

How to Convert Dates to Text in Power BI: A Step-by-Step Guide

Authored by
stats writer

January 29, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Convert Dates to Text in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128520>

In the expansive environment of Power BI, managing and manipulating data types is fundamental to creating effective reports and visualizations. A common requirement for report builders is converting a standard date field into a text or string format. This conversion is often necessary for concatenation with other strings, customized presentation formats, or specialized sorting logic that the native date hierarchy cannot easily handle. Power BI provides several robust mechanisms for this transformation, primarily categorized into two approaches: utilizing the user interface (UI) to change the intrinsic column type, or employing powerful DAX functions for calculated, dynamic formatting.

While the quick UI method offers immediate results, the more flexible DAX approach grants granular control over the output structure. Key DAX tools for this task include the CONVERT function for general type casting, and the superior FORMAT function, which allows users to specify highly customized display strings (e.g., "YYYY-MM-DD" or "Month Name, YYYY"). Furthermore, specialized functions like DATENAME exist to extract specific textual components, such as the full name of the month or day, directly from the date value. Understanding both the simplistic UI conversion and the advanced DAX methods is crucial for mastering date presentation within your reports.

Converting Date to Text in Power BI: A Comprehensive Guide to Multiple Methods

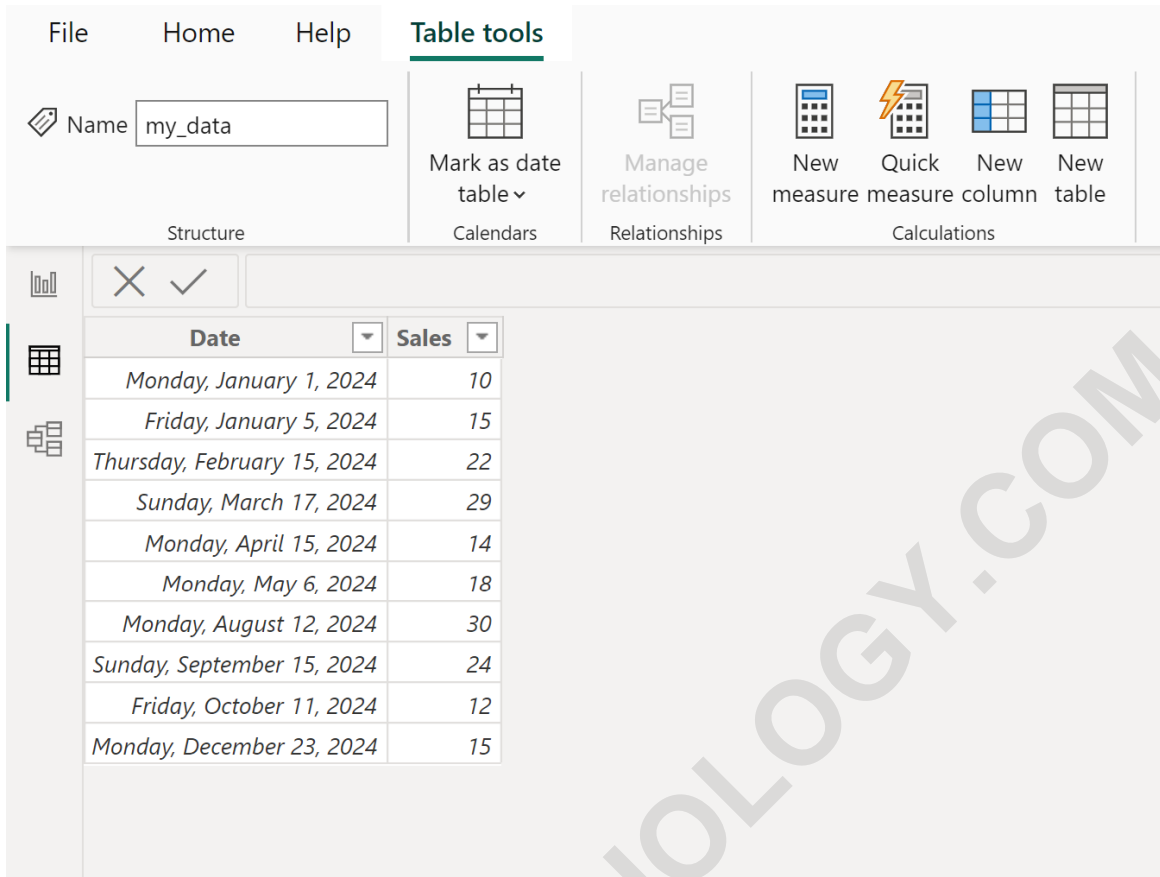
The process of transforming a date column into a text column within the Power BI environment can be achieved through both direct modeling actions and calculated columns using DAX. The choice between these methods depends heavily on whether you need a permanent data type change applied at the source or a dynamic, formatted text value used exclusively within the report calculations.

We will explore two primary practical approaches utilized by analysts:

Method 1: Utilizing the Data Type Dropdown Menu in Column Tools (A permanent, model-level change).

Method 2: Writing a Custom DAX Function (A calculated column providing flexibility).

To illustrate these techniques effectively, consider the following sample data table loaded into the Power BI Model. This table contains a standard date column, which serves as the focus for our conversion examples.



The screenshot displays the Power BI interface with the 'Table tools' ribbon selected. The ribbon includes options for 'Mark as date table', 'Manage relationships', and 'Calculations' (New measure, Quick measure, New column, New table). Below the ribbon, a data table is visible with two columns: 'Date' and 'Sales'. The table contains the following data:

Date	Sales
Monday, January 1, 2024	10
Friday, January 5, 2024	15
Thursday, February 15, 2024	22
Sunday, March 17, 2024	29
Monday, April 15, 2024	14
Monday, May 6, 2024	18
Monday, August 12, 2024	30
Sunday, September 15, 2024	24
Friday, October 11, 2024	12
Monday, December 23, 2024	15

Understanding the Necessity of Date-to-Text Conversion

Dates are inherently complex pieces of data, often stored internally as numerical values representing the number of days or seconds since a fixed point in time (like January 1, 1900, in Excel/Power BI's underlying engine). When a column is defined as a Date or DateTime Data type, Power BI automatically applies specific formatting rules, localization settings, and hierarchical properties (year, quarter, month, day) that are essential for time-intelligence calculations. However, there are scenarios where these built-in features are undesirable or actively interfere with report requirements.

For instance, if you are attempting to display a date in a very specific, non-standard format (e.g., "Wk 42, 2024"), or if you need to concatenate the date value with a descriptive string like a product ID or a report title, the date must first be converted into a standard string or text Data type. Attempting to concatenate a date value directly with a text string without explicit conversion will often result in unexpected behavior or errors, as Power BI struggles to automatically coerce the Date Data type into a string for display purposes. Therefore, explicit conversion is a necessary step to ensure data integrity and desired display output.

The distinction between the two core methods--UI adjustment versus DAX calculated column--is

critical. Changing the Data type via the UI affects the source column itself and applies the change permanently across the model, potentially losing the time-intelligence capabilities associated with the original date type. Conversely, using DAX creates a new column, leaving the original date column intact for any required numerical calculations, thereby offering a non-destructive and generally preferred approach for reporting flexibility.

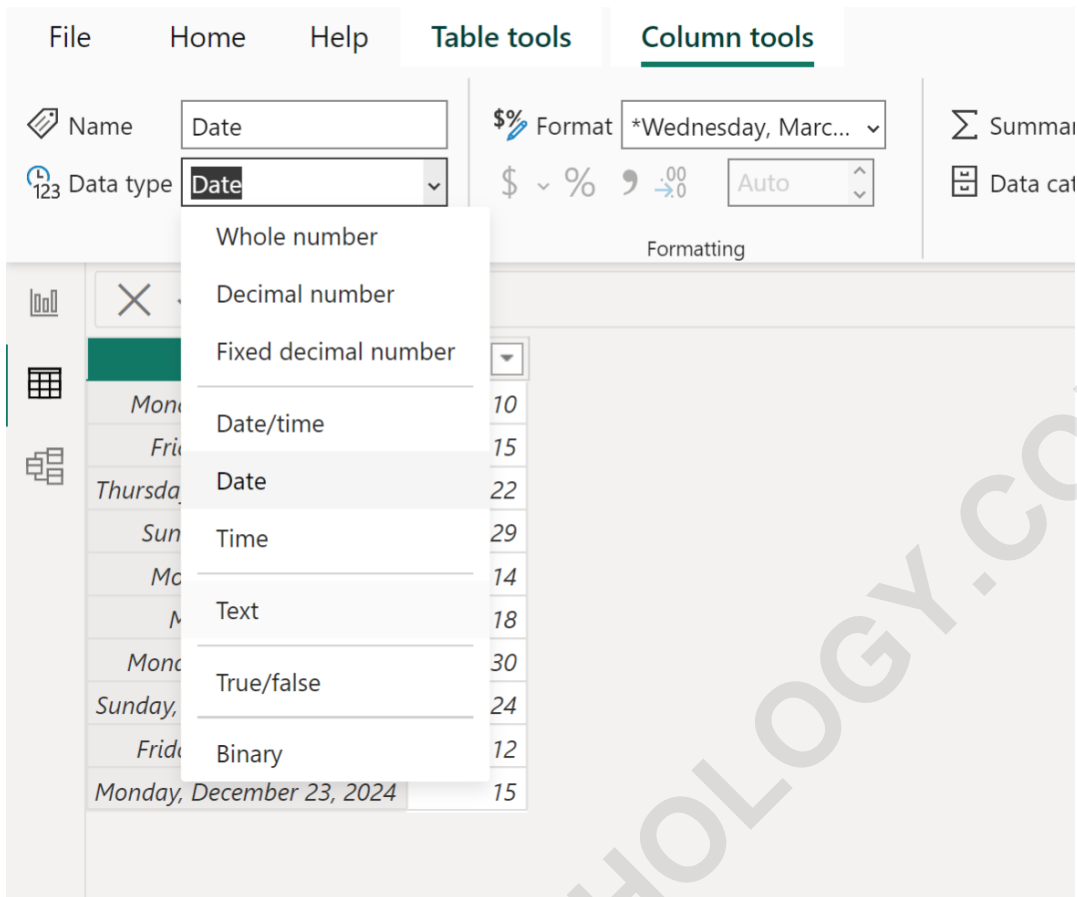
Method 1: Quick Conversion Using the Power BI User Interface

The most straightforward way to convert a Date column to a Text column is by manipulating the Data type directly within the modeling view of Power BI Desktop. This method is fast and requires no knowledge of DAX, making it suitable for simple reports where custom formatting is not a concern and where the original date column's numerical properties are no longer required.

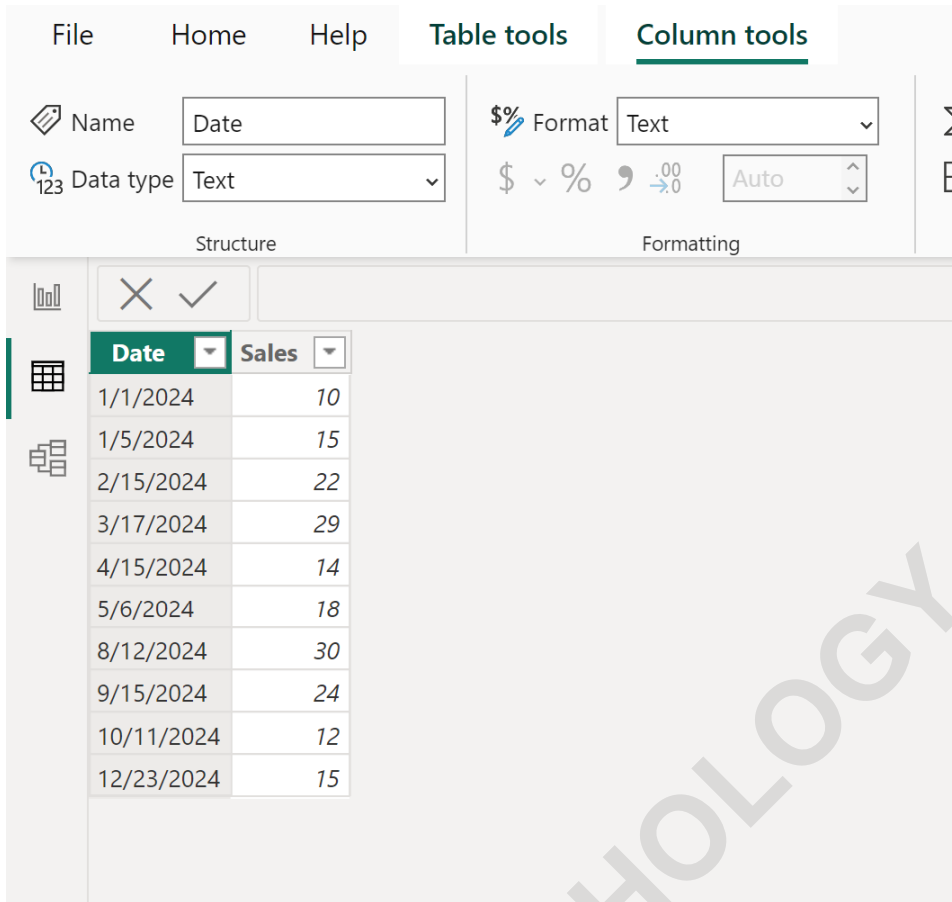
Suppose, for the purposes of a quick visual check or data export, we wish to convert the **Date** column in the provided sample table from its native Date/Time Data type to a Text Data type. The process begins in the Data View or Report View of Power BI Desktop. The user must first locate the column they wish to modify.

To execute this conversion, the user must click directly on the header of the **Date** column within the data preview pane. Selecting the column activates the contextual ribbon menu, specifically bringing up the **Column tools** tab. Within this tab, located prominently in the formatting section, you will find the **Data type** dropdown menu. This menu lists all compatible data types, including Decimal Number, Whole Number, Date/Time, and Text.

We can then click the **Data type** dropdown menu and explicitly select the **Text** option. Power BI will instantly process this change across the entire model. Note that depending on the data size, this operation might take a moment.



Upon successful execution, the column's icon in the Fields pane will change from a calendar symbol (indicating Date) to an 'Abc' symbol (indicating Text). Furthermore, the data displayed in the preview pane confirms that the values are now treated as strings, typically defaulting to the locale's short date format (e.g., mm/dd/yyyy) upon conversion:



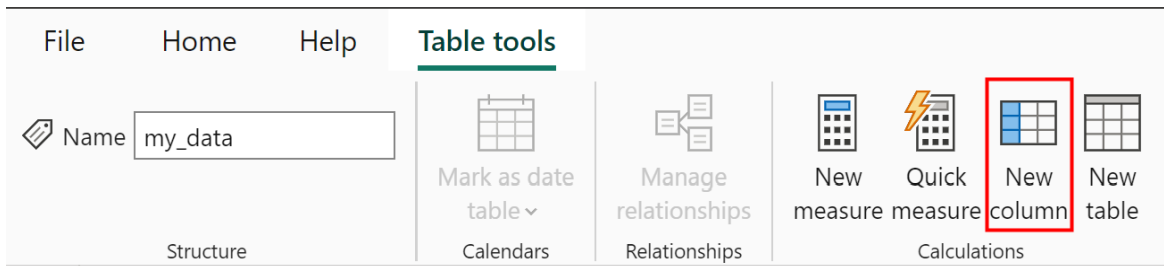
The screenshot shows the Power BI ribbon with the 'Column tools' tab selected. The 'Name' field is set to 'Date' and the 'Data type' is set to 'Text'. The 'Format' dropdown is set to 'Text'. Below the ribbon, a table is displayed with columns 'Date' and 'Sales'. The 'Date' column contains dates from 1/1/2024 to 12/23/2024, and the 'Sales' column contains numerical values from 10 to 30.

Date	Sales
1/1/2024	10
1/5/2024	15
2/15/2024	22
3/17/2024	29
4/15/2024	14
5/6/2024	18
8/12/2024	30
9/15/2024	24
10/11/2024	12
12/23/2024	15

Method 2: Advanced Conversion Using DAX Functions

While the UI method is expedient, it lacks control over the output format and permanently alters the source column's Data type. For scenarios requiring high precision, customized string output, or preservation of the original date column, employing DAX to create a new calculated column is the superior professional practice. DAX (Data Analysis Expressions) provides specific functions designed for type casting and formatting, ensuring the conversion is dynamic and tailored to reporting needs.

The process for this method begins with creating a new column in the existing table. This is achieved by navigating to the **Table tools** tab within the Power BI Desktop ribbon and clicking the **New column** icon. This action opens the formula bar, allowing us to input the desired DAX expression.



The most basic DAX function for explicit type conversion is CONVERT. This function takes an expression and converts the result to a specified Data type. When converting dates to text, we specify the target type as `STRING`. The use of CONVERT is typically used when the exact formatting does not matter, and the default system format (usually mm/dd/yyyy) is acceptable for the resulting string.

DAX Function Deep Dive: Utilizing CONVERT for Simple String Output

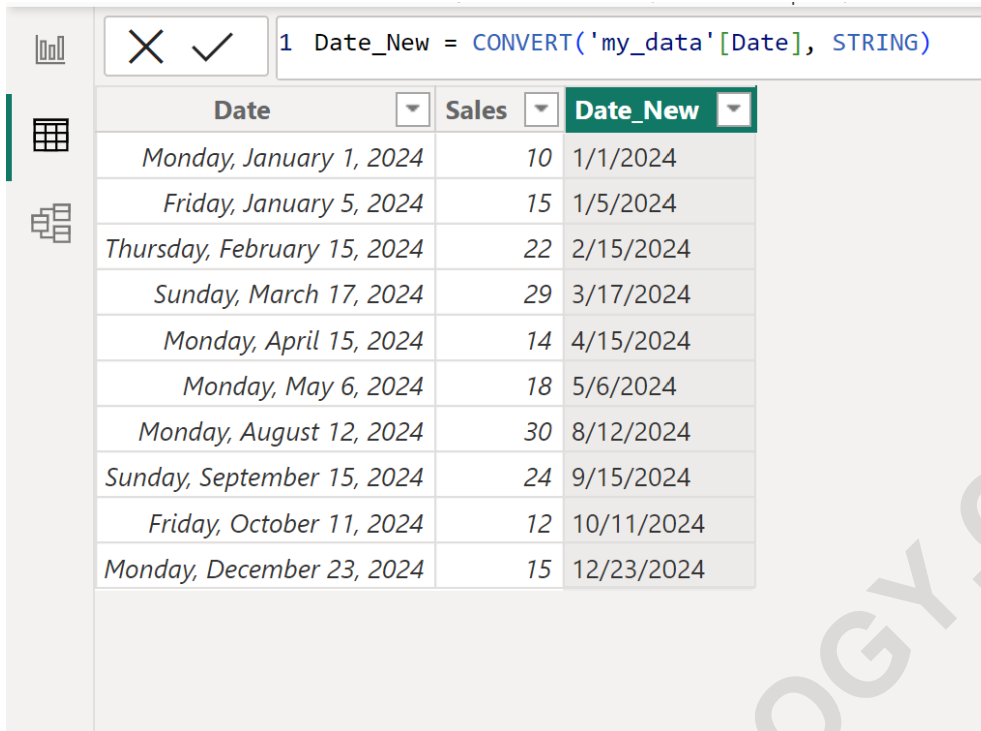
The CONVERT function is highly efficient for quick type casting. Its syntax is straightforward: `CONVERT(expression, dataType)`. When applied to our example, where we want to turn the 'Date' column into a new text column named 'Date_New', the formula is constructed as follows, making sure to specify **STRING** as the target type constant:

Then type the following formula into the formula bar:

Date_New = CONVERT('my_data', STRING)

When this formula is executed, Power BI processes the conversion row by row. This will create a new calculated column named **Date_New** that retains the date values from the original **Date** column but now represents them as text values in the system's default short date format. It is crucial to remember that this default formatting cannot be overridden when using CONVERT; it relies entirely on the model's locale settings.

The resulting table structure visually demonstrates the creation of the new column, showing the date values now behaving like text strings:



The screenshot shows the Power BI interface with a DAX formula bar at the top containing the formula: `1 Date_New = CONVERT('my_data'[Date], STRING)`. Below the formula bar is a table with three columns: Date, Sales, and Date_New. The Date column contains long-form dates, the Sales column contains numerical values, and the Date_New column contains dates in MM/DD/YYYY format.

Date	Sales	Date_New
Monday, January 1, 2024	10	1/1/2024
Friday, January 5, 2024	15	1/5/2024
Thursday, February 15, 2024	22	2/15/2024
Sunday, March 17, 2024	29	3/17/2024
Monday, April 15, 2024	14	4/15/2024
Monday, May 6, 2024	18	5/6/2024
Monday, August 12, 2024	30	8/12/2024
Sunday, September 15, 2024	24	9/15/2024
Friday, October 11, 2024	12	10/11/2024
Monday, December 23, 2024	15	12/23/2024

While [CONVERT](#) is useful for basic transformations, most expert [DAX](#) developers favor the [FORMAT](#) function when explicit date-to-text conversion is required, due to the superior control over output structure.

Note: You can find the complete documentation for the **CONVERT** function in [DAX](#) on the Microsoft Learn platform.

DAX Best Practice: Achieving Custom Formats with the FORMAT Function

For nearly all professional reporting tasks requiring a date to be presented as text, the [FORMAT](#) function is the industry standard. Unlike [CONVERT](#), which relies on the model's locale for output string representation, [FORMAT](#) allows the user to define the exact string pattern using standard formatting codes. This capability is essential for ensuring consistent reporting across different geographical regions or for matching stringent corporate style guides.

The syntax for the [FORMAT](#) function is `FORMAT(value, format_string)`. The `format_string` parameter can accept dozens of predefined formats (like "Short Date" or "Long Date") or custom strings composed of date and time placeholders (e.g., "YYYY", "MM", "DD", "MMM").

Consider a scenario where the default mm/dd/yyyy format is unacceptable, and the requirement is to display the date as the full month name, followed by the year (e.g., "March, 2024"). Using the [FORMAT](#) function, we can create a calculated column:

Date_Formatted = FORMAT('my_data', "MMMM, YYYY")

In this example, "MMMM" compels the function to output the full name of the month (e.g., "March"), and "YYYY" outputs the four-digit year. This precision is unobtainable with the simple UI change or the CONVERT function, making FORMAT the indispensable tool for tailored date presentation in Power BI reports.

Handling Time Components: Using DATENAME for Specific Extractions

In certain analytical contexts, the goal is not to convert the entire date into a text string, but rather to isolate and extract a specific textual component, such as the name of the day of the week or the name of the month, for use in slicers, visual titles, or conditional logic. For these specific extractions, the DATENAME function is employed.

The DATENAME function is specifically designed to return a string representing a named part of a date/time value. Its functionality is similar to the FORMAT function when using specific format strings, but it is often clearer and more direct for these singular extraction tasks. For example, to get the full name of the month from the 'Date' column, one would use:

Month_Name = DATENAME('my_data', MONTH)

Similarly, to extract the name of the day (e.g., "Monday," "Tuesday"), the function can be configured accordingly. This method ensures that the extracted text is standardized and ready for use in text-based filtering or grouping. Although FORMAT can achieve similar results (e.g., `FORMAT(Date, "DDDD")`), DATENAME often provides better readability in complex DAX calculations where the intent is solely to extract a name.

Summary and Use Case Comparison of Conversion Methods

The decision of which conversion method to use hinges on the desired outcome: permanent alteration of the column Data type, or the creation of a dynamically formatted text column.

The UI dropdown method (Method 1) is best used for quick fixes or when the original date values are no longer needed for time intelligence features. Its main drawback is the lack of formatting control and the destructive nature of the type change. Conversely, the DAX methods (Method 2) preserve the source date column, offering superior flexibility.

We can summarize the ideal use cases for the three primary DAX functions involved in date-to-text conversion:

CONVERT: Ideal for type casting when the default system date format is acceptable, prioritizing

simplicity over custom formatting. Useful in nested functions where a string is required but format is secondary.

FORMAT: The go-to function for achieving any custom output structure, allowing granular control over month, day, and year representation using standard format strings. Essential for localized or highly specified reporting requirements.

DATENAME: Used specifically for extracting text components like the full month name or day name, often for categorical visual elements or advanced sorting logic in visuals.

By mastering both the quick UI conversion and the precise DAX functions--especially the versatile FORMAT function--analysts can confidently manage date representations, ensuring reports are both accurate and aesthetically pleasing, regardless of the complexity of the desired output format.

Related Power BI Tutorials for Further Exploration

To continue building expertise in data manipulation and modeling within the Power BI environment, consider exploring tutorials on related topics such as conditional formatting, complex time intelligence functions, and advanced data type handling techniques:

The following tutorials explain how to perform other common tasks in Power BI: