

MODEIF Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *function_col_ref*
 - *test_expression*
- *Examples*
 - *Example - MODEIF function*

Computes the mode (most frequent value) from all row values in a column, according to their grouping. Input column can be of Integer, Decimal, or Datetime type.

- If a row contains a missing or null value, it is not factored into the calculation. If the entire column contains no values, the function returns a null value.
- If there is a tie in which the most occurrences of a value is shared between values, then the lowest value of the evaluated set is returned.
- When used in a `pivot` transform, the function is computed for each instance of the value specified in the `group` parameter. See *Pivot Transform*.

For a non-conditional version of this function, see *MODE Function*.

For a version of this function computed over a rolling window of rows, see *ROLLINGMODE Function*.

Wrangle vs. SQL: This function is part of Wrangle , a proprietary data transformation language. Wrangle is not SQL. For more information, see *Wrangle Language*.

Basic Usage

```
modeif(count_visits, health_status == 'sick')
```

Output: Returns the mode of the values in the `count_visits` column as long as `health_status` is set to `sick`.

Syntax and Arguments

```
modeif(function_col_ref, test_expression) [group:group_col_ref] [limit:limit_count]
```

Argument	Required?	Data Type	Description
<code>function_col_ref</code>	Y	string	Name of column to which to apply the function
<code>test_expression</code>	Y	string	Expression that is evaluated. Must resolve to <code>true</code> or <code>false</code>

For more information on the `group` and `limit` parameters, see *Pivot Transform*.

For more information on syntax standards, see *Language Documentation Syntax Notes*.

function_col_ref

Name of the column the values of which you want to calculate the function. Column must contain Integer, Decimal, or Datetime values.

NOTE: If the input is in Datetime type, the output is in unixtime format. You can wrap these outputs in the DATEFORMAT function to generate the results in the appropriate Datetime format. See *DATEFORMAT Function*.

- Literal values are not supported as inputs.
- Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference)	myValues

test_expression

This parameter contains the expression to evaluate. This expression must resolve to a Boolean (`true` or `false`) value.

Usage Notes:

Required?	Data Type	Example Value
Yes	String expression that evaluates to <code>true</code> or <code>false</code>	<code>(LastName == 'Mouse' && FirstName == 'Mickey')</code>

Examples

Tip: For additional examples, see *Common Tasks*.

Example - MODEIF function

The following data contains a list of weekly orders for 2017 across two regions (`r01` and `r02`). You are interested in calculating the most common order count for the second half of the year, by region.

Source:

NOTE: For simplicity, only the first few rows are displayed.

Date	Region	OrderCount
1/6/2017	r01	78
1/6/2017	r02	97
1/13/2017	r01	92

1/13/2017	r02	90
1/20/2017	r01	97
1/20/2017	r02	84

Transformation:

To assist, you can first calculate the week number for each row:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	weeknum(Date)
Parameter: New column name	'weekNumber'

Then, you can use the following aggregation to determine the most common order value for each region during the second half of the year:

Transformation Name	Pivot columns
Parameter: Row labels	Region
Parameter: Values	modeif(OrderCount, weekNumber > 26)
Parameter: Max number of columns to create	50

Results:

Region	modeif_OrderCount
r01	85
r02	100