Aggregate Functions

Aggregate functions perform a computation against a set of values to generate a single result. For example, you could use an aggregate function to compute the average (mean) order over a period of time. Aggregations can be applied as standard functions or used as part of a transformation step to reshape the data.

Aggregate across an entire column:

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	average(Scores)

Output: Generates a new column containing the average of all values in the `Scores` column.

Transformation Name	Pivot columns
Parameter: Values	average(Score)
Parameter: Max number of columns to create	1

Output: Generates a single-column table with a single value, which contains the average of all values in the `Scores` column. The limit defines the maximum number of columns that can be generated.

> **NOTE:** When aggregate functions are applied as part of a pivot transformation, they typically involve multiple parameters as part of an operation to reshape the dataset. See below.

Aggregate across groups of values within a column:

Aggregate functions can be used with the pivot transformation to change the structure of your data. Example:

Transformation Name	Pivot columns
Parameter: Row labels	StudentId
Parameter: Values	average(Score)
Parameter: Max number of columns to create	1

In the above instance, the resulting dataset contains two columns:

- `studentId` - one row for each distinct student ID value
- `average_Scores` - average score by each student (`studentId`)

> **NOTE:** You cannot use aggregate functions inside of conditionals that evaluate to `true` or `false`.

A pivot transformation can include multiple aggregate functions and group columns from the pre-aggregate dataset.

For more information on the transformation, see *Pivot Data*.

> **NOTE:** Null values are ignored as inputs to these functions.

These aggregate functions are available: