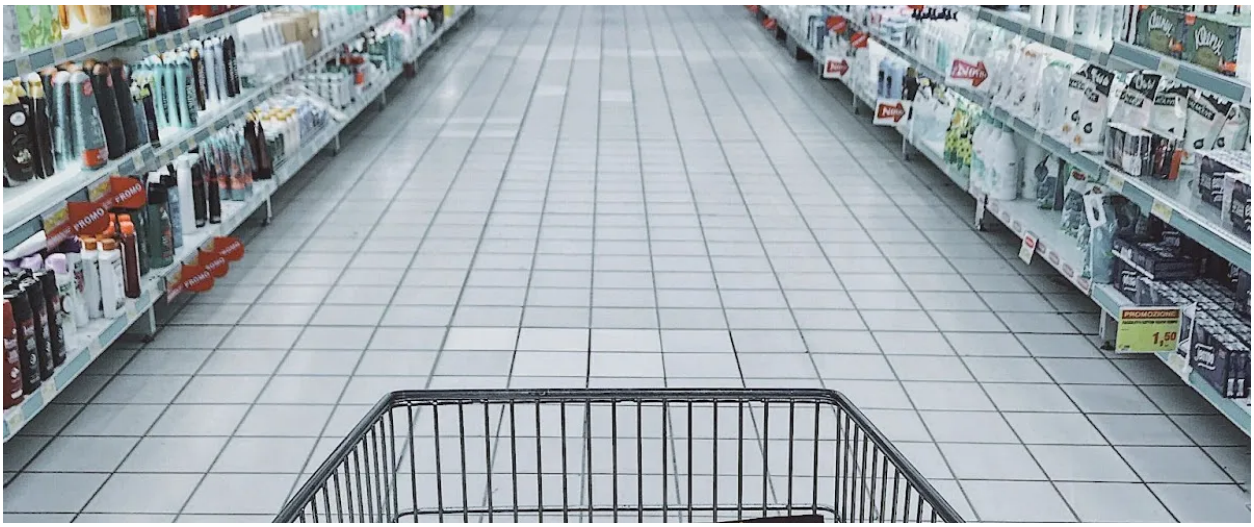

SQL Exploratory Data Analysis of a Grocery Dataset

This is the second article in a series for beginning data analysis. In the last article I went over using basic and intermediate Excel to analyze our grocery dataset.

We will now use SQL to make the same Exploratory Data Analysis, adding in a bit more complexity using a JOIN on item price data that I created. Many people get a bit scared of SQL because they think ‘programming language’, but it is relatively simple. As promised in Part 1, you’re going to see how SQL is so much easier than Excel in a lot of ways, and nothing to be afraid of. If you’re a data nerd like me, you might also find it quite a bit of fun!

For this project I used SQL Server Management Studio (SSMS). You will need to download both SSMS and SQL Server for the server (the Free/Express version is fine).

First we want to import our dataset into SSMS. When you first open up SMSS it will ask you to connect to a server. Use SQL Server to connect to your

desktop/laptop and you'll be good to go.

The Dataset and Data Cleaning

Once again, this dataset is from Kaggle(1).

Let's start by doing a simple query of the data to make sure we have the right dataset. If you expand the Groceries database, under Tables, you will see your table listed. I went ahead and changed mine a bit to make it just a little cleaner.

You will see that all of our data is there. And we have the right number of rows, and the right number of columns. All good there.

```
--preview the table records 38,765
Select * from [grocery data].[dbo].['grocery data']
```

Check for Null values in Column 1. This did not return anything which is of course what we want. Let's do the same for the other 2 columns. No results again. We're golden.

```
SELECT [Member_number]
FROM [grocery data].[dbo].['grocery data']
WHERE [Member_number] IS  NULL;

SELECT [itemdescription]
FROM [grocery data].[dbo].['grocery data']
WHERE [Member_number] IS  NULL;

SELECT [purchasedate]
FROM [grocery data].[dbo].['grocery data']
WHERE [Member_number] IS  NULL;
```

Your first instinct would be to change the datatype under ‘Column Properties’. That should work right? Wrong.

We are going to create a new column for the new date, CAST the old date info as a DATE datatype and then we will use this column rather than our old nvarchar(255) column for our future analysis. CAST is SQL’s most basic conversion function.

Since CAST did not work, we’re going to try something stronger. That something stronger is CONVERT. CONVERT is a more advanced conversion function that allows you to specify the format of the output, in this case the ‘103’ specifies that we want to YYYY-MM-DD date format.

```
ALTER TABLE [grocery data].[dbo].['grocery data']
ADD [new purchase date] date;

---update new column with date
UPDATE [grocery data].[dbo].['grocery data']
SET [new purchase date] = TRY_CONVERT(DATE, [purchase date])
WHERE TRY_CONVERT(DATE, [purchase date]) IS NOT NULL;
```

Now we are in great shape to start analyzing our data. We have checked for null values, changed our column names to fit our naming conventions, and made the date a column one we can actually analyze, again very important as a lot of the analysis and data visualization will use the date columns.

Number of Records: 38,765

Unique Customers: 3898

Unique Items: 167

Date Data Begins: 2014–01–01

Date Data Ends: 2015–12–30

```
--distinct records 38006
select count(*) from
(select distinct * from
[grocery data].[dbo].['grocery data']) as a
```

What Are The Most Popular Items Sold?

```
--find the most popular item sold
select Top1 [itemdescription],
count(itemdescription) as item_count
from [grocery data].[dbo].['grocery data']
group by[itemdescription]
order by count(itemdescription) desc


The most popular item sold is whole milk    2502
```

Let’s start by extracting the year, month, day, and day of the week from our purchase date field. For the purposes of this project and practice, we’ll add some new fields.

```
--to add the columns
ALTER TABLE [grocery data].[dbo].['grocery data']
ADD [new purchase date] date;

ALTER TABLE [grocery data].[dbo].['grocery data']
ADD [ purchase day] date;

ALTER TABLE [grocery data].[dbo].['grocery data']
ADD [ purchase month] date;

ALTER TABLE [grocery data].[dbo].['grocery data']
ADD [new purchase year] date;

ALTER TABLE [grocery data].[dbo].['grocery data']
```

```
ADD [purchase season] date;




---to update date column
UPDATE [grocery data].[dbo].['grocery data']
SET [new purchase date] = TRY_CONVERT(DATE, [purchase date])
WHERE TRY_CONVERT(DATE, [purchase date]) IS NOT NULL;




---update year column
UPDATE [grocery data].[dbo].['grocery data']
SET [purchase year] = year([new purchase date])
WHERE [new purchase date] IS NOT NULL;


--update day column
UPDATE [grocery data].[dbo].['grocery data']
SET [purchase day] = day([new purchase date])
WHERE [new purchase date] IS NOT NULL;


---update Dayofweek(DOW)
UPDATE [grocery data].[dbo].['grocery data']
SET [purchase dow] = DATENAME(weekday, [new purchase date])
WHERE [new purchase date] IS NOT NULL;
```

Most Popular Products in Year 2015

```
-- popular products for 2015

SELECT TOP 1 *
```

```
FROM (
select [itemdescription],
count(itemdescription) as item_count
from [grocery data].[dbo].['grocery data']
where [purchase year] = 2015
group by (itemdescription)
) AS ITEMDESCRIPTION
ORDER BY ITEM_COUNT DESC;


most products sold in 2015 with count is whole milk 1464
```

Most Popular Products in Year 2014

```
--popular products for 2014
SELECT TOP 1 *
FROM (
    SELECT [itemdescription],
    COUNT(itemdescription) AS ITEM_COUNT
    FROM [grocery data].[dbo].['grocery data']
    WHERE [purchase year] = '2014'
    GROUP BY itemdescription
) AS ITEMDESCRIPTION
ORDER BY ITEM_COUNT DESC;

most products sold in 2014 with count is whole milk 1038
```

Most Popular Products sold by Member_number

```
---count based on member_number

select TOP 1*
FROM (
select [member_number],
count(member_number) as item_count
from [grocery data].[dbo].['grocery data']
```

```
group by (member_number)
) AS MEMBER_NUMBERCOUNT
order by ITEM_COUNT DESC

The member with 3180 bought higest items which is   36
```

Most Popular Products by Month

```
select TOP 1*
FROM (
SELECT [purchase month],
COUNT(itemdescription) AS item_count
FROM [grocery data].[dbo].['grocery data']
GROUP BY [purchase month]
) AS MONTHCOUNT
ORDER BY ITEM_COUNT DESC;  -- Sort by purchase month in ascendi



compare to two years AUGUST sold higest items is 3496
```

Most Popular Products by Season

```
-- count based on season
SELECT TOP 1 *
FROM (
    SELECT [purchase season],
    COUNT(itemdescription) AS item_count
    FROM [grocery data].[dbo].['grocery data']
    GROUP BY [purchase season]
) AS seasoncount
ORDER BY item_count DESC;


compare to two years SPRING sold higest items is 11183
```

Most Popular Products by Day

```
---count based on days
SELECT TOP 1 *
FROM (
    SELECT [purchase day],
    COUNT(itemdescription) AS item_count
    FROM [grocery data].[dbo].['grocery data']
    GROUP BY [purchase day]
) AS DailyCounts
ORDER BY item_count DESC;


top purchase day is 3 with the count 1393
```

Most Popular Products by DOW(Dayofweek)

```
--count based on dow
select TOP 1*
FROM (
select [purchase dow] ,
count(itemdescription) as item_count
from [grocery data].[dbo].['grocery data']
group by[purchase dow]
) AS DOW_COUNTS
order by item_count desc


top purchase day is Thursday with the count 5620
```