

# COUNTDISTINCT Function

Generates the count of distinct values in a specified column, optionally counted by group. Generated value is of Integer type.

**NOTE:** Empty string values are counted. Null values are not counted.

**NOTE:** When added to a transform, the function calculates the number of distinct values in the specified column, as displayed in the current sample. Counts are not applied to the entire dataset until you run the job. 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 value for the already computed instance of COUNTDISTINCT.

## Basic Usage

```
pivot value: COUNTDISTINCT(name) group:postal_code limit:1
```

**Output:** Generates a two-column table containing the unique values for `postal_code` and the count of distinct values in the `name` column for that `postal_code` value. The `limit` parameter defines the maximum number of output columns.

## Syntax and Arguments

```
pivot value:COUNTDISTINCT(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` parameter, see *Pivot Transform*.

### function\_col\_ref

Name of the column from which to count values based on the grouping.

- 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*.

## Example - Simple row count

This section provides simple examples for how to use the `COUNTA` and `COUNTDISTINCT` functions. These functions include the following:

- `COUNTA` - Count the number of values within a group that meet a specific condition. See *COUNTA Function*.
- `COUNTDISTINCT` - Count the number of non-null values within a group that meet a specific condition. See *COUNTDISTINCT Function*.

### Source:

In the following example, the seventh row is an empty string, and the eighth row is a null value.

rowId	Val
r001	val1
r002	val1
r003	val1
r004	val2
r005	val2
r006	val3
r007	(empty)
r008	(null)

### Transform:

Apply a `COUNTA` function on the source column:

```
derive type:single value:COUNTA(Val) as:'fctnCounta'
```

Apply a `COUNTDISTINCT` function on the source:

```
derive type:single value:COUNTDISTINCT(Val) as:'fctnCountdistinct'
```

### Results:

Below, both functions count the number of values in the column, with `COUNTDISTINCT` counting distinct values only. The empty value for `r007` is counted by both functions.

rowId	Val	fctnCountdistinct	fctnCounta
r001	val1	4	7
r002	val1	4	7
r003	val1	4	7
r004	val2	4	7
r005	val2	4	7
r006	val3	4	7
r007	(empty)	4	7

r008	(null)	4	7
------	--------	---	---