Formatting Calculations in Tableau

*Version: 2019.4 Applies to: Tableau Desktop, Tableau Online, Tableau Public, Tableau Server*

This article describes how to create and format calculations in Tableau. It lists the basic components of calculations, and explains the proper syntax for each.

Calculation building blocks

There are four basic components to calculations in Tableau:

* **Functions** - Statements used to transform the values or members in a field.
* **Fields** - Dimensions or measures (columns) from your data source.
* **Operators** - Symbols that denote an operation.
* **Literal expressions** - Constant values that are represented “as is”, such as "Profitable" and "Unprofitable".

For example, consider the following calculation:

**IF** [Profit per Day] > 000**THEN**"Highly Profitable"  
**ELSEIF**[Profit per Day] <= 0**THEN**"Unprofitable"  
**ELSE**"Profitable"  
**END**

The components of this calculation can be broken down into the following:

**Functions**: IF, THEN, ELSEIF, ELSE and END

Field(s): Profit per Day

Operators: > and <=

Literal expressions:

* + String literals: "Highly Profitable", "Unprofitable" and "Profitable"
  + Numeric literals: 2000 and 0

Note that not all calculations need to contain all four components. For example, a calculation might not contain a literal expression, such as the calculation SUM([Sales])/SUM([Cost]), which only contains the function, SUM; the division operator (/); and the fields, Sales and Cost.

Additionally, calculations can contain:

* **Parameters** - Placeholder variables that can be inserted into calculations to replace constant values.
* **Comments** - Notes about a calculation or its parts, not included in the computation of the calculation.

For more information about how to use and format each of these components in a calculation, see the sections below.

At a glance: calculation syntax

|  |  |  |
| --- | --- | --- |
| **Components** | **Syntax** | **Example** |
| **Functions** | See [Tableau Functions (Alphabetical)](https://help.tableau.com/current/pro/desktop/en-gb/functions_all_alphabetical.htm) for examples of how to format all functions in Tableau. | SUM (expression) |
| **Fields** | A field in a calculation is often surrounded by brackets [ ]. See [Field syntax](https://help.tableau.com/current/pro/desktop/en-gb/functions_operators.htm#Fields) for more information. | [Sales] |
| **Operators** | +, -, \*, /, %, ==, =, >, <, >=, <=, !=, <>, ^, AND, OR, NOT, ( ). See [Operator syntax](https://help.tableau.com/current/pro/desktop/en-gb/functions_operators.htm#Operators) for information on the types of operators you can use in Tableau calculations, as well as the order they are performed in a formula. | [Profit] - [Sales] |
| **Literal expressions** | Numeric literals are written as numbers.  String literals are written with quotation marks.  Date literals are written with the # symbol.  Boolean literals are written as either true or false.  Null literals are written as null.  See [Literal expression syntax](https://help.tableau.com/current/pro/desktop/en-gb/functions_operators.htm#literals) for more information. | 1.3567 or 27  "Unprofitable"  #August 22, 2005#  true or false  Null |
| **Parameters** | A parameter in a calculation is surrounded by brackets [ ]. See [Create Parameters](https://help.tableau.com/current/pro/desktop/en-gb/parameters_create.htm#Parameters) for more information. | [Profit Bin Size] |
| **Comments** | To enter a comment in a calculation, type two forward slashes // . See [Add comments to a calculation](https://help.tableau.com/current/pro/desktop/en-gb/functions_operators.htm#Comments) for more information. | SUM([Sales])/SUM([Profit]) //John's calculation  //To be used for profit ratio  //Do not edit |

Calculation syntax in detail

See the following sections to learn more about the different components of Tableau calculations and how to format them to work in Tableau.

Function syntax

In Tableau, functions are the main components of a calculation and can be used for a variety of different purposes.

Functions are coloured blue in Tableau calculations, with the exception of logical functions, which are coloured black.

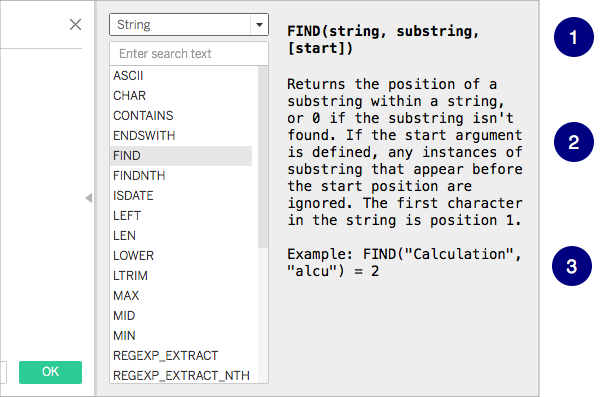
Every function in Tableau requires a particular syntax. For instance, the SUM function requires the following syntax: SUM(expression) (i.e. SUM([Sales]) ). The PERCENTILE function, on the other hand, requires the following syntax: PERCENTILE(expression, number) (i.e. PERCENTILE([Sales],0.90) ).

At any time in Tableau, you can look up how to use and format a particular function.

To open the list of functions in Tableau:

1. Select **Analysis**> **Create Calculated Field**.
2. In the Calculation Editor that opens, click the triangle icon (located on the right edge of the editor).

A list of functions appears for you to browse. When you select a function from the list, the section on the far right updates with information about that function's required syntax (1) and description (2), as well as with one or more examples (3).



Using multiple functions in a calculation

You can use more than one function in a calculation. For example:

ZN(SUM([Order Quantity])) - WINDOW\_AVG(SUM([Order Quantity]))

There are three functions in the calculation: ZN, SUM and WINDOW\_AVG.

The ZN function and the WINDOW\_AVG function are separated with the subtraction operator (-).

A function can also be part of another function (or, nested), as is the case with the ZN(SUM([Order Quantity])) portion of the example above. In this case, the SUM of Order Quantity is computed before the ZN function because it is inside parentheses. For more information on why, see [Parentheses](https://help.tableau.com/current/pro/desktop/en-gb/functions_operators.htm#Parentheses).

Field syntax

Fields can be inserted into your calculations. Often, a function's syntax will indicate where a field should be inserted into the calculation. For example: SUM(expression).

Field names should be encompassed by brackets [ ] in a calculation when the field name contains a space or is not unique. For example, [Sales Categories].

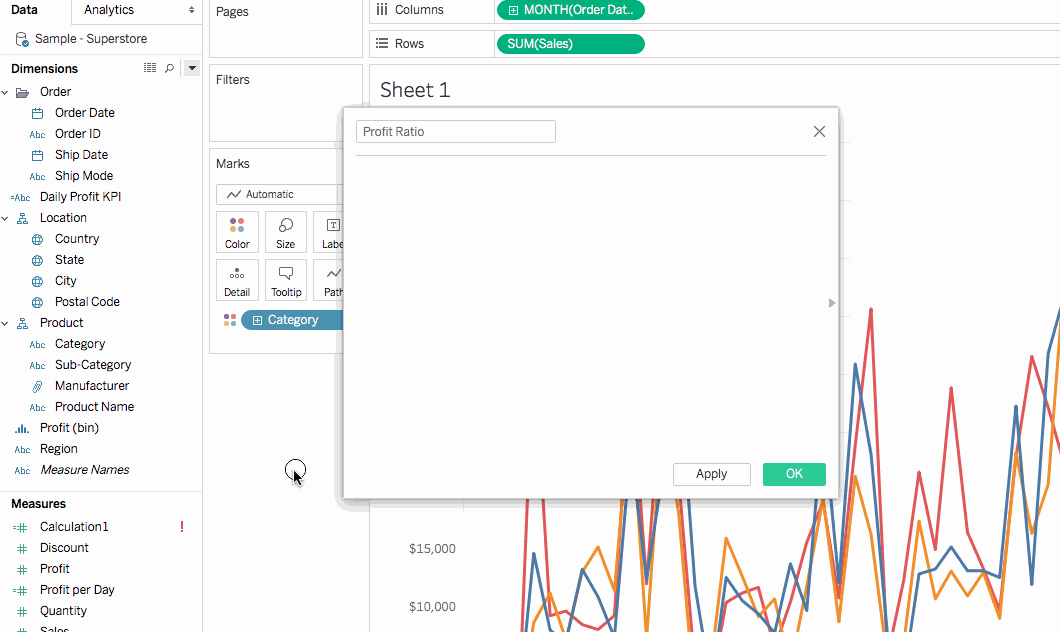
The type of function you use will determine the type of field you use. For example, with the SUM function, you can insert a numerical field, but you cannot insert a date field. For more information, see [Understanding data types in calculations](https://help.tableau.com/current/pro/desktop/en-gb/functions_operators.htm#Data).

The fields you choose to include in your calculations also depends on the purpose of calculation. For example, if you want to calculate profit ratio your calculation will use the Sales and Profit fields from your data source:

SUM([Sales])/SUM([Profit])

To add a field to a calculation, do one of the following:

* Drag it from the Data pane or the view and drop it in the calculation editor.
* In the Calculation Editor, type the field name. Note: The Calculation Editor attempts to auto-complete field names.



Fields are coloured orange in Tableau calculations.

Operator syntax

To create calculations, you need to understand the operators supported by Tableau. This section discusses the basic operators that are available, as well as the order (precedence) they are performed.

Operators are coloured black in Tableau calculations.

+ (addition)

The + operator means addition when applied to numbers and concatenation when applied to strings. When applied to dates, it can be used to add a number of days to a date. For example:

7 + 3

Profit + Sales

'abc' + 'def' = 'abcdef'

#April 15, 2004# + 15 = #April 30, 2004#

– (subtraction)

The - operator means subtraction when applied to numbers and negation if applied to an expression. When applied to dates, it can be used to subtract a number of days from a date. Hence it can also be used to calculate the difference in days between two dates. For example:

7 - 3

Profit - Sales

-(7+3) = -10

#April 16, 2004# - 15 = #April 1, 2004#

#April 15, 2004# - #April 8, 2004# = 7

\* (multiplication)

The \* operator means numeric multiplication. For example, 5 \* 4 = 20.

/ (division)

The / operator means numeric division. For example, 20 / 4 = 5.

% (modulo)

The % operator returns the remainder of a division operation. For example, 9 % 2 returns 1 because 2 goes into 9 four times with a remainder of 1. Modulo can only operate on integers.

==, =, >, <, >=, <=, !=, <>(comparisons)

These are the basic comparison operators that can be used in expressions. Their meanings are as follows: == or = (equal to), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to), != and <> (not equal to).

Each operator compares two numbers, dates or strings and returns a either TRUE, FALSE or NULL.

^ (power)

This symbol is equivalent to the POWER function. It raises a number to the specified power.

For example:

6^3 = 216

AND

This is a logical operator. An expression or a boolean must appear on either side of it. For example,

IIF(Profit =100 AND Sales =1000, "High", "Low")

If both expressions are TRUE (i.e., not FALSE and not NULL), then the result is TRUE. If either expression is NULL, then the result is NULL. In all other cases, the result is FALSE.

If you create a calculation in which the result of an AND comparison is displayed on a worksheet, Tableau displays TRUE and FALSE. If you would like to change this, use the Format area in the format dialog.

The AND operator employs "short circuit evaluation." This means that if the first expression is evaluated to be TRUE, then the second expression is not evaluated at all. This can be helpful if the second expression results in an error when the first expression is TRUE, because the second expression in this case is never evaluated.

OR

This is a logical operator. An expression or a boolean must appear on either side of it. For example,

IIF(Profit =100 OR Sales =1000, "High", "Low")

If either expression is TRUE, then the result is TRUE. If both expressions are FALSE, then the result is FALSE. If both expressions are NULL, then the result is NULL.

If you create a calculation in which the result of an OR comparison is displayed on a worksheet, Tableau displays TRUE and FALSE. If you would like to change this, use the Format area in the format dialog.

The OR operator employs "short circuit evaluation." This means that if the first expression is evaluated to be TRUE, then the second expression is not evaluated at all. This can be helpful if the second expression results in an error when the first expression is TRUE, because the second expression in this case is never evaluated.

NOT

This is a logical operator. It can be used to negate another boolean or an expression. For example,

IIF(NOT(Sales = Profit),"Not Equal","Equal")

Operator precedence

All operators in a calculation are evaluated in a specific order. For example, 2\*1+2 is equal to 4 and not equal to 6, because multiplication is performed before addition (the \* operator is always evaluated before the + operator).

If two operators have the same precedence (such as addition and subtraction (+ or -) they are evaluated from left to right in the calculation.

Parentheses can be used to change the order of precedence. See the [Parentheses](https://help.tableau.com/current/pro/desktop/en-gb/functions_operators.htm#Parentheses) section for more information.

| PRECEDENCE | OPERATOR |
| --- | --- |
| 1 | – (negate) |
| 2 | ^ (power) |
| 3 | \*, /, % |
| 4 | +, – |
| 5 | ==, >, <, >=, <=, != |
| 6 | NOT |
| 7 | AND |
| 8 | OR |

Parentheses

Parentheses can be used as needed to force an order of precedence. Operators that appear within parentheses are evaluated before those outside of parentheses, starting from the innermost parentheses and moving outward.

For example,(1 + (2\*2+1)\*(3\*6/3) ) = 31 because the operators within the innermost parentheses are performed first. The calculation is calculated in the following order:

1. (2\*2+1) = 5
2. (3\*6/3) = 6
3. (1+ 5\*6) = 31

Literal expression syntax

This section describes the proper syntax for using literal expressions in Tableau calculations. A literal expression signifies a constant value that is represented ‘as is’. When you are using functions you will sometimes want to use literal expressions to represent numbers, strings, dates and more.

For example, you may have a function where your input is a date. Rather than just type ‘1 May 2005, which would be interpreted a string, you would type #1 May 2005#. This is equivalent to using a date function to convert the argument from a string to a date (refer to [Date Functions](https://help.tableau.com/current/pro/desktop/en-gb/functions_functions_date.htm)).

You can use numeric, string, date, boolean and null literals in Tableau calculations. Each type, and how to format them, are described below.

Literal expressions are coloured black and grey in Tableau calculations.

Numeric Literals

A numeric literal is written as a number. For example, to input the number one as a numeric literal, enter 1. If you want to input the number 0.25 as a numeric literal, enter 0.25.

String Literals

A string literal can be written either using single quote or double quote. If your string has a single or double quote within it, simply type the symbol twice. For example, to input the string “cat” as a string literal, enter ‘cat’ or “cat”.

Additionally, if you want to type the string “She’s my friend.” as a string literal, type ‘She’s my friend.’ or “She’s my friend.”

Date Literals

Date literals are signified by the pound symbol (#). To input the date “22 August 2005” as a literal date, enter the ISO formatted date, #2005-08-22#.

Boolean Literals

Boolean literals are written as either true or false. To input “true” as a boolean literal, enter true.

Null Literals

Null literals are written as Null. To input “Null” as a Null literal, enter Null.

Add parameters to a calculation

Parameters are placeholder variables that can be inserted into calculations to replace constant values. When you use a parameter in a calculation, you can then expose a parameter control in a view or dashboard to allow users to dynamically change the value.

For details, see [Use a parameter in a calculation](https://help.tableau.com/current/pro/desktop/en-gb/parameters_create.htm#Calc).

Parameters are coloured purple in Tableau calculations.

**Note**: You cannot create parameters as you edit a view on the web.

Add comments to a calculation

You can add comments to a calculation to make notes about it or its parts. Note that comments are not included in the computation of the calculation.

To add a comment to a calculation, type two forward slash (//) characters.

For example:

SUM([Sales])/SUM([Profit]) //John’s calculation

In this example, //John’s calculation is a comment.

A comment starts at the two forward slashes (//) and goes to the end of the line. To continue with your calculation, you must start a new line.

A multiline comment can be written by starting each line with two forward slashes (//). For example:

SUM([Sales])/SUM([Profit]) //John's calculation  
//To be used for profit ratio  
//Do not edit

Comments are coloured grey in Tableau calculations.

Understanding data types in calculations

Tableau supports string, date/datetime, number and boolean data types. If you create calculated fields, you need to know how to use and combine the different data types in calculations. Many functions that are available to you when you define a calculation only work when they are applied to specific data types.

For example, the DATEPART() function can accept only a date/datetime data type as an argument. You can enter DATEPART('year',#2004-04-15#) and expect a valid result: 2004. You cannot enter DATEPART('year',"Tom Sawyer") and expect a valid result. In fact, this example returns an error because "Tom Sawyer" is a string, not a date/datetime.

**Note**: Although Tableau will attempt to fully validate all calculations, some data type errors cannot be found until the query is run against the database. These issues appear as error dialogs at the time of the query rather than in the calculation dialog box.

The data types supported by Tableau are described below. Refer to [Type Conversion](https://help.tableau.com/current/pro/desktop/en-gb/functions_functions_typeconversion.htm) to learn about converting from one data type to another.

STRING

A sequence of zero or more characters. For example, "Wisconsin", "ID-44400", and "Tom Sawyer" are all strings. Strings are recognised by single or double quotes. The quote character itself can be included in a string by repeating it. For example, ‘O''Hanrahan’.

DATE/DATETIME

A date or a datetime. For example "January 23, 1972" or "January 23, 1972 12:32:00 AM". If you would like a date written in long-hand style to be interpreted as a a date/datetime, place the # sign on either side of it. For instance, “January 23, 1972” is treated as a string data type but #January 23, 1972# is treated as a date/datetime data type.

NUMBER

Numerical values in Tableau can be either *integers* or *floating-point numbers*.

With floating-point numbers, results of some aggregations may not always be exactly as expected. For example, you may find that the SUM function returns a value such as -1.42e-14 for a column of numbers that you know should sum to exactly 0. This happens because the Institute of Electrical and Electronics Engineers (IEEE) 754 floating-point standard requires that numbers be stored in binary format, which means that numbers are sometimes rounded at extremely fine levels of precision. You can eliminate this potential distraction by formatting the number to show fewer decimal places. For more information see, [Tableau Functions (by Category)](https://help.tableau.com/current/pro/desktop/en-gb/functions_all_categories.htm#Number) and select **Numbers**.

Operations that test floating point values for equality can behave unpredictably for the same reason. Such comparisons can occur when using level of detail expressions as dimensions, in categorical filtering, creating ad-hoc groups, creating IN/OUT sets and with data blending.

**Note**: The largest signed 64-bit integer is 9,223,372,036,854,775,807. When connecting to a new data source, any column with data type set to **Number (whole)** can accommodate values up to this limit; for larger values, because **Number (whole)** does not use floating-points, Tableau displays ‘Null’ When the data type is set to **Number (decimal)**, larger values can be accommodated.

BOOLEAN

A field that contains the values TRUE or FALSE. An unknown value arises when the result of a comparison is unknown. For example, the expression 7 > Null yields unknown. Unknown booleans are automatically converted to Null