

DAX Calendar

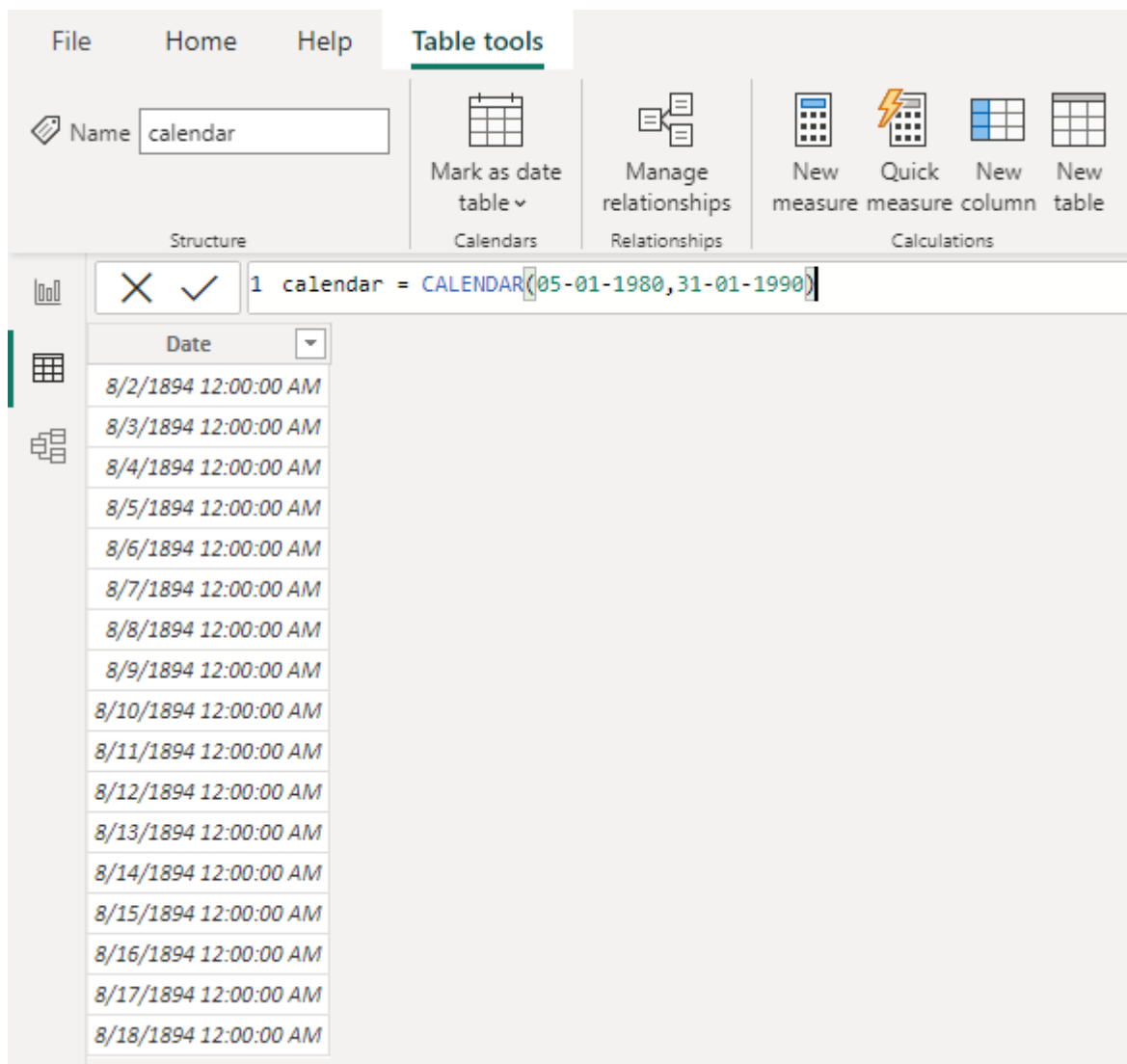
Generates a table with a single “Date” column that is made up of a continuous range of dates. The range of dates is inclusive of those two dates and extends from the specified start date to the specified end date.

Syntax:

`CALENDAR(<start_date>, <end_date>)`

Example:

`calendar = CALENDAR(05-01-1980,31-01-1990)`



The screenshot shows the Microsoft Power BI Desktop interface. The 'Table tools' ribbon is active, with the 'Name' field set to 'calendar'. The formula bar contains the DAX formula: `1 calendar = CALENDAR(05-01-1980,31-01-1990)`. Below the formula bar, a table is displayed with a single column named 'Date'. The table contains 18 rows of dates, starting from 8/2/1894 12:00:00 AM and ending at 8/18/1894 12:00:00 AM. The dates are listed in a continuous sequence, with each row showing a date and time in the format 'MM/DD/YYYY HH:MM:SS AM'.

Date
8/2/1894 12:00:00 AM
8/3/1894 12:00:00 AM
8/4/1894 12:00:00 AM
8/5/1894 12:00:00 AM
8/6/1894 12:00:00 AM
8/7/1894 12:00:00 AM
8/8/1894 12:00:00 AM
8/9/1894 12:00:00 AM
8/10/1894 12:00:00 AM
8/11/1894 12:00:00 AM
8/12/1894 12:00:00 AM
8/13/1894 12:00:00 AM
8/14/1894 12:00:00 AM
8/15/1894 12:00:00 AM
8/16/1894 12:00:00 AM
8/17/1894 12:00:00 AM
8/18/1894 12:00:00 AM

DAX Date

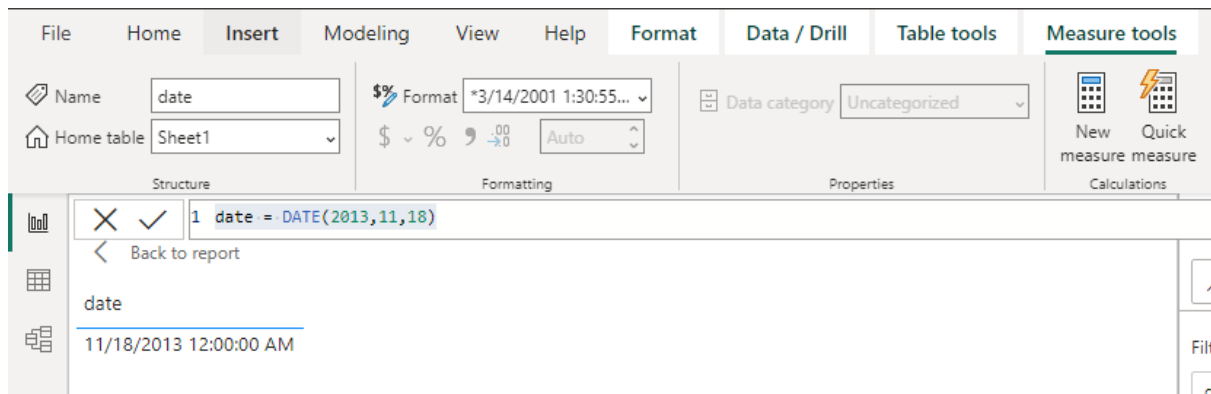
The supplied date is returned in DateTime format by the DATE function. It creates the corresponding date using the integers provided as parameters. When the year, month, and day are determined by formulas, the DATE function is most helpful. The underlying data, for instance, can include dates in a format that isn't accepted as a date, such as YYYYMMDD.

Syntax:

DATE(<year>, <month>, <day>)

Example:

date = DATE(2013,11,18)

**DAX Datevalue**

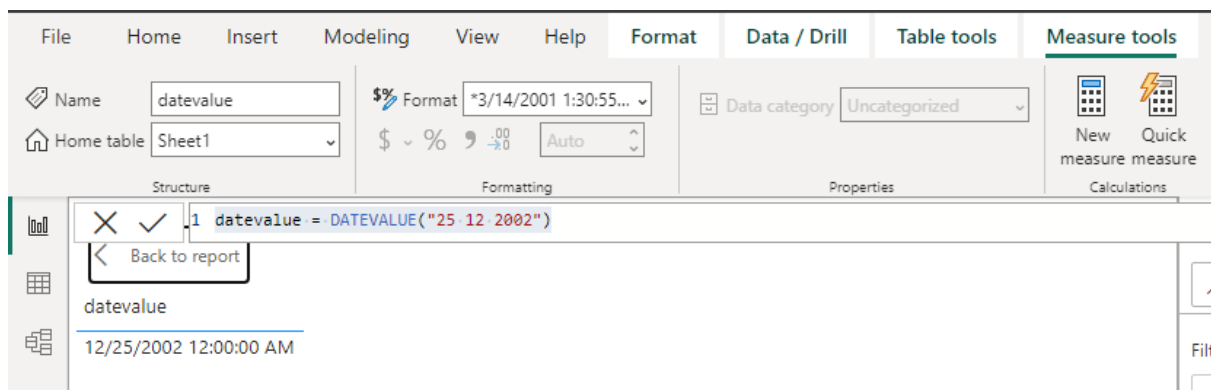
Translates a textual date into a date in DateTime format. The client computer's locale and date/time settings are used by the DATEVALUE function to interpret the text value while executing the conversion.

Syntax:

DATEVALUE(date_text)

Example:

datevalue = DATEVALUE("25 12 2002")

**DAX Datediff**

Returns the number of boundaries between the time frame to compare dates with. Any of the following values may be the value of the Interval:

- Second,

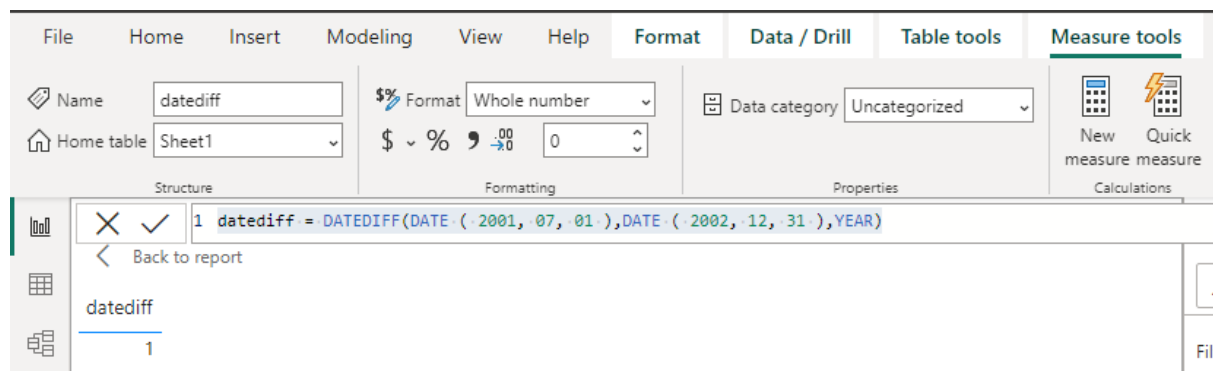
- Minute,
- Hour,
- Day,
- Week,
- Month,
- Quarter,
- Year

Syntax:

DATEDIFF(<Date1>, <Date2>, <Interval>)

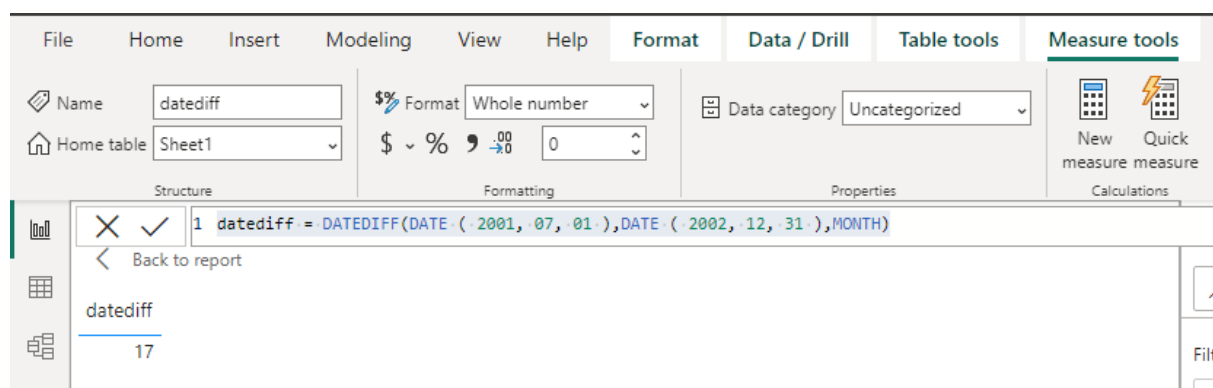
Example:

datediff = DATEDIFF(DATE (2001, 07, 01),DATE (2002, 12, 31),YEAR)



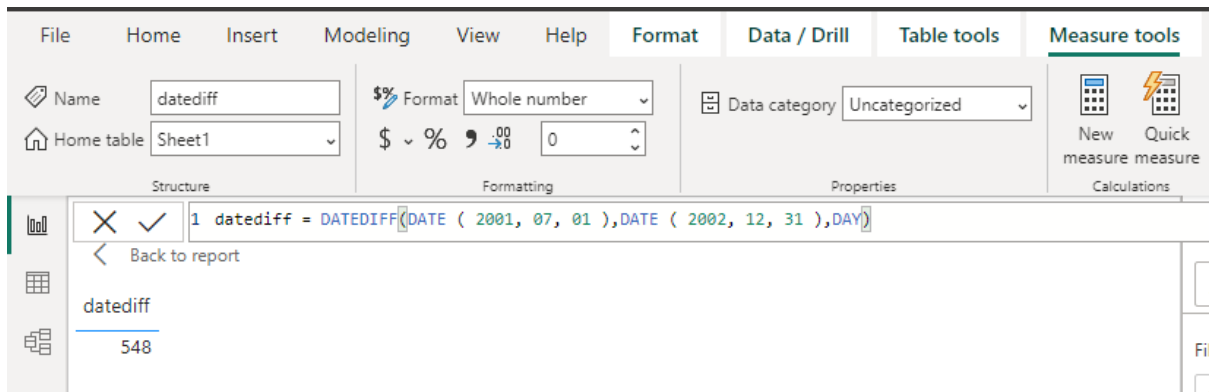
Example:

datediff = DATEDIFF(DATE (2001, 07, 01),DATE (2002, 12, 31),MONTH)



Example:

datediff = DATEDIFF(DATE (2001, 07, 01),DATE (2002, 12, 31),DAY)



DAX Day

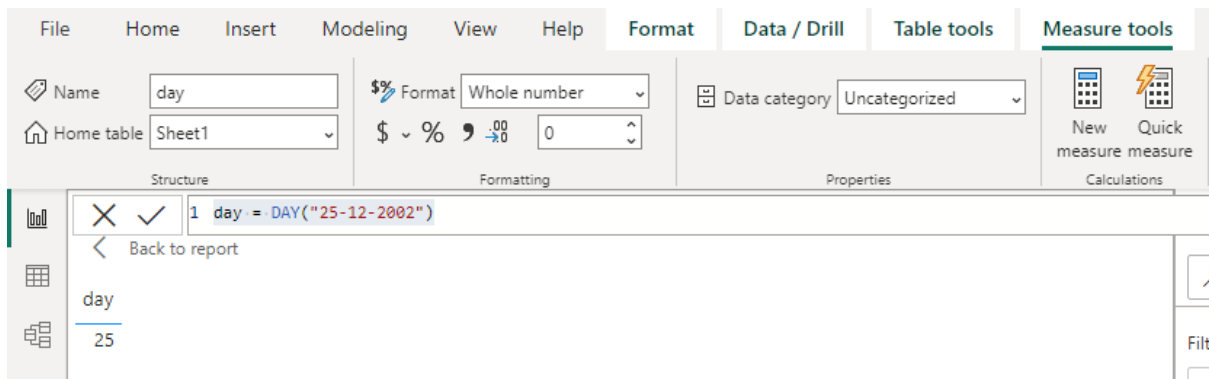
A number between 1 and 31 represents the day of the month. The date of the day you're looking for is an argument for the DAY function. Dates can be passed to the method by inputting them in a DateTime format, using another date function, or using an expression that yields a date. Additionally, you can type a date in one of the acceptable date string formats.

Syntax:

DAY(<date>)

Example:

day = DAY("25-12-2002")



DAX Edate

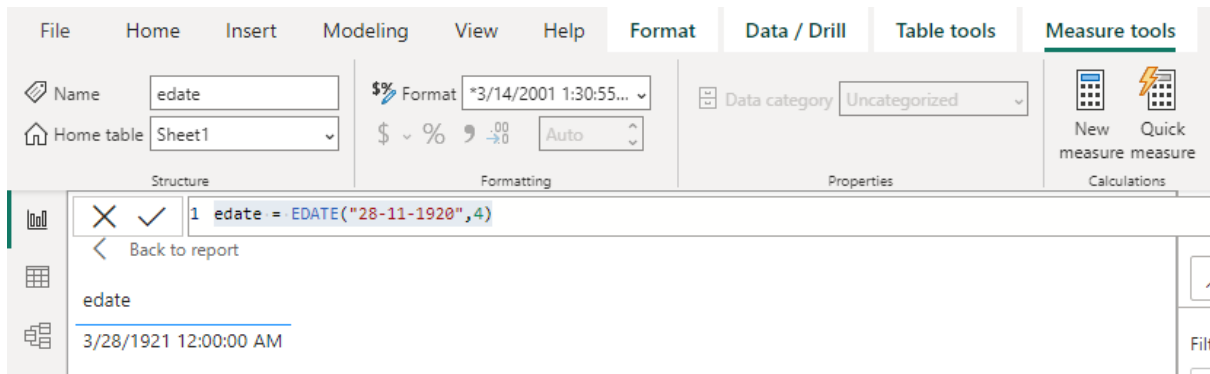
a date that is the specified number of months before or after the start date is returned. To determine maturity dates or due dates that occur on the same day of the month as the date of issue, use the EDATE function.

Syntax:

EDATE(<start_date>, <months>)

Example:

edate = EDATE("28-11-1920",4)



DAX Eomonth

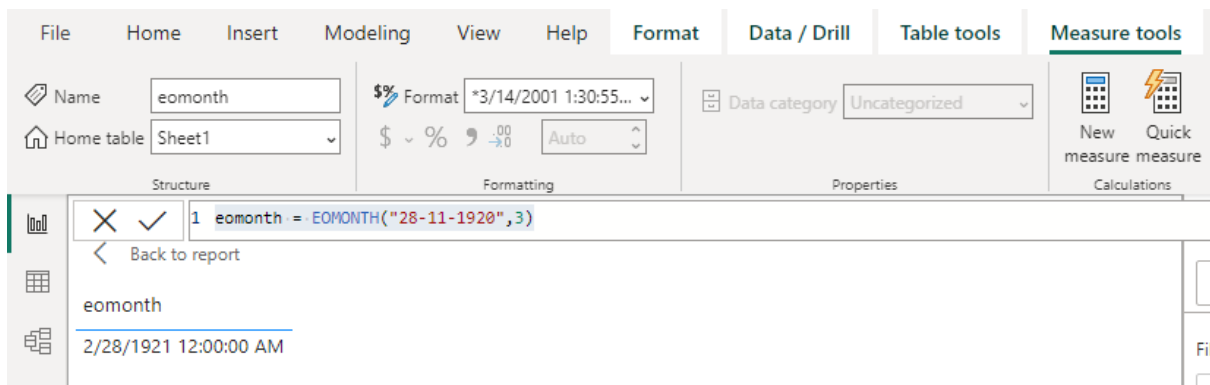
Returns the date of the final day of the month, before or after a specified number of months, in DateTime format. To get maturity dates or due dates that fall on the last day of the month, use the EOMONTH function.

Syntax:

EOMONTH(<start_date>, <months>)

Example:

eomonth = EOMONTH("28-11-1920",3)



DAX Month

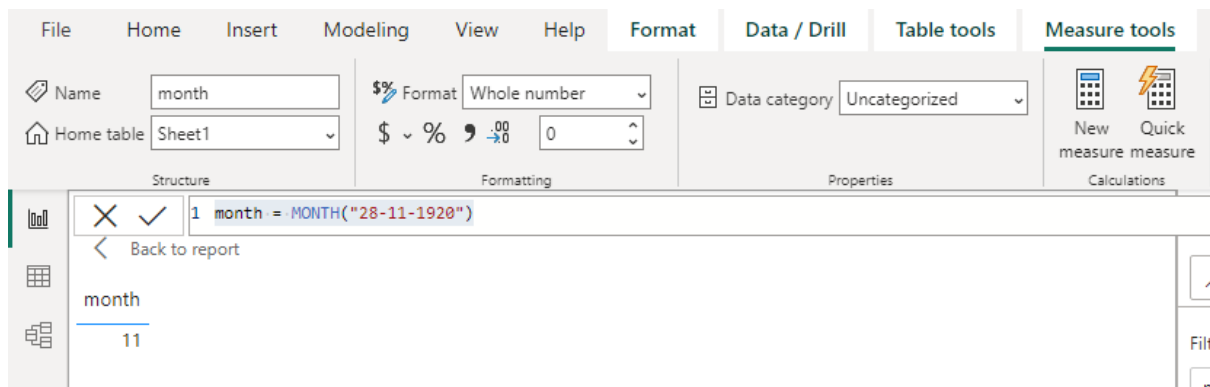
The month is returned as a number ranging from 1 (January) to 12 (December). DAX works with dates in a datetime format as opposed to Microsoft Excel, which maintains dates as serial numbers. You can type a date in a supported datetime format, refer to a column that includes dates, or use an expression that returns a date to enter the date that will be used as the parameter to the MONTH function.

Syntax:

MONTH(<datetime>)

Example:

month = MONTH("28-11-1920")



DAX Now

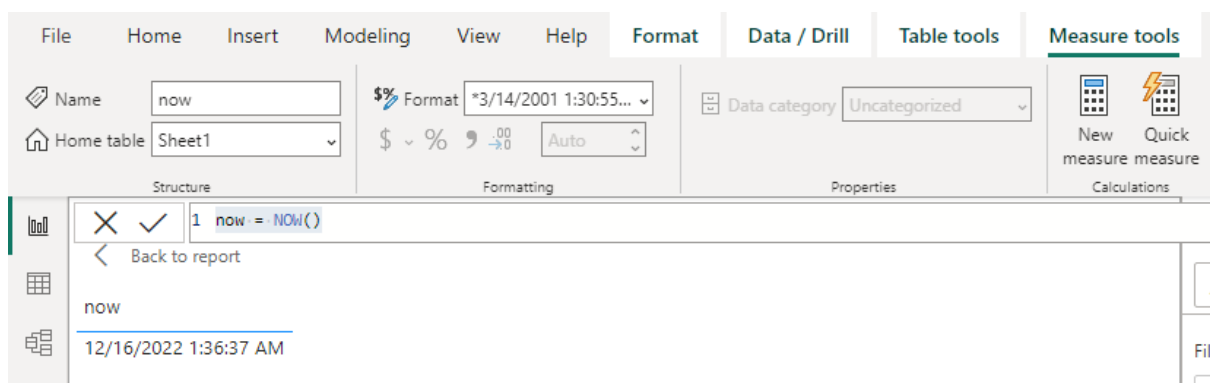
A datetime format string that contains the current date and time. When you need to show the current date and time on a worksheet or compute a number based on the current date and time and have that value updated every time you view the worksheet, the NOW function comes in handy.

Syntax:

NOW()

Example:

now = NOW()



DAX Quarter

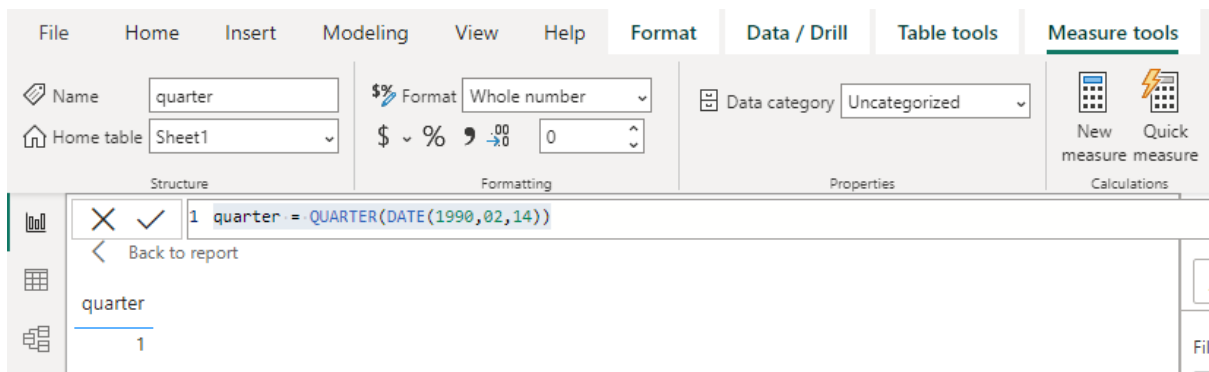
Gives back the quarter as a number between 1 (January through March) and 4. (October – December). The output value is also blank if the input value is blank.

Syntax:

QUARTER(<date>)

Example:

quarter = QUARTER(DATE(1990,02,14))



DAX Today

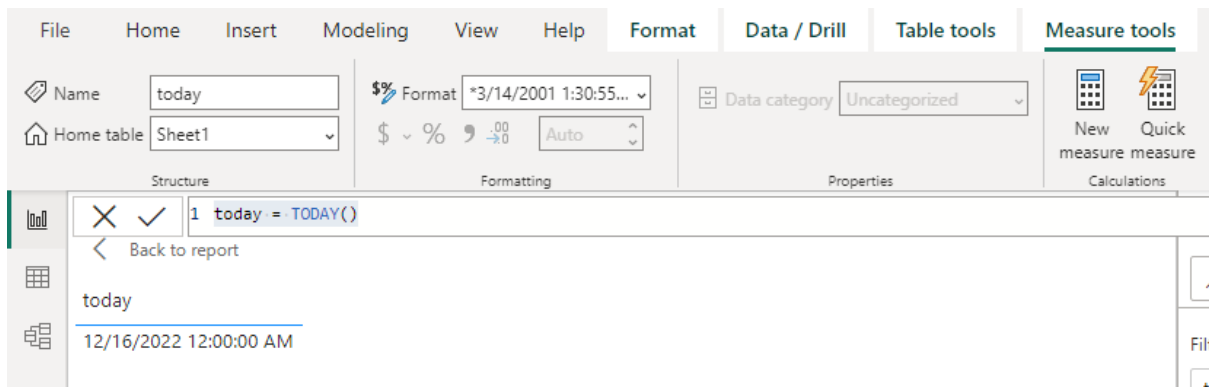
Gives the current date back. The TODAY function comes in handy when you want the current date to be shown on a worksheet regardless of when the workbook is opened. It can be used to compute intervals as well.

Syntax:

`TODAY()`

Example:

`today = TODAY()`



DAX Utcnow

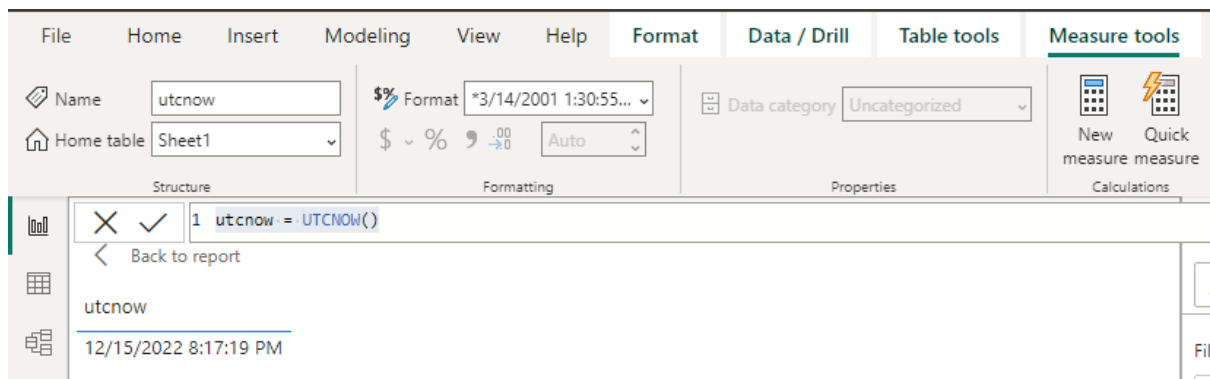
Gives the current date and time in UTC. The UTCNOW function's output only varies when the formula is updated. It isn't always being updated.

Syntax:

`UTCNOW()`

Example:

`utcnow = UTCNOW()`



DAX UtcToday

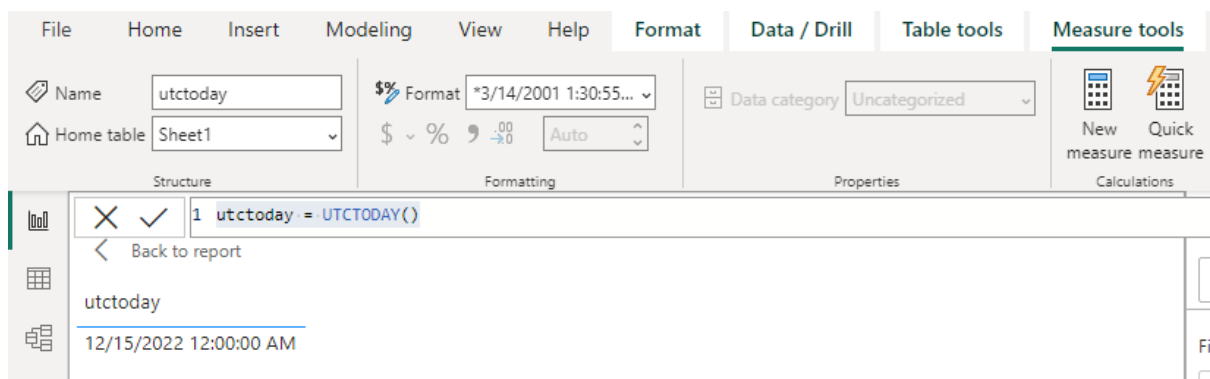
Gives the current date in UTC.

Syntax:

UTCTODAY()

Example:

utctoday = UTCTODAY()



DAX Weekday

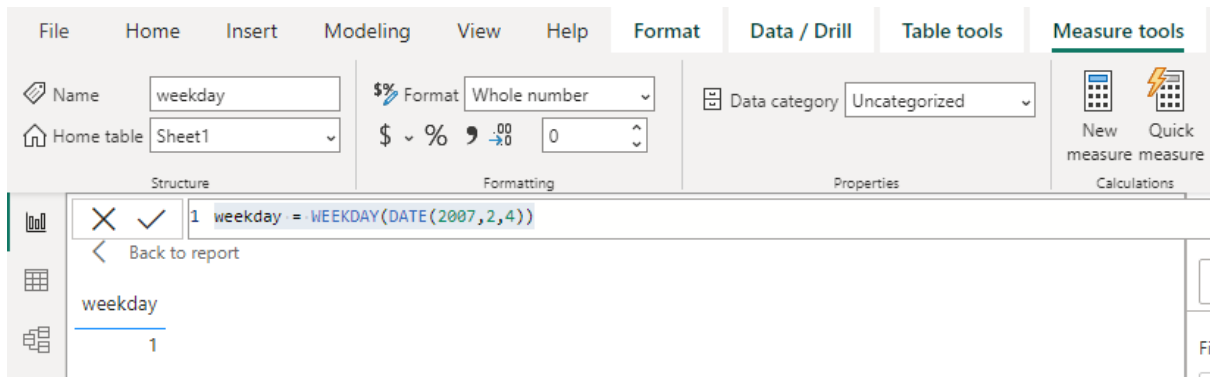
Gives a number between 1 and 7 indicating the date's weekday. The day defaults to being from 1 (Sunday) to 7 (Saturday).

Syntax:

WEEKDAY(<date>, <return_type>)

Example:

weekday = WEEKDAY(DATE(2007,2,4))



DAX Weeknum

According to the return type value, returns the week number for the specified date. The week number shows the week's numerical position within a year. For this purpose, two systems are employed:

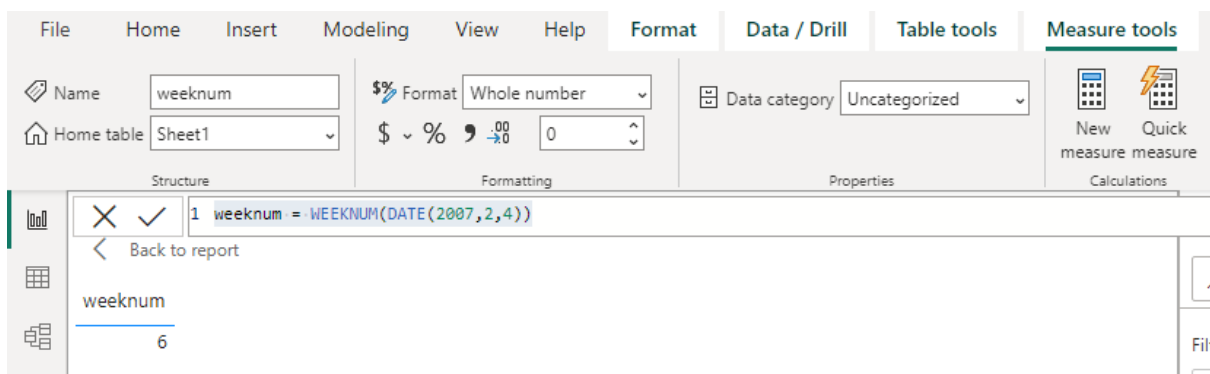
- **System 1** – Week 1 is the year's first week and corresponds to the week beginning January 1.
- **System 2** – The week that includes the first Thursday of the year is week 1 and is the first week of the year. The mechanism used in this system is described in ISO 8601, also referred to as the European week numbering scheme.

Syntax:

WEEKNUM(<date>[, <return_type>])

Example:

weeknum = WEEKNUM(DATE(2007,2,4))



DAX Year

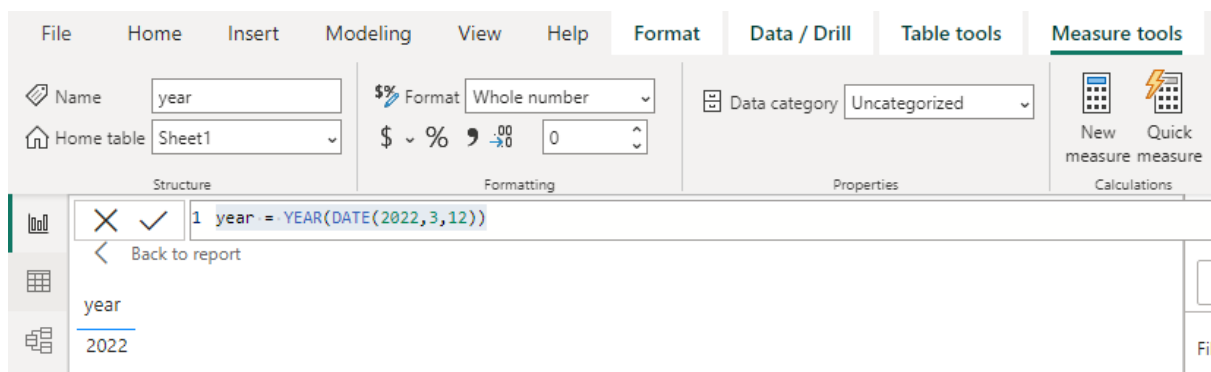
Provides a four-digit integer in the range of 1900 to 9999 as the year of a given date. The date in text or datetime format that includes the desired year.

Syntax:

YEAR(<date>)

Example:

year = YEAR(DATE(2022,3,12))



DAX YearFrac

Determines the fraction of the year that the number of full days between two dates corresponds to. To determine the percentage of benefits or responsibilities for an entire year to be assigned to a certain term, use the YEARFRAC spreadsheet function.

Syntax:

`YEARFRAC(<start_date>, <end_date>, <basis>)`

Example:

yearfrac = YEARFRAC(DATE(2022,3,12),DATE(2022,6,2),2)

