

ANYIF Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *col_ref*
 - *test_expression*
- *Examples*
 - *Example - ANYIF and LISTIF Functions*

Selects a single non-null value from rows in each group that meet a specific 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 counting of non-nulls without conditionals, use the `ANY` function. See *ANY 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

```
anyif(custId, donation = 10000)
```

Output: Returns a single value from `custId` when the `donation` value is greater than 10000.

Syntax and Arguments

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

Argument	Required?	Data Type	Description
<code>col_ref</code>	Y	string	Reference to the column you wish to evaluate.
<code>test_expression</code>	Y	string	Expression that is evaluated. Must resolve to <code>true</code> or <code>false</code>

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

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 - ANYIF and LISTIF Functions

This section provides simple examples for how to use the `ANYIF` and `LISTIF` functions. These functions include the following:

- `ANYIF` - Identifies a single value from a group that meets a specific condition. See *ANYIF Function*.
- `LISTIF` - Lists all values within a group that meet a specified condition. See *LISTIF Function*.

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	