

# VAR Function

Computes the variance among all values in a column. Input column can be of Integer or Decimal. If no numeric values are detected in the input column, the function returns 0.

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

$$\text{Var}(X) = [\text{Sum} ((X - \text{mean}(X))^2)] / \text{Count}(X)$$

If a row contains a missing or null value, it is not factored into the calculation.

## 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> .<br><br>These function names include SAMP in their name.<br><br><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 the entire population.
- For more information on a sampled version of this function, see *VAR\_SAMP Function*.

The square root of variance is standard deviation, which is used to measure variance under the assumption of a bell curve distribution. See *STDEV Function*.

For a version of this function computed over a rolling window of rows, see *ROLLINGVAR 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

```
var(myRating)
```

**Output:** Returns the variance of the group of values from the `myRating` column.

## Syntax and Arguments

```
var(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

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

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 transformations to calculate the average (mean), minimum, and maximum scores:

| Transformation Name | 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>Parameter: New column name</b> | var_Score          |

|                                   |                    |
|-----------------------------------|--------------------|
| <b>Transformation Name</b>        | New formula        |
| <b>Parameter: Formula type</b>    | Single row formula |
| <b>Parameter: Formula</b>         | STDEV(Score)       |
| <b>Parameter: New column name</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>         | $((\text{Score} - \text{avg\_Score}) / \text{stdev\_Score})$ |
| <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 transformation 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>      | <code>IF((stDevs &gt; 1), 'A', IF((stDevs &lt; -2), 'F', IF((stDevs &lt; -1), 'D', IF((stDevs &gt; 0.5), 'B', 'C'))))</code> |

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>  | <code>stdev_Score</code>                     |
| <b>Parameter: Formula</b>  | <code>NUMFORMAT(stdev_Score, '##.00')</code> |

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

|                                   |                          |
|-----------------------------------|--------------------------|
| <b>Transformation Name</b>        | New formula              |
| <b>Parameter: Formula type</b>    | Single row formula       |
| <b>Parameter: Formula</b>         | <code>MODE(Score)</code> |
| <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       | 87.000000000000001 | 9.33        | 0.21   | C     |
| Ben     | 71    | 85        | 82       | 68       | 99       | 87.000000000000001 | 9.33        | -1.18  | D     |
| Caleb   | 76    | 85        | 82       | 68       | 99       | 87.000000000000001 | 9.33        | -0.64  | C     |

|          |    |    |    |    |    |                       |      |       |   |
|----------|----|----|----|----|----|-----------------------|------|-------|---|
| Danielle | 87 | 85 | 82 | 68 | 99 | 87.000000000<br>00001 | 9.33 | 0.54  | B |
| Evan     | 85 | 85 | 82 | 68 | 99 | 87.000000000<br>00001 | 9.33 | 0.32  | C |
| Faith    | 92 | 85 | 82 | 68 | 99 | 87.000000000<br>00001 | 9.33 | 1.07  | A |
| Gabe     | 85 | 85 | 82 | 68 | 99 | 87.000000000<br>00001 | 9.33 | 0.32  | C |
| Hannah   | 99 | 85 | 82 | 68 | 99 | 87.000000000<br>00001 | 9.33 | 1.82  | A |
| Ian      | 73 | 85 | 82 | 68 | 99 | 87.000000000<br>00001 | 9.33 | -0.96 | C |
| Jane     | 68 | 85 | 82 | 68 | 99 | 87.000000000<br>00001 | 9.33 | -1.50 | D |