

# STDEV Function

Computes the standard deviation across all column values of Integer or Decimal type.

The **standard deviation** of a set of values attempts to measure the spread in values around the mean and is used to measure confidence in statistical results. A standard deviation of zero means that all values are the same, and a small standard deviation means that the values are closely bunched together. A high value for standard deviation indicates that the numbers are spread out widely. Standard deviation is always a positive value.

Standard deviation comes in two flavors:

- **Population standard deviation** computes the variance from all possible values.
- **Sample standard deviation** computes from a subset or sample of all values.
- Since Trifacta® Wrangler has access to all available values, the computation for population standard deviation is used across the entire dataset.

If a row contains a missing or null value, it is not factored into the calculation. If no numeric values are detected in the input column, the function returns 0 .

The square of standard deviation is variance. See *VAR Function*.

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

## Basic Usage

```
stdev(myRating)
```

**Output:** Returns the standard deviation of the values from the `myRating` column.

## Syntax

```
stdev(function_col_ref) [group:group_col_ref] [limit:limit_count]
```

Argument	Required?	Data Type	Description
function_col_ref	Y	string	Name of column to which to apply the function

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 variance. Column must contain Integer or Decimal values.

- 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
-----	---------------------------	----------

## Examples

This example illustrates how you can apply statistical functions to your dataset. Calculations include average (mean), max, min, standard deviation, and variance.

### Source:

Students took a test and recorded the following scores. You want to perform some statistical analysis on them:

Student	Score
Anna	84
Ben	71
Caleb	76
Danielle	87
Evan	85
Faith	92
Gabe	85
Hannah	99
Ian	73
Jane	68

### Transformation:

You can use the following transforms to calculate the average (mean), minimum, and maximum scores:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	AVERAGE(Score)
<b>Parameter: New column name</b>	'avgScore'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	MIN(Score)
<b>Parameter: New column name</b>	'minScore'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	MAX(Score)
<b>Parameter: New column name</b>	'maxScore'

To apply statistical functions to your data, you can use the VAR and STDEV functions, which can be used as the basis for other statistical calculations.

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	VAR(Score)

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	STDEV(Score)

For each score, you can now calculate the variation of each one from the average, using the following:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	((Score - avg_Score) / stdev_Score)
<b>Parameter: New column name</b>	'stDevs'

Now, you want to apply grades based on a formula:

Grade	standard deviations from avg (stDevs)
A	stDevs > 1
B	stDevs > 0.5
C	-1 <= stDevs <= 0.5
D	stDevs < -1
F	stDevs < -2

You can build the following transform using the IF function to calculate grades.

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	IF((stDevs > 1), 'A', IF((stDevs < -2), 'F', IF((stDevs < -1), 'D', IF((stDevs > 0.5), 'B', 'C'))))

For more information, see *IF Function*.

To clean up the content, you might want to apply some formatting to the score columns. The following reformats the stdev\_Score and stDevs columns to display two decimal places:

<b>Transformation Name</b>	Edit column with formula
<b>Parameter: Columns</b>	stdev_Score
<b>Parameter: Formula</b>	NUMFORMAT(stdev_Score, '##.00')

<b>Transformation Name</b>	Edit column with formula
<b>Parameter: Columns</b>	stDevs
<b>Parameter: Formula</b>	NUMFORMAT(stDevs, '##.00')

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	MODE(Score)
<b>Parameter: New column name</b>	'modeScore'

**Results:**

Student	Score	modeScore	avgScore	minScore	maxScore	var_Score	stdev_Score	stDevs	Grade
Anna	84	85	82	68	99	87.00000000000001	9.33	0.21	C
Ben	71	85	82	68	99	87.00000000000001	9.33	-1.18	D
Caleb	76	85	82	68	99	87.00000000000001	9.33	-0.64	C
Danielle	87	85	82	68	99	87.00000000000001	9.33	0.54	B
Evan	85	85	82	68	99	87.00000000000001	9.33	0.32	C
Faith	92	85	82	68	99	87.00000000000001	9.33	1.07	A
Gabe	85	85	82	68	99	87.00000000000001	9.33	0.32	C
Hannah	99	85	82	68	99	87.00000000000001	9.33	1.82	A
Ian	73	85	82	68	99	87.00000000000001	9.33	-0.96	C
Jane	68	85	82	68	99	87.00000000000001	9.33	-1.50	D