

## Summary: Data Cleaning with Power BI

### **SESSION OVERVIEW:**

By the end of this session, students will be able to:

- Understand various ways to do data cleaning and manipulation in Power BI
- Understand the features available under the Add Column, View, Tools and Help tabs in the ribbon
- Use various AI tools in Power BI

### **KEY TOPICS AND EXAMPLES:**

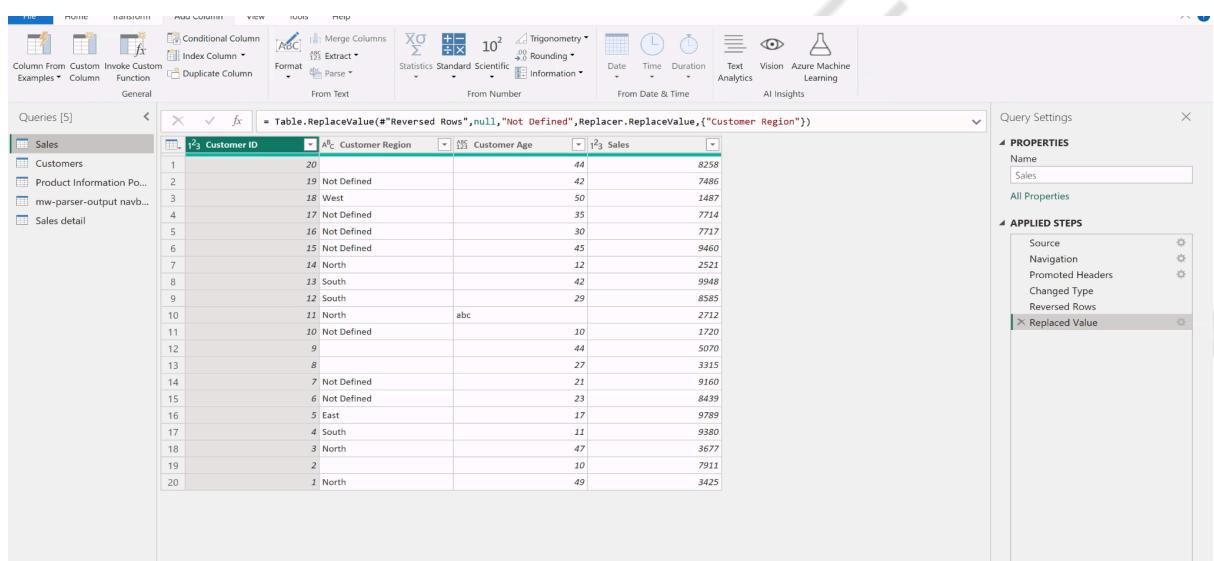
#### **Using Power Query Editor for Data Cleaning**

While doing data cleaning through Power Query, it is essential to realize that Power BI just reads data, it does not write data back to the source. In other words, the original dataset remains unchanged while you transform/clean data on Power BI.

Whenever we perform any cleaning/transformation function, it gets added as a new **step** in the window on the right side: in the Query Settings window. Each step can be deleted to reverse the previous action. This helps you manage your data manipulations (for transformation/cleaning).

Power Query Editor provides some of the features and functionality given below for cleaning data:

1. The **Add Column** tab has multiple options to clean the data. These are explained below.

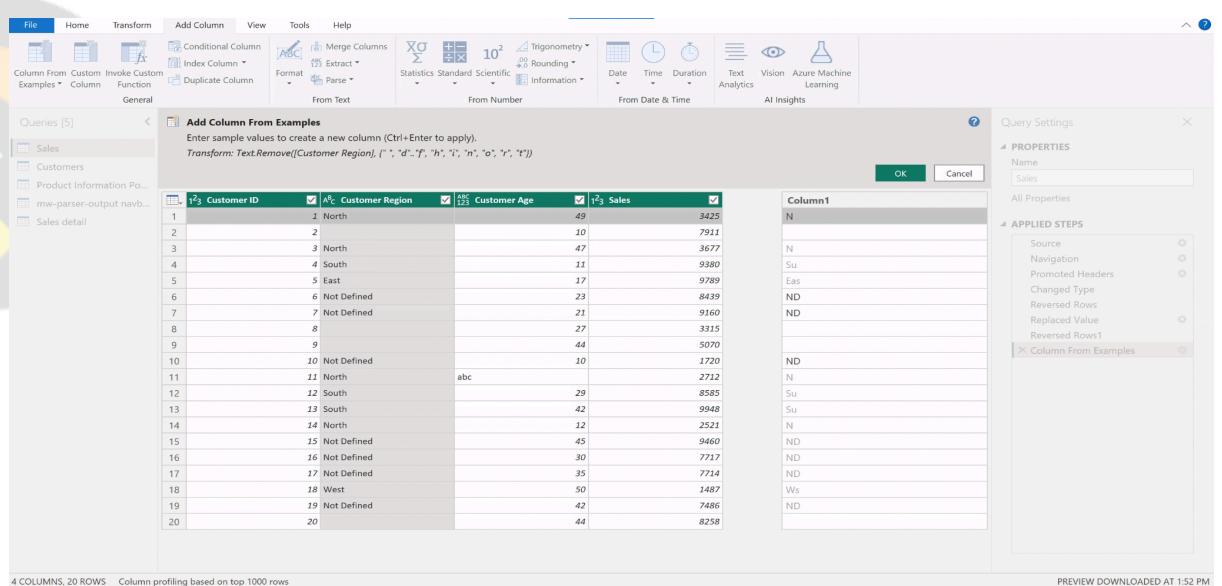


The screenshot shows the Microsoft Power Query Editor interface. The top ribbon is visible with the 'Add Column' tab selected. The main area displays a table with columns: Customer ID, Customer Region, Customer Age, and Sales. The formula bar at the top shows the formula: = Table.ReplaceValue(#"Reversed Rows",null,"Not Defined",Replacer.ReplaceValue,{"Customer Region"}). The 'APPLIED STEPS' pane on the right lists the following steps: Source, Navigation, Promoted Headers, Changed Type, Reversed Rows, and Replaced Value. The 'Replaced Value' step is currently highlighted.

Customer ID	Customer Region	Customer Age	Sales
1	20	44	8258
2	19 Not Defined	42	7486
3	18 West	50	1487
4	17 Not Defined	35	7714
5	16 Not Defined	30	7717
6	15 Not Defined	45	9460
7	14 North	12	2521
8	13 South	42	9948
9	12 South	29	8585
10	11 North abc		2712
11	10 Not Defined	10	1720
12	9	44	5070
13	8	27	3315
14	7 Not Defined	21	9160
15	6 Not Defined	23	8439
16	5 East	17	9789
17	4 South	11	9380
18	3 North	47	3677
19	2	10	7911
20	1 North	49	3425

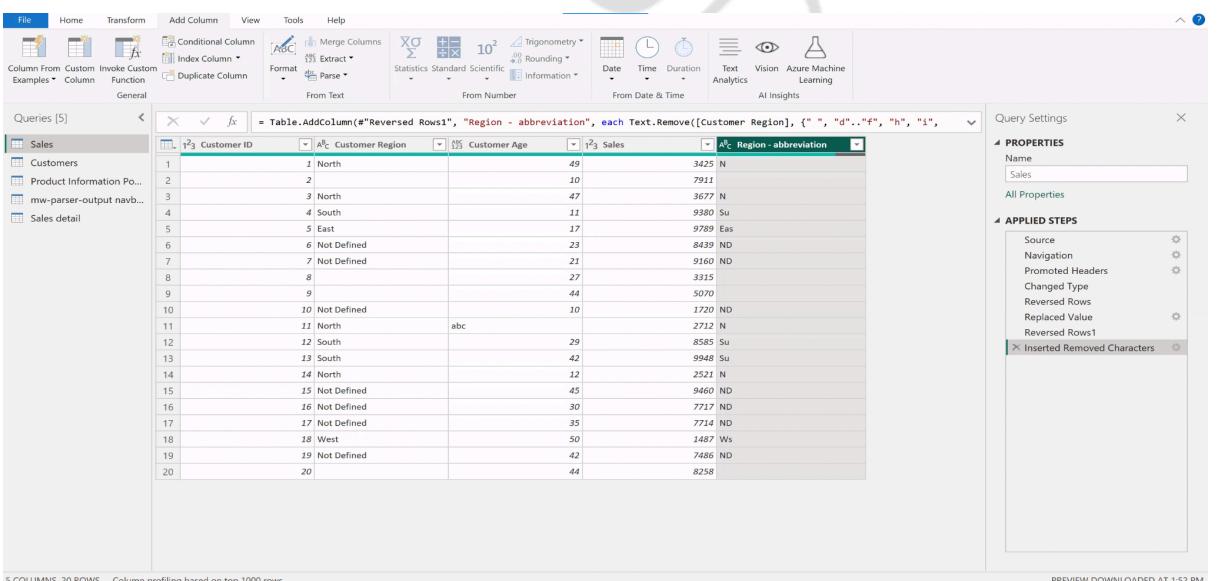
- **Add Column From Examples:** This is used to create a new column from an existing column. Example: From date we can get month, year etc.

**Example:** The steps and image below demonstrates its use from [this Dataset](#).



The screenshot shows the Power BI desktop interface with the 'Add Column' dialog open. The 'From Selection' tab is selected. A preview pane on the right shows the original 'Customer Region' column and the resulting 'Column1' with values like 'N', 'Su', 'Eas', 'ND', etc. The formula bar at the bottom contains the expression: `=Table.AddColumn(#"Reversed Rows1", "Region - abbreviation", each Text.Remove([Customer Region], {",", "d", "f", "h", "i"}))`. The 'APPLIED STEPS' pane on the right lists the 'Column From Examples' step.

- Select the column from which you intend to generate a new column.
- Select 'Column From Examples' → 'From Selection'.
- Enter sample values into the new column. **Power BI will intelligently recognise the formula that may help you get those values.**
- Click OK or press Ctrl+Enter to apply.



The screenshot shows the Power BI desktop interface after applying the 'Column From Examples' step. The 'Region - abbreviation' column has been added to the dataset. The 'APPLIED STEPS' pane on the right shows the 'Inserted Removed Characters' step.

- **Custom Column:** We can also create a new column based on a custom formula. These formulas are written in M Query. **M Query is Power BI language used in Power Query editor to transform, clean and manipulate data.** A detailed list of M Query functions is given here: <https://learn.microsoft.com/en-us/powerquery-m/power-query-m-function-reference>

**Example:** In the [Product Information dataset](#), we can calculate the sales for each customer by multiplying the **Qty bought** and **Price per item**. This will give us the sales value for each customer ID. The image below demonstrates this formula.

### Some other Formulas:

To create a new column "**Bulk Purchase**", which will mark orders as "**High**" if Qty bought is greater than 50, otherwise mark them as "**Low**".

```
if [Qty bought] > 50 then "High" else "Low"
```

To categorize the "Bulk Purchase" column into **3 categories**.

```
if [Qty bought] > 50 then "High"
else if [Qty bought] >= 30 and [Qty bought] <= 50 then "Medium"
else "Low"
```

Query for adding a new column named "**Price**" that **concatenates** "\$" with the existing "**Price per item**" column.

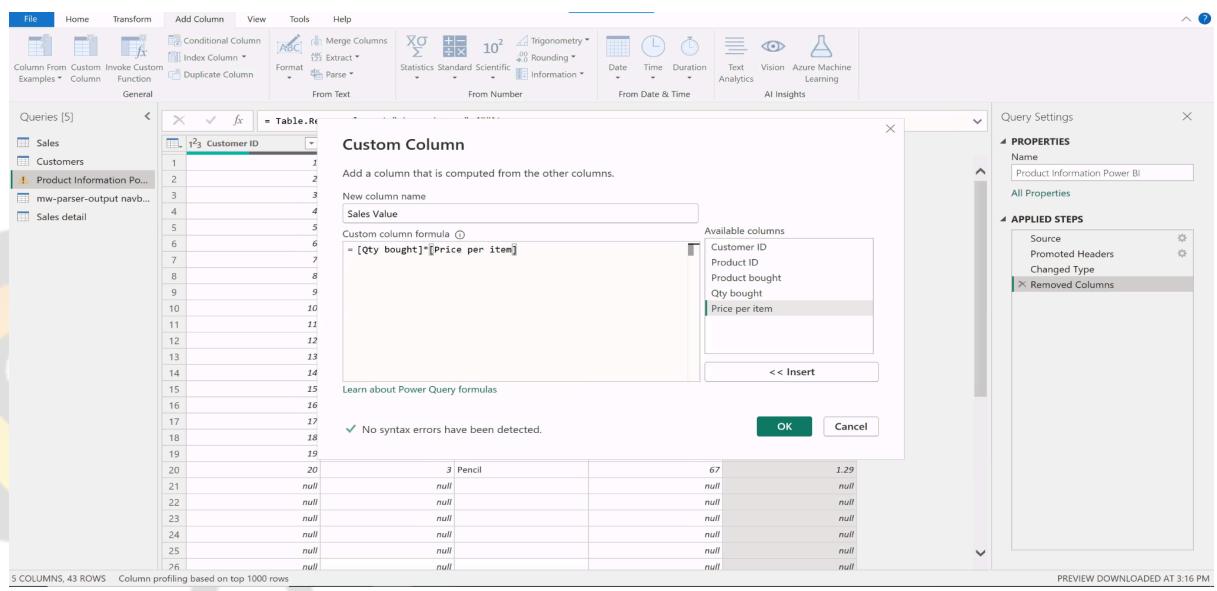
```
"$" & Number.ToString([Price per item])
```

To categorize products based on their **price per item**, capitalize the **category labels**, and concatenate them with their respective **price per item** values to create a more descriptive and readable output.

```
= if [Price per item] > 10 then Text.Proper("expensive") & " - $" &
Number.ToString([Price per item])

else if [Price per item] >= 5 and [Price per item] <= 10 then
Text.Proper("affordable") & " - $" & Number.ToString([Price per item])

else Text.Proper("cheap") & " - $" & Number.ToString([Price per item])
```



**Custom Column**

Add a column that is computed from the other columns.

New column name: Sales Value

Custom column formula: = [Qty bought]\*[Price per item]

Available columns:

- Customer ID
- Product ID
- Product bought
- Qty bought
- Price per item

Learn about Power Query formulas

✓ No syntax errors have been detected.

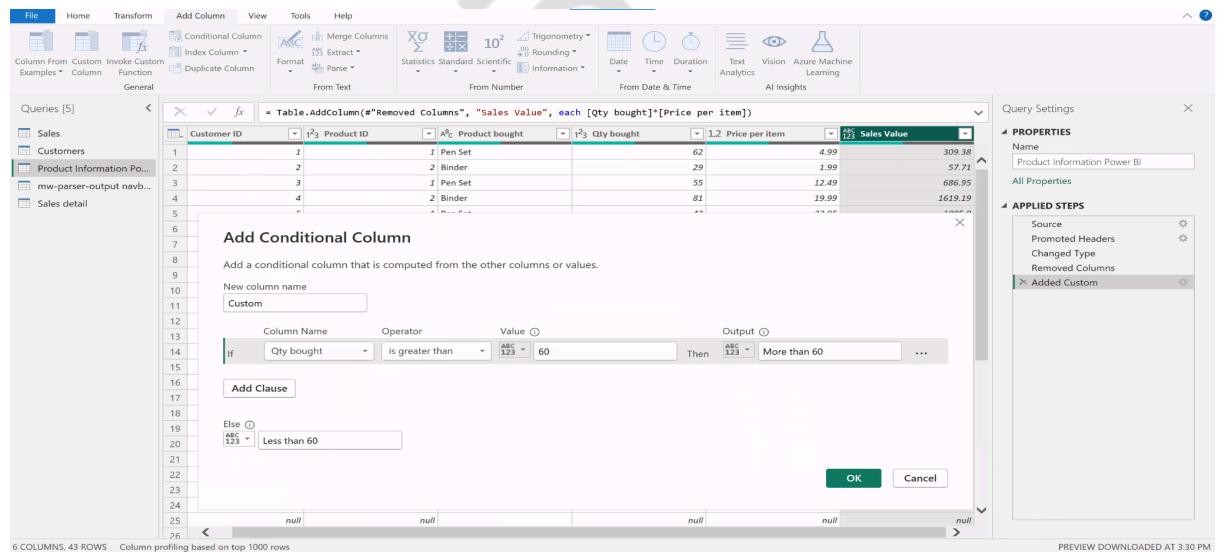
OK Cancel

5 COLUMNS, 43 ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED AT 3:16 PM

- **Conditional Column:** It is used to create a new column using a condition that is based on another column.

**Example:** In the [Product Information dataset](#), if we want to create a bucket for customers that bought more than 60 items, then we can use a conditional column for that.



**Add Conditional Column**

Add a conditional column that is computed from the other columns or values.

New column name: Custom

Column Name	Operator	Value	Output	
if	Qty bought	is greater than	60	Then 123 More than 60
Else	123	Less than 60		

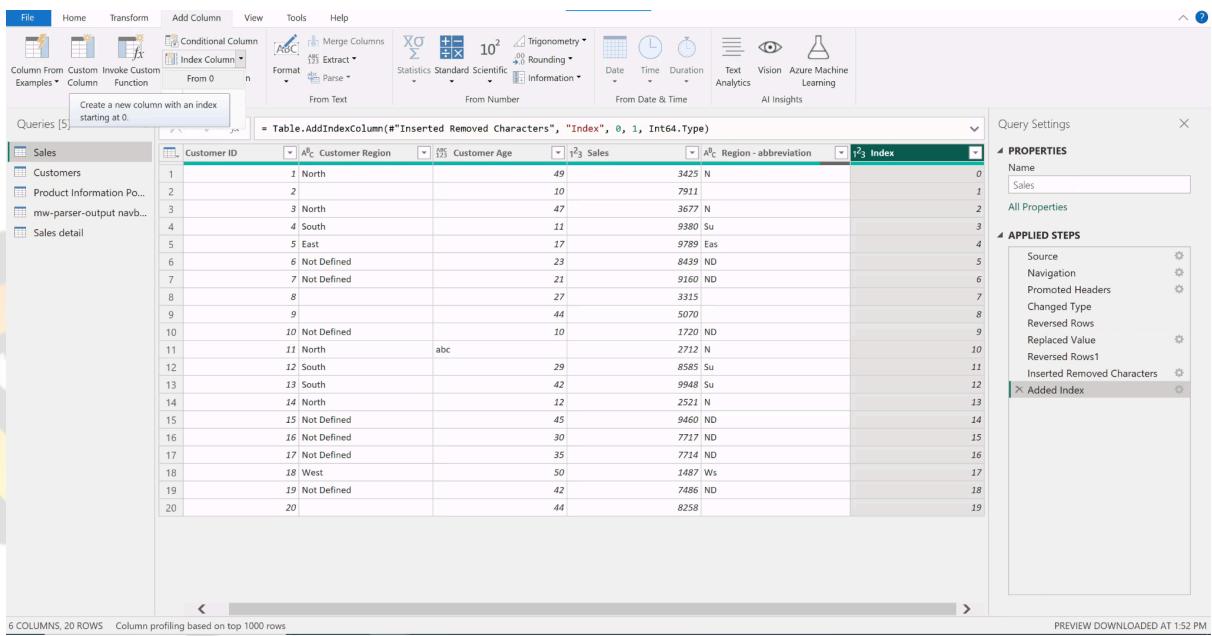
OK Cancel

6 COLUMNS, 43 ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED AT 3:30 PM

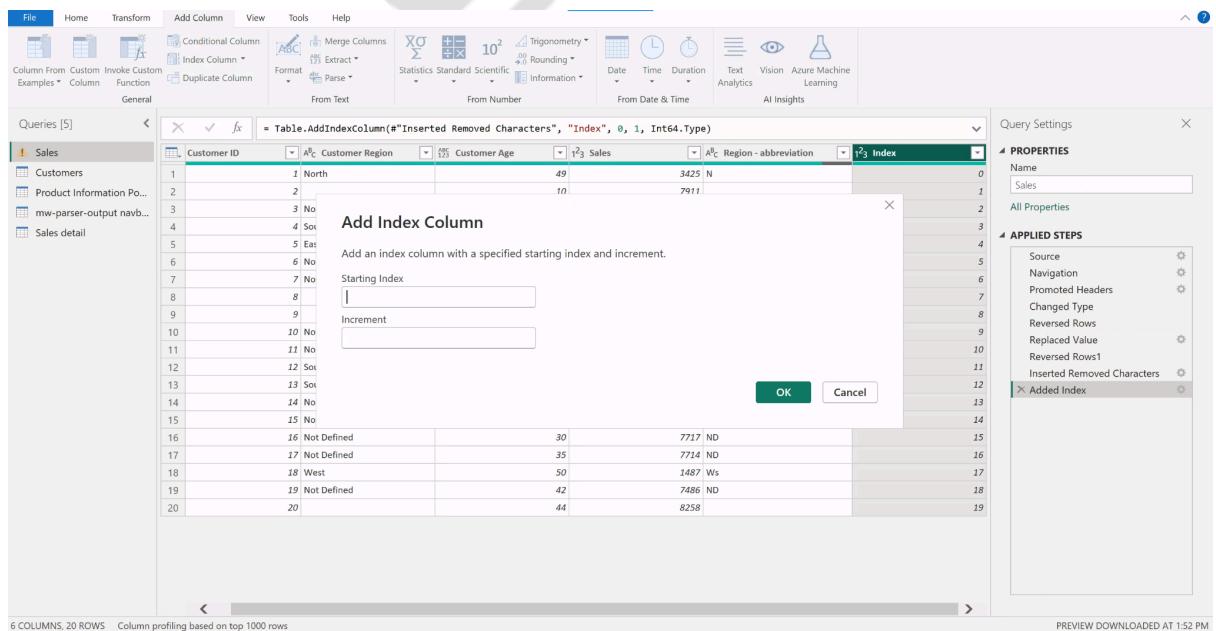
- **Index Column:** It creates a new column (from 0 or 1).

**Example:** In our Sales dataset, we do not have an index column. It can be added from the 'Add Column' tab → 'Index Column' → 'From 0'.



The screenshot shows the Power BI desktop interface. A query named "Sales" is selected. In the ribbon, under the "Transform" tab, the "Add Column" section is open, specifically the "Index Column" option. A tooltip indicates "Create a new column with an index starting at 0." The main area displays a table with columns: Customer ID, Customer Region, Customer Age, Sales, Region - abbreviation, and Index. The Index column contains values from 0 to 19. The "APPLIED STEPS" pane on the right shows the step "Added Index".

Index column option provides us with the flexibility to start from 0, 1 or any custom number with a custom increment (see image below).

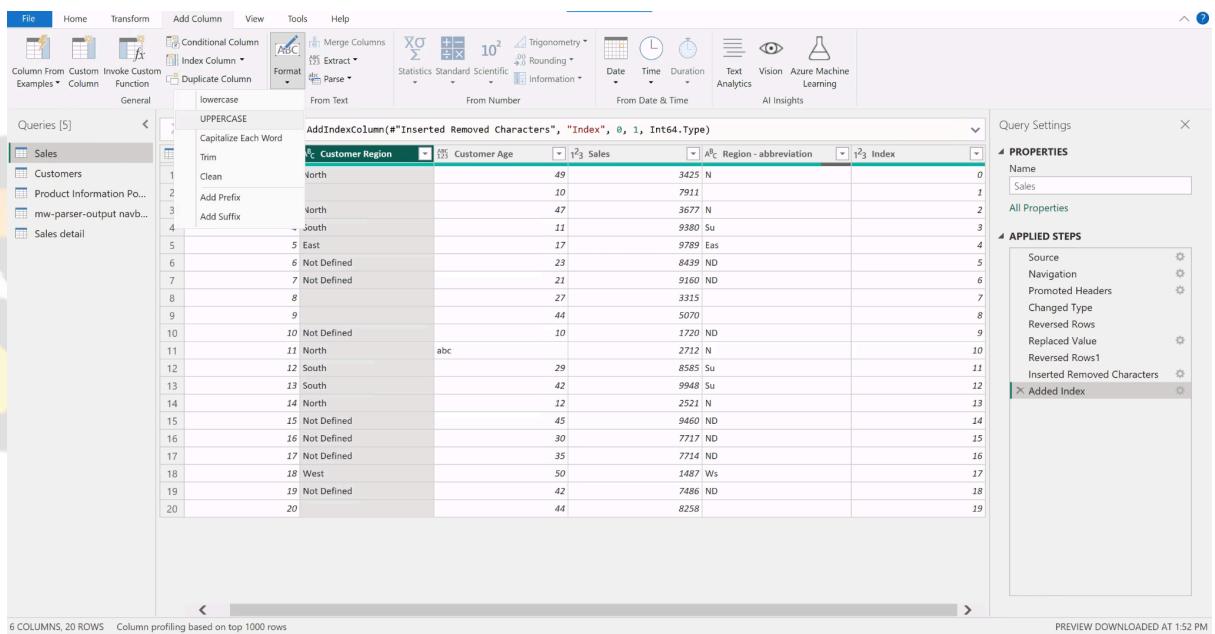


The screenshot shows the Power BI desktop interface with the "Add Index Column" dialog box open. The dialog has two input fields: "Starting Index" (set to 0) and "Increment" (set to 1). The main table view shows the same data as the previous screenshot, with the "Index" column now starting at 0. The "APPLIED STEPS" pane on the right shows the step "Added Index".

**Note:** By default, Power BI adds an Index column at the end but it can be moved to the beginning by simply dragging the header of the column.

- **Format:** It is used to change the case of the text. It can be used to convert it to lowercase, uppercase, capitalize each word, trim, clean or add prefix/suffix.

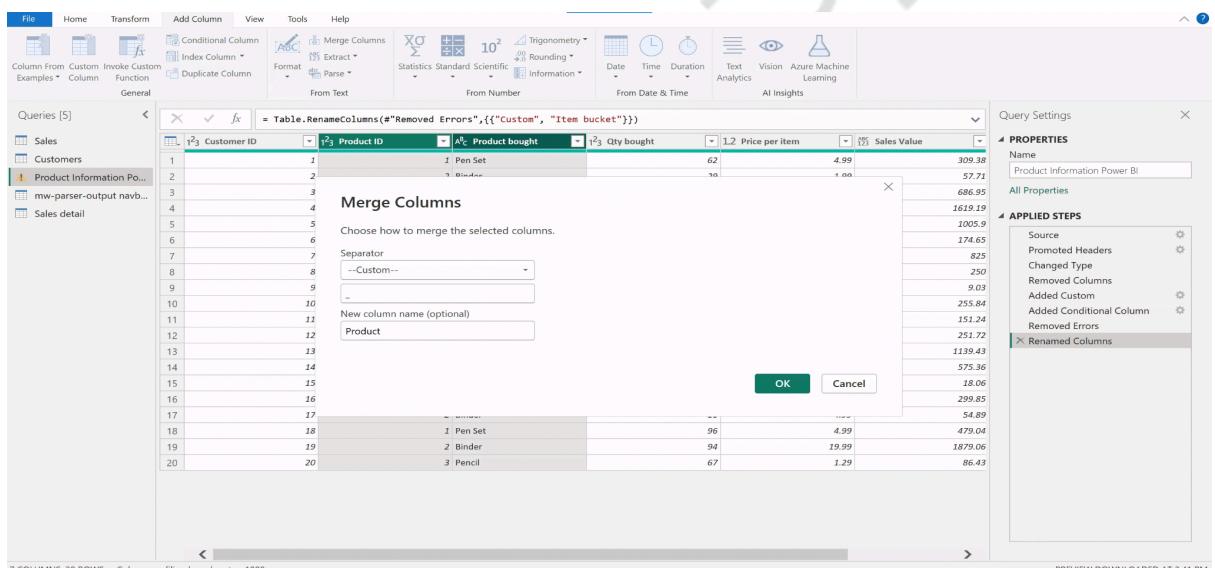
**Example:** We can choose from the options below in the [dataset](#). We can convert the ‘Customer Region’ column to UPPERCASE.



The screenshot shows the Power BI desktop interface with the 'Query Editor' open. The 'Transform' tab is selected. A table is displayed with columns: 'Customer Region', 'Customer Age', 'Sales', 'Region - abbreviation', and 'Index'. The 'Customer Region' column contains values like 'North', 'South', 'East', etc. The 'Region - abbreviation' column contains abbreviations like 'N', 'S', 'E', etc. The 'Index' column is a numerical sequence from 1 to 20. On the left, there's a list of queries: Sales, Customers, Product Information Po..., mw-parser-output navb..., and Sales detail. On the right, the 'Properties' pane shows the query name is 'Sales' and the 'Applied Steps' pane lists steps like 'Source', 'Navigation', 'Promoted Headers', and 'Inserted Removed Characters'. A tooltip 'Added Index' is visible over one of the steps.

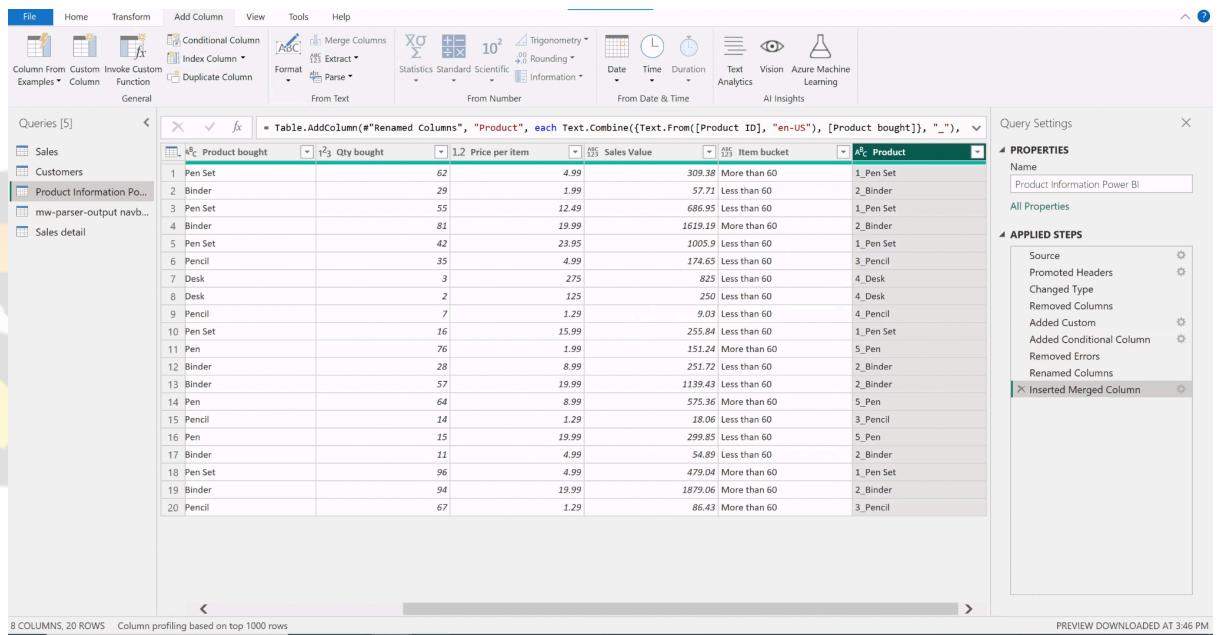
- **Merge Columns:** This is used to merge multiple columns. To perform this operation we need to select multiple columns, click on the Merge Columns option and then select the type of separator we require.

**Example:** Suppose, in the [Product Information dataset](#) we want to merge the ‘Product ID’ and ‘Product bought’ column to create a single column that will be in this format: ‘ProductID\_ProductBought’. To do this, we can use the ‘Merge Columns’ operation, as shown below.



The screenshot shows the Power BI desktop interface with the 'Query Editor' open. The 'Transform' tab is selected. A table is displayed with columns: 'Customer ID', 'Product ID', 'Product bought', 'Qty bought', 'Price per item', and 'Sales Value'. The 'Product ID' and 'Product bought' columns are selected, and a 'Merge Columns' dialog box is open. The dialog box asks 'Choose how to merge the selected columns.' under 'Separator' with '---Custom---' selected. It also has a 'New column name (optional)' field containing 'Product'. At the bottom are 'OK' and 'Cancel' buttons. The 'Properties' pane on the right shows the query name is 'Product Information Power BI' and the 'Applied Steps' pane lists steps like 'Source', 'Promoted Headers', 'Changed Type', 'Removed Columns', 'Added Custom', and 'Renamed Columns'. A tooltip 'Renamed Columns' is visible over one of the steps.

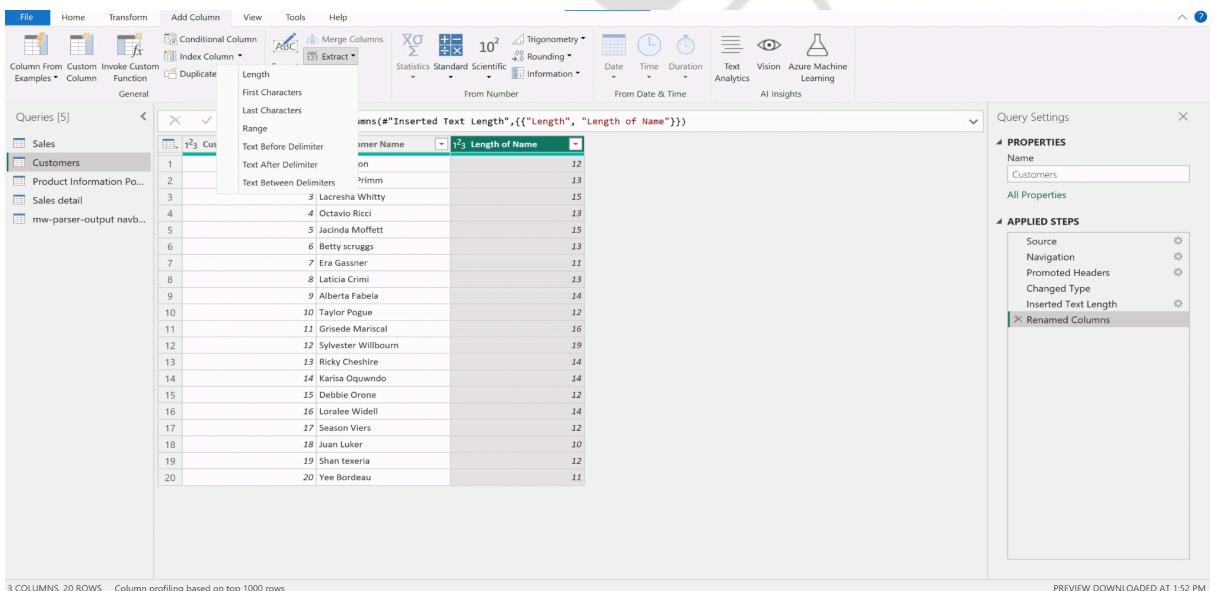
The final, merged column will look like this:



This screenshot shows the Power BI Editor interface. The top ribbon has tabs like File, Home, Transform, Add Column, View, Tools, and Help. The 'Transform' tab is selected. The 'Extract' button is highlighted. The main area shows a table with columns: Product bought, Qty bought, Price per item, Sales Value, Item bucket, and Product. The 'Product' column contains values like 'Pen Set', 'Binder', etc. The bottom right corner shows 'PREVIEW DOWNLOADED AT 3:46 PM'.

- **Extract:** Using **Extract** we can extract the first characters, last characters, length, Range, Text before delimiter, Text after delimiter and Text Between Delimiters.

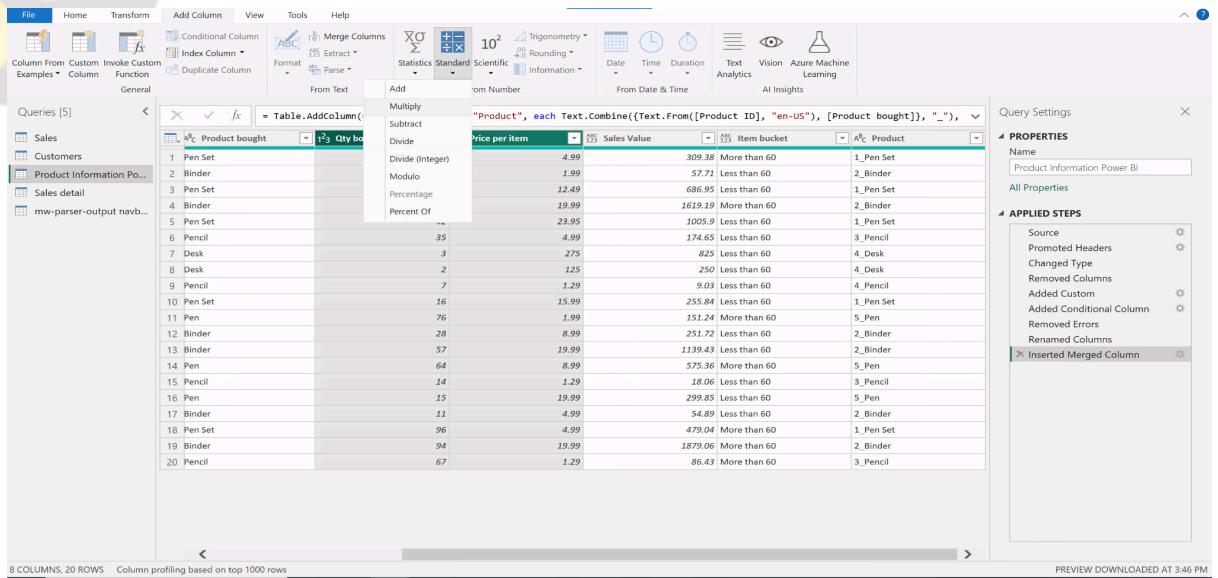
**Example:** Using the [Customers table in the Sales dataset](#), if we want to find the length of the name of the customer, we can use the following steps: click 'Extract' → Select 'Length' → rename the field as per your convenience. The final result as well as the options Extract provides are shown in the image below.



This screenshot shows the Power BI Editor interface with the 'Extract' button selected in the ribbon. A dropdown menu is open for the 'Customer Name' field, showing options like Length, First Characters, Last Characters, Range, Text Before Delimiter, Text After Delimiter, and Text Between Delimiters. The 'Length' option is selected. The bottom right corner shows 'PREVIEW DOWNLOADED AT 1:52 PM'.

- **Statistics:** We have numerical operations like ‘Statistics’ that can be used to create outputs that are based on statistical measures like mean, median, mode etc.
- **Standard:** It enables you to perform basic mathematical operations on your data, like Add, Subtract, Multiply, Divide etc. We can either perform these operations on the selected column with a fixed value or with another column.

**Example:** In our [Product Information dataset](#), we can multiply “Qty bought” with “Price per item” using this operation. Image below shows this operation being performed.



The screenshot shows the Power BI Query Editor interface. A context menu is open over a column named "Product bought". The menu is expanded to show the "Multiply" option under the "Add" section. The formula bar at the top shows the formula: `"Product", each Text.Combine([Text.From([Product ID], "en-US"), [Product bought]), "_"]`. The main grid displays data for various products, and the right pane shows the "APPLIED STEPS" history.

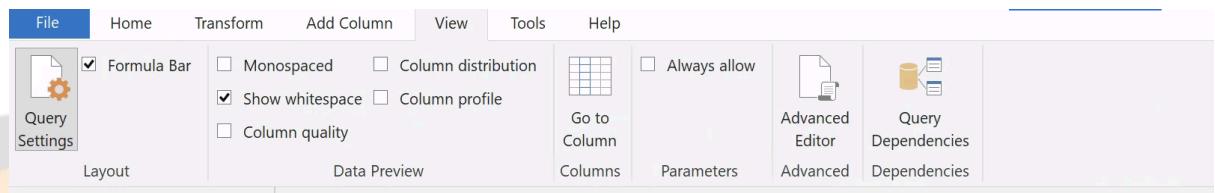
Product	Price per Item	Sales Value	Item bucket	Product	
Pen Set	4.99	309.38	More than 60	1_Pen Set	
Binder	1.99	57.71	Less than 60	2_Binder	
Pen Set	12.49	686.95	Less than 60	1_Pen Set	
Binder	19.99	1619.19	More than 60	2_Binder	
Pen Set	23.95	1005.9	Less than 60	1_Pen Set	
Pencil	35	174.65	Less than 60	3_Pencil	
Desk	3	825	Less than 60	4_Desk	
Desk	2	250	Less than 60	4_Desk	
Pencil	7	1.29	9.03	Less than 60	4_Pencil
Pen Set	16	15.99	255.84	Less than 60	1_Pen Set
Pen	76	1.99	151.24	More than 60	5_Pen
Binder	28	8.99	251.72	Less than 60	2_Binder
Binder	57	19.99	1139.43	Less than 60	2_Binder
Pen	64	8.99	575.36	More than 60	5_Pen
Pencil	14	1.29	18.06	Less than 60	3_Pencil
Pen	15	19.99	299.85	Less than 60	5_Pen
Binder	11	4.99	54.89	Less than 60	2_Binder
Pen Set	96	4.99	479.04	More than 60	1_Pen Set
Binder	94	19.99	1879.06	More than 60	2_Binder
Pencil	67	1.29	86.43	More than 60	3_Pencil

- **Scientific:** It is used to perform scientific operations like square root, power, absolute value and exponent.

## AI TOOLS IN POWER BI:

- **Text Analytics:** Helps in detecting the language of a column, extract key phrases and generate sentiment scores.
- **Vision:** It helps us analyse the images and return tags that are based on what the picture contains.
- **Azure Machine Learning:** If your organisation has subscribed for Azure Machine Learning, then some of its options can be found here through which the BI analyst can directly use those models in Power Query Editor.

2. The **View** tab enables us to add/disable some settings and helps us view data in a better format. It has some useful options for cleaning data as listed below.



- **Monospaced:** It displays data in a fixed-width font, making each character the same width.

**Example:** The image below from the Sales dataset shows data with the monospace check-box ticked.

Index	Customer ID	Customer Region	Customer Age	Sales	Region - abbreviation
1	0	1 North	49	3425	N
2	1	2	10	7911	
3	2	3 North	47	3677	N
4	3	4 South	11	9380	Su
5	4	5 East	17	9789	Eas
6	5	6 Not Defined	23	8439	ND
7	6	7 Not Defined	21	9160	ND
8	7	8	27	3315	
9	8	9	44	5070	
10	9	10 Not Defined	10	1720	ND
11	10	11 North	ERROR	2712	N
12	11	12 South	29	8585	Su
13	12	13 South	42	9948	Su
14	13	14 North	12	2521	N
15	14	15 Not Defined	45	9460	ND
16	15	16 Not Defined	30	7717	ND
17	16	17 Not Defined	35	7714	ND
18	17	18 West	50	1487	Ws
19	18	19 Not Defined	42	7486	ND
20	19	20	44	8258	

- **Column Quality:** It shows the percentage of valid, error, or empty values in a column. It helps us identify errors easily and fix them.

**Example:** The image below shows the column quality for some of the columns in the dataset.

- The Region-abbreviation column has 20% empty values, so we need to handle these missing values separately. For example, we can right click on the column and replace empty values by some fixed value.
- The customer age column has 5% error values. We can remove errors by right clicking on the column and replacing errors by a fixed value.

Screenshot of Power BI Query Editor showing the 'Sales' query. The 'Customer Age' column has been transformed to type Int64. The 'Customer Region' column is highlighted, showing its distribution: 100% Valid, 0% Error, and 0% Empty. The 'Region - abbreviation' column shows 80% Valid, 0% Error, and 20% Empty. The 'Customer ID' and 'Sales' columns also show their respective distributions.

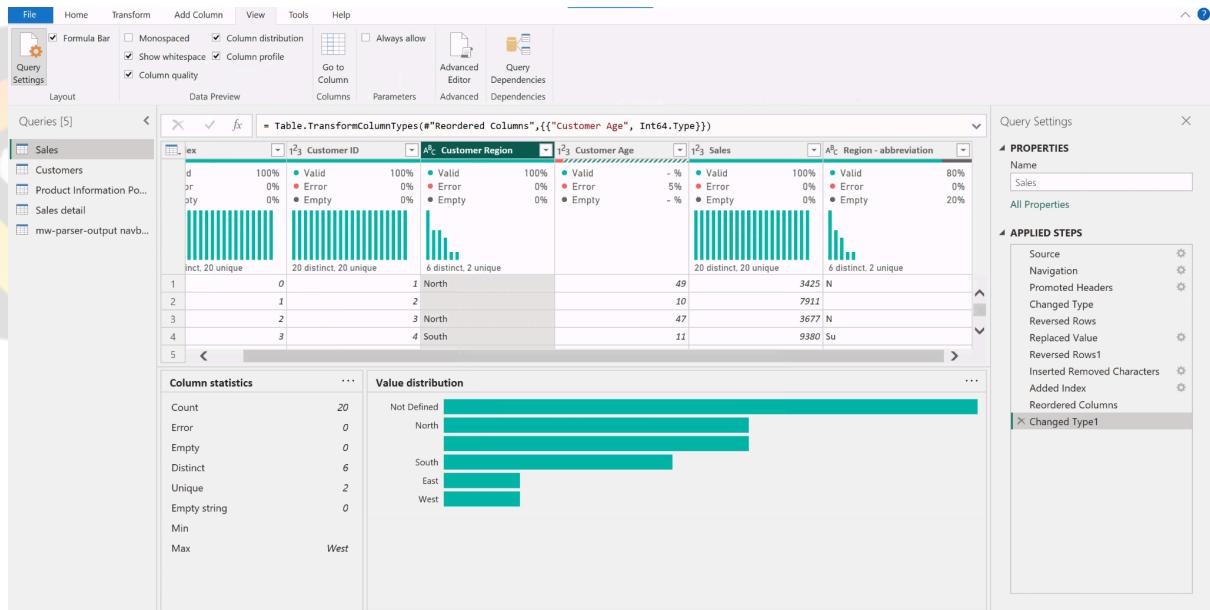
- **Column Distribution:** It is used to view the distribution of the data in each column. In particular, it displays the number of distinct and unique values in each column.

**Example:** In the Sales dataset, it enables one to identify the different regions and the number of distinct/unique values.

Screenshot of Power BI Query Editor showing the 'Sales' query. The 'Customer Age' column has been transformed to type Int64. The 'Customer Region' column is highlighted, showing its distribution: 100% Valid, 0% Error, and 0% Empty. The 'Region - abbreviation' column shows 80% Valid, 0% Error, and 20% Empty. The 'Customer ID' and 'Sales' columns also show their respective distributions. The 'Customer Region' column has 20 distinct, 20 unique values.

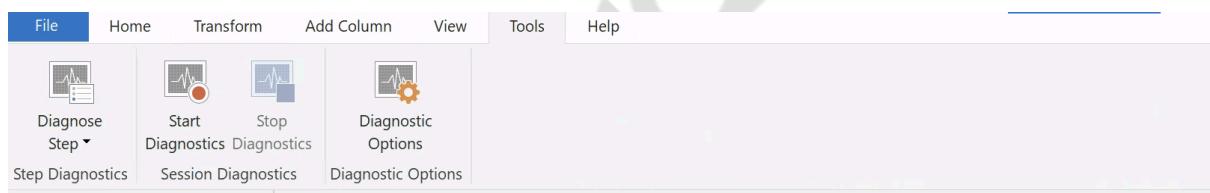
- **Column Profile:** It provides statistical information about the distribution of data, including errors, empty values, distinct values, minimum values, and maximum values. It provides a snapshot of the column and can be very useful to understand the column data distribution.

**Example:** In the Sales dataset, if we want to understand the distribution of each region as well as the errors, empty values, distinct values, minimum values, and maximum values, **Column profile** would be the best option, as displayed below. It also displays the quantum in each of these categories of region.



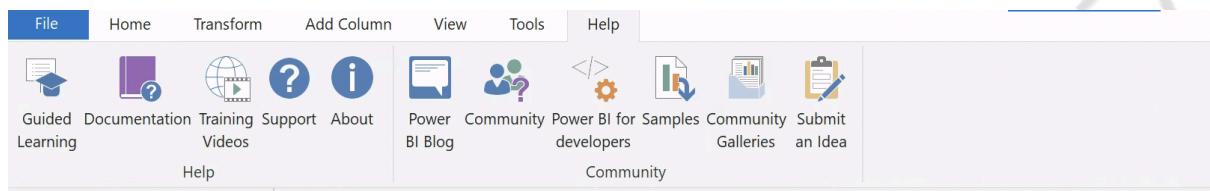
**Note:** Notice that on the right side we can see all the applied steps from the beginning. At any point we can remove any of these steps or go back and change the manner in which we performed any of these steps.

### 3. The **Tools** tab: It helps us diagnose and troubleshoot data transformations.



The image above shows the options available in the Tools tab.

### 4. The **Help** tab: It provides us with documentation and other information to help us understand the features of Power Query Editor.



The image above shows the options available in the Help tab.