LISTIF Function

Contents:

Returns list of all values in a column for rows that match a specified condition.

> **NOTE:** When added to a transformation, this function is applied to the current sample. If you change your sample or run the job, the computed values for this function are updated. Transformations that change the number of rows in subsequent recipe steps do not affect the values computed for this step.

To perform a simple extraction of values without conditionals, use the LIST function. See *LIST 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

```
listif(hotChocolateLiters, 500, temperature &lt; 0)
```

Output: Returns the values from the `hotChocolateLiters` column when the `temperature` value is less than 0. Maximum number of values is 500.

Syntax and Arguments

```
listif(col_ref, limit, test_expression) [group:group_col_ref] [limit:limit_count]
```

Argument	Required?	Data Type	Description
col_ref	Y	string	Reference to the column you wish to evaluate.
limit_int	N	integer	Maximum number of values to extract into the list array. From `1` to `1000`.
test_expression	Y	string	Expression that is evaluated. Must resolve to `true` or `false`

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

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

col_ref

Name of the column whose values you wish to use in the calculation. Column must be a numeric (Integer or Decimal) type.

Usage Notes:

Required?	Data Type	Example Value
Yes	String that corresponds to the name of the column	`myValues`

limit_int

Non-negative integer that defines the maximum number of values to extract into the list array.

> **NOTE:** If specified, this value must between 1 and 1000, inclusive.

> **NOTE:** Do not use the limiting argument in a `LISTIF` function call on a flat aggregate, in which all values in a column have been inserted into a single cell. In this case, you might be able to use the limit argument if you also specify a `group` parameter. Misuse of the `LISTIF` function can cause the application to crash.

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 `true` or `false`	`(LastName == 'Mouse' && FirstName == 'Mickey')`

Examples

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

Example - ANYIF and LISTIF Functions

This example illustrates you to identify and list all values within a group that meet a specified condition.

Functions:

Item	Description
ANYIF Function	Selects a single non-null value from rows in each group that meet a specific condition.
LISTIF Function	Returns list of all values in a column for rows that match a specified condition.
WEEKDAY Function	Derives the numeric value for the day of the week (1, 2, etc.). Input must be a reference to a column containing Datetime values.

Source:

The following data identifies sales figures by salespeople for a week:

EmployeeId	Date	Sales
S001	1/23/17	25
S002	1/23/17	40
S003	1/23/17	48
S001	1/24/17	81
S002	1/24/17	11
S003	1/24/17	25
S001	1/25/17	9
S002	1/25/17	40
S003	1/25/17	
S001	1/26/17	77
S002	1/26/17	83
S003	1/26/17	
S001	1/27/17	17
S002	1/27/17	71
S003	1/27/17	29
S001	1/28/17	
S002	1/28/17	
S003	1/28/17	14
S001	1/29/17	2
S002	1/29/17	7
S003	1/29/17	99

Transformation:

In this example, you are interested in the high performers. A good day in sales is one in which an individual sells more than 80 units. First, you want to identify the day of week:

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`WEEKDAY(Date)`
Parameter: New column name	`'DayOfWeek'`

Values greater than 5 in `DayOfWeek` are weekend dates. You can use the following to identify if anyone reached this highwater marker during the workweek (non-weekend):

Transformation Name	`Pivot columns`
Parameter: Rows labels	`EmployeeId,Date`
Parameter: Values	`ANYIF(Sales, (Sales > 80 && DayOfWeek < 6))`
Parameter: Max number of columns to create	`1`

Before adding the step to the recipe, you take note of the individuals who reached this mark in the `anyif_Sales` column for special recognition.

Now, you want to find out sales for individuals during the week. You can use the following to filter the data to show only for weekdays:

Transformation Name	`Pivot columns`
Parameter: Rows labels	`EmployeeId,Date`
Parameter: Values	`LISTIF(Sales, 1000, (DayOfWeek < 6))`
Parameter: Max number of columns to create	`1`

To clean up, you might select and replace the following values in the listif_Sales column with empty strings:

```
["
"]
[]
```

Results:

EmployeeId	Date	listif_Sales
S001	1/23/17	25
S002	1/23/17	40
S003	1/23/17	48
S001	1/24/17	81
S002	1/24/17	11
S003	1/24/17	25
S001	1/25/17	40
S002	1/25/17	
S003	1/25/17	66
S001	1/26/17	77
S002	1/26/17	83
S003	1/26/17	
S001	1/27/17	17
S002	1/27/17	71
S003	1/27/17	29
S001	1/28/17	

S002	1/28/17	
S003	1/28/17	
S001	1/29/17	
S002	1/29/17	
S003	1/29/17	