

In Power Query Editor in Power BI, **Custom Columns** and **Invoke Custom Function** are powerful tools that allow you to extend the data transformation process with custom calculations and reusable functions. Power BI uses **M Query** as its data transformation language in Power Query Editor, enabling complex data manipulations.

Let's break down each concept and see how to implement them.

---

## 1. Custom Column

A **Custom Column** allows you to create a new column based on a formula or calculation applied to existing columns. This is useful when you need to apply specific logic that Power Query doesn't provide directly.

### Steps to Create a Custom Column:

#### 1. Load Data into Power Query Editor:

- Open Power BI and go to **Home** → **Transform Data** to open Power Query Editor.

#### 2. Add Custom Column:

- Go to the **Add Column** tab.
- Select **Custom Column**.

#### 3. Enter the Formula:

- In the **Custom Column** dialog box, enter your formula using **M Query**.
- Provide a name for your new column and write the formula based on the fields and values in your dataset.

### Example of Custom Column Using M Query:

Suppose you have a table with columns **Price** and **Quantity**, and you want to calculate the **Total Sales** as `Price * Quantity`.

In the **Custom Column** formula box, you would write:

```
m
= [Price] * [Quantity]
```

This M Query expression multiplies the values in the **Price** column by the values in the **Quantity** column and returns the result in the new column named **Total Sales**.

---

## 2. Invoke Custom Function

An **Invoke Custom Function** allows you to create reusable functions in Power Query and apply them to multiple rows or datasets. Once created, these functions can be used on-demand to apply specific transformations.

### Steps to Invoke a Custom Function:

## 1. Define the Custom Function:

- Create a new blank query in Power Query Editor.
- Write your function using M Query syntax.
- Functions in M Query are defined with a `let` statement, where parameters are passed and used within the function.

## 2. Invoke the Function:

- Once the function is created, go back to the main table.
- Use **Invoke Custom Function** to apply the function to a specific column or value.

## Example of Creating and Invoking a Custom Function:

Suppose you want to create a function that calculates **Profit** by subtracting **Cost** from **Price**.

### Step 1: Create the Custom Function

- Go to **Home** → **Advanced Editor** and enter the following M Query:

```
m
(price as number, cost as number) =>
let
    profit = price - cost
in
    profit
```
- Name this query `CalculateProfit`.

### Step 2: Invoke the Custom Function

- Return to your original table containing **Price** and **Cost** columns.
- Go to the **Add Column** tab and select **Invoke Custom Function**.
- Choose `CalculateProfit` as the function.
- Map the **Price** column to `price` and **Cost** column to `cost`.

The result will be a new column showing the **Profit** for each row.

---

## 3. Using M Query in Power Query with Examples

**M Query** is the functional language used in Power Query to transform and manipulate data. It's case-sensitive and employs a combination of functions and steps for data manipulation.

### Example of M Query in Power Query

Suppose we have a table with columns **Product**, **Price**, and **Quantity**. We want to:

1. Add a **Total Cost** column calculated as `Price * Quantity`.
2. Filter the table to show only rows where **Total Cost** is greater than 100.

Here's how we can achieve this using M Query:

```
m
let
    // Load data into Power Query
    Source = YourDataSource,

    // Step 1: Add Custom Column for Total Cost
    AddTotalCost = Table.AddColumn(Source, "Total Cost", each
[Price] * [Quantity]),

    // Step 2: Filter rows where Total Cost is greater than
    100
    FilteredRows = Table.SelectRows(AddTotalCost, each [Total
Cost] > 100)
in
    FilteredRows
```

Explanation of M Query Steps:

- 1. `Source` : Loads the data from the source.
- 2. `AddTotalCost` : Adds a custom column called **Total Cost** by multiplying **Price** and **Quantity**.
- 3. `FilteredRows` : Filters the table to only include rows where **Total Cost** is greater than 100.
- 4. `in` : Specifies the final output of the M Query, which will be the filtered table.

Summary Table of Key Steps

Feature	Purpose	Steps Summary
Custom Column	Adds a new column based on a formula or calculation.	<b>Add Column</b> → <b>Custom Column</b> → Enter M Query formula (e.g., <code>[Price] * [Quantity]</code> ).
Invoke Custom Function	Applies a reusable function to rows or datasets.	<b>Create Function</b> → Define parameters in M Query → <b>Invoke Custom Function</b> on main table.
M Query Example	Performs data transformations and custom calculations.	Use <b>let</b> expressions to load data, add columns, filter rows, and specify final output with <b>in</b> .

These features in Power Query Editor, combined with M Query, allow for dynamic and powerful data transformation capabilities, making Power BI a robust tool for complex data analysis.

What is "Column from Examples"?

**Column from Examples** is a feature that allows you to generate new columns based on example data you type in for each row. Power Query analyzes these examples and infers a pattern, which it then applies to the entire dataset. This feature is available in Power BI's Power Query Editor under the **Add Column** tab.

Use Cases for "Column from Examples"

- Extracting specific parts of text, such as first names, last names, or domain names from email addresses.
  - Combining data from multiple columns, like creating full names by combining first and last names.
  - Formatting data, such as changing date formats or capitalizing text.
  - Transforming text, such as adding prefixes, suffixes, or removing special characters.
- 

## How to Use "Column from Examples"

### Step-by-Step Guide:

#### 1. Load Data into Power Query Editor:

- Open Power BI and load your dataset into Power Query Editor by selecting **Home → Transform Data**.

#### 2. Select "Column from Examples":

- Go to the **Add Column** tab in the Power Query Editor.
- Select **Column from Examples** and choose either **From All Columns** (if you want to use data from multiple columns) or **From Selection** (if you only want to use data from selected columns).

#### 3. Enter Example Values:

- A new column titled **Column1** (or another name if specified) will appear on the right.
- In this new column, start typing the desired value for the first few rows.
- As you type examples for different rows, Power Query tries to detect a pattern and generate a formula to replicate it across the entire column.

#### 4. Review Suggested Formula:

- Power Query will display a suggested formula at the top based on your examples.
- Check if the suggested formula correctly replicates your desired transformation. If it doesn't, you may need to adjust your example values or type additional examples until it identifies the pattern accurately.

#### 5. Apply Changes:

- Once the formula accurately reflects the transformation you want, click **OK** to add the new column with the applied transformation.
- 

## Example Scenarios for "Column from Examples"

### Example 1: Extracting the Domain from Email Addresses

Suppose you have a column **Email** with values like `jane.doe@example.com`, and you want to create a new column **Domain** with only the domain part, like `example.com`.

- **Steps:**

1. Go to **Add Column → Column from Examples → From All Columns**.

2. In the new column, type `example.com` for the first email address.
3. Power Query will detect the pattern and apply it to all rows in the dataset.
4. Click **OK** to create the new column with the extracted domain.

## Example 2: Concatenating First and Last Names

If you have separate columns for **First Name** and **Last Name** and want to create a new column **Full Name** by combining these, you can use Column from Examples to achieve this.

- **Steps:**
  1. Select **Add Column** → **Column from Examples** → **From All Columns**.
  2. Type the full name in the new column (e.g., type `John Smith` for rows with **First Name** `John` and **Last Name** `Smith`).
  3. Power Query will generate a concatenation formula to combine first and last names for each row.

## Behind the Scenes: How Power Query Creates Formulas

When using **Column from Examples**, Power Query translates your example data into **M Code** (M Query Language) formulas. Here's what happens:

1. **Pattern Recognition:** Power Query analyzes your input examples to detect a consistent pattern. It attempts to deduce the logic behind the transformation based on your examples.
2. **Formula Generation:** Once it identifies the pattern, Power Query writes the equivalent M Query formula. For example:
  - If you're extracting the domain from an email, it might use the `Text.AfterDelimiter` function in M Code.
  - If you're combining text from multiple columns, it might use `Text.Combine`.
3. **Formula Testing:** The formula is applied to all rows in the preview to ensure it produces the expected results. You can edit or refine it by adding more examples.

## M Code Functions Commonly Used in Column from Examples

Below are some M Code functions that Power Query might use to implement transformations based on your examples:

- **Text.BeforeDelimiter:** Extracts text before a specified delimiter.  

```
m
Text.BeforeDelimiter([Email], "@") // Extracts username from email
```
- **Text.AfterDelimiter:** Extracts text after a specified delimiter.  

```
m
Text.AfterDelimiter([Email], "@") // Extracts domain from email
```
- **Text.Combine:** Concatenates multiple columns into one.  

```
m
Text.Combine({[First Name], [Last Name]}, " ") // Combines First Name and Last Name
```

- **Text.Upper, Text.Lower:** Converts text to uppercase or lowercase.  
m  
Text.Upper([Product]) // Converts product name to uppercase
- **Date.Year, Date.Month, Date.Day:** Extracts year, month, or day from a date.  
m  
Date.Year([Order Date]) // Extracts the year from the order date

## Benefits of Using Column from Examples

- **User-Friendly:** No need to know M Query; you just provide examples, and Power Query generates the formula.
- **Efficient:** Quickly create columns without having to manually code transformations.
- **Versatile:** Can be used for a wide range of data manipulations, including text transformations, date formatting, and data extraction.

## Limitations

- **Complex Transformations:** For very complex calculations, you may still need to use the **Custom Column** option and manually write M Query.
- **Pattern Recognition:** Sometimes, Power Query may not correctly identify a pattern, especially if the transformations are inconsistent or highly specific.

## Summary Table of Key Points

Feature	Purpose	Example Usage
Column from Examples	Creates new columns based on examples provided by the user	Extracting domain from email, combining first and last names
Pattern Recognition	Power Query detects patterns based on examples	Recognizes [Username]@[Domain] in emails to extract just the domain
Generated M Query	Power Query generates an M Query formula automatically	Uses functions like Text.BeforeDelimiter, Text.Combine, etc.
Benefits	Simplifies column creation without needing M Code knowledge	Quick, user-friendly, versatile
Limitations	May not handle very complex patterns	Manually written M Code may be needed for advanced transformations

**Column from Examples** is a highly efficient way to perform common data transformations in Power BI, making data modeling more accessible and intuitive.

In [ ]: