Microsoft Power Tools for Data Analysis #10

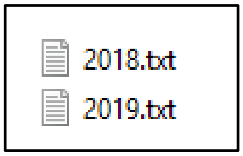
Power BI M Code: Helper Table to Calculate MAT By Month & Product

Notes from Video:

Table of Contents:

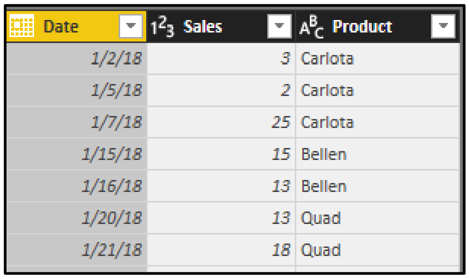
1. **Intermediate Fact Table / Helper Table:**
    1) Intermediate Fact Table
    2) Synonyms:
        i. Helper Table
        ii. Snap Shot Table
        iii. Intermediate Fact Table
        iv. Table with intermediate calculations
    3) Intermediate Fact Table make intermediate calculations that might otherwise be too complicated or calculation intensive if we included them in a Mega-Formula
2. **Goal of our Intermediate Fact Table Calculation**:
    1) Create a Table that cross joins Month and Product and then calculates the Monthly Total and the Moving Annual Total (MAT) for 12 Months Back.
    2) Advantage of a Table like this is that it might eliminate a Measure in DAX that is calculation intensive.
3. We will create a multiply Query transformation with five separate queries.

4. **Step 01 Query (already created in Power BI Download File)**:
    1) We have Text Files that contain sales numbers for each year, as seen here:



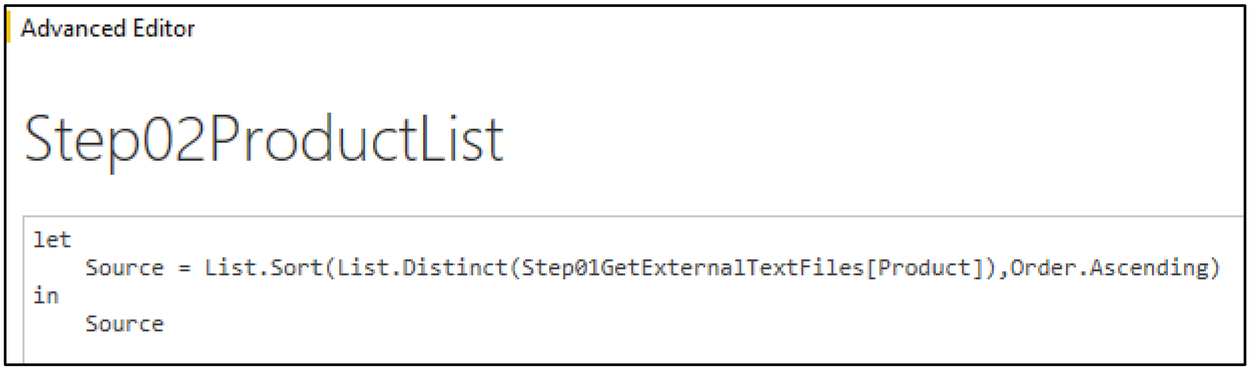
2018.txt
2019.txt

2) They have been imported into Power BI Desktop using the From Folder Feature. The Sales Table looks like this:



Date	Sales	Product
1/2/18	3	Carlota
1/5/18	2	Carlota
1/7/18	25	Carlota
1/15/18	15	Bellen
1/16/18	13	Bellen
1/20/18	13	Quad
1/21/18	18	Quad

5. **Step 02 Query**:
   1) Goal of this query is to create a sorted, unique list of Product names that can be used in Query 03 to do a crossjoin to get a listing of all combinations of Months and Product Names.
   2) Steps:
      i. Reference Step 01 Query and then lookup Product Column
      ii. Use List. Distinct function to create a unique list of product names
      iii. Use Table.Sort function to sort list.
   3) M Code is here:

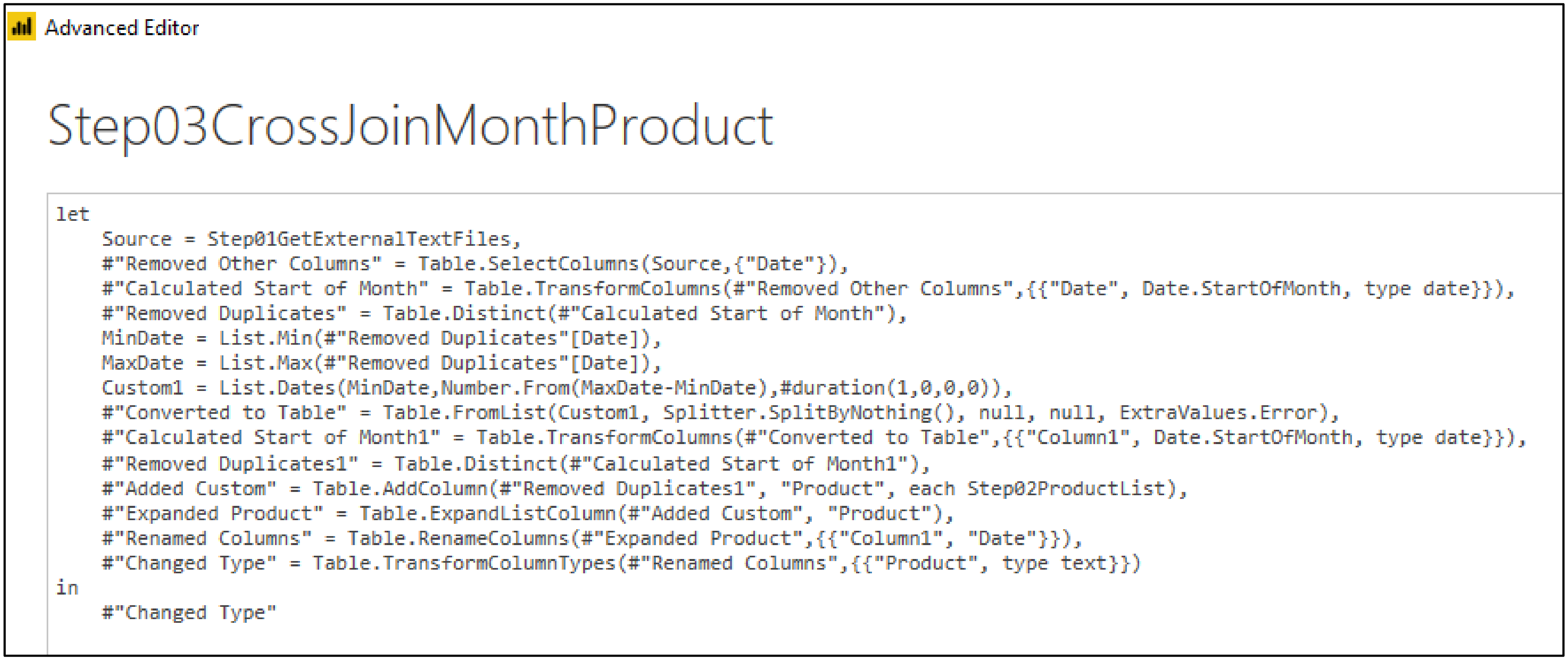


Advanced Editor

Step02ProductList

```
let
    Source = List.Sort(List.Distinct(Step01GetExternalTextFiles[Product]),Order.Ascending)
in
    Source
```

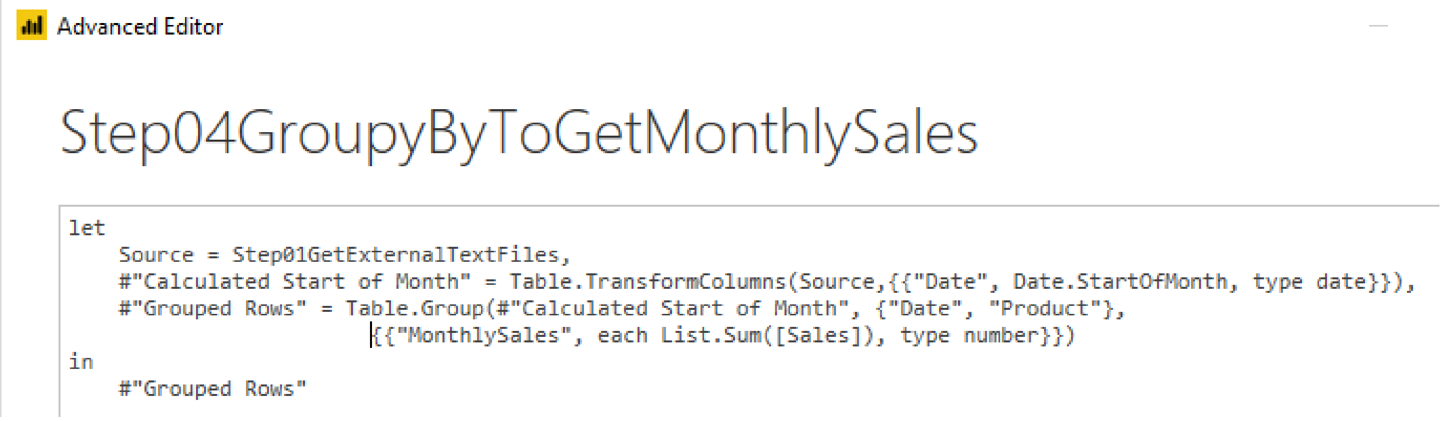
6. **Step 03 Query**:
   1) Goal of this query is to create a cross join (listing of all combinations) of all moths and products that we can use in Query 5 to then make the calculations of Sales Totals and MAT for each row.
   2) Steps:
      i. Reference Step 01 Table
      ii. Remove all columns except for the Date column.
      iii. Convert Date column to Start Of Month Dates.
      iv. Lookup or Extract the Min Date.
      v. Lookup or Extract the Max Date.
      vi. Create a List of all possible dates between min and max dates.
      vii. Convert list of dates to a table.
      viii. Convert Date column to Start Of Month Dates.
      ix. Remove Duplicates.
      x. Add Custom Column to bring list of product names into each row.
      xi. Expand Custom Column to perform crossjoin of Start Dates and Product Names.
      xii. Rename Columns
      xiii. Change Data Types.
   3) M Code on Next Page:



Step03CrossJoinMonthProduct

```
let
    Source = Step01GetExternalTextFiles,
    #"Removed Other Columns" = Table.SelectColumns(Source,{"Date"}),
    #"Calculated Start of Month" = Table.TransformColumns(#"Removed Other Columns",{{"Date", Date.StartOfMonth, type date}}),
    #"Removed Duplicates" = Table.Distinct(#"Calculated Start of Month"),
    MinDate = List.Min(#"Removed Duplicates"[Date]),
    MaxDate = List.Max(#"Removed Duplicates"[Date]),
    Custom1 = List.Dates(MinDate,Number.From(MaxDate-MinDate),#duration(1,0,0,0)),
    #"Converted to Table" = Table.FromList(Custom1, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
    #"Calculated Start of Month1" = Table.TransformColumns(#"Converted to Table",{{"Column1", Date.StartOfMonth, type date}}),
    #"Removed Duplicates1" = Table.Distinct(#"Calculated Start of Month1"),
    #"Added Custom" = Table.AddColumn(#"Removed Duplicates1", "Product", each Step02ProductList),
    #"Expanded Product" = Table.ExpandListColumn(#"Added Custom", "Product"),
    #"Renamed Columns" = Table.RenameColumns(#"Expanded Product",{{"Column1", "Date"}}),
    #"Changed Type" = Table.TransformColumnTypes(#"Renamed Columns",{{"Product", type text}})
in
    #"Changed Type"
```

7. **Step 04 Query**:
    1) Goal of Query is to calculate Monthly Sales Totals for each month and product
    2) Steps:
        i. Reference Step 01 Table
        ii. Convert Date column to Start Of Month Dates.
        iii. Group By Date and Product to get Monthly Sales Totals
    3) M Code is here:



Step04GroupyByToGetMonthlySales

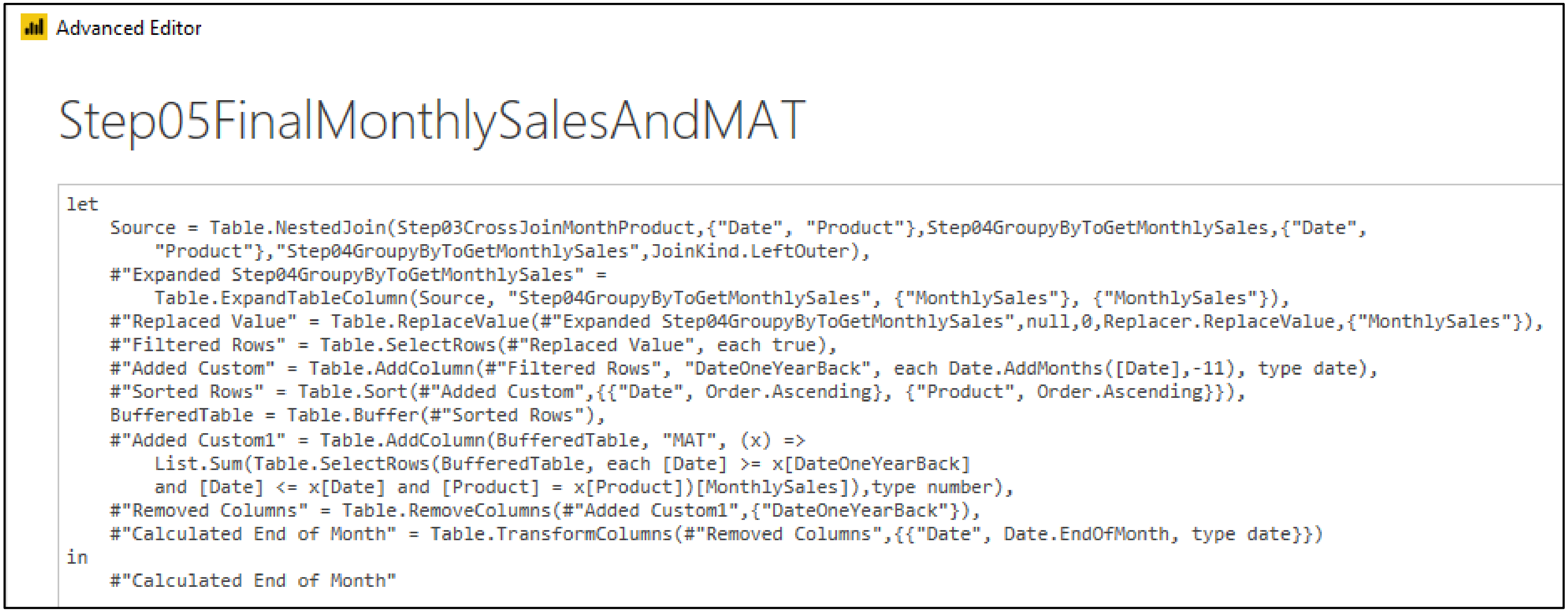
```
let
    Source = Step01GetExternalTextFiles,
    #"Calculated Start of Month" = Table.TransformColumns(Source,{{"Date", Date.StartOfMonth, type date}}),
    #"Grouped Rows" = Table.Group(#"Calculated Start of Month", {"Date", "Product"},
                        {{"MonthlySales", each List.Sum([Sales]), type number}})
in
    #"Grouped Rows"
```

8. **Step 05 Query**:
    1) Goal of Query: Create Final Intermediate Fact Table that lists all combinations of Month and Product and shows Month Sales and Monthly Annual Totals (MAT).
    2) Steps:
        i. Merge Query 03 & Query 04
        ii. Expand to show Monthly Sales Totals for each Product
        iii. Replace null values with zero values
        iv. Add Custom Column to calculate Date that is 12 months back
        v. Sort based on Date and Product, so MAT calculation will be accurate
        vi. Use Table.Buffer to prevent sales table from changing during MAT Calculation in the next step.
        vii. Add Custom Column with Function that will calculate the MAT
        viii. Remove unwanted columns
        ix. Rename Columns or Create End Of Month Date.

3) M Code is here:

  i. Sorting in Power Query (M Code) using the Table.Sort function: If you use the drop down arrows to sort, the first column sorted will be the Major Sort (this is the opposite of the Sort Buttons in Excel).

  ii. The Custom Function in the below M Code is calculated for each row in the Custom Column Table. However, the Custom Function uses two tables for each row in the Custom Column Table: 1) the Inner Table (BufferedTable in first argument of Table.SelectRows function) and 2) the Outer Table (the BufferedTable which gets the Custom Column added to it). To prevent the Inner Table from making a call for each row in the Outer Table, we can use the Table.Buffer Function to Buffer the Internal Table to prevent a call to the source table for every row in the table.



Step05FinalMonthlySalesAndMAT

```
let
    Source = Table.NestedJoin(Step03CrossJoinMonthProduct,{"Date", "Product"},Step04GroupyByToGetMonthlySales,{"Date",
        "Product"},"Step04GroupyByToGetMonthlySales",JoinKind.LeftOuter),
    #"Expanded Step04GroupyByToGetMonthlySales" =
        Table.ExpandTableColumn(Source, "Step04GroupyByToGetMonthlySales", {"MonthlySales"}, {"MonthlySales"}),
    #"Replaced Value" = Table.ReplaceValue(#"Expanded Step04GroupyByToGetMonthlySales",null,0,Replacer.ReplaceValue,{"MonthlySales"}),
    #"Filtered Rows" = Table.SelectRows(#"Replaced Value", each true),
    #"Added Custom" = Table.AddColumn(#"Filtered Rows", "DateOneYearBack", each Date.AddMonths([Date],-11), type date),
    #"Sorted Rows" = Table.Sort(#"Added Custom",{{"Date", Order.Ascending}, {"Product", Order.Ascending}}),
    BufferedTable = Table.Buffer(#"Sorted Rows"),
    #"Added Custom1" = Table.AddColumn(BufferedTable, "MAT", (x) =>
        List.Sum(Table.SelectRows(BufferedTable, each [Date] >= x[DateOneYearBack]
        and [Date] <= x[Date] and [Product] = x[Product])[MonthlySales]),type number),
    #"Removed Columns" = Table.RemoveColumns(#"Added Custom1",{"DateOneYearBack"}),
    #"Calculated End of Month" = Table.TransformColumns(#"Removed Columns",{{"Date", Date.EndOfMonth, type date}})
in
    #"Calculated End of Month"
```