

MINIF Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *col_ref*
 - *test_expression*
- *Examples*
 - *Example - Conditional Calculation Functions*

Generates the minimum value of rows in each group that meet a specific condition.

NOTE: When added to a transform, 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. Transforms that change the number of rows in subsequent recipe steps do not affect the values computed for this step.

To calculate the minimum value of rows without conditionals, use the `MIN` function. See *MIN Function*.

Basic Usage

```
pivot value: MINIF(testScores, testCount >= 3) group:studentId limit:1
```

Output: Generates a two-column table containing the unique values for `studentId` and the minimum of the `testScores` column for that `studentId` value when the `testCount` is greater than or equal to 3. The `limit` parameter defines the maximum number of output columns.

Syntax and Arguments

```
pivot value:MINIF(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	<code>myValues</code>

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 - Conditional Calculation Functions

This example illustrates how you can use the following conditional calculation functions to analyze weather data:

- **AVERAGEIF** - Average of a set of values by group that meet a specified condition. See *AVERAGEIF Function*.
- **MINIF** - Minimum of a set of values by group that meet a specified condition. See *MINIF Function*.
- **MAXIF** - Maximum of a set of values by group that meet a specified condition. See *MAXIF Function*.
- **VARIF** - Variance of a set of values by group that meet a specified condition. See *VARIF Function*.
- **STDEVIF** - Standard deviation of a set of values by group that meet a specified condition. See *STDEVIF Function*.

Source:

Here is some example weather data:

date	city	rain_cm	temp_C	wind_mph
1/23/17	Valleyville	0.00	12.8	6.7
1/23/17	Center Town	0.31	9.4	5.3
1/23/17	Magic Mountain	0.00	0.0	7.3
1/24/17	Valleyville	0.25	17.2	3.3
1/24/17	Center Town	0.54	1.1	7.6
1/24/17	Magic Mountain	0.32	5.0	8.8
1/25/17	Valleyville	0.02	3.3	6.8
1/25/17	Center Town	0.83	3.3	5.1
1/25/17	Magic Mountain	0.59	-1.7	6.4
1/26/17	Valleyville	1.08	15.0	4.2
1/26/17	Center Town	0.96	6.1	7.6
1/26/17	Magic Mountain	0.77	-3.9	3.0
1/27/17	Valleyville	1.00	7.2	2.8
1/27/17	Center Town	1.32	20.0	0.2

1/27/17	Magic Mountain	0.77	5.6	5.2
1/28/17	Valleyville	0.12	-6.1	5.1
1/28/17	Center Town	0.14	5.0	4.9
1/28/17	Magic Mountain	1.50	1.1	0.4
1/29/17	Valleyville	0.36	13.3	7.3
1/29/17	Center Town	0.75	6.1	9.0
1/29/17	Magic Mountain	0.60	3.3	6.0

Transform:

The following computes average temperature for rainy days by city:

```
derive type:single value:AVERAGEIF(temp_C, rain_cm > 0) group:city as:'avgTempWRain'
```

The following computes maximum wind for sub-zero days by city:

```
derive type:single value:MAXIF(wind_mph,temp_C < 0) group:city as:'maxWindSubZero'
```

This step calculates the minimum temp when the wind is less than 5 mph by city:

```
derive type:single value:MINIF(temp_C,wind_mph<5) group:city as:'minTempWind5'
```

This step computes the variance in temperature for rainy days by city:

```
derive type:single value:VARIF(temp_C,rain_cm >0) group:city as:'varTempWRain'
```

The following computes the standard deviation in rainfall for Center Town:

```
derive type:single value:STDEVIF(rain_cm,city=='Center Town') as:'stDevRainCenterTown'
```

You can use the following transforms to format the generated output. Note the \$col placeholder value for the multi-column transforms:

```
set col:stDevRainCenterTown,maxWindSubZero value:numformat($col,'##.##')
```

Since the following rely on data that has only one significant digit, you should format them differently:

```
set col:varTempWRain,avgTempWRain,minTempWind5 value:numformat($col,'##.#')
```

Results:

Here is some example weather data:

date	city	rain_cm	temp_C	wind_mph	avgTempWRain	maxWindSubZero	minTempWind5	varTempWRain
1/23/17	Valleyville	0.00	12.8	6.7	8.3	5.1	7.2	63.8
1/23/17	Center Town	0.31	9.4	5.3	7.3		5	32.6

1/23 /17	Magic Mount ain	0.00	0.0	7.3	1.6	6.43	-3.9	12
1/24 /17	Valley ville	0.25	17.2	3.3	8.3	5.1	7.2	63.8
1/24 /17	Cente r Town	0.54	1.1	7.6	7.3		5	32.6
1/24 /17	Magic Mount ain	0.32	5.0	8.8	1.6	6.43	-3.9	12
1/25 /17	Valley ville	0.02	3.3	6.8	8.3	5.1	7.2	63.8
1/25 /17	Cente r Town	0.83	3.3	5.1	7.3		5	32.6
1/25 /17	Magic Mount ain	0.59	-1.7	6.4	1.6	6.43	-3.9	12
1/26 /17	Valley ville	1.08	15.0	4.2	8.3	5.1	7.2	63.8
1/26 /17	Cente r Town	0.96	6.1	7.6	7.3		5	32.6
1/26 /17	Magic Mount ain	0.77	-3.9	3.0	1.6	6.43	-3.9	12
1/27 /17	Valley ville	1.00	7.2	2.8	8.3	5.1	7.2	63.8
1/27 /17	Cente r Town	1.32	20.0	0.2	7.3		5	32.6
1/27 /17	Magic Mount ain	0.77	5.6	5.2	1.6	6.43	-3.9	12
1/28 /17	Valley ville	0.12	-6.1	5.1	8.3	5.1	7.2	63.8
1/28 /17	Cente r Town	0.14	5.0	4.9	7.3		5	32.6
1/28 /17	Magic Mount ain	1.50	1.1	0.4	1.6	6.43	-3.9	12
1/29 /17	Valley ville	0.36	13.3	7.3	8.3	5.1	7.2	63.8
1/29 /17	Cente r Town	0.75	6.1	9.0	7.3		5	32.6
1/29 /17	Magic Mount ain	0.60	3.3	6.0	1.6	6.43	-3.9	12