

# VARSAPIF Function

## Contents:

- *Basic Usage*
- *Syntax and Arguments*
  - *col\_ref*
  - *test\_expression*
- *Examples*
  - *Example - Conditional Calculation Functions*

Generates the variance of values by group in a column that meet a specific condition using the sample statistical method.

**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.

**NOTE:** This function applies to a sample of the entire population. More information is below.

## Relevant terms:

Term	Description
Population	Population statistical functions are computed from all possible values. See <a href="https://en.wikipedia.org/wiki/Statistical_population">https://en.wikipedia.org/wiki/Statistical_population</a> .
Sample	Sample-based statistical functions are computed from a subset or sample of all values. See <a href="https://en.wikipedia.org/wiki/Sampling_(statistics)">https://en.wikipedia.org/wiki/Sampling_(statistics)</a> .  These function names include SAMP in their name.  <b>NOTE:</b> Statistical sampling has no relationship to the samples taken within the product. When statistical functions are computed during job execution, they are applied across the entire dataset. Sample method calculations are computed at that time.

- This function is calculated across a sample of all values.
- For more information on a population version of this function, see *VARIF 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

```
varsampif(testScores, ((testScores > 0) && (testScores < 90)))
```

**Output:** Returns the variance of the `testScores` column when the `testScores` value is between 0 and 90 using the sample method of calculation.

## Syntax and Arguments

```
varsampif(col_ref, 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.
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

### 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 - Conditional Calculation Functions

This example shows some of the statistical functions that use the sample method of computation.

#### Functions:

Item	Description
STDEVSAMP Function	Computes the standard deviation across column values of Integer or Decimal type using the sample statistical method.
VARSAAMP Function	Computes the variance among all values in a column using the sample statistical method. Input column can be of Integer or Decimal. If no numeric values are detected in the input column, the function returns 0.
STDEVSAMP IF Function	Generates the standard deviation of values by group in a column that meet a specific condition using the sample statistical method.

<b>VARSAPIF Function</b>	Generates the variance of values by group in a column that meet a specific condition using the sample statistical method.
<b>ROUND Function</b>	Rounds input value to the nearest integer. Input can be an Integer, a Decimal, a column reference, or an expression. Optional second argument can be used to specify the number of digits to which to round.

**Source:**

Students took tests on three consecutive Saturdays:

Student	Date	Score
Andrew	11/9/19	81
Bella	11/9/19	84
Christina	11/9/19	79
David	11/9/19	64
Ellen	11/9/19	61
Fred	11/9/19	63
Andrew	11/16/19	73
Bella	11/16/19	88
Christina	11/16/19	78
David	11/16/19	67
Ellen	11/16/19	87
Fred	11/16/19	90
Andrew	11/23/19	76
Bella	11/23/19	93
Christina	11/23/19	81
David	11/23/19	97
Ellen	11/23/19	97
Fred	11/23/19	91

**Transformation:**

You can use the following transformations to calculate standard deviation and variance across all dates using the sample method. Each computation has been rounded to three digits.

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>round(stdevsamp(Score), 3)</code>
<b>Parameter: New column name</b>	'stdevSamp'
<b>Transformation Name</b>	New formula

<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	round(varsamp(Score), 3)
<b>Parameter: New column name</b>	'varSamp'

You can use the following to limit the previous statistical computations to the last two Saturdays of testing:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	round(stdevsampilf(Score, Date != '11\9\2019'), 3)
<b>Parameter: New column name</b>	'stdevSampIf'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	round(varsampilf(Score, Date != '11\9\2019'), 3)
<b>Parameter: New column name</b>	'varSampIf'

## Results:

Student	Date	Score	varSampIf	stdevSampIf	varSamp	stdevSamp
Andrew	11/9/19	81	94.515	9.722	131.673	11.475
Bella	11/9/19	84	94.515	9.722	131.673	11.475
Christina	11/9/19	79	94.515	9.722	131.673	11.475
David	11/9/19	64	94.515	9.722	131.673	11.475
Ellen	11/9/19	61	94.515	9.722	131.673	11.475
Fred	11/9/19	63	94.515	9.722	131.673	11.475
Andrew	11/16/19	73	94.515	9.722	131.673	11.475
Bella	11/16/19	88	94.515	9.722	131.673	11.475
Christina	11/16/19	78	94.515	9.722	131.673	11.475
David	11/16/19	67	94.515	9.722	131.673	11.475
Ellen	11/16/19	87	94.515	9.722	131.673	11.475
Fred	11/16/19	90	94.515	9.722	131.673	11.475
Andrew	11/23/19	76	94.515	9.722	131.673	11.475
Bella	11/23/19	93	94.515	9.722	131.673	11.475
Christina	11/23/19	81	94.515	9.722	131.673	11.475
David	11/23/19	97	94.515	9.722	131.673	11.475
Ellen	11/23/19	97	94.515	9.722	131.673	11.475
Fred	11/23/19	91	94.515	9.722	131.673	11.475

