

Transforms

NOTE: Transforms are a part of the underlying language, which is not directly accessible to users. This content is maintained for reference purposes only. For more information on the user-accessible equivalent to transforms, see *Transformation Reference*.

In Wrangle , a **transform** is an action applied to your dataset. Each step of your recipe corresponds to a fully specified transform.

Tip: To see transforms by category, click the sort buttons in the Category header in the online documentation.

Transform	Category	Description
<i>Case Transform</i>	Other	Perform if/then/else or case logic on the rows in your dataset.
<i>Comment Transform</i>	Other	Inserts a non-functional comment as a recipe step.
<i>Countpattern Transform</i>	Search and Replace	Counts the number of instances of a specified pattern in a column and writes that value into a newly generated column. Source column is unchanged.
<i>Deduplicate Transform</i>	Manage Rows	Removes exact duplicate rows from your dataset. Duplicate rows are identified by exact matches between values. For example, two strings with different capitalization do not match.
<i>Delete Transform</i>	Manage Rows	Deletes a set of rows in your dataset, based on a condition specified in the <code>row</code> expression. If the conditional expression is <code>true</code> , then the row is deleted.
<i>Derive Transform</i>	Manage Columns	Generate a new column where the values are the output of the <code>value</code> expression. Expression can be calculated based on values specified in the <code>group</code> parameter. Output column can be named as needed.
<i>Drop Transform</i>	Manage Columns	Removes the specified column or columns permanently from your dataset.
<i>Extract Transform</i>	Search and Replace	Extracts a subset of data from one column and inserts it into a new column, based on a specified string or pattern. The source column is unmodified.
<i>Extractkv Transform</i>	Search and Replace	Extracts key-value pairs from a source column and writes them to a new column. Source column must be of String type, although the data can be formatted as other data types.
<i>Extractlist Transform</i>	Search and Replace	Extracts a set of values based on a specified pattern from a source column of any data type. The generated column contains an array of occurrences of the specified pattern. While the new column contains array data, the data type of the new column is sometimes inferred as String.
<i>Filter Transform</i>	Manage Rows	Keep or delete rows in your dataset based on a defined type of filter.
<i>Flatten Transform</i>	Nested Data	Unpacks array data into separate rows for each value.
<i>Header Transform</i>	Initial Parsing	Uses one row from the dataset sample as the header row for the table. Each value in this row becomes the name of the column in which it is located.
<i>Keep Transform</i>	Manage Rows	Retains a set of rows in your dataset, which are specified by the conditional in the <code>row</code> expression. All other rows are removed from the dataset.
<i>Merge Transform</i>	Manage Columns	Merges two or more columns in your dataset to create a new column of String type. Optionally, you can insert a delimiter between the merged values.
<i>Move Transform</i>	Manage Columns	Moves the specified column or columns before or after another column in your dataset.

<i>Nest Transform</i>	Nested Data	Creates an Object or Array of values using column names and their values as key-value pairs for one or more columns. Generated column type is determined by the <code>into</code> parameter.
<i>Pivot Transform</i>	Nested Data	<p>The <code>pivot</code> transform can be used to aggregate or pivot your data into columns and aggregate the results. Reshape your dataset into summary information.</p> <p>When you aggregate data, calculations are performed on column values, which are then grouped and ordered based on specified parameters.</p> <p>When you pivot data, the values of a selected column become new columns in the dataset, each of which contains a summary calculation that you specify. This calculation can be based on all rows for totals across the dataset or based on group of rows you define in the transform.</p>
<i>Rename Transform</i>	Manage Columns	Renames one or more columns to specified names or append or prepend column names with specific values.
<i>Replace Transform</i>	Search and Replace	Replaces values within the specified column or columns based on the string literal, pattern, or location within the cell value, as specified in the transform.
<i>Set Transform</i>	Search and Replace	Replaces all values in the specified column with the specified value, which can be a literal or an expression. You can specify an optional <code>row :</code> parameter, containing a conditional test to identify the rows where the replacement is to be made within the column.
<i>Settype Transform</i>	Manage Columns	Sets the data type of the specified column. This transform does not modify the source values. The data in the column is re-inferred against the specified data type, which can change the results of column profiling.
<i>Split Transform</i>	Initial Parsing	Splits the specified column into separate columns of data based on the delimiters in the transform. Delimiters can be specified in a number of methods described below.
<i>Splitrows Transform</i>	Initial Parsing	Splits a column of values into separate rows of data based on the specified delimiter. You can split rows only on String literal values. Pattern-based row splitting is not supported.
<i>Unnest Transform</i>	Nested Data	Unpacks nested data from an Array or Object column to create new rows or columns based on the keys in the source data. This transform works differently on columns of Object or Array type.
<i>Unpivot Transform</i>	Nested Data	Reshapes the layout of data by merging one or more columns into key and value columns. Keys are the names of input columns, and the values are the cell values from the source columns. Rows of data are duplicated, once for each input column.
<i>Valuestocols Transform</i>	Manage Columns	For each unique value in a column, a separate column is created. For each row that contains the value in the source column, an indicator value is inserted in the new column. This value can be a literal value or the output of a function. If no indicator value is generated, a null value is written.
<i>Window Transform</i>	Aggregation	The <code>window</code> transform enables you to perform summations and calculations based on a rolling window of data relative to the current row. For example, you can compute the rolling average for a specified column for the current row value and the nine preceding rows. This transform is particularly useful for processing time or otherwise sequential data.