# Easy Data Transform v1.39.1 for Windows

© 2023 Oryx Digital Ltd, all rights reserved

1. Getting started 7

1.1 Introduction ..... 8
1.2 System requirements ..... 8
1.3 Quick start guide ..... 8

2. Reference 21

2.1 User Interface ..... 22
2.1.1 Main window ..... 22
2.1.2 Left pane ..... 23
2.1.3 Center pane ..... 23
2.1.4 Right pane ..... 23
2.1.5 Log pane ..... 23
2.1.6 Preferences window ..... 23
2.2 Input ..... 26
2.2.1 Input data ..... 26
2.3 Transforms ..... 29
2.3.1 Transform data ..... 29
2.3.2 Calculate ..... 29
2.3.3 Case ..... 37
2.3.4 Chop ..... 39
2.3.5 Clone ..... 41
2.3.6 Cluster ..... 41
2.3.7 Compare Cols ..... 45
2.3.8 Concat Cols ..... 48
2.3.9 Concat Rows ..... 50
2.3.10 Copy Cols ..... 56
2.3.11 Count ..... 58
2.3.12 Cross ..... 60
2.3.13 DateTime Format ..... 61
2.3.14 Decode ..... 68
2.3.15 Dedupe ..... 70
2.3.16 Extract ..... 73
2.3.17 Fill ..... 79
2.3.18 Filter ..... 81
2.3.19 Gather ..... 85
2.3.20 Hash ..... 87
2.3.21 Header ..... 90
2.3.22 If ..... 92
2.3.23 Impute ..... 96



*© 2023 Oryx Digital Ltd*

2.3.24 Insert ..... 101
2.3.25 Interpolate ..... 103
2.3.26 Intersect ..... 106
2.3.27 Javascript ..... 107
2.3.28 Join ..... 111
2.3.29 Lookup ..... 115
2.3.30 Moving ..... 118
2.3.31 New Col ..... 122
2.3.32 New Rows ..... 124
2.3.33 Ngram ..... 127
2.3.34 Num Base ..... 131
2.3.35 Num Format ..... 134
2.3.36 Offset ..... 138
2.3.37 Outliers ..... 139
2.3.38 Pad ..... 144
2.3.39 Pivot ..... 146
2.3.40 Remove Cols ..... 149
2.3.41 Rename Cols ..... 151
2.3.42 Reorder Cols ..... 152
2.3.43 Replace ..... 154
2.3.44 Row Num ..... 163
2.3.45 Sample ..... 165
2.3.46 Scale ..... 167
2.3.47 Slice ..... 175
2.3.48 Sort ..... 177
2.3.49 Split Col ..... 182
2.3.50 Split Rows ..... 186
2.3.51 Spread ..... 189
2.3.52 Stack ..... 191
2.3.53 Stamp ..... 193
2.3.54 Stats ..... 197
2.3.55 Substitute ..... 199
2.3.56 Subtract ..... 201
2.3.57 Summary ..... 202
2.3.58 Total ..... 206
2.3.59 Transpose ..... 209
2.3.60 Unfill ..... 210
2.3.61 Unique ..... 213
2.3.62 Units ..... 215
2.3.63 Whitespace ..... 218
**2.4 Output ..... 221**


*© 2023 Oryx Digital Ltd*

2.4.1 Output data ..... 221
**2.5 File formats ..... 227**
2.5.1 File formats ..... 227
2.5.2 CSV format ..... 227
2.5.3 Excel format ..... 230
2.5.4 Fixed width format ..... 231
2.5.5 JSON format ..... 233
2.5.6 HTML format ..... 236
2.5.7 Markdown format ..... 238
2.5.8 Plain text format ..... 238
2.5.9 TSV format ..... 239
2.5.10 vCard format ..... 241
2.5.11 XML format ..... 241
2.5.12 YAML format ..... 246
**2.6 Processing ..... 246**
**2.7 Comments ..... 248**
**2.8 Headers ..... 250**
**2.9 Connections ..... 251**
**2.10 Text ..... 253**
**2.11 Dates ..... 253**
**2.12 Numbers ..... 255**
**2.13 Booleans ..... 255**
**2.14 Locale ..... 256**
**2.15 Meta Information ..... 257**
**2.16 Column variables ..... 259**
**2.17 Regular expressions ..... 260**
**2.18 Fuzzy matching ..... 261**
**2.19 Drilldown ..... 262**
**2.20 Batch processing ..... 264**
**2.21 Command line arguments ..... 266**
**2.22 File name variables ..... 268**
**2.23 .transform files ..... 270**
**2.24 Keyboard shortcuts ..... 271**
**2.25 Dark mode ..... 274**

## 3. How do I? 276

**3.1 Add a transform between existing items ..... 277**


*© 2023 Oryx Digital Ltd*

3.2 Add metadata to a dataset ..... 277
3.3 Add missing data values ..... 278
3.4 Add or remove a header ..... 280
3.5 Change a connection ..... 281
3.6 Change encoding ..... 281
3.7 Clean a dataset ..... 282
3.8 Convert to percentages ..... 285
3.9 Duplicate a series of transforms ..... 287
3.10 Dedupe a dataset ..... 287
3.11 Find the difference between dates/datetimes ..... 293
3.12 Handle column name/order changes in inputs ..... 295
3.13 Handle datasets with many columns ..... 297
3.14 Handle large datasets ..... 298
3.15 Input a fixed width format file ..... 299
3.16 Merge datasets ..... 299
3.17 Move a .transform file ..... 304
3.18 Open multiple main windows ..... 304
3.19 Output nested JSON or XML ..... 304
3.20 Output to Excel ..... 305
3.21 Profile a dataset ..... 307
3.22 Perform the same transforms on many files ..... 308
3.23 Replace empty values ..... 312
3.24 Run jobs on a schedule ..... 314
3.25 Scrape web data ..... 315
3.26 See changes from a transform ..... 317
3.27 Set output file name from input file ..... 319
3.28 Show whitespace ..... 320
3.29 Split a dataset into multiple files ..... 321
3.30 Trigger an update when an input changes ..... 325
3.31 Visualize data ..... 326
3.32 Write to multiple sheets of an Excel file ..... 329

4. Expert tips 330

4.1 Expert tips ..... 331

5. Support 340


© 2023 Oryx Digital Ltd

5.1 Forum ..... 341
5.2 Contact support ..... 341
5.3 Report a bug ..... 341
5.4 Request an enhancement ..... 341

Index 342


© 2023 Oryx Digital Ltd

# Getting started

Easy Data Transform data wrangling software for Windows and Mac.

# 1 Getting started

## 1.1 Introduction

Easy Data Transform allows you to quickly transform table and list data into new and more useful forms, without programming. The step-by-step visual transformation is quicker, more interactive, more repeatable and less error prone than other approaches.

Please take a couple of minutes to read the Quick Start Guide.

## 1.2 System requirements

The suggested requirements for running this software are:

- Operating system: Windows 11, 10, 8 or 7 (64 bit versions only).
- Screen resolution: 1280x720 pixels or better.

If your operating system is more recent than the above, check our website to find a compatible version of Easy Data Transform.

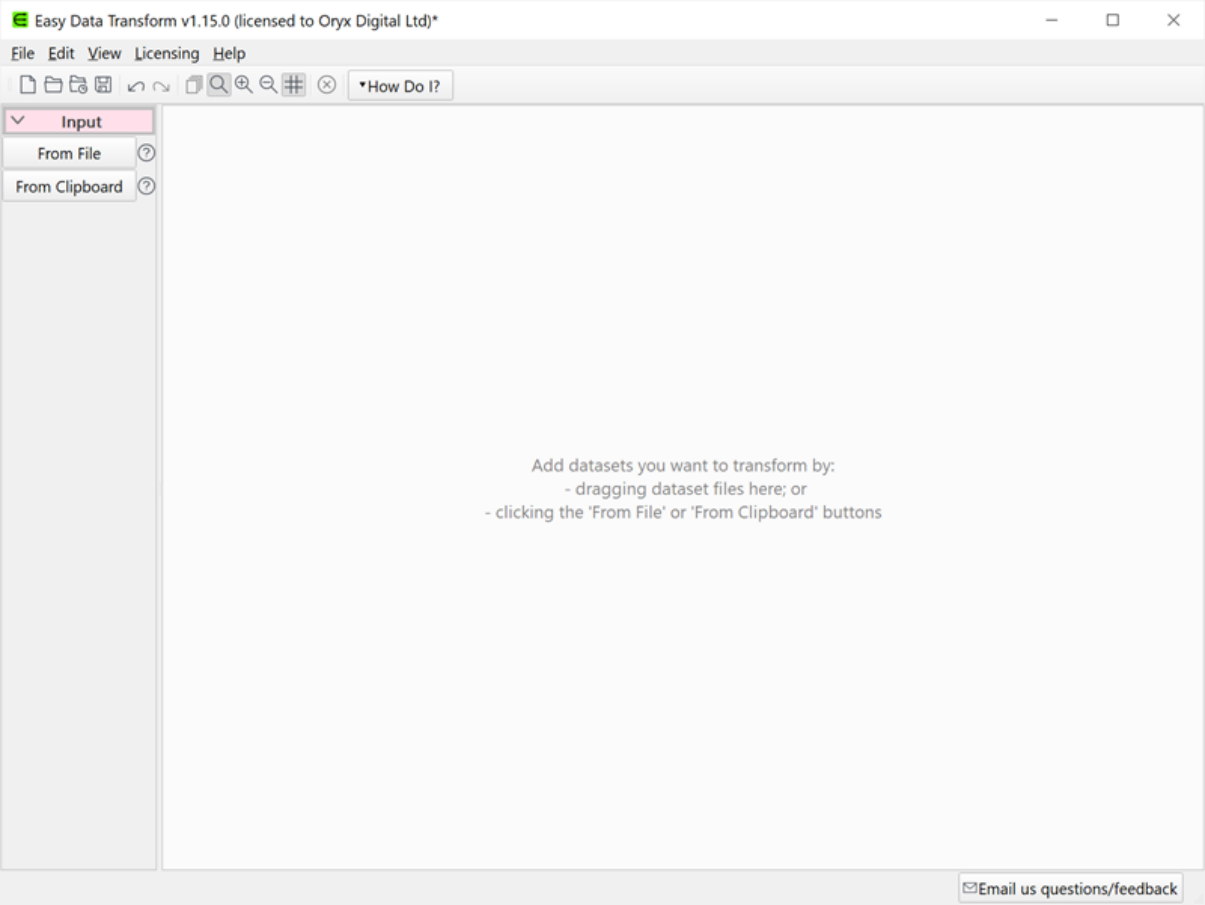
You may be able to run the software satisfactorily on lower specification systems or more operating systems, but we can't guarantee it. If in doubt, try running an unlicensed trial version before you buy a license.

## 1.3 Quick start guide

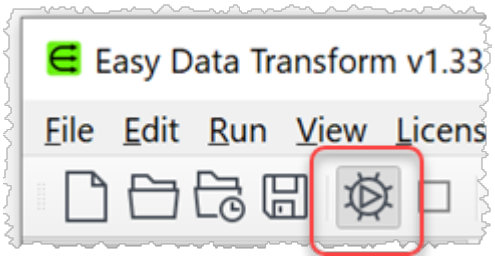
This is a quick tour of some of Easy Data Transform's features. It should only take a few minutes to complete.

Start Easy Data Transform. If the **Free Trial** window appears, click **Continue free trial**. If the Getting Started window appears, click **I got this!** (or you will just end up back on this page). You should now see the main window.


*© 2023 Oryx Digital Ltd*

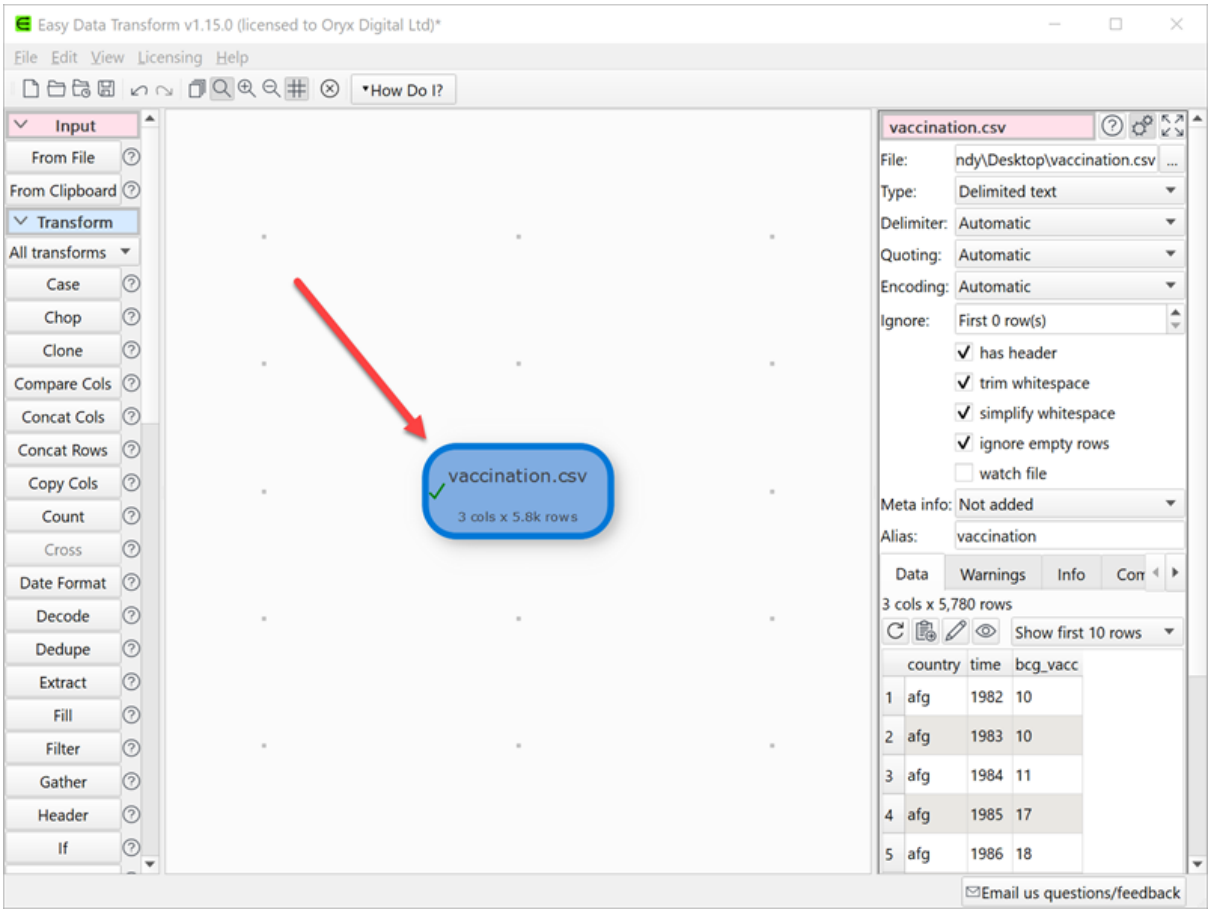



Make sure that **Auto Run** tool bar button is checked (pressed in), so that any changes you make are processed automatically after a short delay.




Drag a data file you want to transform onto Easy Data Transform. For example a CSV file or an Excel .xlsx/.xls file. XML, JSON, fixed width, plain text and vCard formats are also supported.

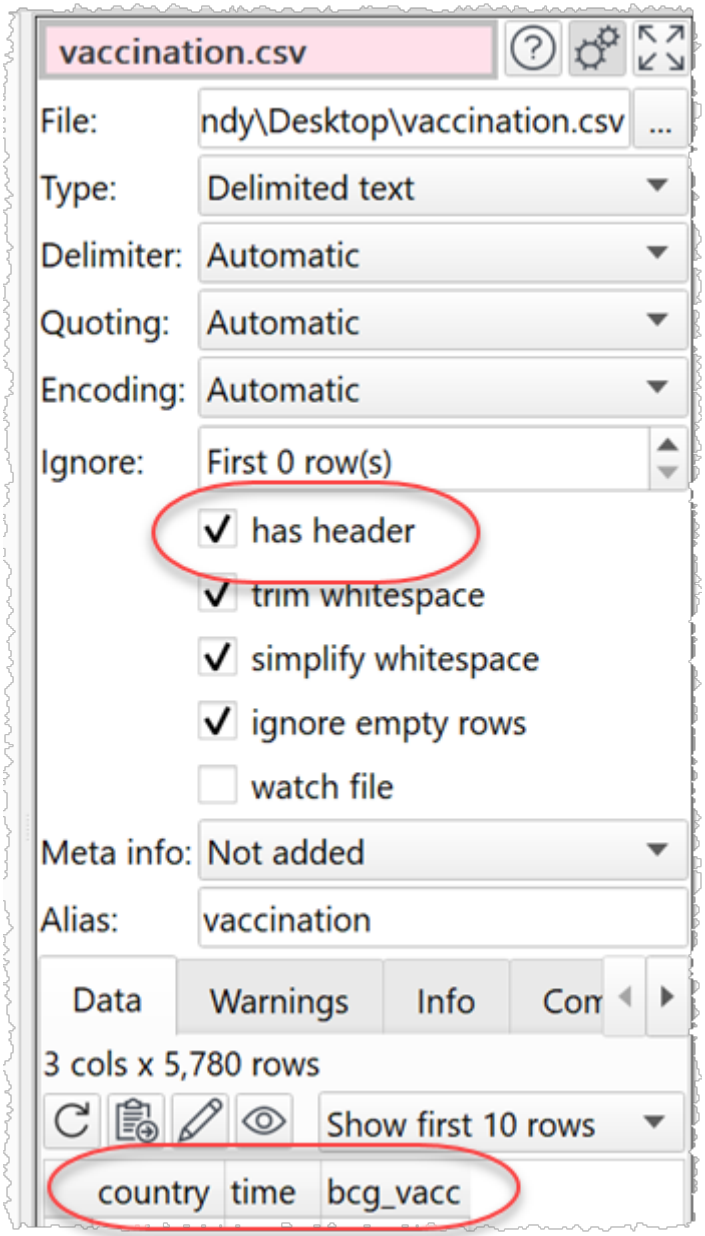

*© 2023 Oryx Digital Ltd*




Notice that the available transforms are shown in the **Left** pane and the selected dataset is shown in the **Right** pane.

In the **Right** pane, you can check **has header**, depending on whether you want to treat the first row of the dataset as a header.

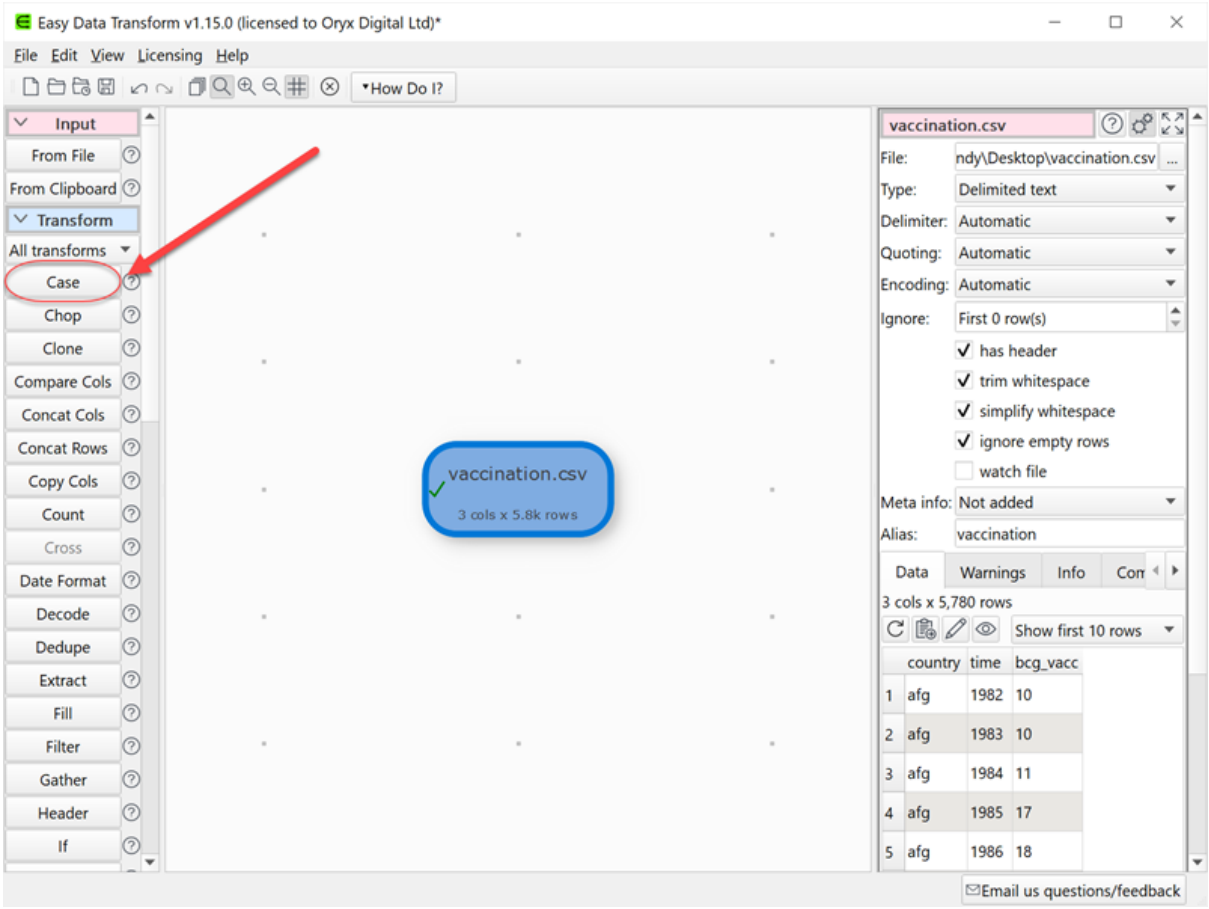

*© 2023 Oryx Digital Ltd*




All the transforms available for a single dataset are now enabled in the **Left** pane. Hover over the transforms to see tooltips explaining what they do. Click on the **?** next to a transform button for more details.

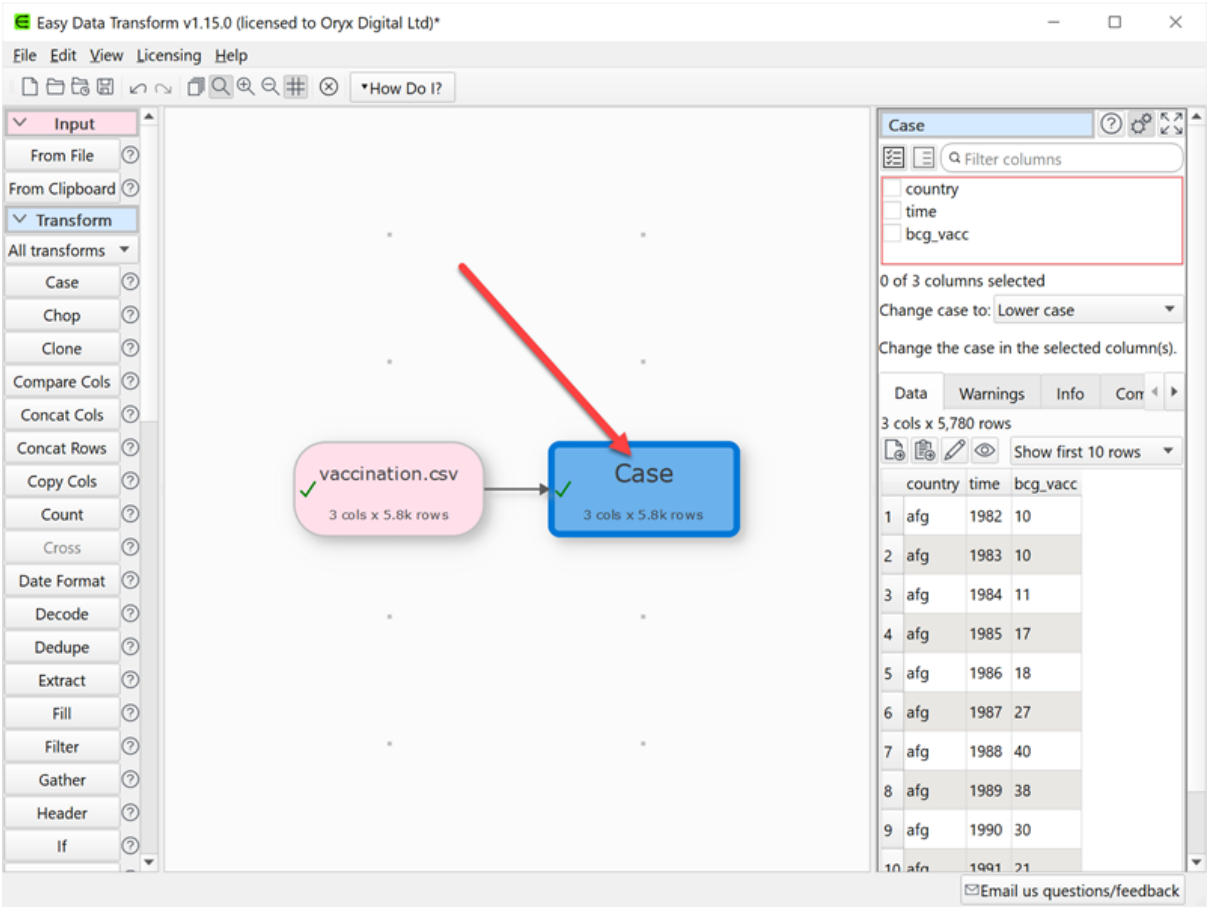
Ensure the input item is selected and click on the **Case** transform button to change the case of your data.


*© 2023 Oryx Digital Ltd*



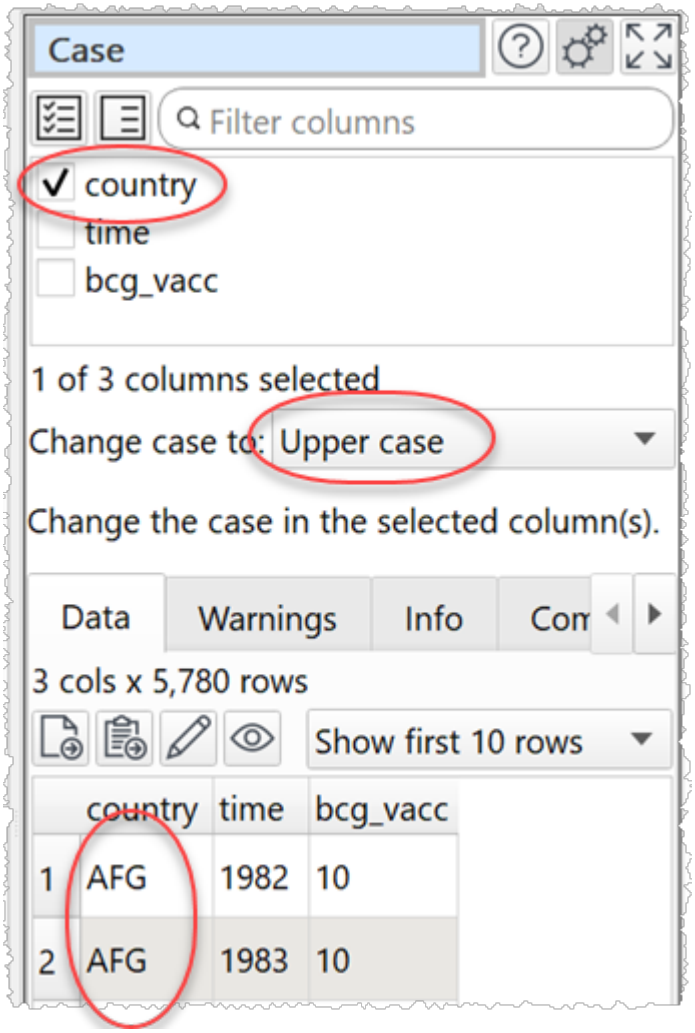

A **Case** transform item will now be added.


*© 2023 Oryx Digital Ltd*




In the **Right** pane, check one of the columns and set **Change case to** to **Upper case**. All the text in that column will now be converted to upper case. You don't have to click a 'run' button.


*© 2023 Oryx Digital Ltd*

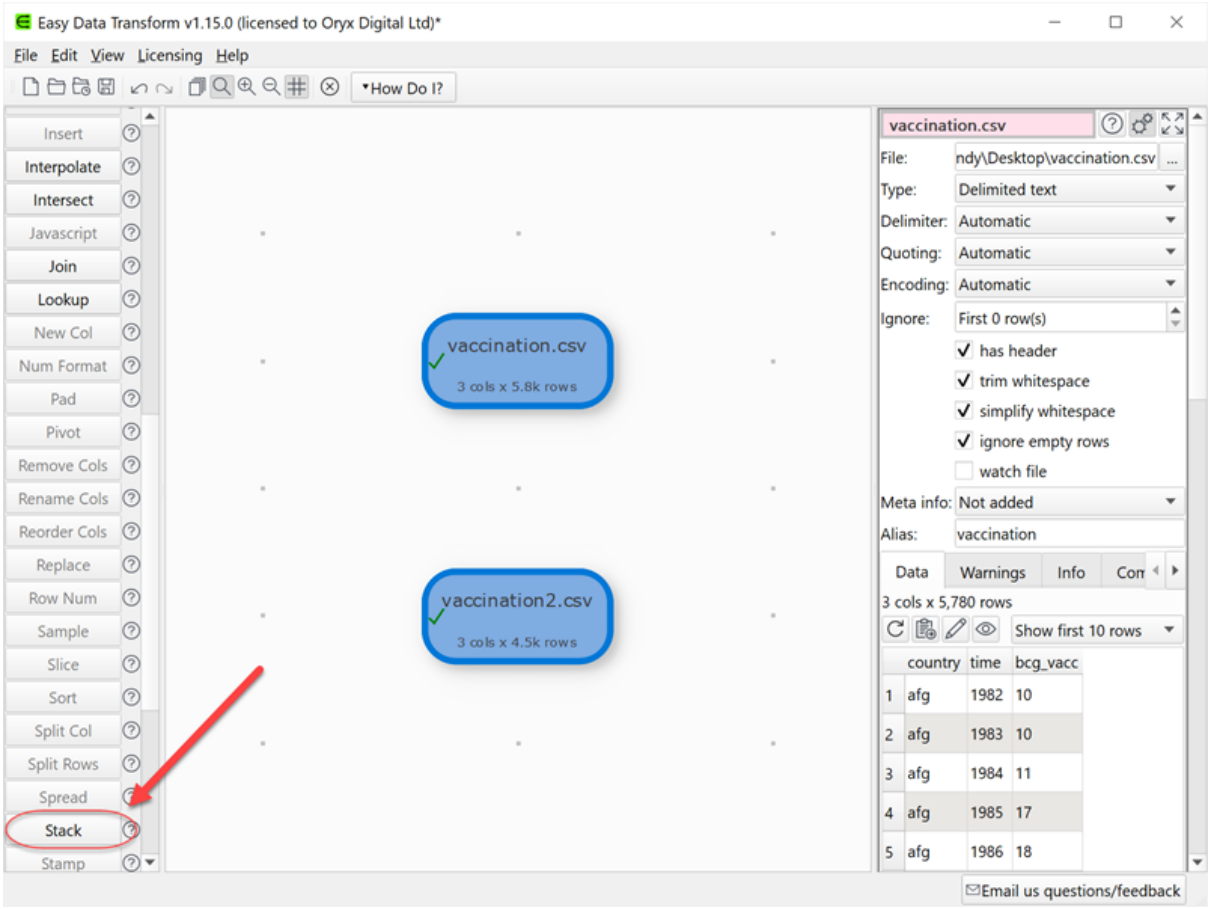



You can create a sequence of transforms to perform complex manipulations.

Some transforms require more than one input dataset. For example, to stack two datasets, one on top of the other:

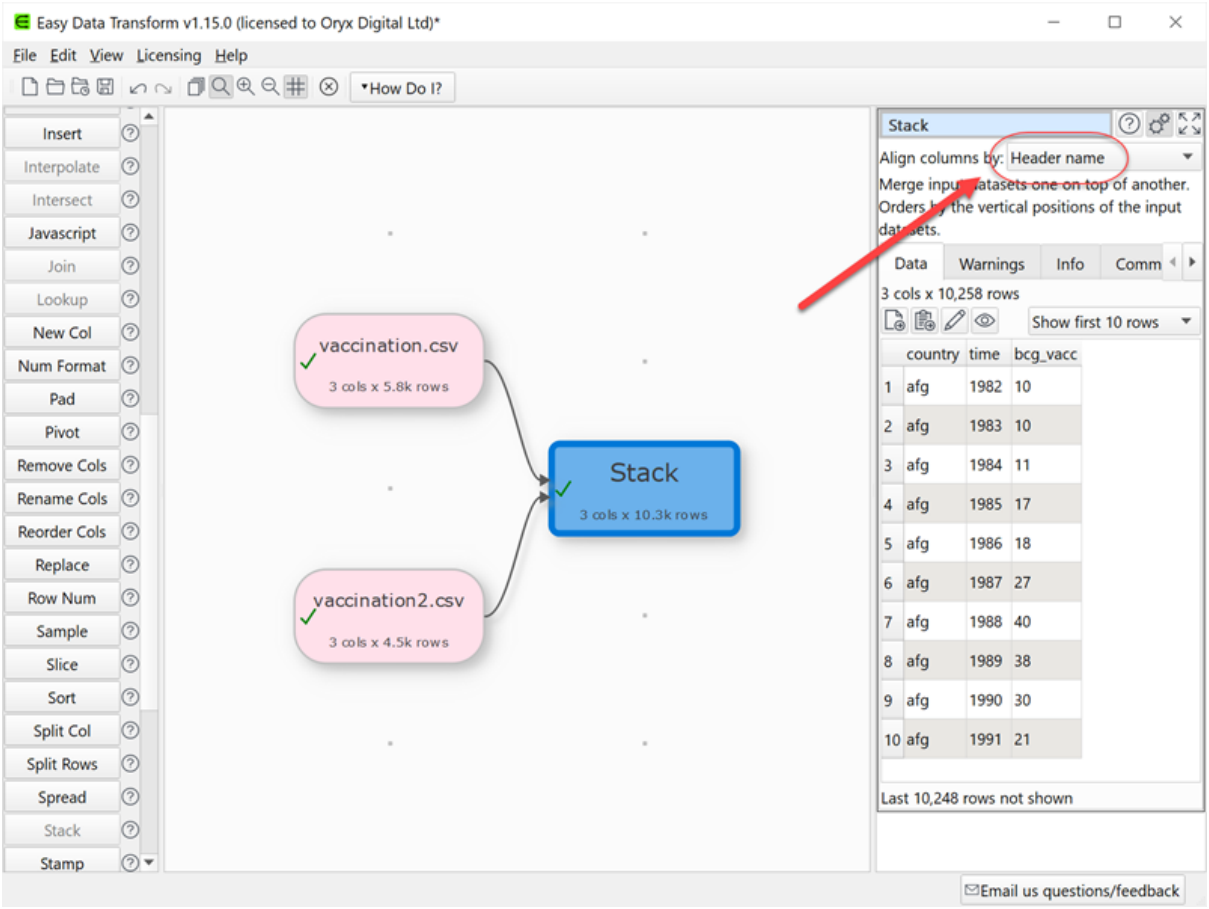
- Select **File>New** to start again. Don't save the changes.
- Drag two data files onto the **Center** pane.
- Select both input items (by dragging a box around them or using `Ctrl`+click).
- Click the **Stack** transform button (you may need to scroll the **Left** pane to see the button).


*© 2023 Oryx Digital Ltd*




The tables are now stacked one on top of the other in a new dataset item. You can choose to match the columns by **Header name** or **Column number**.


*© 2023 Oryx Digital Ltd*

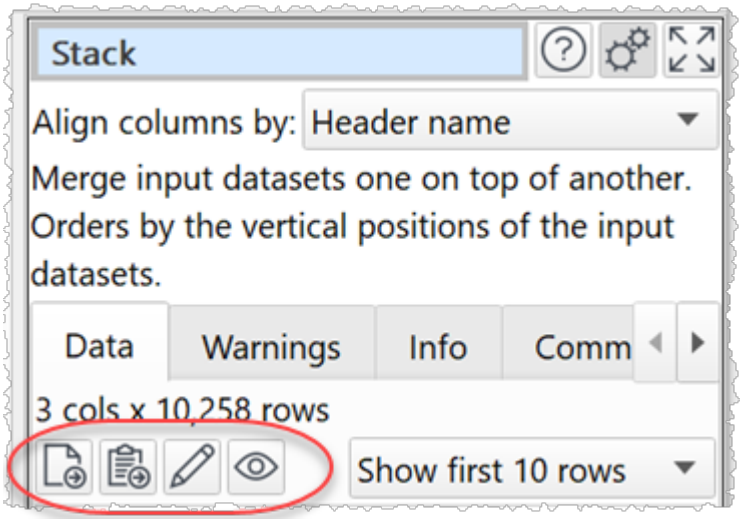



Note that the vertical (Y) position of the inputs affects the order the datasets are stacked. Try swapping the two inputs around and re-select **Stack** to see the affect.

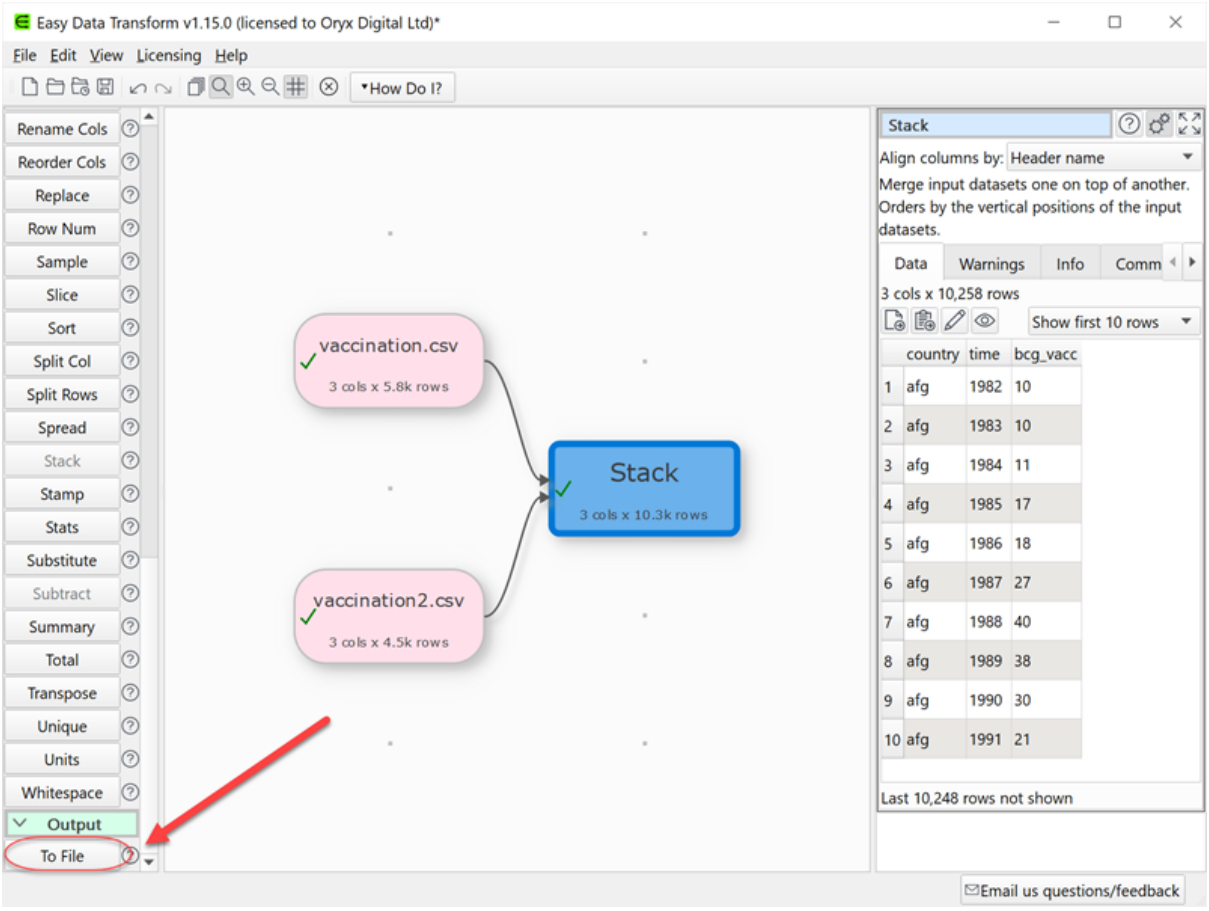
Any changes to input files will be automatically read in. Any changes to input datasets or transform options will be automatically propagated 'downstream'.

To export your transformed dataset to a file or the clipboard, view it in a local editor or see it with whitespace displayed, click on the appropriate button in the **Right** pane.


*© 2023 Oryx Digital Ltd*

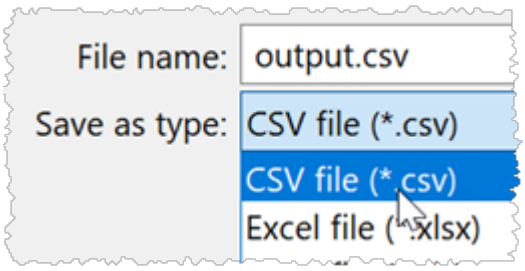



You can also add an output item to automatically write a dataset to file whenever it changes.

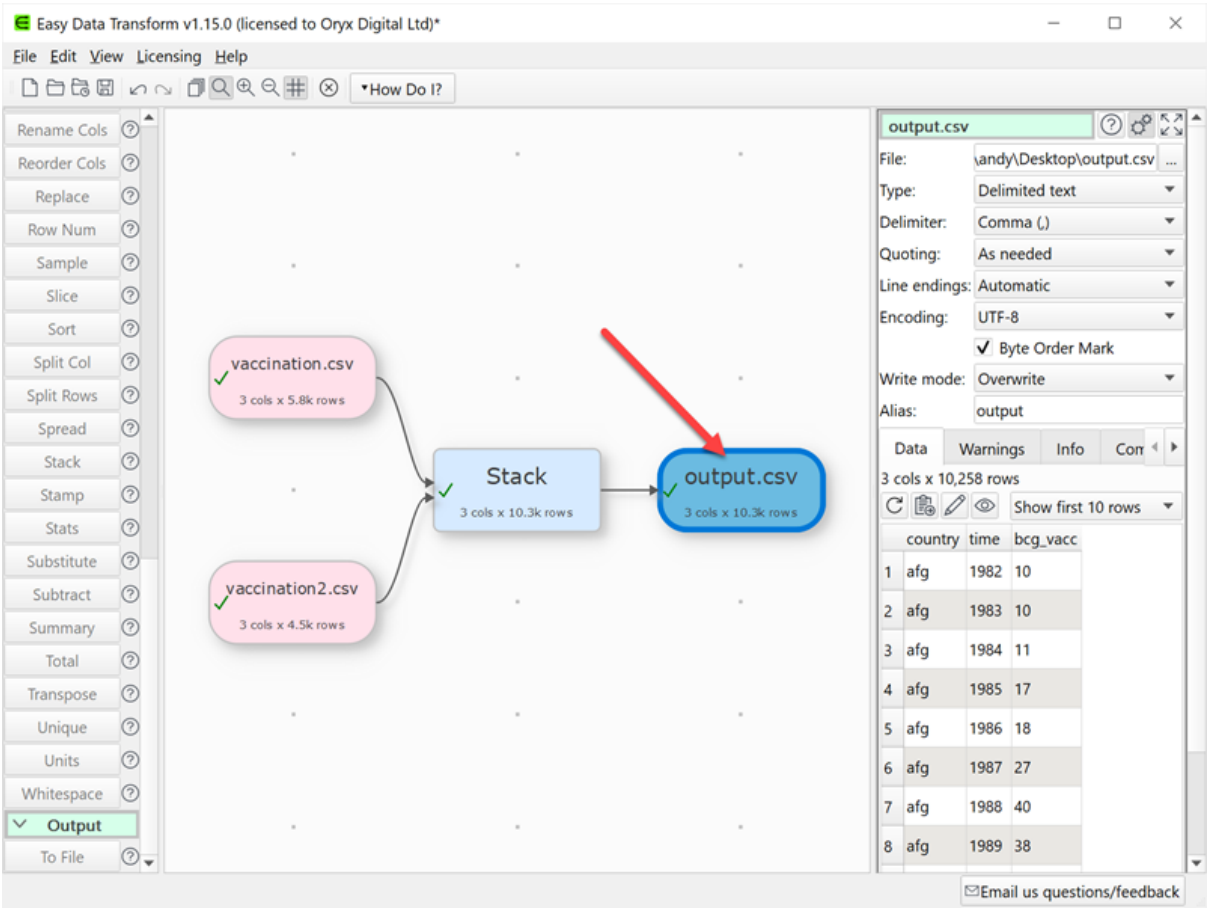



You will be asked for a file to write to. You can choose amongst CSV, Excel, HTML, JSON, Markdown, Plain text, TSV, vCard, XML and YAML file formats. Select **CSV file**.


*© 2023 Oryx Digital Ltd*

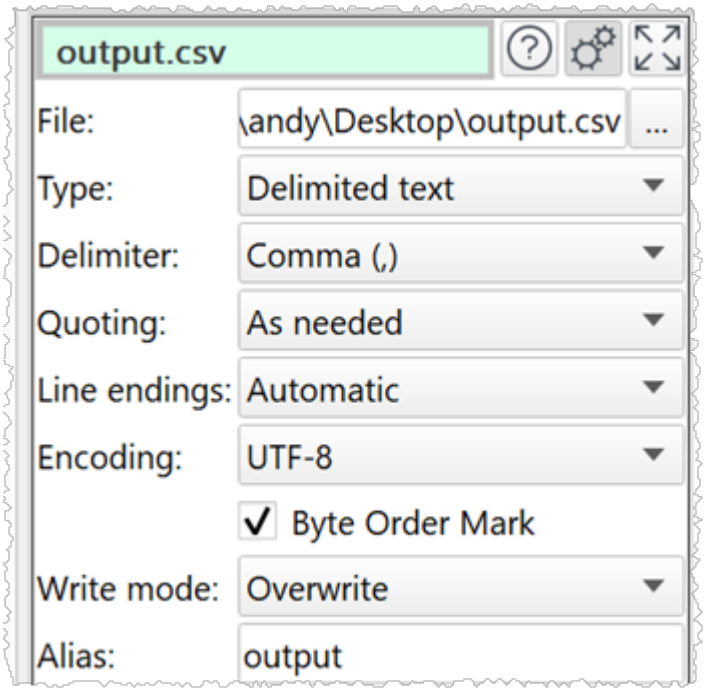



Your dataset will then be written to this file every time it changes.




You can also specify the **Delimiter**, **Quoting**, **Encoding, Line endings** etc for your CSV files in the **Right** pane.


*© 2023 Oryx Digital Ltd*




You can save your transforms to a `.transform` document to use again with **File>Save**.

Have a play!

Tips:

- You can also watch a short introductory video.
- Check the **show advanced** check box in the **Right** pane to see all the available transforms.
- You can also paste in data from the clipboard (for example, a table from a web page or Word document).
- Transforms such as **Compare Cols**, **Filter**, **If** and **Sort** take account of dates, numbers and text. You can define what date formats to recognize in the Preferences window.
- New columns are always added to the right of a dataset.
- Comparisons of text are always sensitive to case, unless stated otherwise. E.g. "CASE", "case" and "Case" are treated differently.
- Comparisons of text are always sensitive to whitespace (e.g. spaces and tabs), unless stated otherwise. You can use the Whitespace transform to remove leading and trailing whitespace.
- The contents of input and output data files are not saved in Easy Data Transform, only their locations.
- As well as stacking two datasets, you can also Join them, side-by-side, if they have a common ('key') column.
- You can insert a new transform between existing items by selecting the connection between the items and then adding the transform.
- To be more productive with Easy Data Transform see expert tips.


*© 2023 Oryx Digital Ltd*

We are interested in your feedback, so please contact us to ask a question, report a bug or request an enhancement.


*© 2023 Oryx Digital Ltd*

# Reference

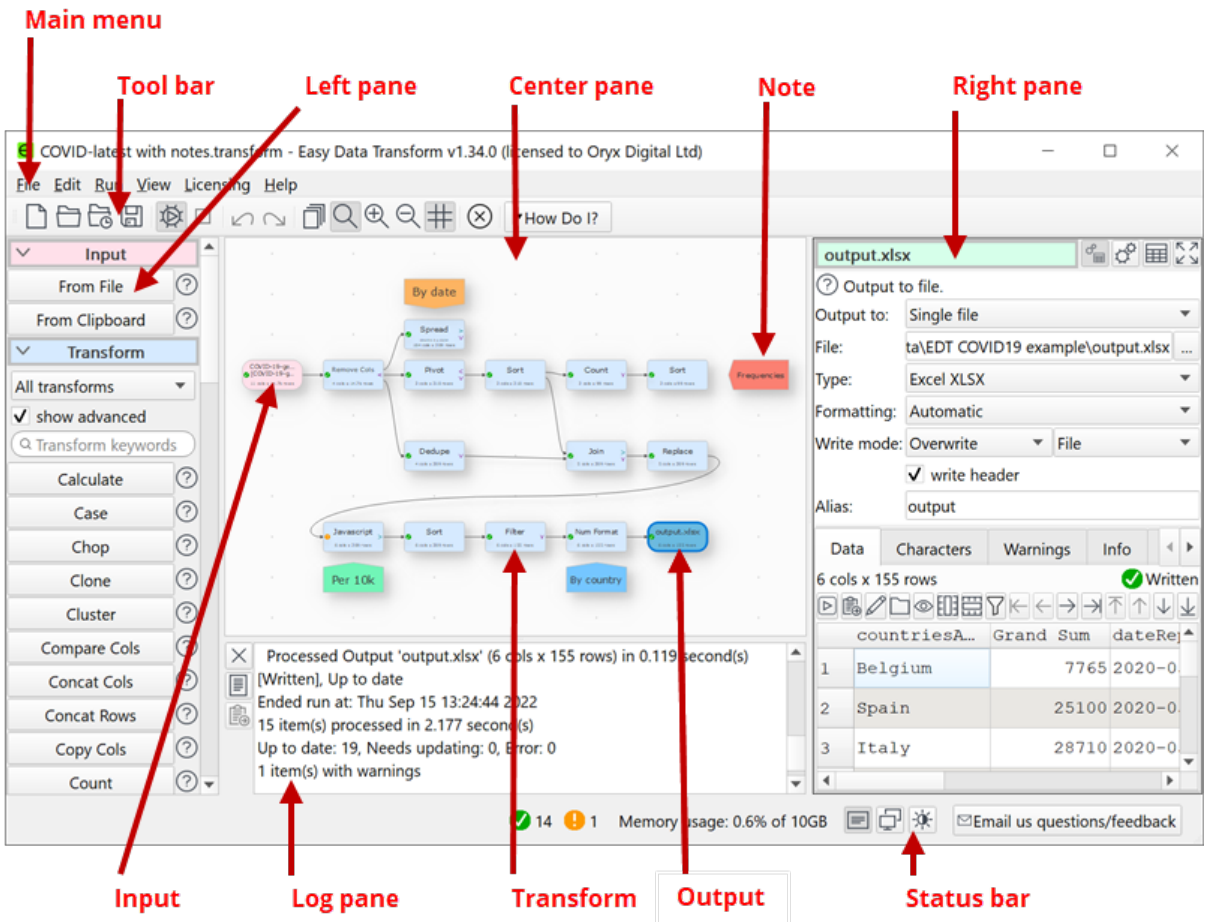
# 2 Reference

## 2.1 User Interface

### 2.1.1 Main window

The Main window comprises:

- **Main** menu
- **Tool** bar
- Left pane
- Center pane
- Right pane
- Log pane
- **Status** bar




You can split the Right pane onto a separate monitor (if available) by selecting **View>Two Screen Mode**.


*© 2023 Oryx Digital Ltd*

### 2.1.2 Left pane

The **Left** pane shows all the available actions you can perform. Which actions are visible will depend on what is shown in the Center pane. Which actions are enabled depends on what is selected in the **Center** pane.

### 2.1.3 Center pane

The **Center** pane show the inputs, transforms and outputs you are using to transform your data.

### 2.1.4 Right pane

The **Right** pane shows details of an input, transform or output item you have selected in the Center pane.

### 2.1.5 Log pane

The **Log** pane shows a text representation of processing on the current .transform file. It shows a maximum of 5000 rows.

- Click the ✕ button to clear all text in the pane.
- Click the button to select all text in the pane.
- Click the button to copy selected text in the pane to the clipboard.
- Select **View>Log** (or the corresponding status bar button) to hide/show the pane.

### 2.1.6 Preferences window

## General tab

Check **give option to disable outputs when opening a .transform file** if you want the option to disable any outputs with write mode set to overwrite or append. This prevents accidentally writing over existing files. Note that this check is never made when you open .transform files with **Run>Auto Run** checked or using the -cli command line argument.

Check **use native file windows** to use the native Windows file open/save windows.

Check **make a sound when processing completed** if you want to make a system sound every time processing is completed.

Set **At start-up** depending on what you want to happen when Easy Data Transform starts.

Set **User interface theme** to **Automatic** to follow the theme of the operating system or override it by setting **Light** or **Dark**. This is only available if your operating system supports a dark theme. Changing **User interface theme** does not update the Center pane colors. You can also toggle between themes using a keyboard shortcut.


*© 2023 Oryx Digital Ltd*

Set **Tool bar icon size** to the size of the icons you wish to display in the tool bar.

Set **Copy data to clipboard text/plain as** according to how you want data to be formatted, when it is copied to the clipboard with MIME type text/plain.

- **Comma Separated Value** uses comma as a column delimiter and quotes carriage returns, commas and quotes within data.
- **Tab Separated Value** uses tab as a column delimiter and converts carriage returns and tabs to spaces within data.
- **Plain text** outputs without any column delimiter.

Data is also copied to the clipboard with MIME type text/csv in Comma Separated Value format and to MIME type text/html as an HTML fragment.

Set **Optimize processing for** to:

- **Minimum memory use** to use memory compression to reduce the memory used. This is usually recommended. It works best when columns contain repeated values.
- **Maximum speed** to turn off memory compression. This might be faster, but uses a lot more memory. It might also be slower if the memory required is more than your RAM. It might also cause **Maximum memory usage** to be exceeded.

Set **Maximum memory usage** to the maximum amount of memory that you want Easy Data Transform to be allowed to use. If this limit is exceeded the program will try to exit cleanly.

Set **Right pane processing delay** depending on how long you want to wait after changes in the **Right** pane before starting processing, when **Run>Auto Run** is checked. Setting the value to 0 is generally not recommended, as this means that every single click in the **Right** pane will cause processing.

Set **Zoom wheel behavior** according to how you want the mouse wheel to work in the **Center** pane. Hold down the `Ctrl` key while moving the mouse wheel to switch between zoom and scroll. Hold down the `Alt` key while moving the mouse wheel to switch between up/down and left/right scroll.

**User interface font** shows the font used for the application user interface, apart from data tables (see below). Click **Choose...** to choose a new font. Click **Default** to set it back to the operating system default.

**Data table font** shows the font used in the data tables in the **Right** pane. Click **Choose...** to choose a new font. You might prefer a monospaced (fixed width) font such as Consolas, Lucida Console or Courier New. Click **Default** to set it back to the operating system default.

The **Locale** language and country setting affects how some numbers and dates are displayed. Consequently it may have an affect on some transforms. It does not change the language of the user interface, which is English only.


*© 2023 Oryx Digital Ltd*

## Dates tab

Set **Supported date formats** to the date formats you wish to recognize. List the date formats in order of preference, with the most likely to be used first.

## Input Extensions tab

Set the default file types corresponding to input file extensions.

Click in the **Extension** column and type to change an input file extension. The text will be trimmed of whitespace, converted to lower case and any '.' characters removed.

Click in the **Default type** column and change the drop-down list to change the file type to associate with an input file extension.

Click **Add** to add a new input file extension.

Click **Remove** to remove the selected input file extension(s).

Click **Default** to set the input file extensions back to the default setting.

The order in which input file extensions are shown is not significant.

## Output Extensions tab

Set the default file types corresponding to output file extensions.

Click in the **Extension** column and type to change an output extension. The text will be trimmed of whitespace, converted to lower case and any '.' characters removed.

Click in the **Default type** column and change the drop-down list to change the file type to associate with an output file extension.

Click **Add** to add a new output file extension.

Click **Remove** to remove the selected output file extension(s).

Click **Default** to set the output file extensions back to the default setting.


*© 2023 Oryx Digital Ltd*

The order in which output file extensions are shown is not significant.

### Colors tab

Select a **Color scheme** for the **Center** pane. A **Preview** is shown in the **Preferences** window. Choose **Custom** to choose your own color scheme.

## 2.2 Input

#### 2.2.1 Input data

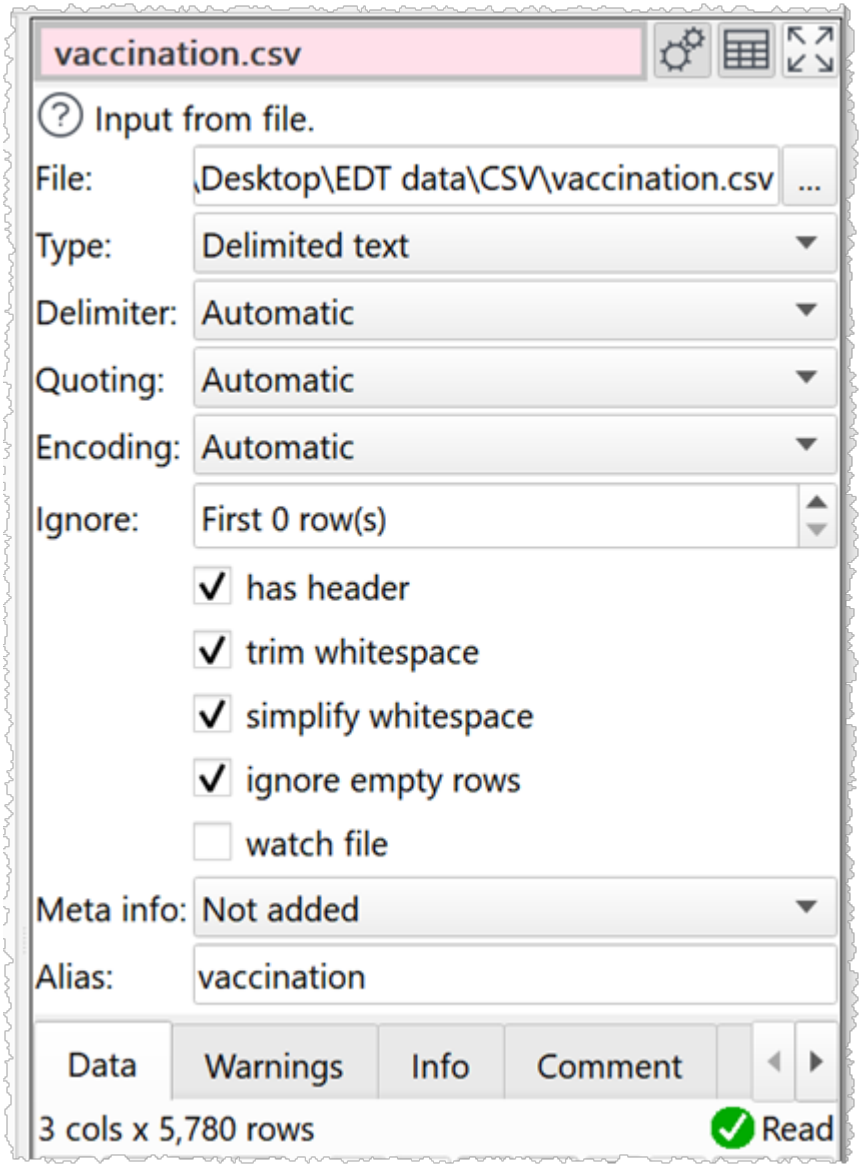
You need to input data before you can transform it. Data can be input by:
- dragging a file onto the Center pane; or
- clicking the **From File** or **From Clipboard** button in the Left pane

Enter the file location in **File** or click the browse button. For Excel spreadsheets you also need to add a sheet name, e.g. 'MySpreadsheet.xlsx[Sheet1]'.

Easy Data Transform can input data from files in the following formats:
- delimited text file (e.g. CSV or TSV) with various delimiters
- Excel .xlsx or .xls
- fixed width
- Plain text
- JSON
- vCard
- XML

You can select the input item in the **Center** pane and change any related options in the Right pane.


*© 2023 Oryx Digital Ltd*




Set **Type** to the file type. The default type will be set according to the file extension and the settings in the **Input Extensions** tab of the Preferences window.

For text files Easy Data Transform will treat `CRLF`, `LF` or `CR` control characters as line endings.

Easy Data Transform will make an intelligent guess at the:
- column delimiter and quoting for delimited text files (e.g. comma)
- column widths for fixed width text files
- text encoding for text files (e.g. UTF-8)
- presence of a header row in the data

But you can also do this manually by selecting the input item and changing the **Delimiter, Columns**, **Encoding** and **has header** fields in the Right pane.


*© 2023 Oryx Digital Ltd*

Set **Ignore** to the number of rows you want to skip before you start inputting. Note that this takes place before any empty rows are removed by **Ignore empty rows**.

For delimited text you can:
- Set **Delimiter** to the delimiter character.
- Check **ignore repeated delimiters** if you want to treat 2 or more consecutive delimiters as a single delimiter.
- Set **Quoting** according to whether " quoting is used so that the delimiter can appear within values.

See delimited text for more details.

For fixed width set **Columns** to **Manual** to set column widths. See fixed width for more details.

For JSON or XML set **Format** to **Long (more rows)** or **Wide (more columns)** depending on how you want to treat arrays/repeat values. See JSON or XML for more details.

Check **trim whitespace** to trim any whitespace (e.g. tabs or spaces) off the start or end of data values.

Check **simplify whitespace** to:
- Replace any tabs or line feeds within data values with spaces.
- Replace non-standard spaces (such as non-breaking space, thin space etc) within data values with spaces.
- Remove carriage returns within data values.

Check **ignore empty rows** to remove any rows that have only empty values (whitespace is not considered empty).

Check **ignore empty columns** to remove any columns that have only empty values (whitespace is not considered empty). Header values are ignored when deciding if a column is empty.

Check **ignore hidden rows and columns** to remove any hidden rows or columns (Excel files only).

Check **watch file** if you want :
- **Run>Auto Run** selected: the input to automatically reload the file every time that Easy Data Transform detects that it has been changed.
- **Run>Auto Run** not selected: the input to be marked as 'Needs update' every time that Easy Data Transform detects that it has been changed.


*© 2023 Oryx Digital Ltd*

Use **Meta info** if you wish to add some meta information about the input dataset, e.g. the name of an input file or the date it was created.

Use **Alias** to identify the file for batch processing.

To change the file being used by an input, select the input item and change the file location in the **Right** pane (e,g, by clicking the '...' browse file button), rather the disconnecting the input and connecting a new one. Otherwise column-related parameters downstream will be reset.

## 2.3 Transforms

### 2.3.1 Transform data

Transforms operate on datasets from input data or other transforms. Some transforms only have a single input (e.g. Case), some transforms have two inputs (e.g. Join) and some transforms have two or more inputs (e.g. Stack).

To create a transform, select one or more input and/or transform items in the Center pane and then click the appropriate button in the Left pane.

Select from the drop-down list in the **Left** pane to choose which types of transform are displayed, e.g. select **Merge Transforms** to show only transforms related to blending data.

Check the **show advanced** checkbox in the **Left** pane to see all available transforms.

You can select the transform item in the **Center** pane and change any options related to the transform (e.g. which columns it acts on) in the Right pane.

The transform will be updated automatically if any input or transform 'upstream' of it changes.

### 2.3.2 Calculate

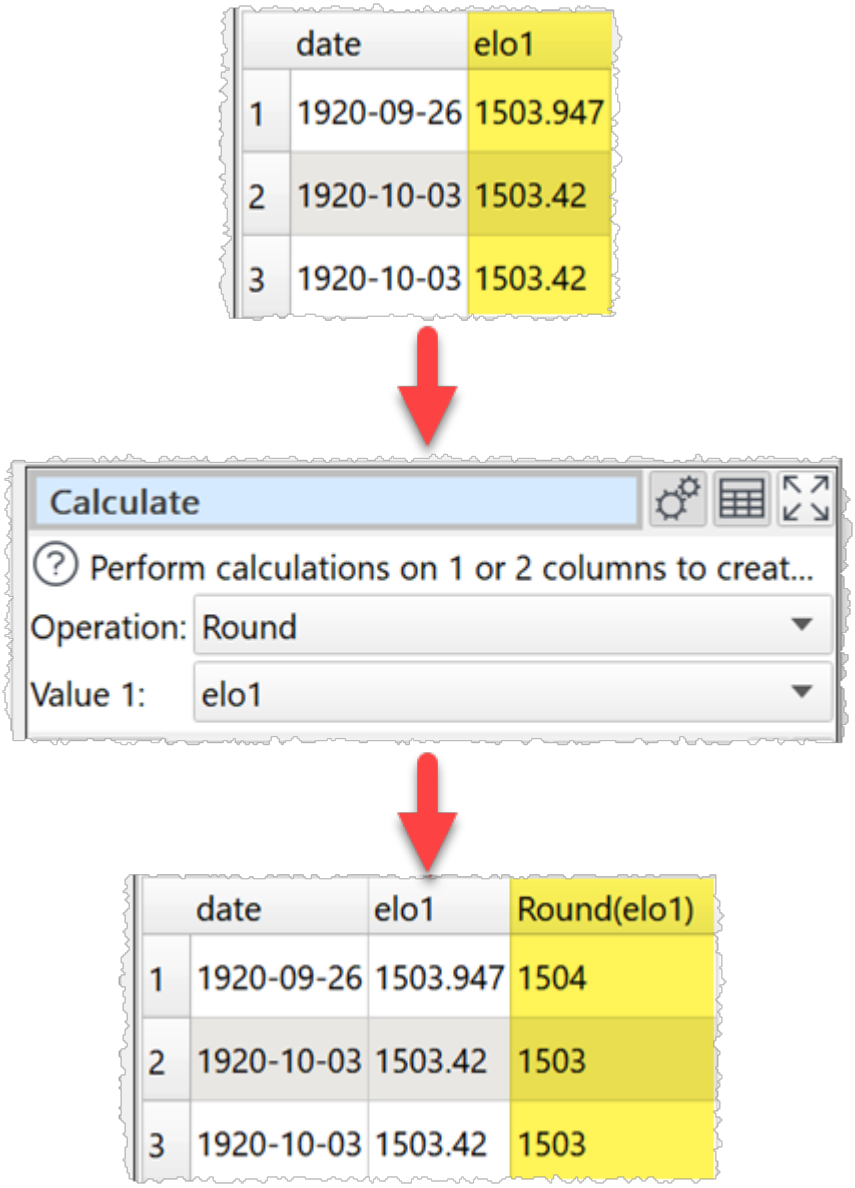
#### Description

Performs a calculation on 1 or 2 columns and creates a new column.

#### Examples

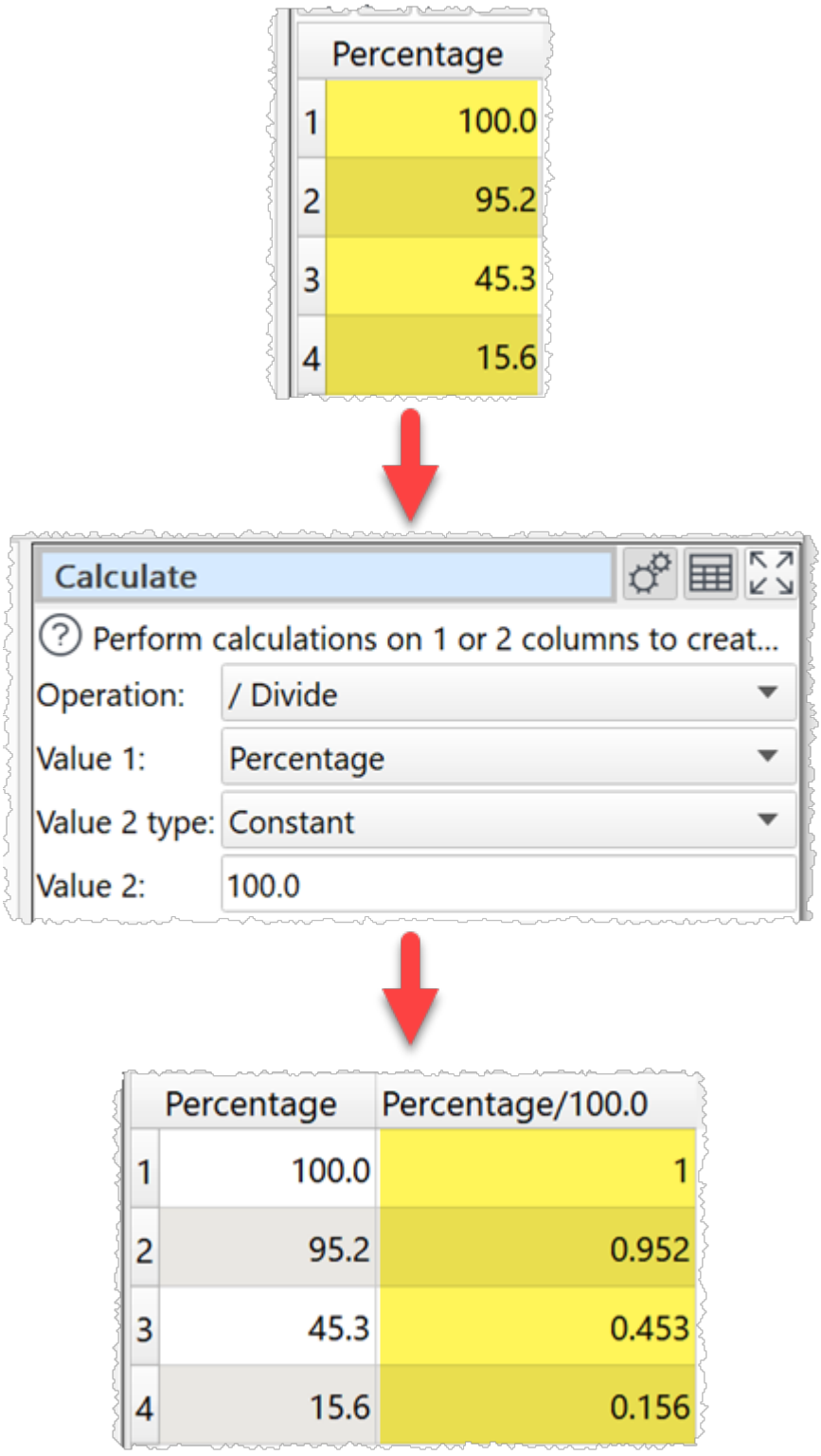
Round a column of numbers to the nearest integer:


*© 2023 Oryx Digital Ltd*



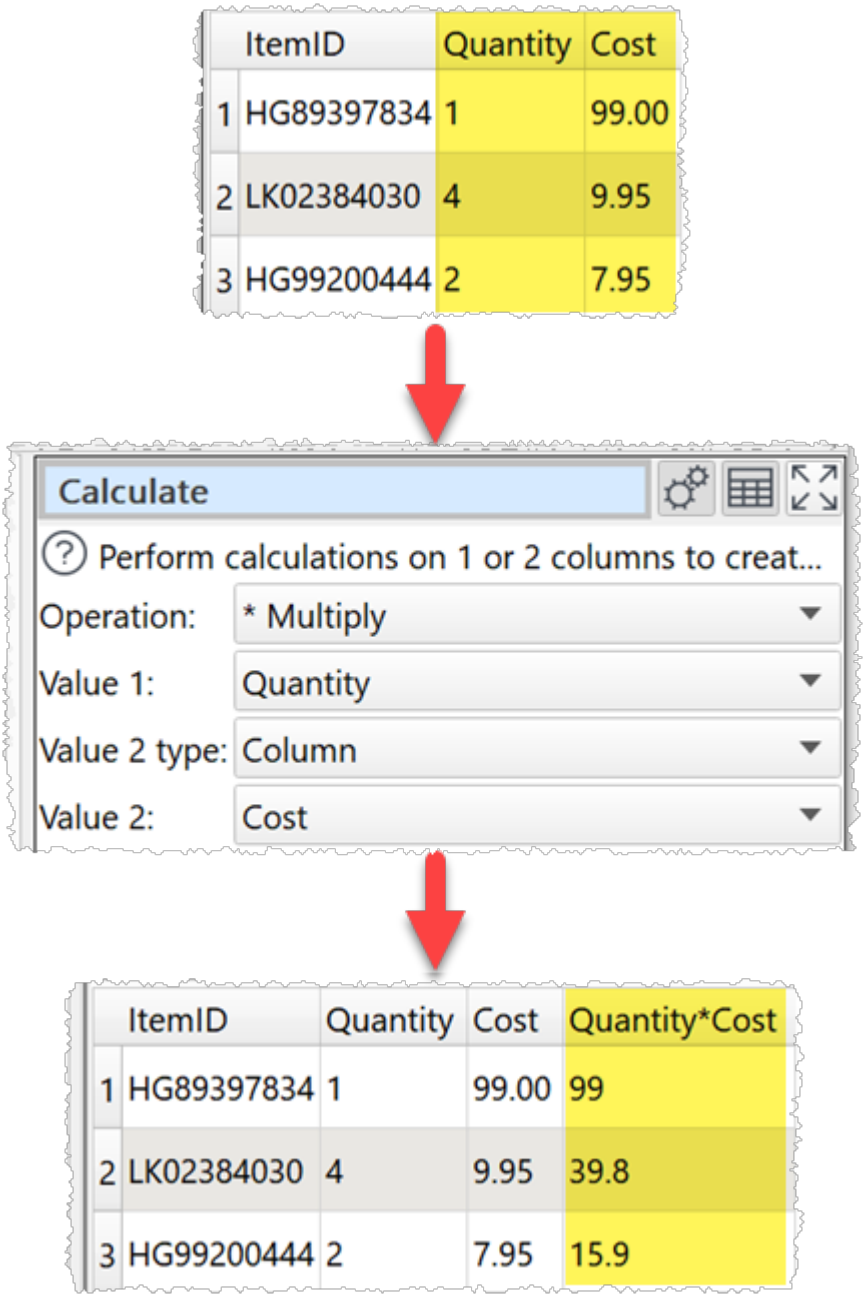

Divide a column of numbers by 100:


*© 2023 Oryx Digital Ltd*



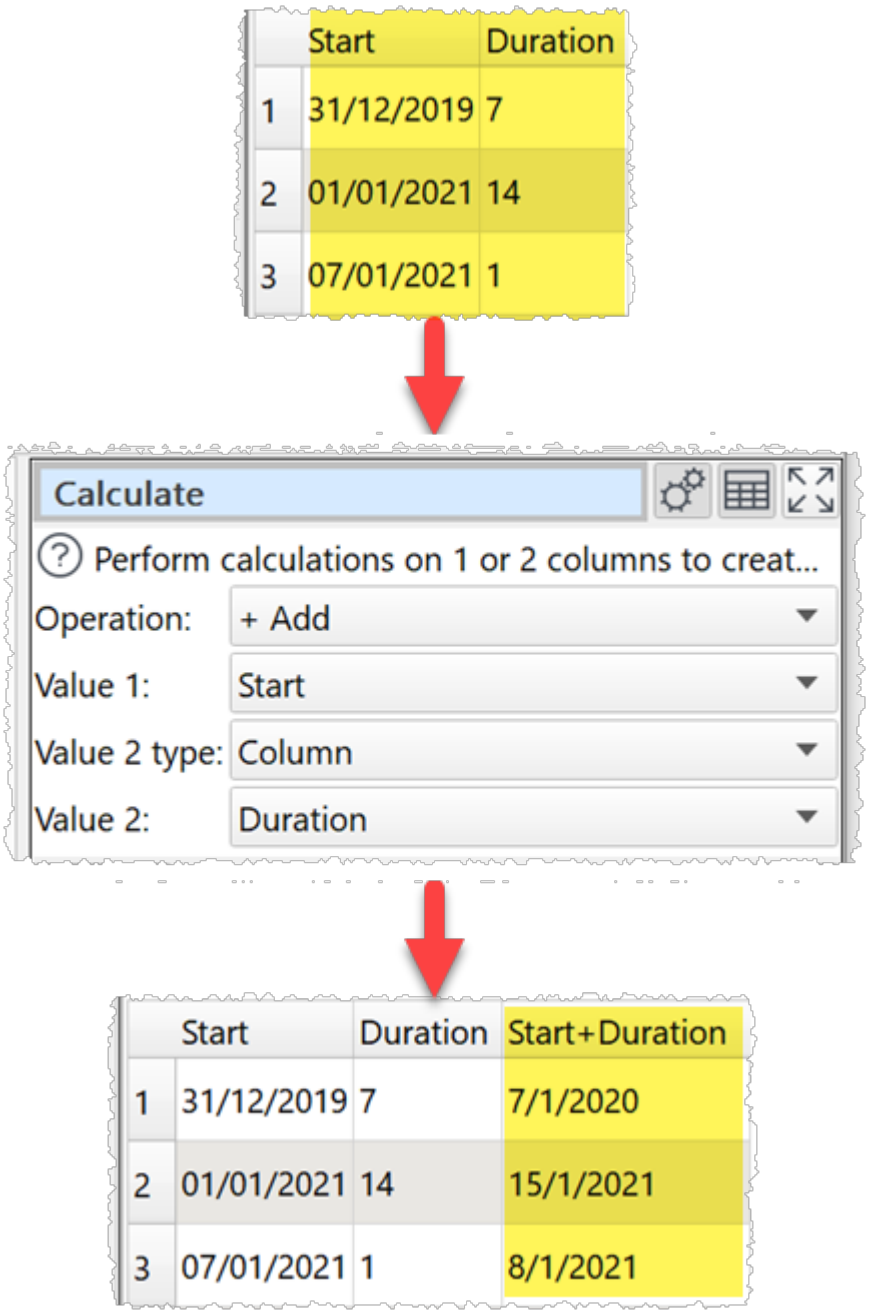

Multiply 2 columns of numeric values:


*© 2023 Oryx Digital Ltd*



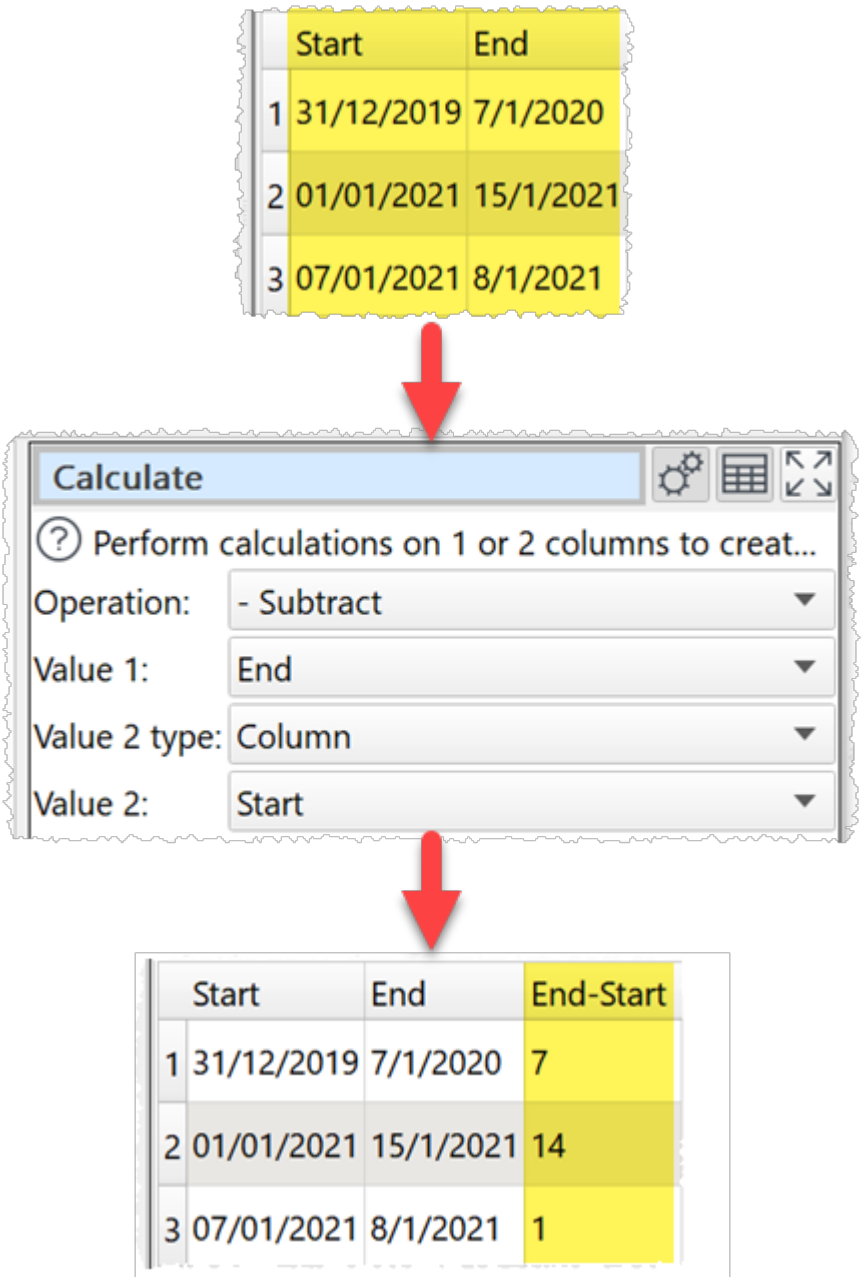

Add a days column to a date column:


*© 2023 Oryx Digital Ltd*




Calculate the difference in days between 2 date columns:


*© 2023 Oryx Digital Ltd*




## Inputs

One.

## Options

- Set **Operation** to the operation you wish to perform.
- Set **Value 1** to the first column of values you wish to operate on.
- Set **Value 2 type** depending on whether you wish to use a single value or a column of values for the second value (binary operations only).


*© 2023 Oryx Digital Ltd*

- Set **Value 2** to the second column of values or constant value you wish to operate on (binary operations only).

## Notes

- The following operations are supported.


*© 2023 Oryx Digital Ltd*

| Operation | Value 1 | Value 2 | Notes | Examples |
|---|---|---|---|---|
| **+ Add** | Number | Number | Numerical addition. | `1.2 + 3 = 4.2` |
| | Date | Integer number | Adds days to the date. | `31/01/2021 + 7 = 07/02/2021` |
| **- Subtract** | Number | Number | Numerical subtraction. | `1.2 - 3 = -1.8` |
| | Date | Integer number | Subtracts days from the date. | `31/01/2021 - 7 = 24/01/2021` |
| | Date | Date | Days the date in **Value 1** is after the date in **Value 2** (negative if before). | `31/01/2021 - 24/01/2021 = 7`<br>`24/01/2021 - 31/01/2021 = -7` |
| *** Multiply** | Number | Number | Numerical multiplication. | `1.2 * 3 = 3.6` |
| **/ Divide** | Number | Number | Numerical division. | `1.2 / 3 = 0.4`<br>`1 / 0 = Error` |
| **^ Power** | Number | Number | **Value 1** to the power **Value 2** (**Value 1**$^{\textbf{Value 2}}$). | `1.2 ^ 3 = 1.728`<br>`2 ^ -3 = 0.125` |
| **% Modulus** | Integer number | Integer number | Remainder after **Value 1** is divided by **Value 2.** | `7 % 2 = 1`<br>`7 % 0 = Error` |
| **Abs** | Number | N/A | The absolute value. | `Abs( 1.2 ) = 1.2`<br>`Abs( -1.2 ) = 1.2` |
| **And** | Boolean | Boolean | Logical AND. | `And( true, false ) = false`<br>`And( 1, 1 ) = true`<br>`And( 0, FALSE ) = false` |
| **Ceiling** | Number | N/A | The smallest integer that is not less than the value. | `Ceiling( 1 ) = 1`<br>`Ceiling( 1.2 ) = 2`<br>`Ceiling( -1.2 ) = -1` |
| **DateTimeToMSecs** | ISO datetime | Integer number | Convert an ISO datetime to the number of milliseconds since 1970-01-01:00:00:00.000 UTC. | `DateTimeToMSecs( 1970-01-01 ) = 0`<br>`DateTimeToMSecs( 2022-09-15T21:06:10 ) = 1663272370000`<br>`DateTimeToMSecs( 1969-12-30T13:42:23.211Z ) = -123456789`<br>`DateTimeToMSecs( 2022-09-16T13:00:02+01:00 ) = 1663329602000` |


© 2023 Oryx Digital Ltd

- Each row is calculated separately.
- Whether values are interpreted as number, dates or text depends on **Supported date formats** and locale.
- To concatenate text in different columns use the Concat Cols transform.
- To compare 2 columns use the Compare Cols transform.
- For logical operations use the If transform.
- For operations on whole rows or columns use the Stats transform.
- For more complex number/date calculations use the Javascript transform.
- To modify the numerical precision of the results use the Num Format transform.
- To add a new column of values use the New Col transform.

See also:
- Video: How to convert a Unix timestamp

### 2.3.3 Case

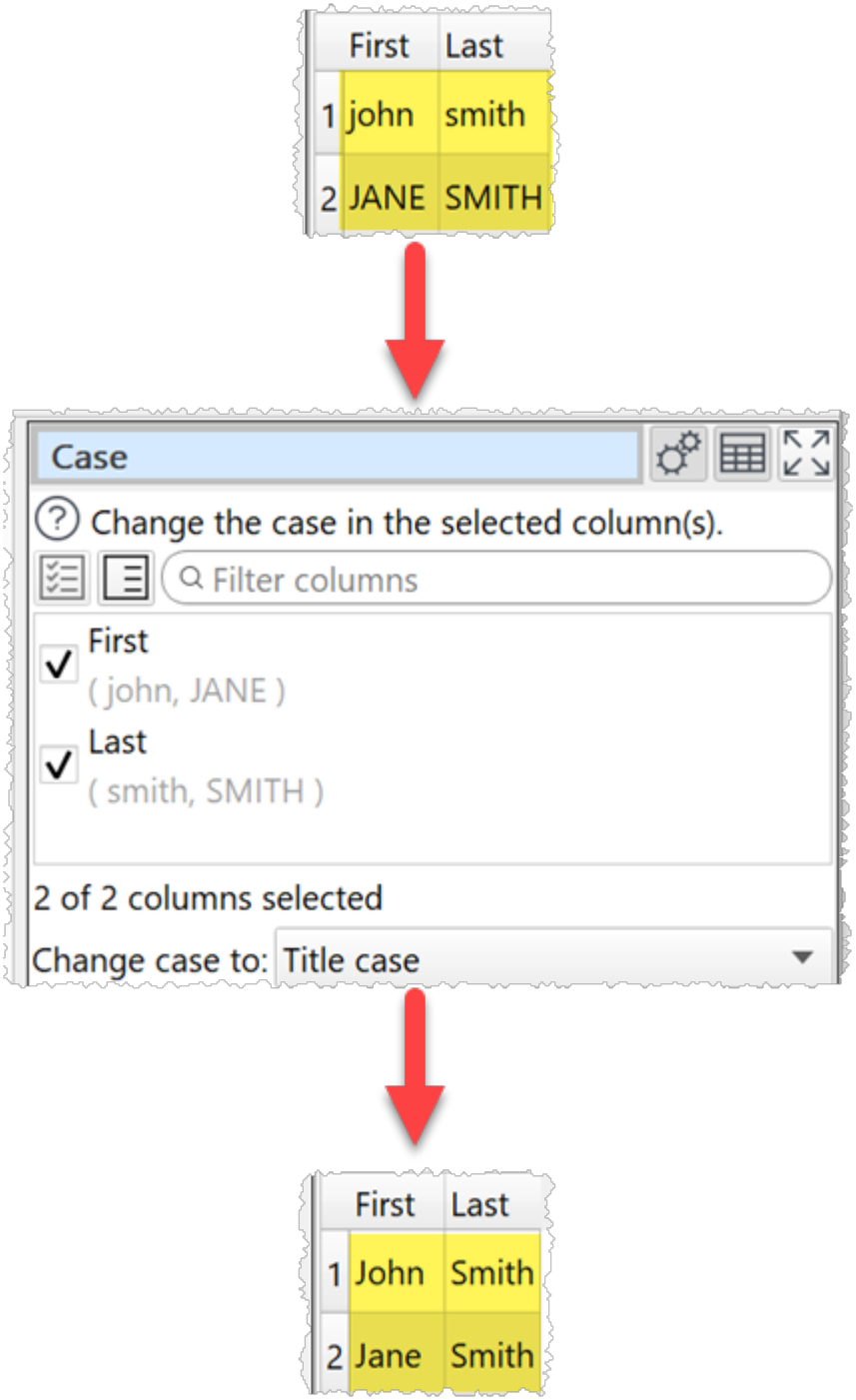
## Description

Changes the case of text in one or more columns.

## Example

Set names to title case:


© 2023 Oryx Digital Ltd




## Inputs

One.

## Options

- Check the column(s) you wish to transform.


*© 2023 Oryx Digital Ltd*

- Set **Change case to** to **Lower case** (e.g. "text"), **Upper case** (e.g. "TEXT") or **Title case** (e.g. "Text").

## See also

- Whitespace

### 2.3.4 Chop

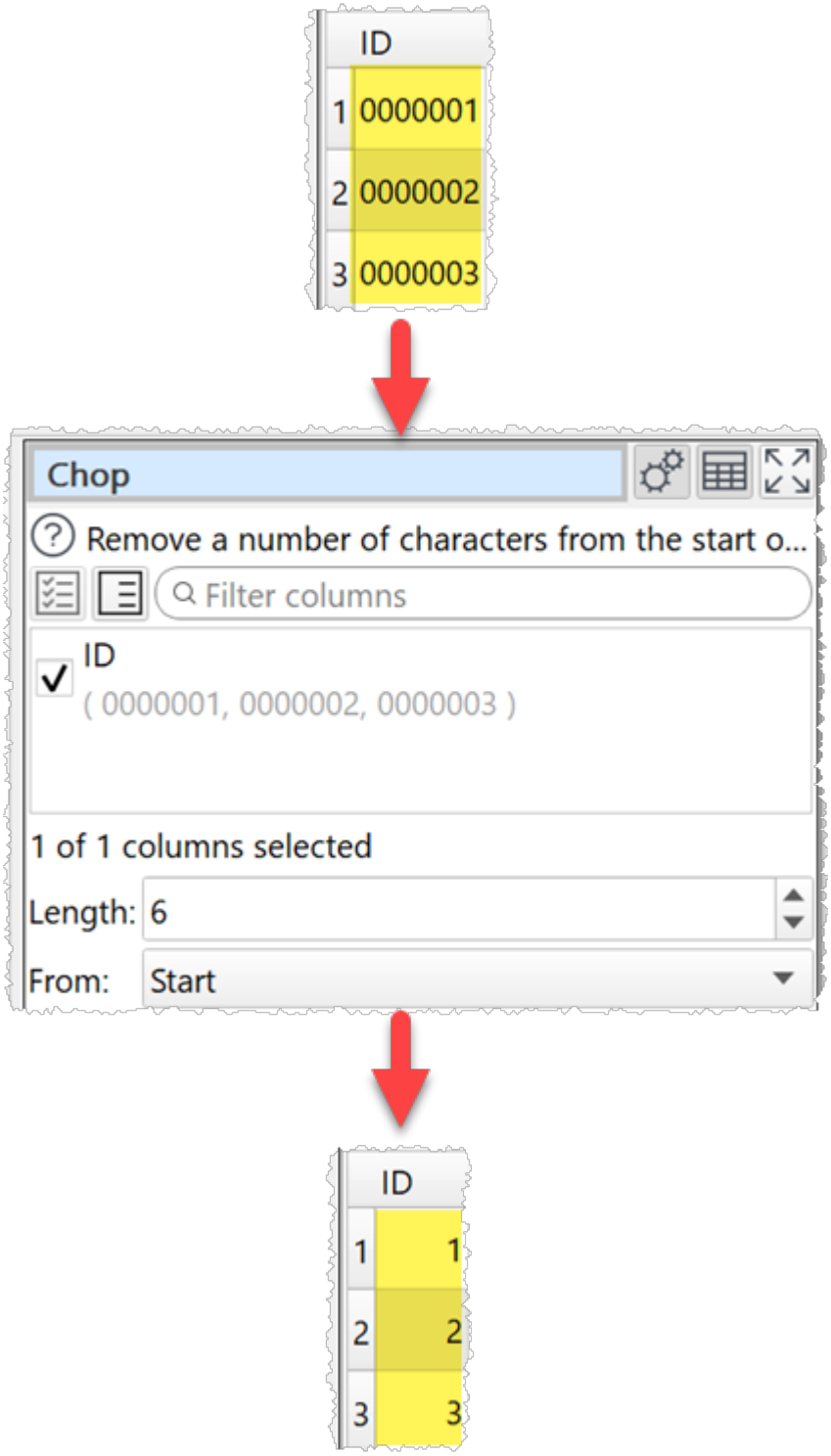
## Description

Remove characters from the start or end in one or more columns.

## Example

Remove leading zeros:


*© 2023 Oryx Digital Ltd*




## Inputs

One.


*© 2023 Oryx Digital Ltd*

### Options

- Check the column(s) you wish to transform.
- Set **Length** to the number of characters you want to remove.
- Set **From** to **Start** or **End** depending on whether you want to remove characters from the start or end.

### Notes

- If you can to keep the original column use Copy Cols to copy the column first.
- Whitespace is counted when calculating length. You can use Whitespace to remove whitespace before chopping.
- If you want to set a column to a fixed length use Pad and Chop together.

### See also

- Extract

## 2.3.5 Clone

### Description

Makes an exact copy of the input dataset.

### Inputs

One.

### Options

- None.

### Notes

- Clone can be useful to simplify complicated layouts.

## 2.3.6 Cluster

### Description

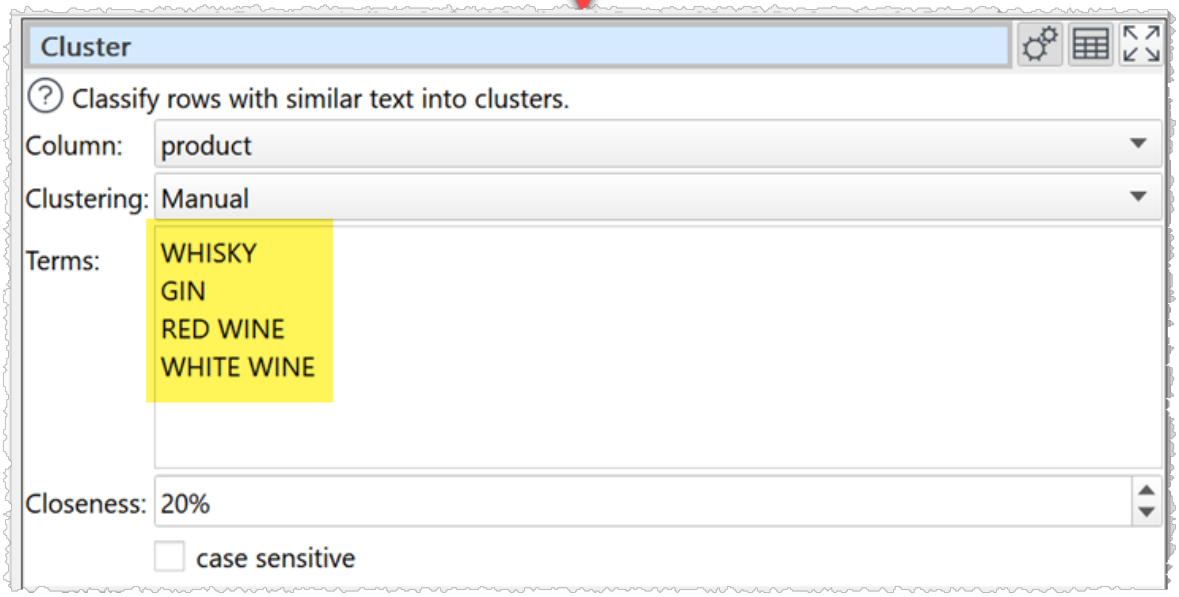
Classify rows with similar text into clusters

### Examples

Clustering products by type:


© 2023 Oryx Digital Ltd

| | product | unit cost | stock |
|---|---|---|---|
| 1 | French red wine | 23.95 | 9 |
| 2 | BULGARIAN RED WN | 11.95 | 13 |
| 3 | Red Label Whisky | 38.45 | 1 |
| 4 | Gordon's Gin | 25 | 7 |
| 5 | German White Wine | 14.95 | 6 |
| 6 | Larios Gin | 19.90 | 50 |
| 7 | Black Label Whisky | 45.50 | 8 |
| 8 | Wine, White, Dessert | 30.00 | 1 |
| 9 | Cocoa Cola | 1.95 | 200 |

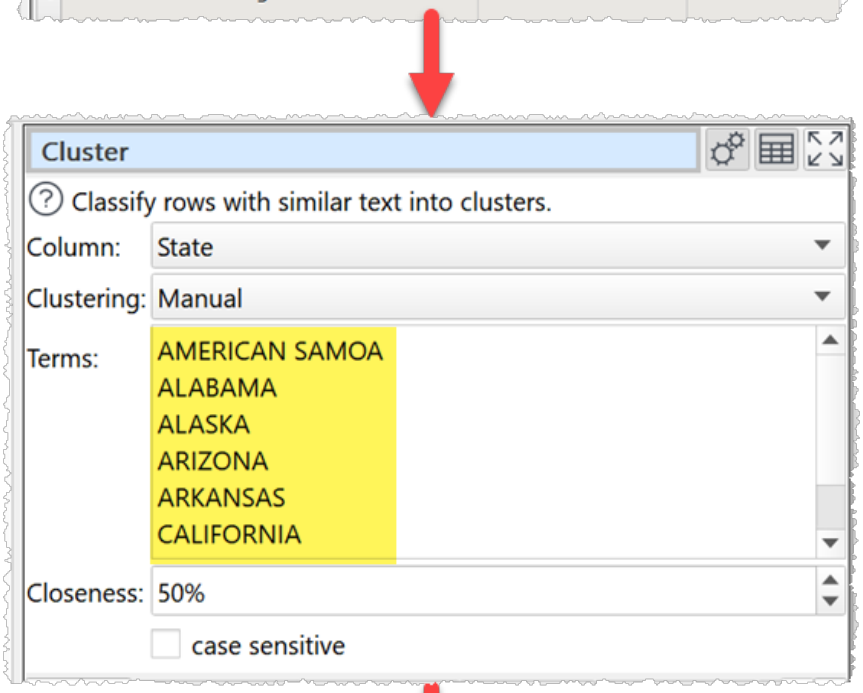



| | product | unit cost | stock | Cluster | Closeness |
|---|---|---|---|---|---|
| 1 | French red wine | 23.95 | 9 | RED WINE | 53 |
| 2 | BULGARIAN RED WN | 11.95 | 13 | RED WINE | 25 |
| 3 | Red Label Whisky | 38.45 | 1 | WHISKY | 37 |
| 4 | Gordon's Gin | 25 | 7 | GIN | 25 |
| 5 | German White Wine | 14.95 | 6 | WHITE WINE | 58 |
| 6 | Larios Gin | 19.90 | 50 | GIN | 30 |

Correcting the state in an address:


© 2023 Oryx Digital Ltd

| | Street | City | State |
|---|---|---|---|
| 1 | 589 Timbercrest Road | Haines | Alaska |
| 2 | 1289 Jerry Toth Drive | | ALASKA |
| 3 | 3597 Blackwell Street | Anchorage | Alaksa |
| 4 | 344 George Avenue | Mobile | Alabam |
| 5 | 2555 Lonely Oak Drive | Mobile | ALABAMA |
| 6 | 2816 Barrington Court | Pine Bluff | arknsas |




| | Street | City | State | Cluster | Closeness |
|---|---|---|---|---|---|
| 1 | 589 Timbercrest Road | Haines | Alaska | ALASKA | 100 |
| 2 | 1289 Jerry Toth Drive | | ALASKA | ALASKA | 100 |
| 3 | 3597 Blackwell Street | Anchorage | Alaksa | ALASKA | 66 |
| 4 | 344 George Avenue | Mobile | Alabam | ALABAMA | 85 |
| 5 | 2555 Lonely Oak Drive | Mobile | ALABAMA | ALABAMA | 100 |
| 6 | 2816 Barrington Court | Pine Bluff | arknsas | ARKANSAS | 87 |


*© 2023 Oryx Digital Ltd*

## Inputs
One.

## Options
- Select the **Column** whose values you wish to use for clustering.
- Select **Clustering** as:
  - **Manual** to cluster by **Terms** you provide (one per line).
  - **Guided** to find clustering terms automatically, starting with the **Terms** you provide (one per line).
  - **Automatic** to find clustering terms automatically.
- Set the minimum **Closeness** percentage for a value to a term to be clustered with a term.
- Set **Max clusters** to the maximum number of clustering terms (**Guided** and **Automatic** only).
- Set **Max time** to the maximum amount of time allowed for trying to improve the terms (**Guided** and **Automatic** only).
- Uncheck **case sensitive** to ignore case.

## Notes
- Fuzzy matching is used to classify each row value according to which of the user supplied **Terms** it is closest to.
- Additional **Cluster** and **Closeness** columns are added. The **Cluster** column shows the closest matching of the **Terms** (or the one highest one in the **Terms** list if 2 or more match equally). The **Closeness** column shows the fuzzy match score between 0 (no match) and 100 (exact match).
- Row values that do not meet the minimum **Closeness** score for any term are set to `<None>`.
- Set the minimum **Closeness** score to 0% to force all values to be assigned.
- You can use a Filter to list all the rows with `<None>` in the **Cluster** column to guide you on any additional **Terms** you might want to add in **Manual** mode.

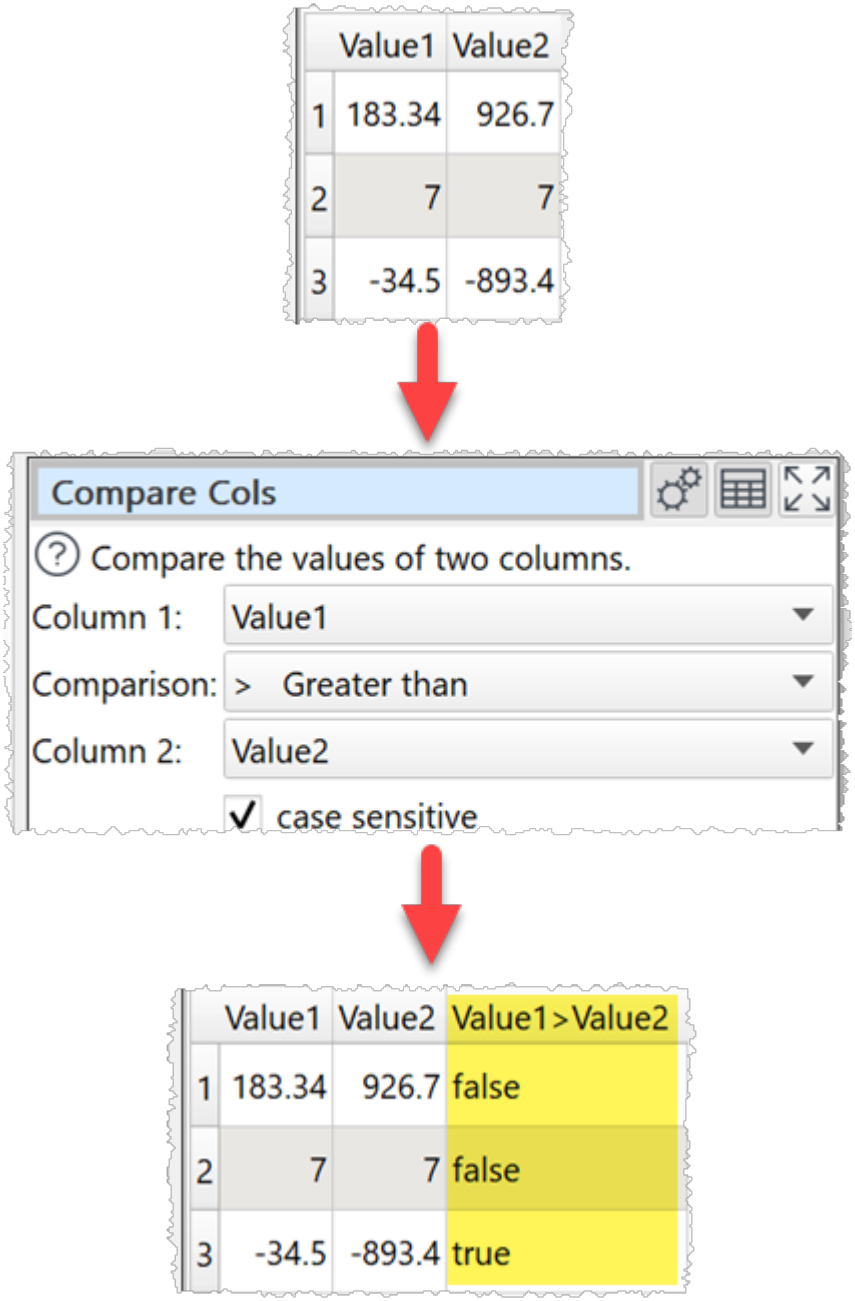
See also:
- Ngram
- Video: How to cluster text

### 2.3.7 Compare Cols

## Description
Creates a new column with a comparison of two other columns.

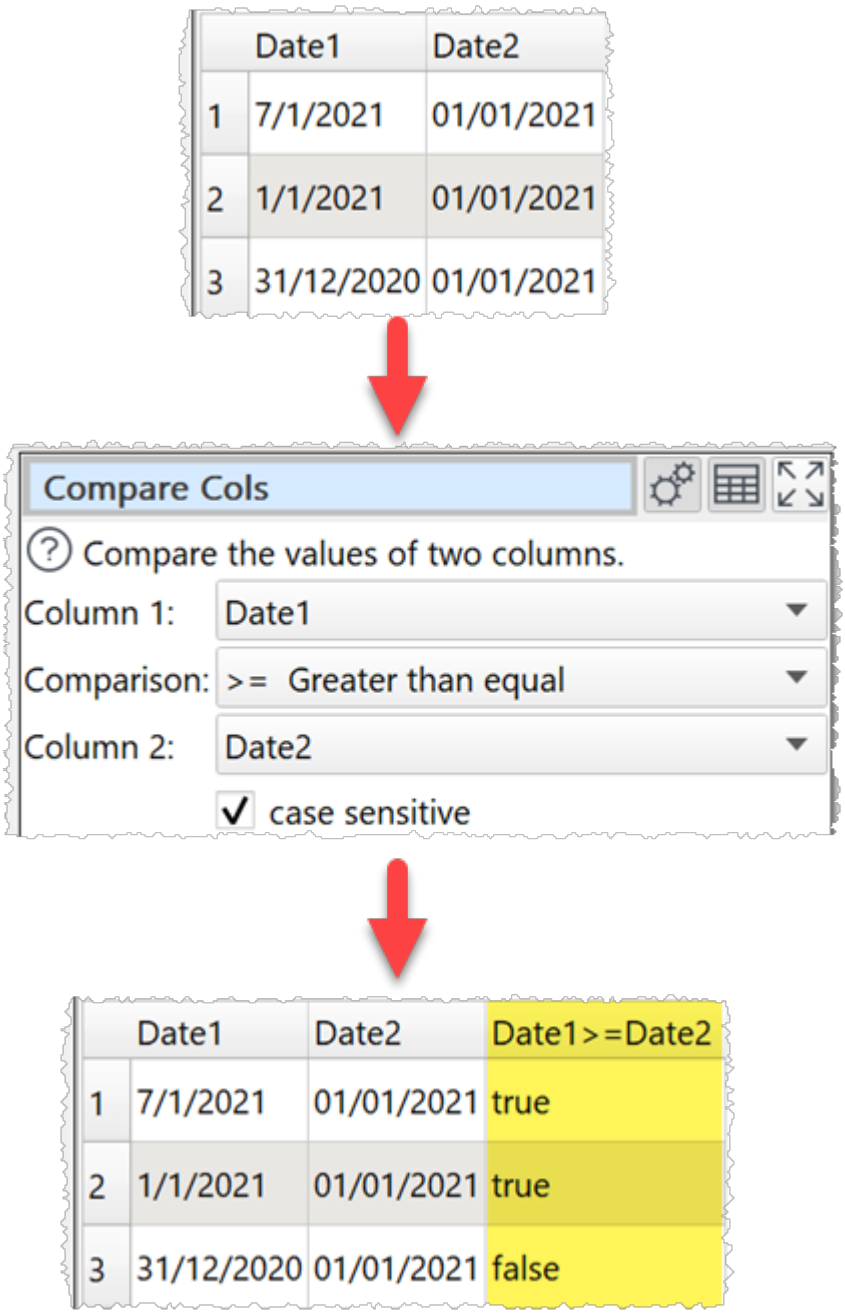
## Examples
Compare 2 numerical columns:


© 2023 Oryx Digital Ltd




Compare 2 date columns:


*© 2023 Oryx Digital Ltd*




## Inputs

One.

## Options

- Select the two columns you wish to compare as **Column 1** and **Column 2** and the **Comparison** operator.
- Check **case sensitive** to use case sensitive matching for text.


*© 2023 Oryx Digital Ltd*

## Notes

- Number, date and text values are treated differently for **Equal to**, **Greater than**, **Less than**, **Greater than equal, Less than equal** and **Not equal to** operations.
  - If both values are numeric, a numeric comparison will be carried out.
  - If both values match a supported date formats in Preferences, a date comparison will be carried out.
  - Otherwise the values will be treated as text.
  - For example, an empty value is considered less than 0, because they will be compared as text. So you might want to replace empty values with 0 or remove those rows with Filter, before comparing them.
- Comparisons of text are whitespace sensitive. You can use Whitespace to remove whitespace before filtering and Replace to get of other unwanted characters (e.g. whitespace inside the text).
- The new column is added at the right end. You can change the column order with Reorder Cols and the column name with Rename Cols.
- See here for more details on Regular expressions (regex).

## See also

- Split Cols

### 2.3.8 Concat Cols

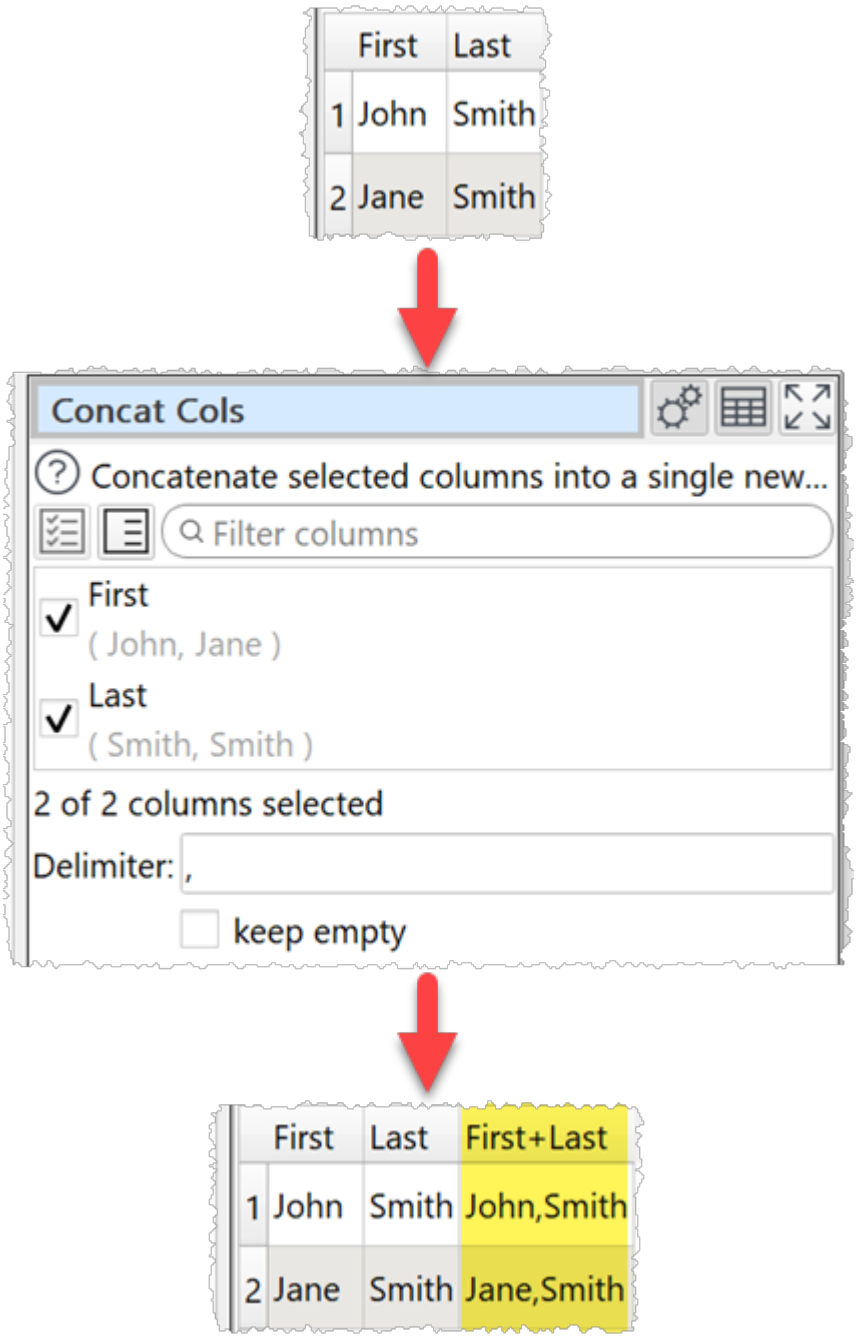
## Description

Creates a new column by concatenating text from existing columns.

## Example

Concatenate 'First' and 'Last' columns with a comma between:


*© 2023 Oryx Digital Ltd*




## Inputs

One.

## Options

- Check the columns you wish to concatenate.
- Supply the **Delimiter** you wish to place between concatenated text (optional). For example ",".
- Check **keep empty** if you wish to keep the delimiter for empty columns.


*© 2023 Oryx Digital Ltd*

## Notes

- If there is a header, the header of the new column is formed from the header of the concatenated columns. You can use Rename Cols to change the new column name.
- Concatenating a single column makes a copy of the column.
- The values in the column are in the order of the original columns. You can change the column order before concatenation with Reorder Cols.
- The new column is added at the right end. You can change the column order with Reorder Cols and the column name with Rename Cols.
- The opposite of **Concat Cols** is Split Col.

## See also

- Concat Rows
- Substitute

### 2.3.9 Concat Rows

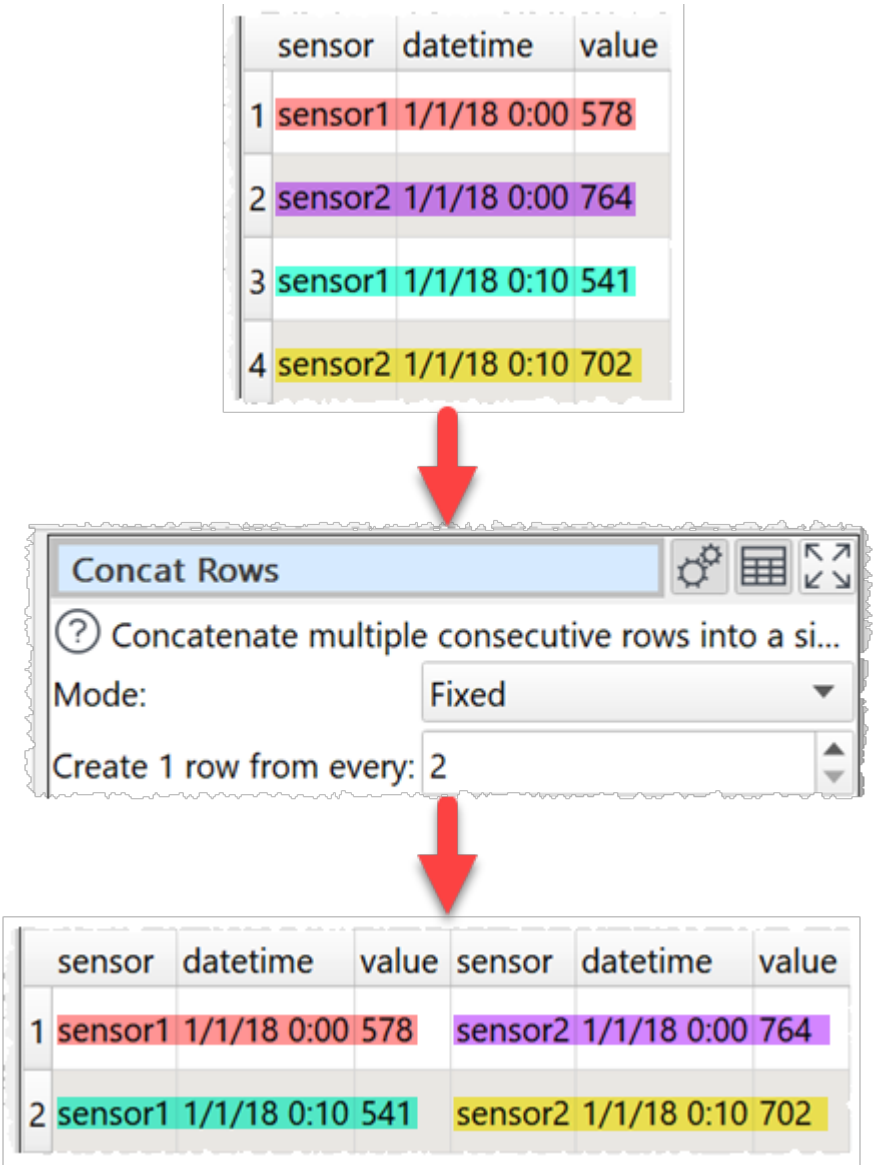
## Description

Concatenate multiple consecutive rows into a single row.

## Examples

Concatenate every 2 rows to 1:

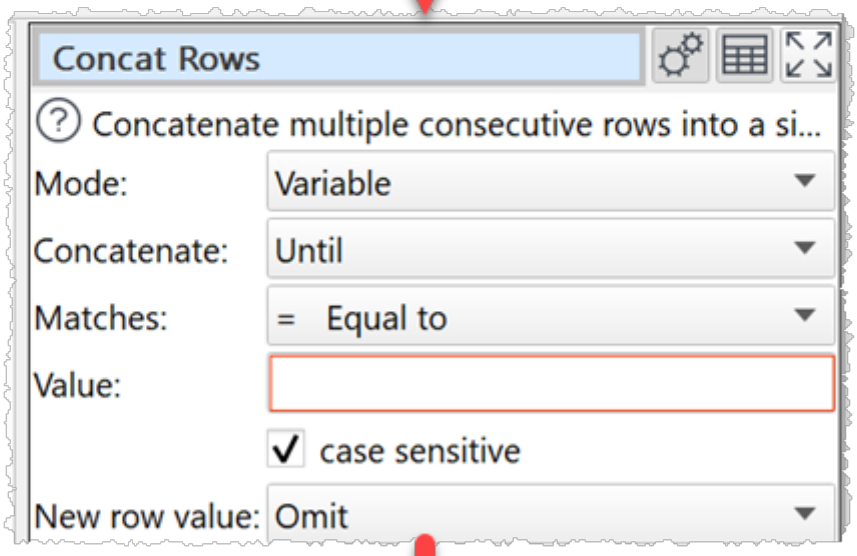

*© 2023 Oryx Digital Ltd*




Concatenate values until empty:


*© 2023 Oryx Digital Ltd*

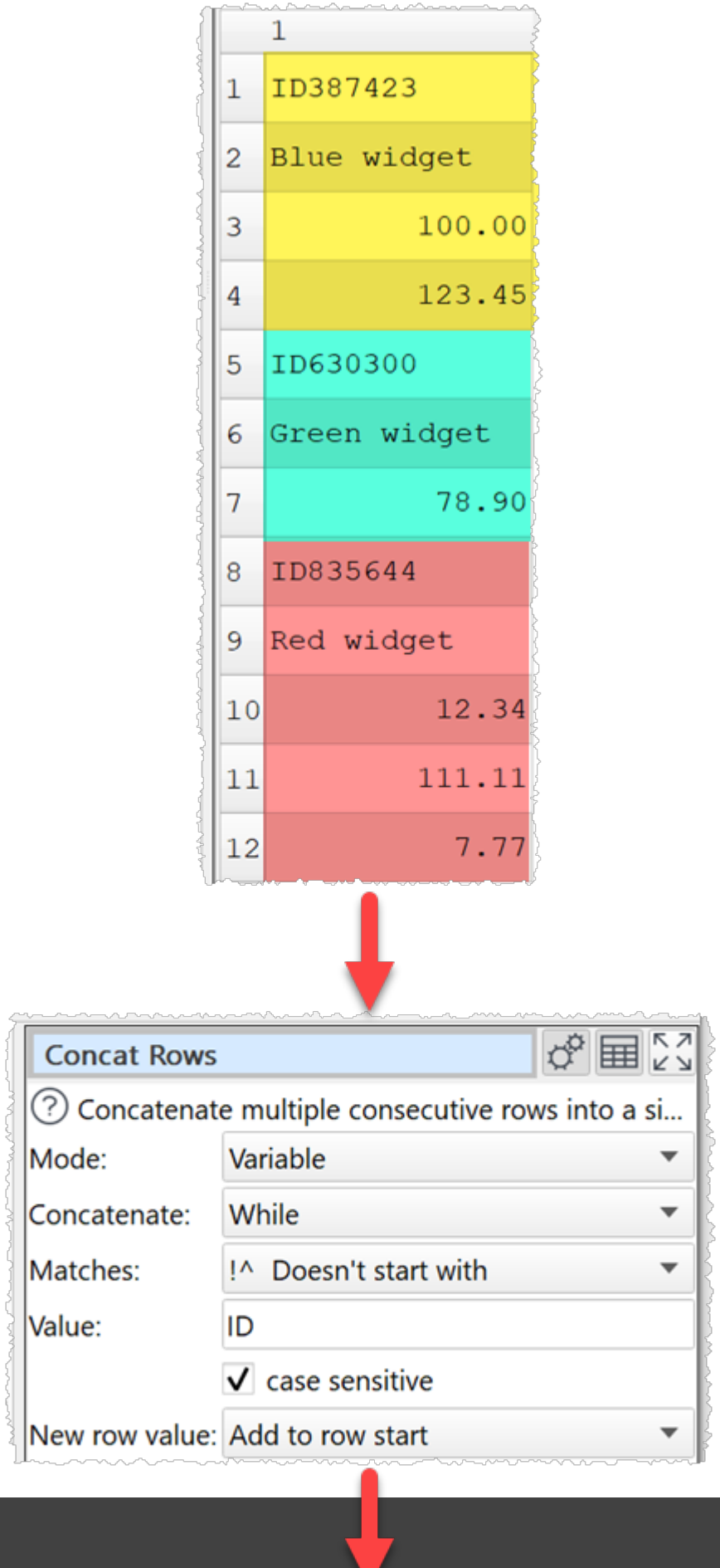
| | 1 |
|---|---|
| 1 | John Smith |
| 2 | 1 The Street |
| 3 | Townsville |
| 4 | Wiltshire |
| 5 | |
| 6 | Jane Brown |
| 7 | 7 The Avenue |
| 8 | Cityville |
| 9 | |




| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | John Smith | 1 The Street | Townsville | Wiltshire |
| 2 | Jane Brown | 7 The Avenue | Cityville | |

Concatenate values while not starting with 'ID':

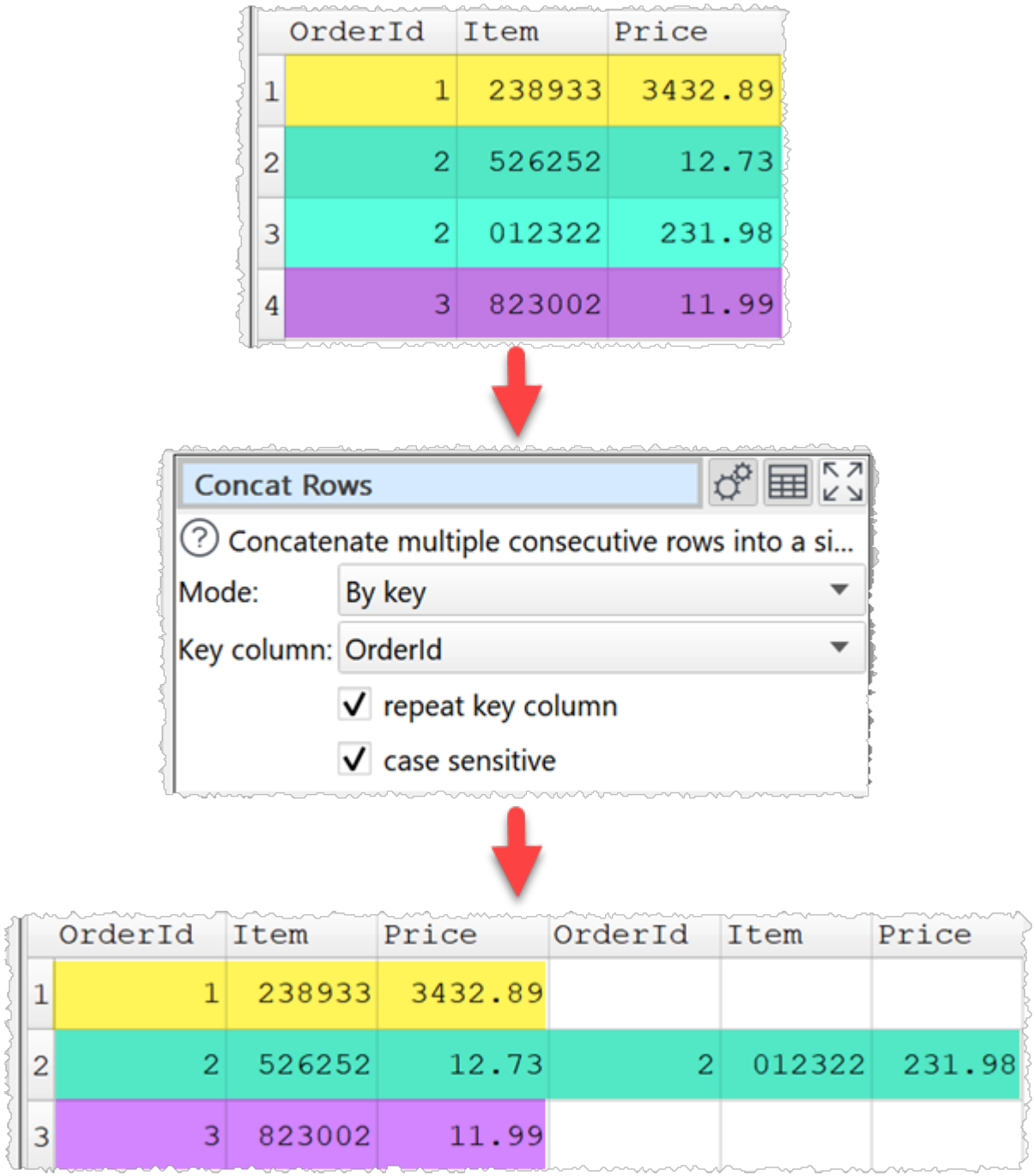

© 2023 Oryx Digital Ltd





© 2023 Oryx Digital Ltd

Concatenate consecutive rows with the same OrderId:




## Inputs

One.


*© 2023 Oryx Digital Ltd*

## Options

- Set **Mode** to:
  - **Fixed** to concatenate a fixed number of rows from the current dataset to make each new row.
  - **Variable** to concatenate values while or until a match to make each new row.
  - **By key** to concatenate consecutive rows with the same value in a key column to make each new row.
- Set **Create 1 row from every** to N, to concatenate every N rows into 1 row (**Mode**=**Fixed** only).
- Set **Concatenate** to **Until** or **While** to keep concatenating until/while a match is found for **Matches** and **Value** (**Mode**=**Variable** only).
- Check **case sensitive** to use case sensitive matching for text (**Mode**=**Variable** or **By Key** only).
- Set **New row value** depending on what you want to do with the value that denotes a new row (**Mode**=**Variable** only).
- Uncheck **repeat key column** to keep only the first key column (**Mode**=**By Key** only).

## Notes

- Use New Col if you need to add additional columns before concatenating rows.
- Use Filter if you need to remove rows before concatenating row.
- Use Rename Cols if you need to change column names after concatenating rows.
- The opposite of **Concat Rows** is Split Rows.
- Row order is important. Use Sort if you need to change the order.

## See also

- Spread
- Unique
- Concat Cols

### 2.3.10 Copy Cols

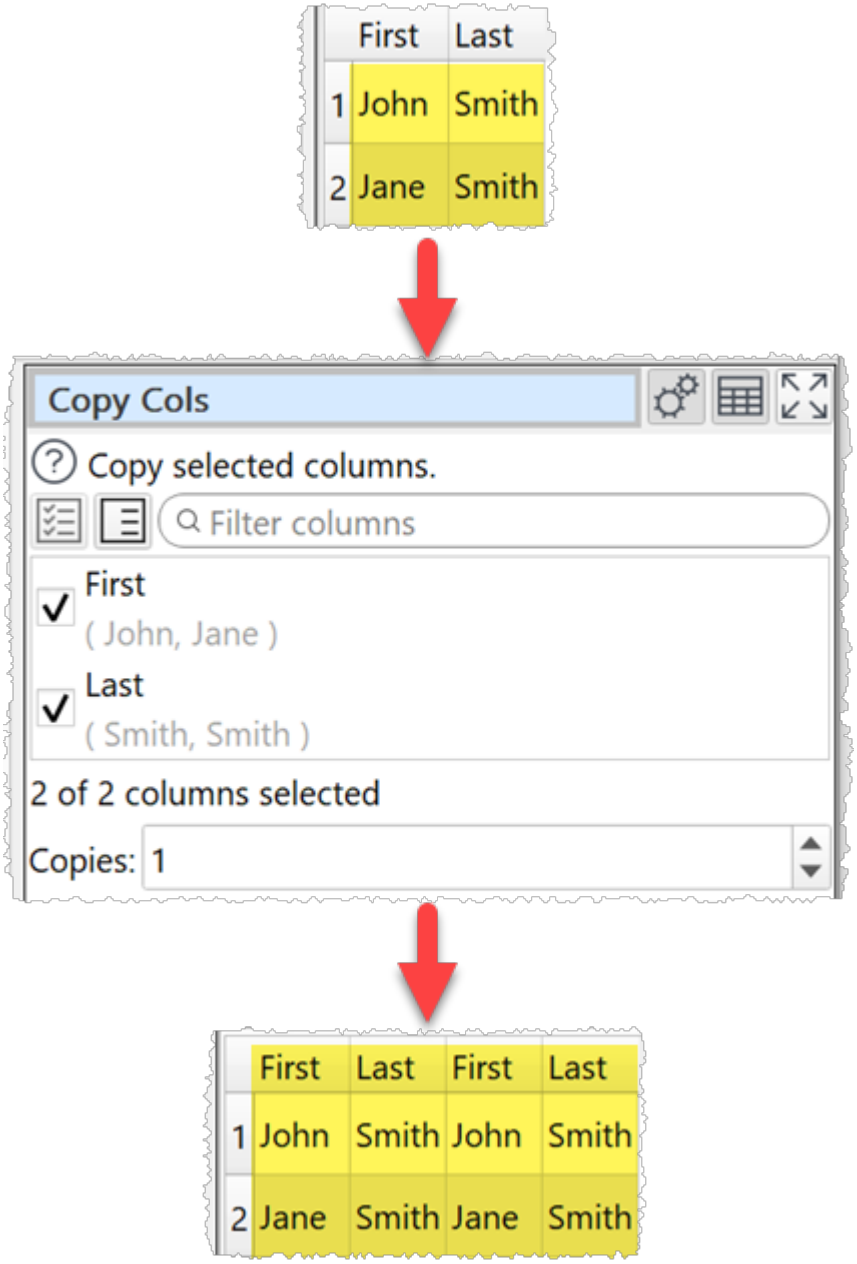
## Description

Creates one or more copies of the selected column(s).

## Example

Copy 2 columns, once each:


© 2023 Oryx Digital Ltd




## Inputs

One.

## Options

- Check the columns you wish to copy.
- Set **Copies** to the number of copies you want to make of each checked column.


*© 2023 Oryx Digital Ltd*

### Notes

- If there is a header, the header of each new column is the original column name. You can rename columns with Rename Cols.
- The new columns are added at the right end. You can change the column order with Reorder Cols.

### See also

- New Col

## 2.3.11 Count

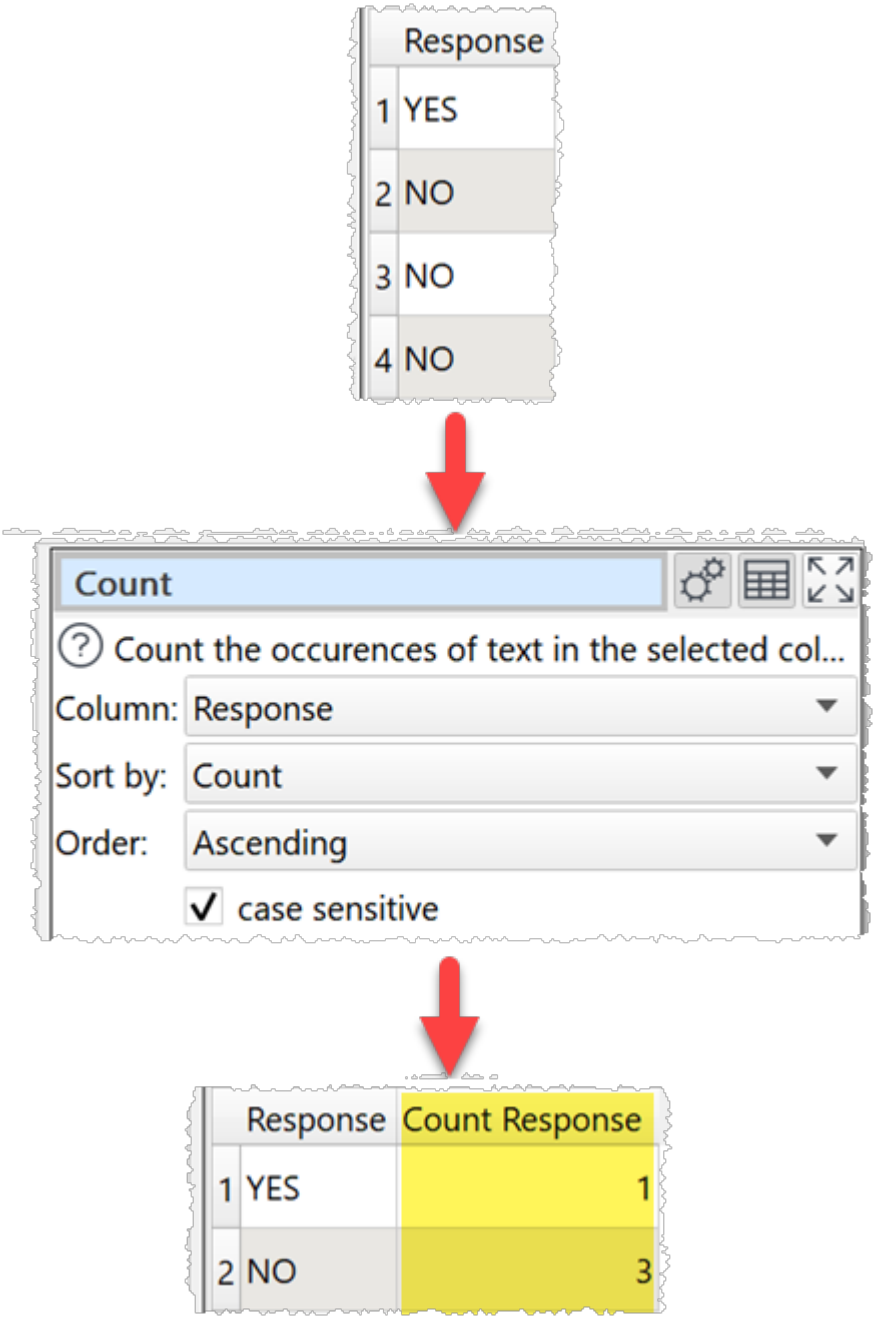
### Description

Counts the number of occurrences of each item of text in the selected column.

### Example

Count unique values in the 'Response' column:


© 2023 Oryx Digital Ltd




## Inputs

One.

## Options

- Select the **Column** whose values you wish to count.
- Set **Sort by** depending on whether you wish to sort alphabetically by the **Text** in the left column or numerically by the **Count** in the right column.
- Set **Order** depending on whether you wish to sort **Ascending** or **Descending**.


*© 2023 Oryx Digital Ltd*

- Uncheck **case sensitive** to convert everything to lower case before counting.
- Check **drilldown** to allow double-clicking a row in the data table to drilldown to the rows that contributed to this value in the upstream data.

## Notes

- Date and number values are treated as text.
- Use Rename Cols to change the new column name.
- Use Scale to convert the results to percentages.

## See also

- Ngram
- Unique
- Total
- Pivot
- Stats
- Summary

### 2.3.12 Cross

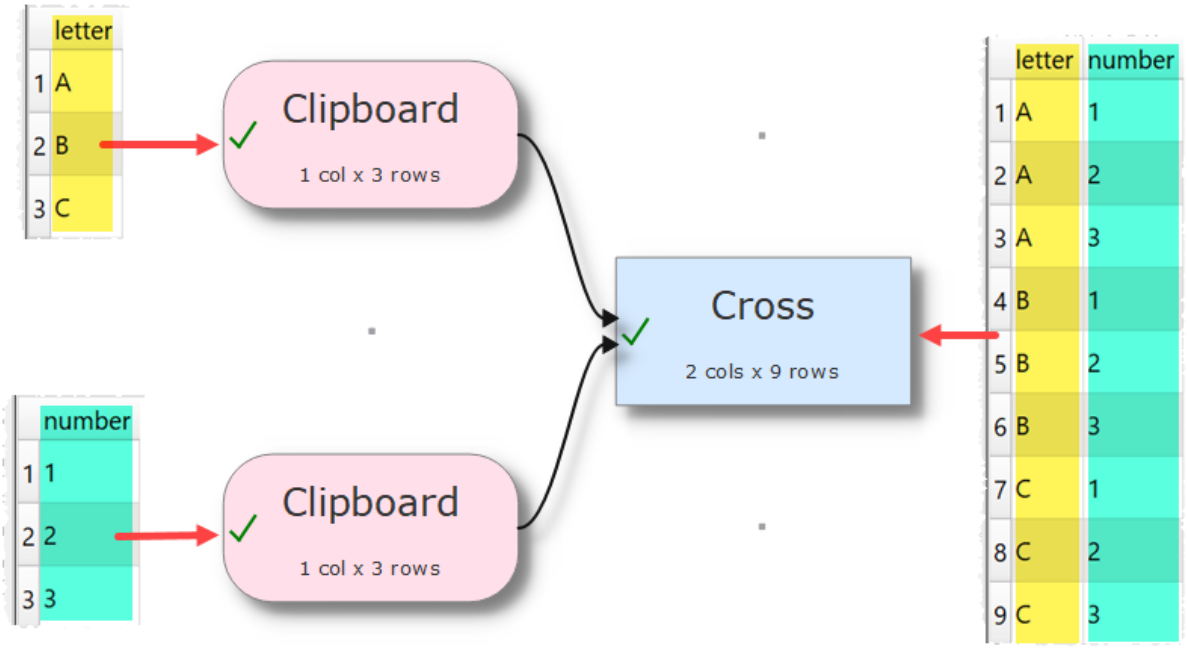
## Description

Creates an output from combining every possible row combination of each input. E.g. if the first input has N1 rows and the second input has N2 rows, then the result will have N1 X N2 rows. Also known as a 'Cartesian product' or 'cross join'.

## Example

Cross 2 small datasets:


© 2023 Oryx Digital Ltd




## Inputs

Two or more.

## Options

- The output depends on the vertical (Y-axis) position of the inputs.

## Notes

- It can create a very large number of rows!

## See also

- Join
- Stack

### 2.3.13 DateTime Format

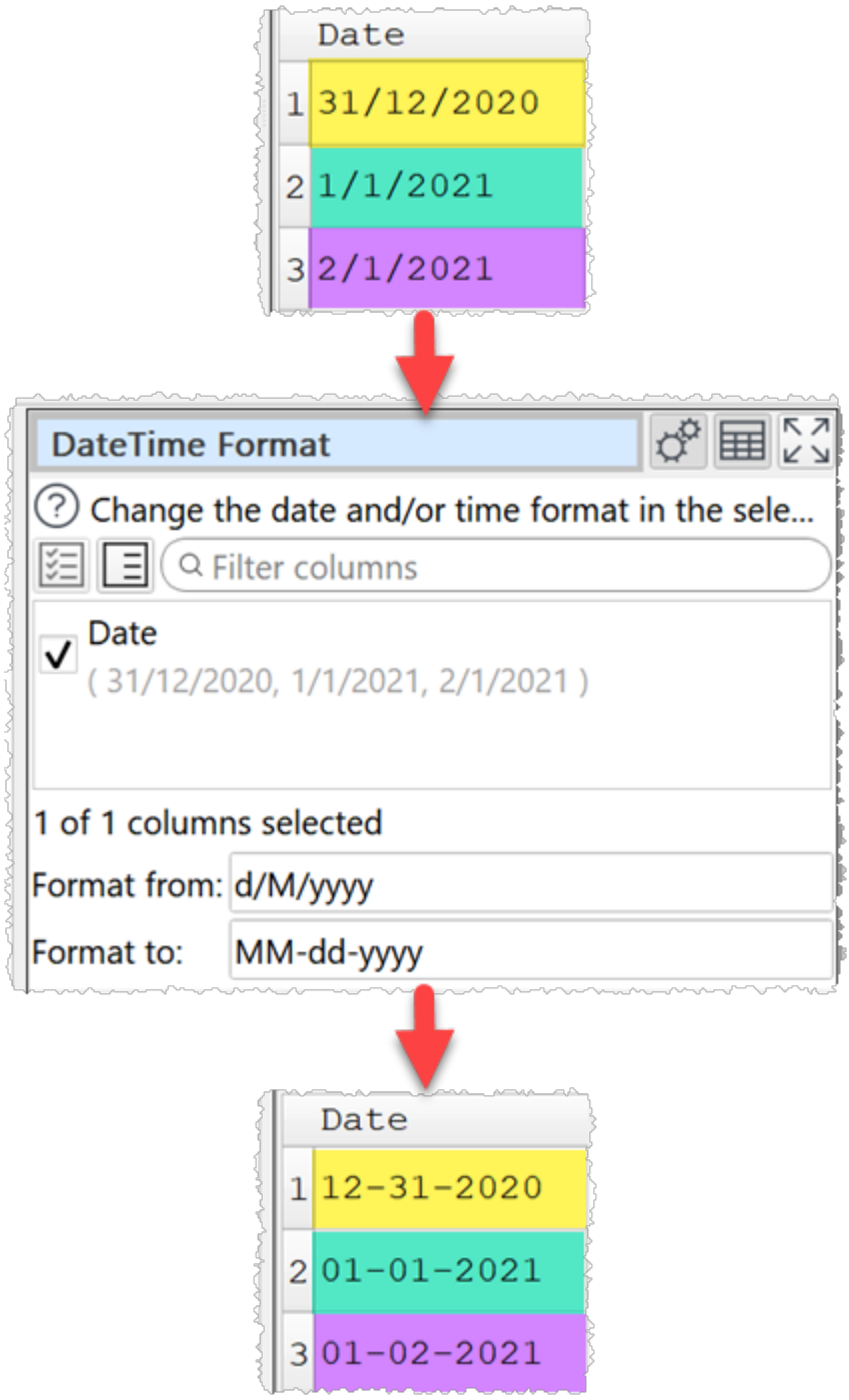
## Description

Changes the date and/or time format in one or more columns.

## Examples

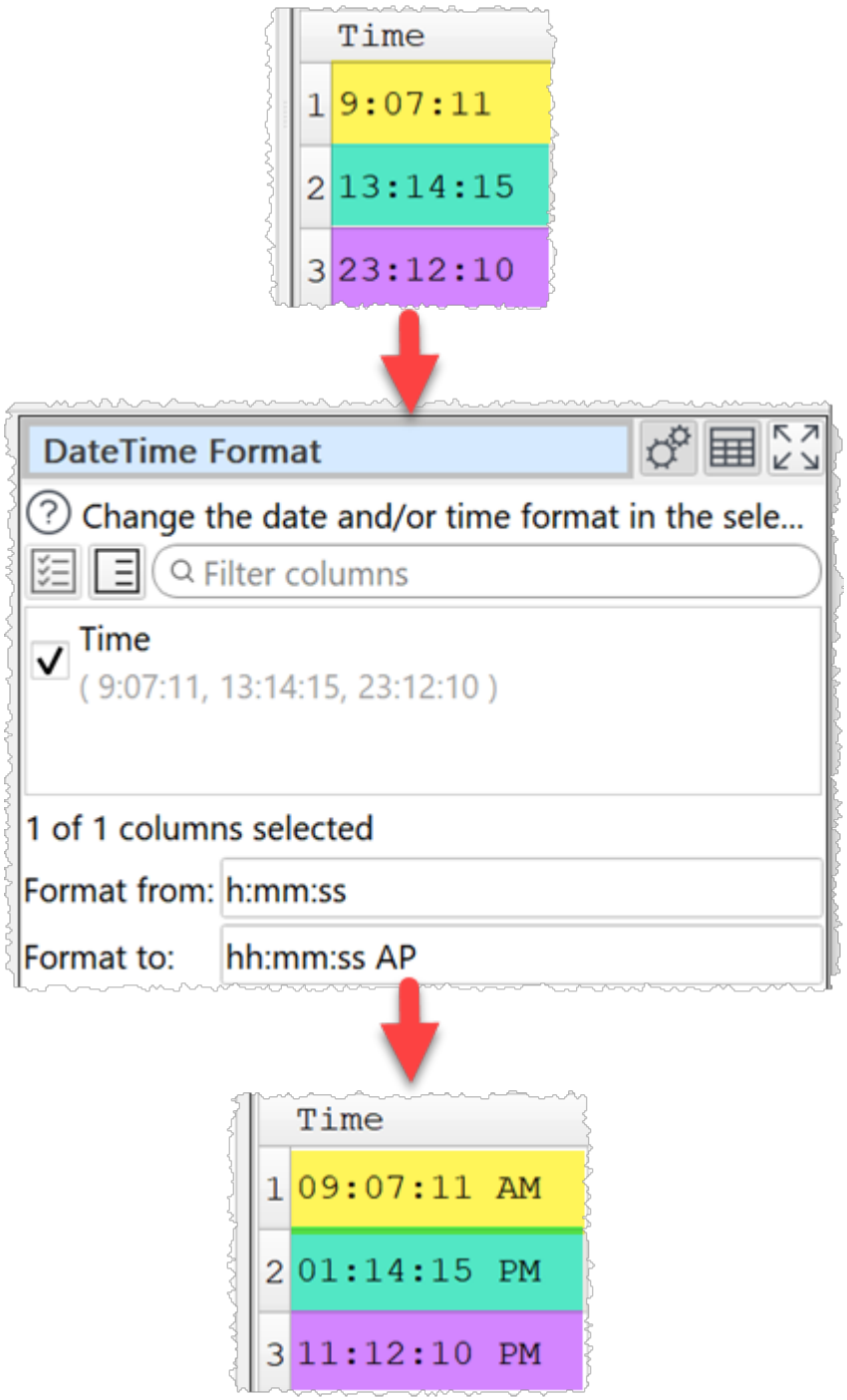
Change day/month/year to month-day-year:


*© 2023 Oryx Digital Ltd*



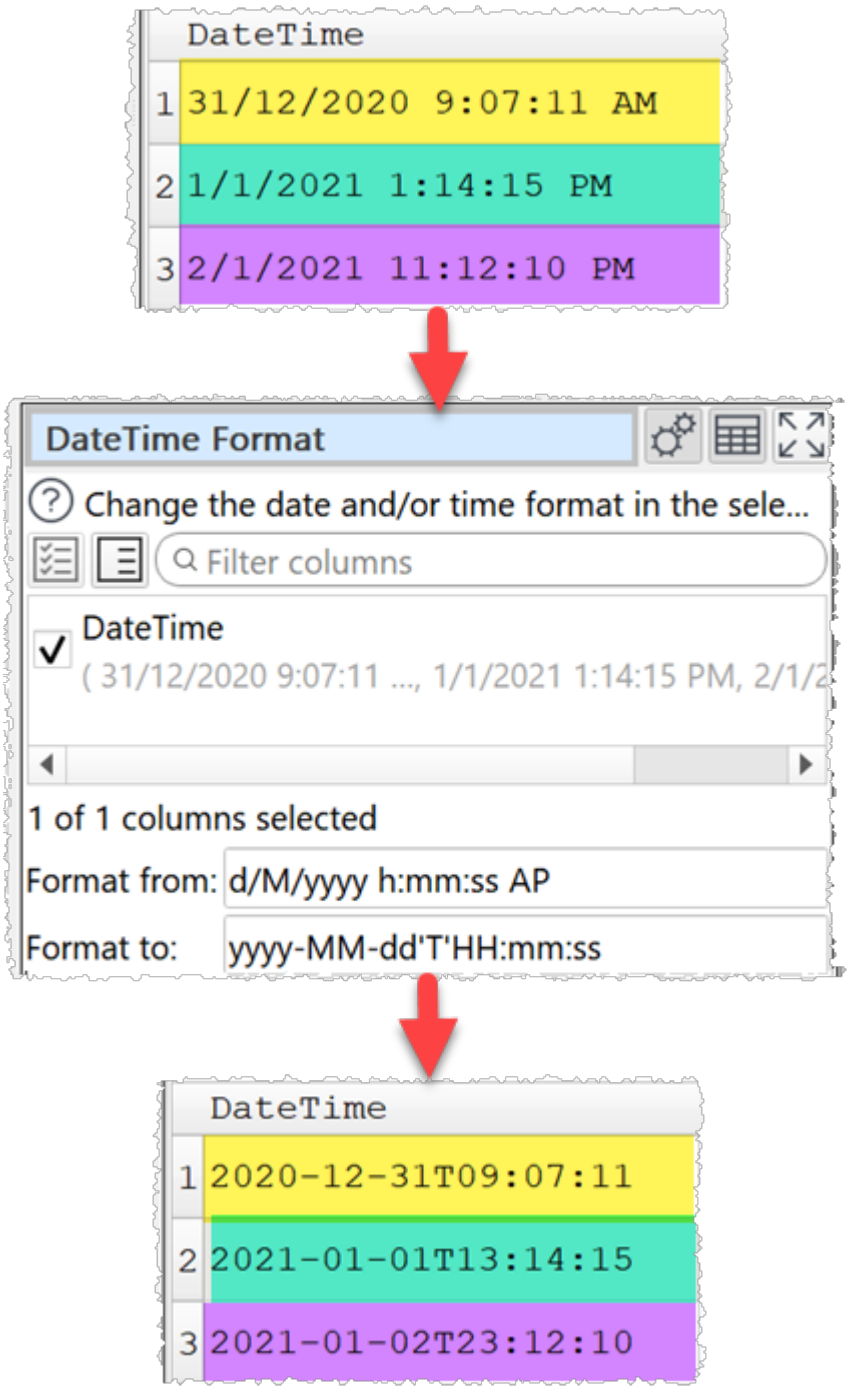

Change 24 hour time to AM/PM time:


*© 2023 Oryx Digital Ltd*




Change datetime to ISO format:


*© 2023 Oryx Digital Ltd*




Example formats for 21 May 2001 14:13:09.120 with locale set to English/United States:


*© 2023 Oryx Digital Ltd*

| Format to | Output |
|---|---|
| dd.MM.yyyy | 21.05.2001 |
| yyyy-d-M | 2001-21-5 |
| ddd MMMM d yy | Mon May 21 01 |
| hh:mm:ss.zzz | 14:13:09.120 |
| hh:mm:ss.z | 14:13:09.12 |
| hhmmss | 141309 |
| h:m:s ap | 2:13:9 pm |
| yyyy-MM-dd'T'HH:mm:ss.zzz | 2001-05-21T14:13:09.120 |
| HH:mm:ss 'in yyyy-MM-dd HH:mm:ss format' | 14:13:09 in HH:mm:ss format |

## Inputs

One.

## Options

- Check the columns you wish to transform.
- Supply the existing date and/or time format in **Format from** (see below).
- Supply the new date and/or time format in **Format to** (see below).
- The following date and/or time formats are supported for input and output:


*© 2023 Oryx Digital Ltd*

| Relates to | | Format | Meaning |
|---|---|---|---|
| Date | Years | `yy` | The year as a two digit number (00 to 99). |
| | | `yyyy` | The year as a four digit number. If the year is negative, a minus sign is prepended in addition. |
| | Months | `M` | The month as number without a leading zero (1 to 12). |
| | | `MM` | The month as number with a leading zero (01 to 12) |
| | | `MMM` | The abbreviated localized month name (e.g. 'Jan' to 'Dec'). Uses the locale to localize the name. |
| | | `MMMM` | The long localized month name (e.g. 'January' to 'December'). Uses the locale to localize the name. |
| | Days | `d` | The day as number without a leading zero (1 to 31). |
| | | `dd` | The day as number with a leading zero (01 to 31). |
| | | `ddd` | The abbreviated localized day name (e.g. 'Mon' to 'Sun'). Uses the locale to localize the name. |

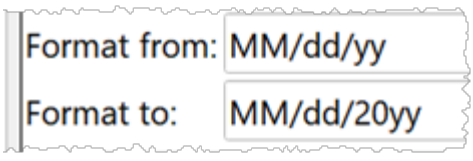

© 2023 Oryx Digital Ltd

If not set, the default values are:

| Value | Default |
|---|---|
| Year | 1900 |
| Month | 1 |
| Day | 1 |
| Hour | 0 |
| Minute | 0 |
| Second | 0 |
| Milliseconds | 0 |

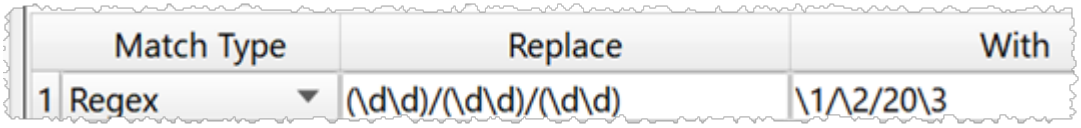
## Notes

- An exact match is expected for the input date. E.g. **Format from** is `yyyy-MM-dd` then `2020-12-17` will be converted, but `2020-12-17T14:58` won't. You can extract just the date or time part here using Split Col with `T` as the delimiter.
- Use the correct case for the format (e.g. `yyyy-MM-dd`, not `YYYY-mm-DD`).
- The locale is used to decide how the date is represented (e.g. names of months and days).
- Any non-empty sequence of characters enclosed in single quotes will be included verbatim in the output string (stripped of the quotes), even if it contains formatting characters. Two consecutive single quotes (`''`) are replaced by a single quote in the output. All other characters in the format string are included verbatim in the output string.
- Formats without separators (e.g. `ddMM`) are supported but must be used with care, as the resulting strings aren't always reliably readable (e.g. if `dM` produces `212` it could mean either the 2nd of December or the 21st of February).
- You can also use Split Col to split a date into its component parts. For example to split "31/1/2019" into day, month and year components using the "/" delimiter.
- If the date to be converted has only two year digits, it is treated as a date between 1900 and 1999. E.g. "31/1/19" is interpreted in d-M-yy format as 31st January 1919. You can avoid that issue by explicitly setting the century, e.g.:




Or you can use Replace. For example to replace `01/15/18` with `01/15/2018`:


© 2023 Oryx Digital Ltd



| | Match Type | Replace | With |
|---|---|---|---|
| 1 | Regex | (\d\d)/(\d\d)/(\d\d) | \1/\2/20\3 |

- Warnings are shown in the **Warnings** tab for values that don't match **Format from**.
- You can also do date and time format conversion using the Javascript transform.

## See also

- Num Format

### 2.3.14 Decode

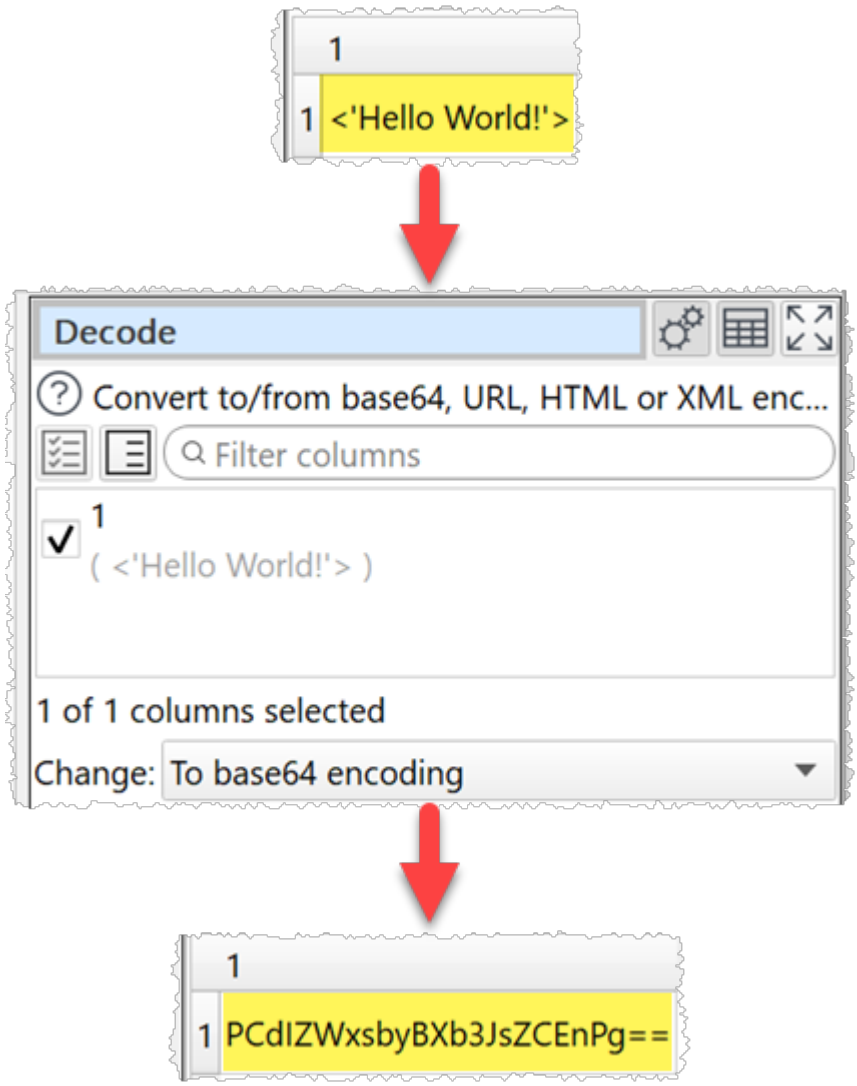
## Description

Convert to/from Base64 encoding, URL encoding, escaped HTML or escaped XML.

## Example

Convert `<'Hello World!'>` to base64 encoding:


© 2023 Oryx Digital Ltd




| Type | Plaintext | Encoded |
|---|---|---|
| Based64 encoding | `<'Hello World!'>` | `PCdIZWxsbyBXb3JsZCEnPg==` |
| URL encoding | `<'Hello World!'>` | `%3C%27Hello%20World%21%27%3E` |
| HTML escaping | `<'Hello World!'>` | `<'Hello World!'>` |
| XML escaping | `<'Hello World!'>` | `<'Hello World!'>` |

## Inputs

One.


*© 2023 Oryx Digital Ltd*

### Options

- Check the column(s) you wish to transform.
- Set **Change** according to how you want to decode/encode the selected columns.

### Notes

- This is distinct from text encoding (e.g. UTF-8 vs UTF-16) which is handled on input and output.
- Use Copy Cols to copy columns before you transform them.
- Data input from XML files and output to HTML and XML files is unescaped/escaped automatically.

## 2.3.15 Dedupe

### Description

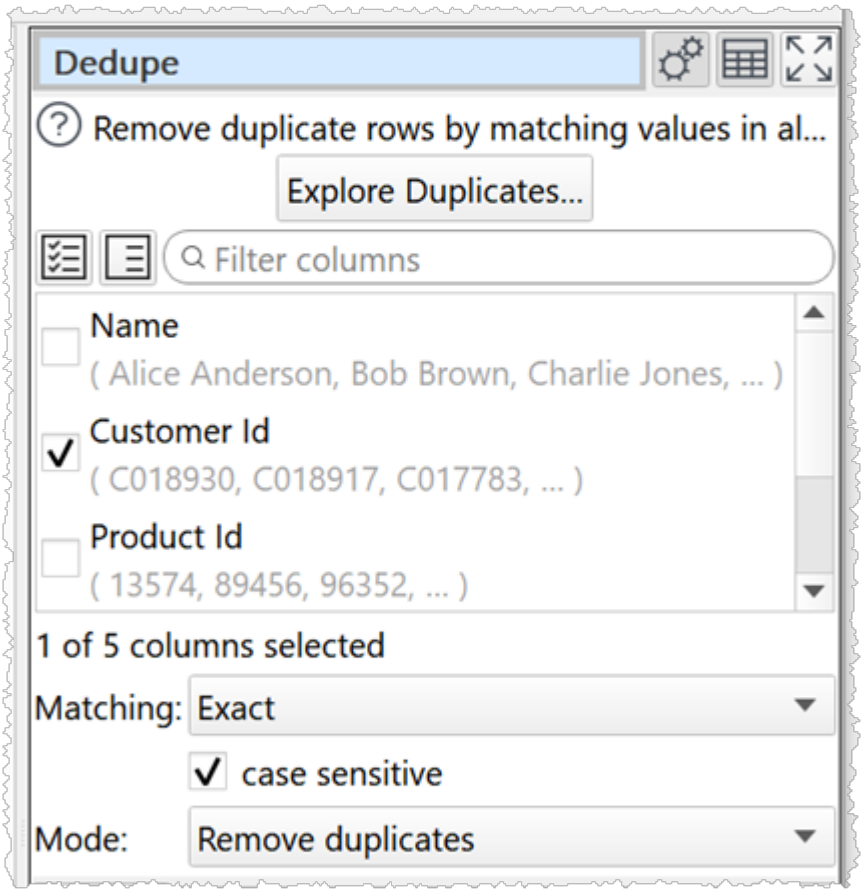
Remove duplicate rows.

### Example

Keep the first record for each unique 'Customer Id':


*© 2023 Oryx Digital Ltd*

| | Name | Customer Id | Product Id | Cost | Date |
|---|---|---|---|---|---|
| 1 | Alice Anderson | C018930 | 13574 | 29.95 | 01/10/2020 |
| 2 | Bob Brown | C018917 | 89456 | 10.55 | 01/10/2020 |
| 3 | Charlie Jones | C017783 | 96352 | 19.95 | 02/10/2020 |
| 4 | Robert Brown | C018917 | 98526 | 10.00 | 02/10/2020 |
| 5 | Charles Jones | C017783 | 38746 | 25.00 | 03/10/2020 |



| | Name | Customer Id | Product Id | Cost | Date |
|---|---|---|---|---|---|
| 1 | Alice Anderson | C018930 | 13574 | 29.95 | 01/10/2020 |
| 2 | Bob Brown | C018917 | 89456 | 10.55 | 01/10/2020 |


© 2023 Oryx Digital Ltd

## Inputs

One.

## Options

- Click **Explore Duplicates...** to explore which rows are duplicates.
- Check the column(s) you wish to look for duplicate values in.
- Set **Matching** to **Exact** to use exact matching to find duplicates and **Fuzzy** to match with some differences. Fuzzy matching is much slower.
- Set **Closeness** to how close a fuzzy match has to be to be considered a match. E.g. set it to 80% to make 2 values that are 80% the same a match. For **Fuzzy** matching only.
- Check **case sensitive** to use case sensitive matching.
- Check **ignore whitespace** to ignore whitespace characters when matching.
- Check **ignore punctuation** to ignore punctuation characters when matching.
- Set **Mode** according to what you want to do with duplicates.
  - **Remove duplicates** removes all duplicate rows.
  - **Keep only duplicates** keeps only the duplicate rows (the inverse of **Remove duplicates**).
  - **Add duplicate information** adds extra columns with information about duplicate rows at the end. Reorders rows, but does not remove them. The additional columns are:
    - **Group**: Each row and its duplicates are given a unique group number, starting at 1.
    - **Row**: The number of the row in the input dataset.
    - **Closeness**: How close a match the duplicate is as a percentage. 100% = exact match.
    - **Duplicate**: Set to `true` if the row is a duplicate and `false` if not.
    - **Duplicates**: Set to the number of duplicates a non-duplicate has.
- Check **drilldown** to allow double-clicking a row in the data table to drilldown to this row and it's duplicates in the upstream data. This is only available when **Matching** is set to **Exact** and **Mode** is set to **Remove duplicates**.

## Notes

- Rows are considered duplicates if they match in all the checked columns.
- Comparisons are whitespace sensitive. You can use Whitespace to remove whitespace before deduping and Replace to get of other unwanted characters (e.g. whitespace inside the text).
- When several rows are duplicates, only the top one is retained. So you may want to Sort your dataset first.
- The Unique transform is a more powerful (but more complex and slower) alternative to Dedupe.

## See also

- Dedupe a dataset


*© 2023 Oryx Digital Ltd*

### 2.3.16 Extract

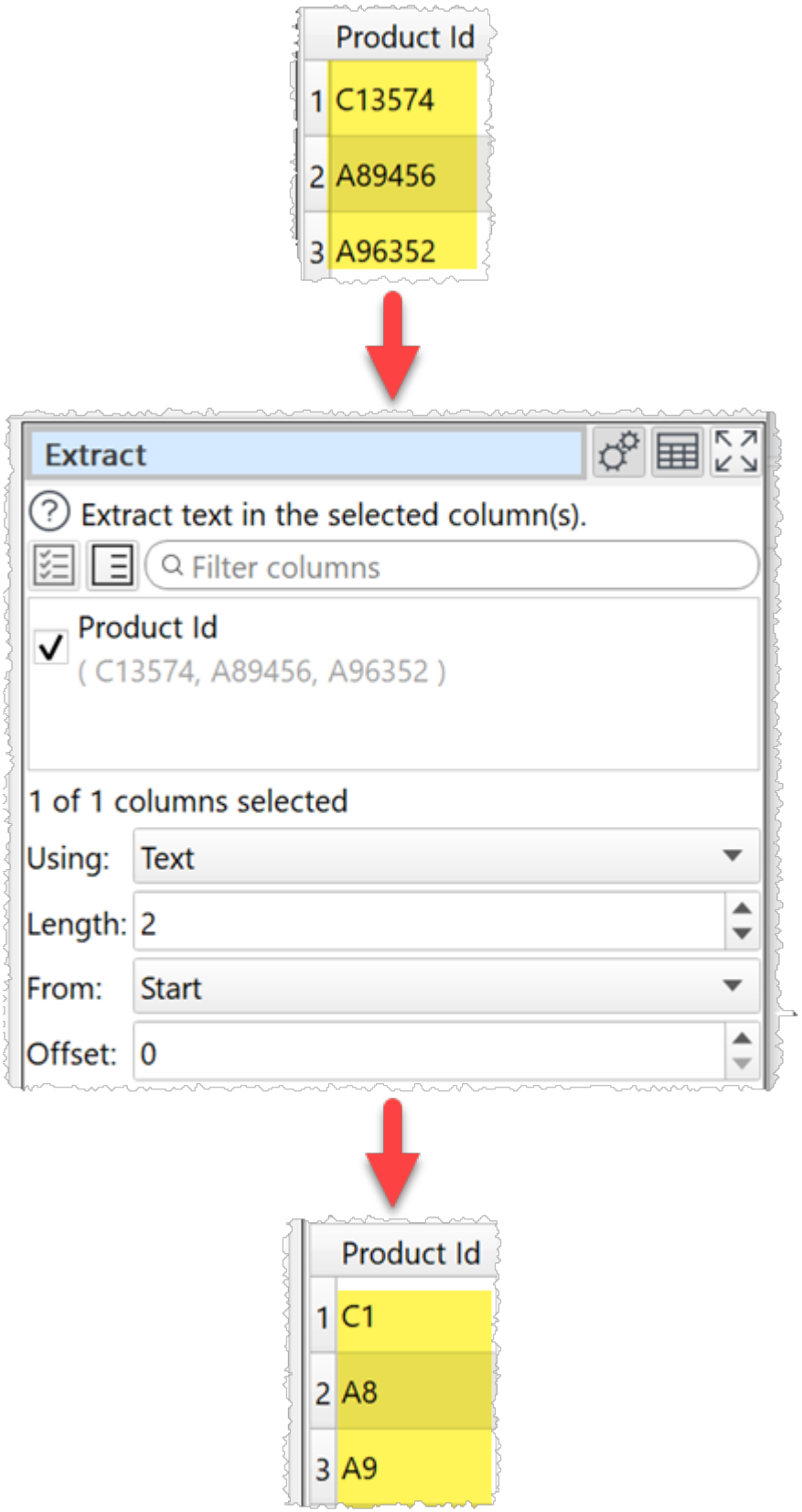
## Description

Extract text in one or more columns.

## Examples

Extract the first 2 digits from the 'Product Id' column:

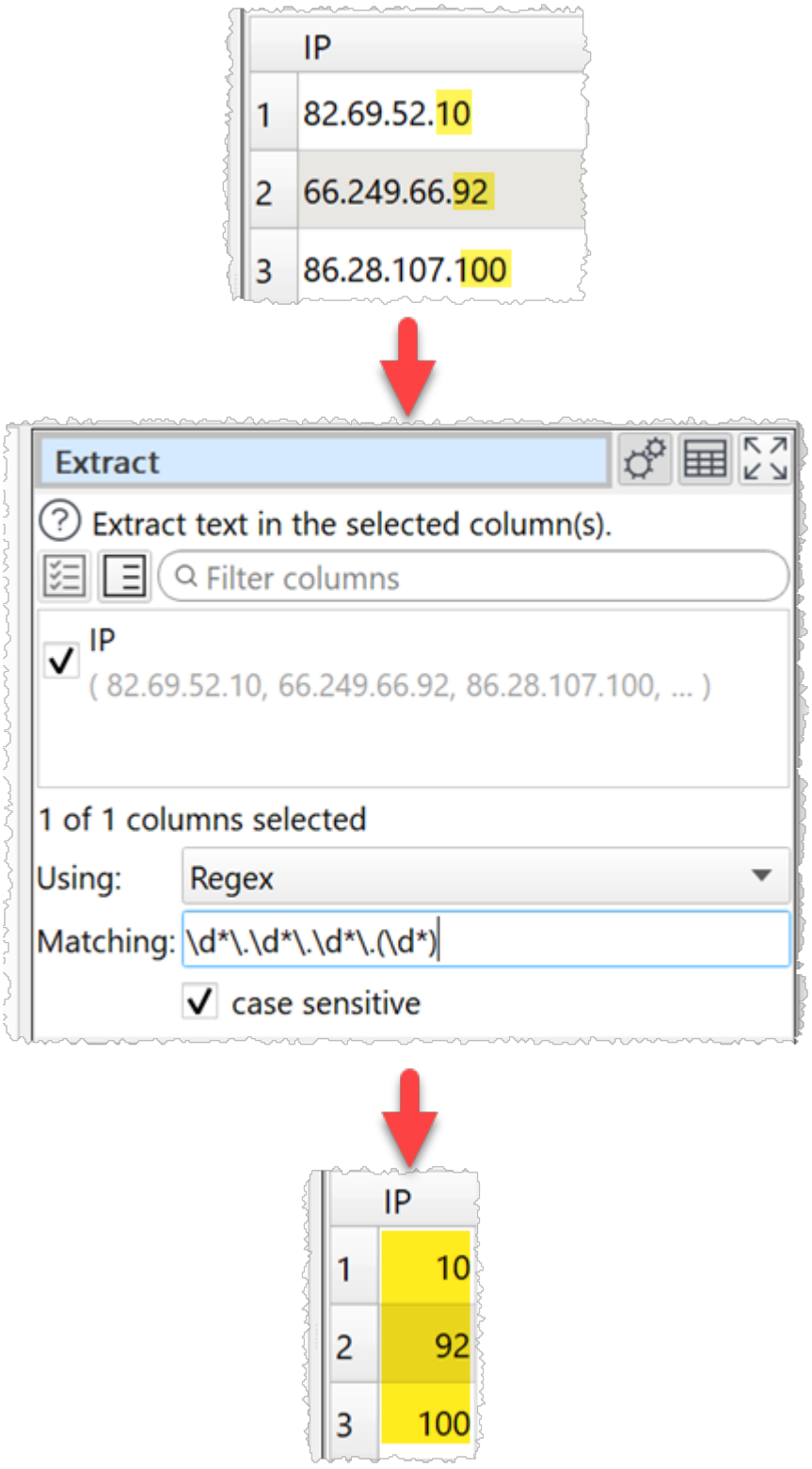

*© 2023 Oryx Digital Ltd*





© 2023 Oryx Digital Ltd

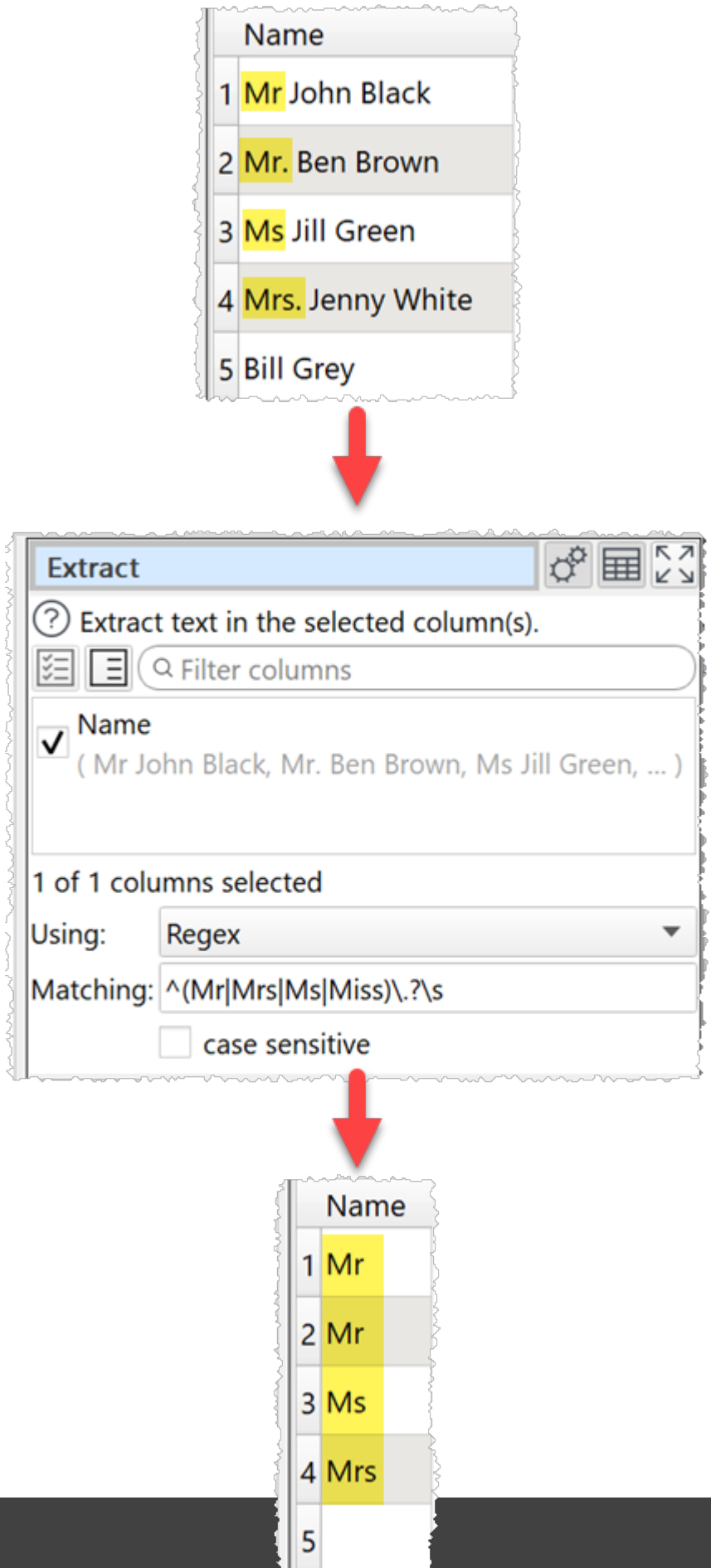
Extract the last group of digits from an IP address:




Extract titles from a name:


© 2023 Oryx Digital Ltd

© 2023 Oryx Digital Ltd





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) you wish to transform.
- Set **Using** depending on whether you want to extract a fixed length of text or with a regular expression.
- Set **Length** to the length you want values in selected columns shortened to. **Text** match only.
- Set **From** to **Start** or **End** depending on whether you want to take from the start or end. **Text** match only.
- If **From** is **Start** then **Offset** is the offset of the first character from the start (0 to start with the first character). If **From** is **End** then **Offset** is the offset of the last character from the end (0 to start with the last character). **Text** match only.
- Set **Matching** to a regular expression. **Regex** match only. If the expression contains capture parentheses the first capture will be extracted. Otherwise it will try to capture the whole regular expression. For example:

| Value | Regular expression | Extracted |
|---|---|---|
| `12.34.567.89` | `(\d*)\.\d*\.\d*\.\d*` | `12` |
| `12.34.567.89` | `\d*\.\d*\.\d*\.(\d*)` | `89` |
| `12.34.567.89` | `\d*\.\d*` | `12.34` |
| `ip 12.34.567.89` | `\d*\.\d*\.\d*\.\d*` | `12.34.567.89` |
| `ip 12.34.567.89` | `^\d*\.\d*\.\d*\.\d*$` | |

- Check **case sensitive** to use case sensitive matching. **Regex** match only.

## Notes
- If you can to keep the original column use Copy Cols to copy the column first.
- Whitespace is counted when calculating length. You can use Whitespace to remove whitespace before extracting.
- If you want to set a column to a fixed length use Pad and Extract together.
- Use Replace If you want to extract multiple captures from a value using a regular expression.

## See also
- Chop


*© 2023 Oryx Digital Ltd*

### 2.3.17 Fill

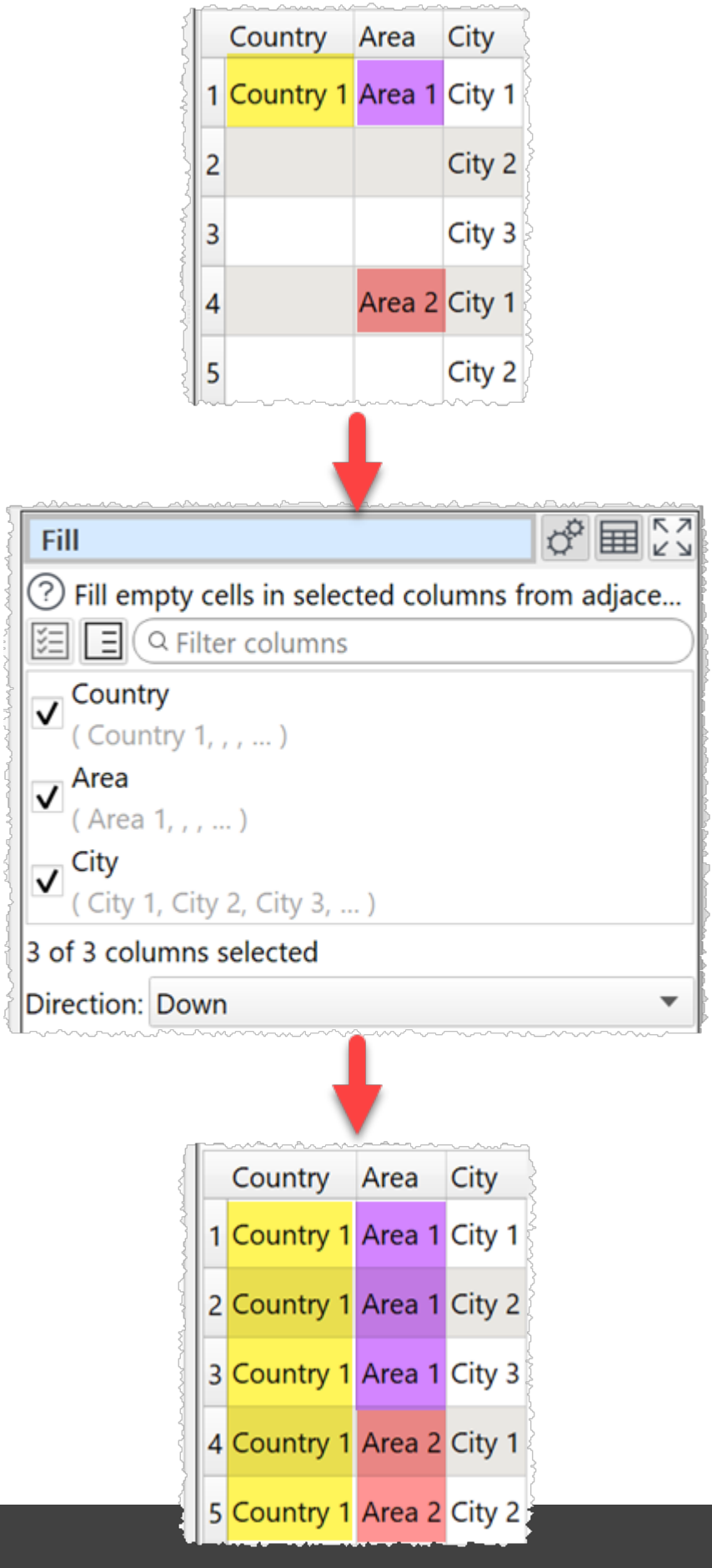
## Description

Fill empty cells in selected columns with the next adjacent non-empty cells. The direction is the direction to fill to (not from).

## Example

Fill down:


*© 2023 Oryx Digital Ltd*





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) you wish to fill.
- Select **Direction** depending on the direction you wish to fill to.

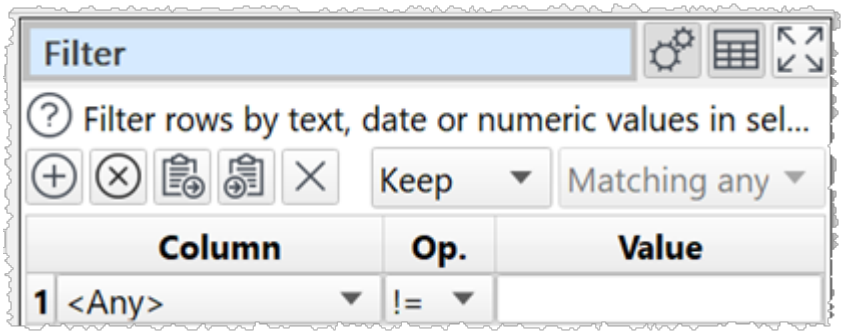
## Notes
- To fill empty values with something else use the **Replace** transform.

## See also
- Unfill
- Impute

### 2.3.18 Filter

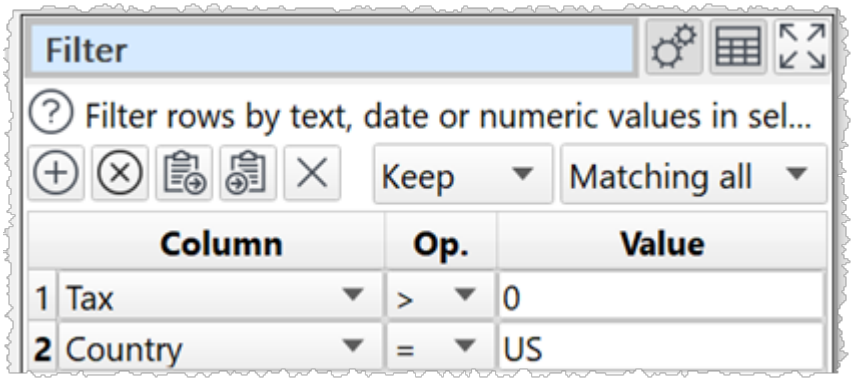
## Description
Removes rows based on number, date and text values in selected columns.

## Examples
Remove rows where all values are empty.

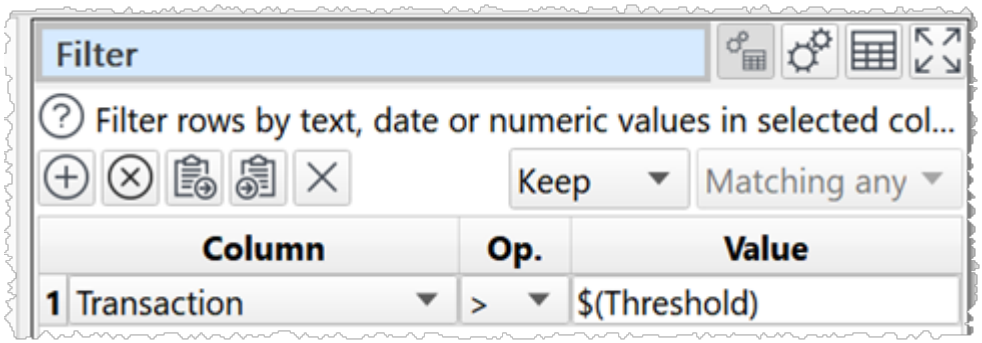



Keep only rows where the 'Tax' value is greater than 0 and the 'Country' value is 'US'.


© 2023 Oryx Digital Ltd




Remove rows where the 'Transaction' value is greater than the 'Threshold' value (using a column variable).




## Inputs

One.

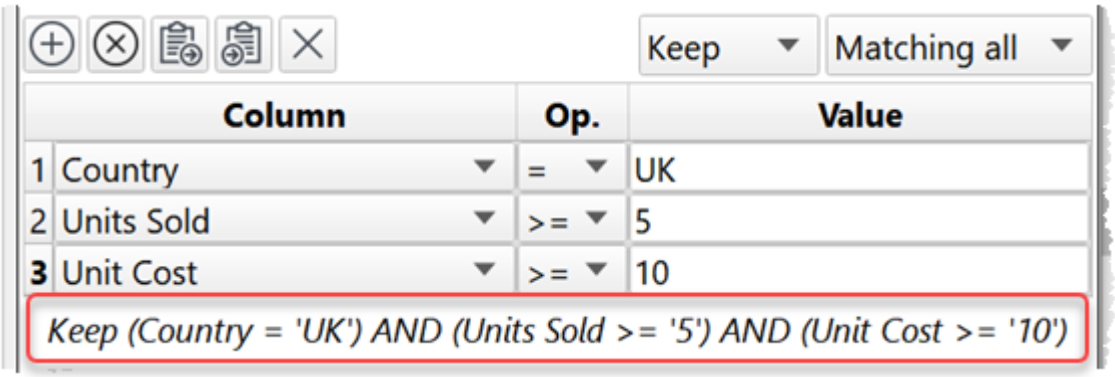
## Options

- Click the ⊕ button to add a new filter criteria.
- Click the ⊗ button to delete the selected filter criteria.
- Click the button to copy terms to the clipboard.
- Click the button to paste terms from the clipboard.
- Click the ✕ button to clear all terms.
- Select **Keep** if you want to keep matching rows and **Remove** to remove matching rows.
- Select **Matching all** to match on all criteria (e.g. criteria 1 and criteria 2). Select **Matching any** to require a match on one or more criteria (e.g. criteria 1 or criteria 2).
- Set **Column** to the column you wish to match.
- Set **Op.** to the comparison operator.
- Set **Value** to the value you wish to compare. You can use a column variable.
- Check **case sensitive** to use case sensitive matching for text.
- Check **disable filtering** to turn off filtering. If sampling is disabled, the transform does nothing.


*© 2023 Oryx Digital Ltd*

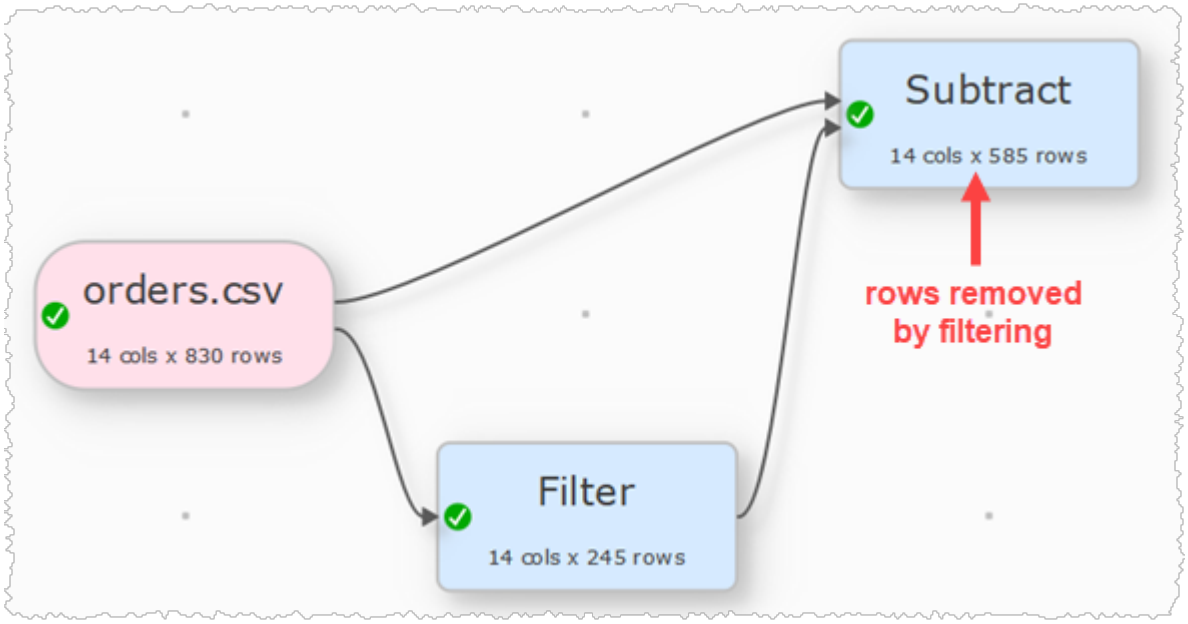
## Notes

- A text summary of the filters is shown:




- A filter row is ignored if the **Value** column is empty , except when **Op.** is **Equal to**, **Not equal to**, **Matches regex** or **Doesn't match regex.**
- Number, date and text values are treated differently for **Equal to**, **Greater than**, **Less than**, **Greater than equal, Less than equal** and **Not equal to** operations.
  - If both values are numeric, a numeric comparison will be carried out.
  - If both values match a supported date formats in Preferences, a date comparison will be carried out.
  - Otherwise the values will be treated as text.
  - For example, an empty value is considered less than 0, because they will be compared as text. So you might want to replace empty values with 0 or remove those rows with Filter, before comparing them.
- All values are treated as text for **Contains**, **Starts with**, **Ends with**, **Matches regex**, **Is not**, **Doesn't start with, Doesn't end with** and **Doesn't match regex** operations.
- Comparisons of text are whitespace sensitive. Cells with whitespace will not match **Is empty**. You can use Whitespace to remove whitespace before filtering and Replace to get of other unwanted characters (e.g. whitespace inside the text).
- See here for more details on Regular expressions (regex).
- Y can use Subtract to see rows removed by filtering if there is unique key column.


*© 2023 Oryx Digital Ltd*




- If you are pasting in filter terms, the format expected is comma delimited text with 3 columns.
  - The first column is the 0-based index of the **Column** drop-down list.
  - The second column is the 0-based index of the **Op.** drop-down list.
  - The third column is the **Value**.

For example, pasting in:

```
0,0,YES
```

Will add:

You can also try copying terms to the clipboard and pasting into a text file to see the format. If you only paste in 1 column of text, it will be assumed to be a list of **Value**. You can, of course, use Easy Data Transform to create the appropriate filter terms and paste them into memory.

## See also

- Compare Cols
- Slice


*© 2023 Oryx Digital Ltd*

### 2.3.19 Gather

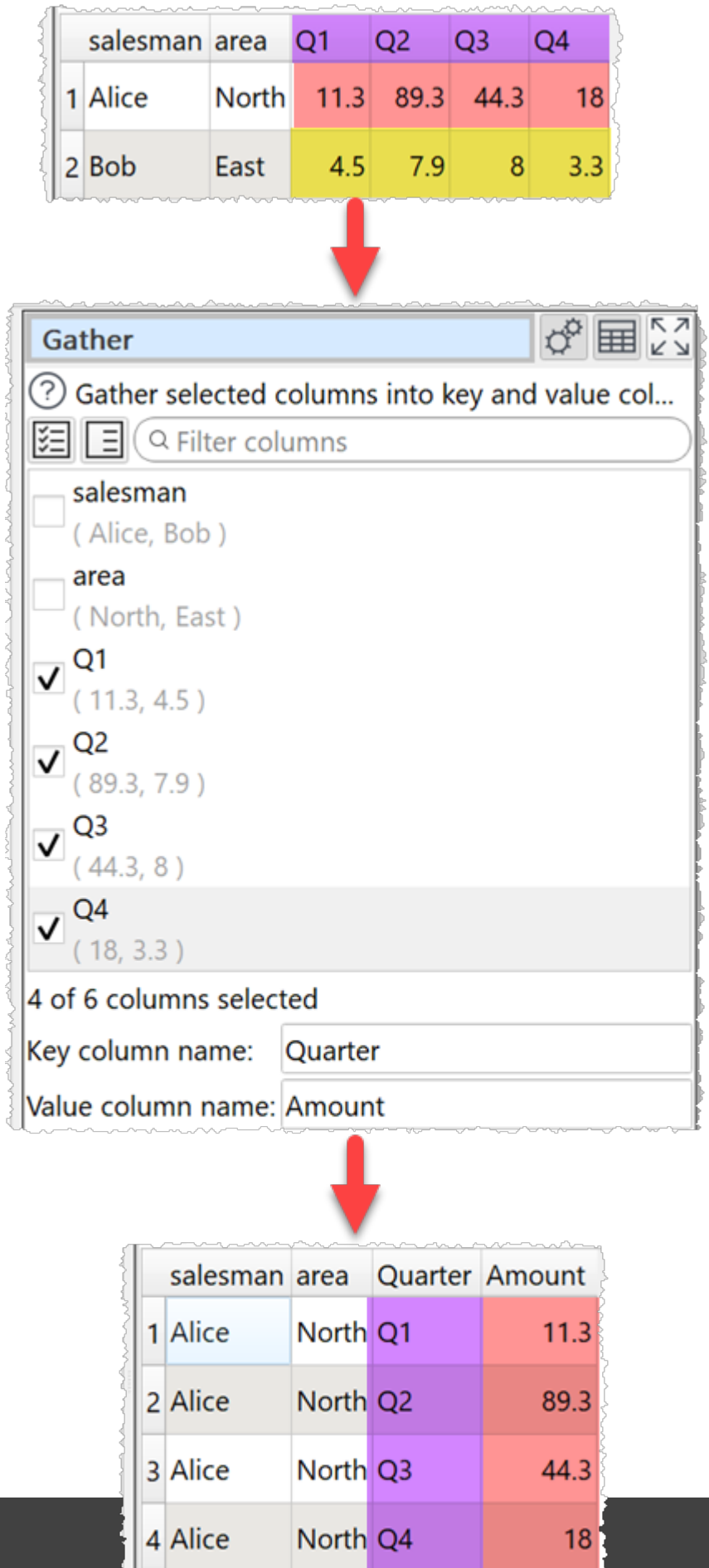
## Description

Gather multiple columns into new key and value columns. Also called unpivot, long pivot or group by.

## Example

Gather 'Q1', 'Q2', 'Q3' and 'Q4' columns into 1 column.


*© 2023 Oryx Digital Ltd*





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Select the **Columns** you wish to gather.
- Set **Key column name** to the name of the new key column, which will have values based on the names of the columns selected.
- Set **Value column name** to the name of the new value column, which will have values based on the values in the columns selected.

## Notes
- New columns are added at the right end. You can change the column order with Reorder Cols.
- You can merge the value and key columns into a single column with Concat Cols.
- The opposite of Gather is Spread.

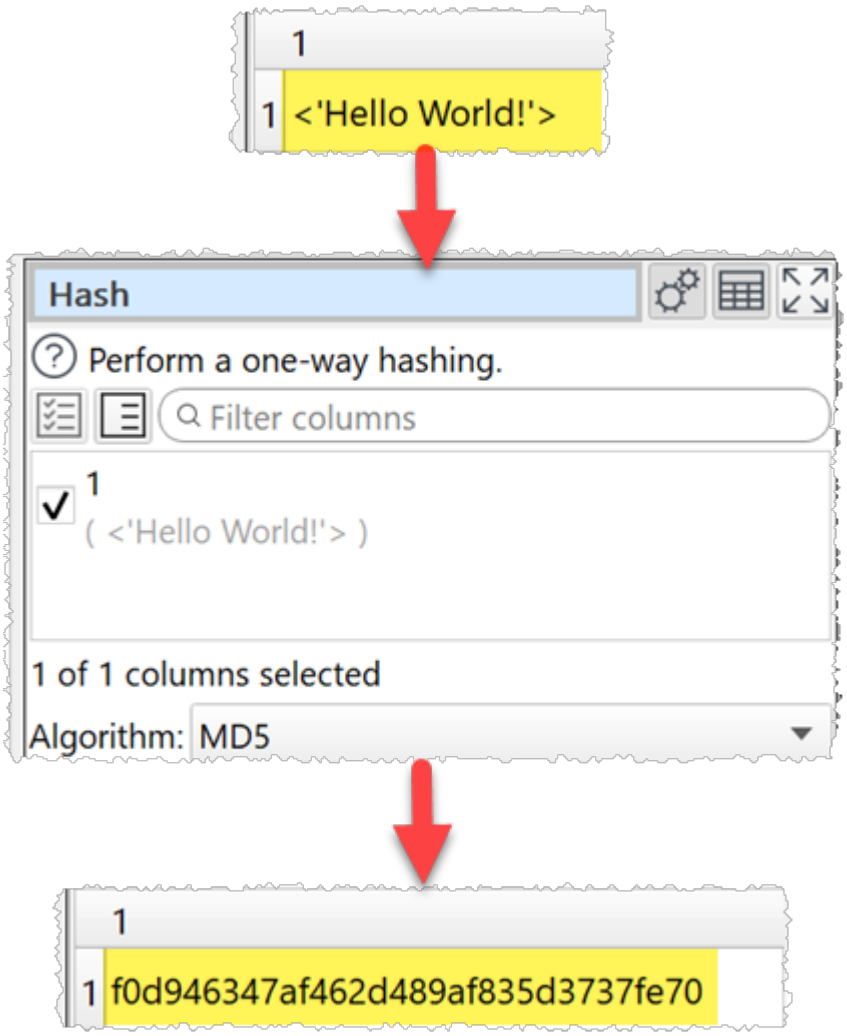
## See also
- Video: How to unpivot Excel
- Split Rows

### 2.3.20 Hash

## Description
Perform one-way hashing on selected columns.

## Example
Compute the MD5 hash of `<'Hello World!'>`:


© 2023 Oryx Digital Ltd





© 2023 Oryx Digital Ltd

| Type | Plaintext | Secret | Hashed |
|---|---|---|---|
| SHA1 | <'Hello World!'> | | 4a700d8555aae497cc54d32f6f3d7478200a5dea |
| SHA-256 | <'Hello World!'> | | 2c6139a5d874b9626e3ae2443fa85cdfe1fc18f1480348267f8bf71a783f738a |
| SHA-512 | <'Hello World!'> | | 23728b865af0f17bb9bfaaf064f711bf00626c07e5389dcbb18365ace19c2394d7a00066c0b6a7112dab528e62106a855dd6bc41d0c262fb72e9f47e10f1652c |
| SHA3-256 | <'Hello World!'> | | 0701263723cdbdf11e1f7118e62d7d29d61f01f8490ef775e4991558634bd719 |
| SHA3-512 | <'Hello World!'> | | 61cbf6f11e864af592626d9430271a5ad6f39edf7e09719a97c57b5c3222cdf24bd70f1f6927b7b41eb948bbddfe0a4ae965688a4a6882df7be36492ad653ab0 |
| Keccak-256 | <'Hello World!'> | | 1a45dd8df9de65049692d9192509b30560b69c1fd881c57b2cf4607e85b3569f |
| Keccak-512 | <'Hello World!'> | | 2741b7ddfe0d4c4726d476cc719fdbb3b7db88fe6df1a1df00723547c546496ac49bd82e94009f4572d3a31fe0fc50bce1ee3c76d983f8ddc480b7da9cd1ed02 |
| MD5 | <'Hello World!'> | | f0d946347af462d489af835d3737fe70 |
| HMAC-MD5 | <'Hello World!'> | ABC123 | 5940ca9b5f2b47ae1972ab81db8030ac |


© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) you wish to transform.
- Set **Algorithm** to the hashing algorithm you wish to use
- Set **Secret** to the secret key you wish to use (HMAC-MD5 only). The default value is randomly generated.

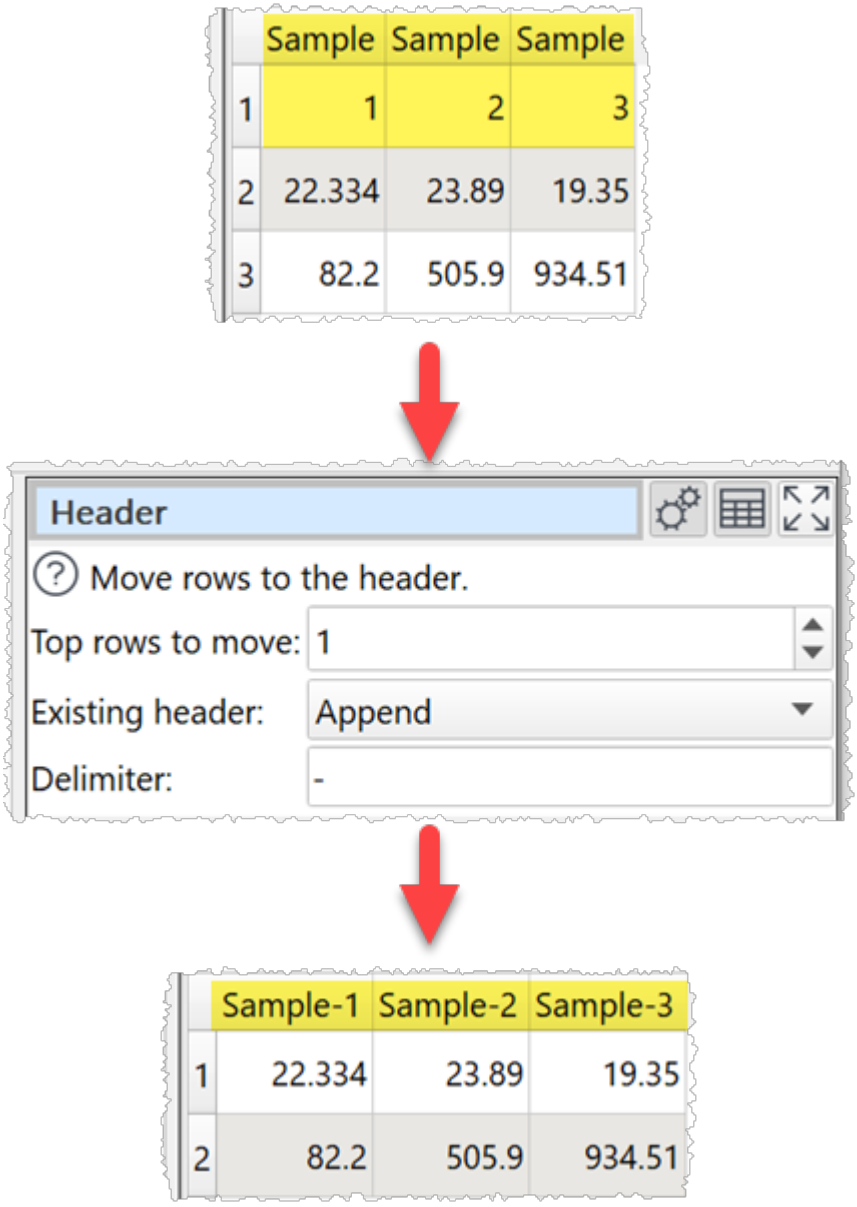
## Notes
- There is no known way to reverse these hashes.
- As the chance of 2 different inputs generating the same hash value by chance (a 'collision') is infinitesimally small, hashing can be used to anonymize data or to help find duplicates of long data values.
- The MD5 algorithm is now considered weak as a cryptographic hash. But remains suitable for other non-cryptographic purposes.
- Use Copy Cols to copy columns before you transform them.

### 2.3.21 Header

## Description
Move rows from the top of the dataset into the header.

## Example
Append the first row to the header:


*© 2023 Oryx Digital Ltd*




## Inputs

One.

## Options

- Set **Top rows to move** to the number of rows you want to move from the top dataset into the header. Setting it to 0 means the transform does nothing.
- Set **Existing header** to **Overwrite** to ignore the existing header values and **Append** to add to the existing header values.
- Set **Delimiter** to any text you want to put between column elements. It can be left empty. Ignored if **Existing header** set to **Overwrite** and **Top rows to move** set to **1.**


*© 2023 Oryx Digital Ltd*

## Notes

- Empty cells are ignored.
- You can Sort and Filter your dataset to change the top rows.
- You can add the header from one dataset to another dataset using Stack.

## See also

- Headers

### 2.3.22 If

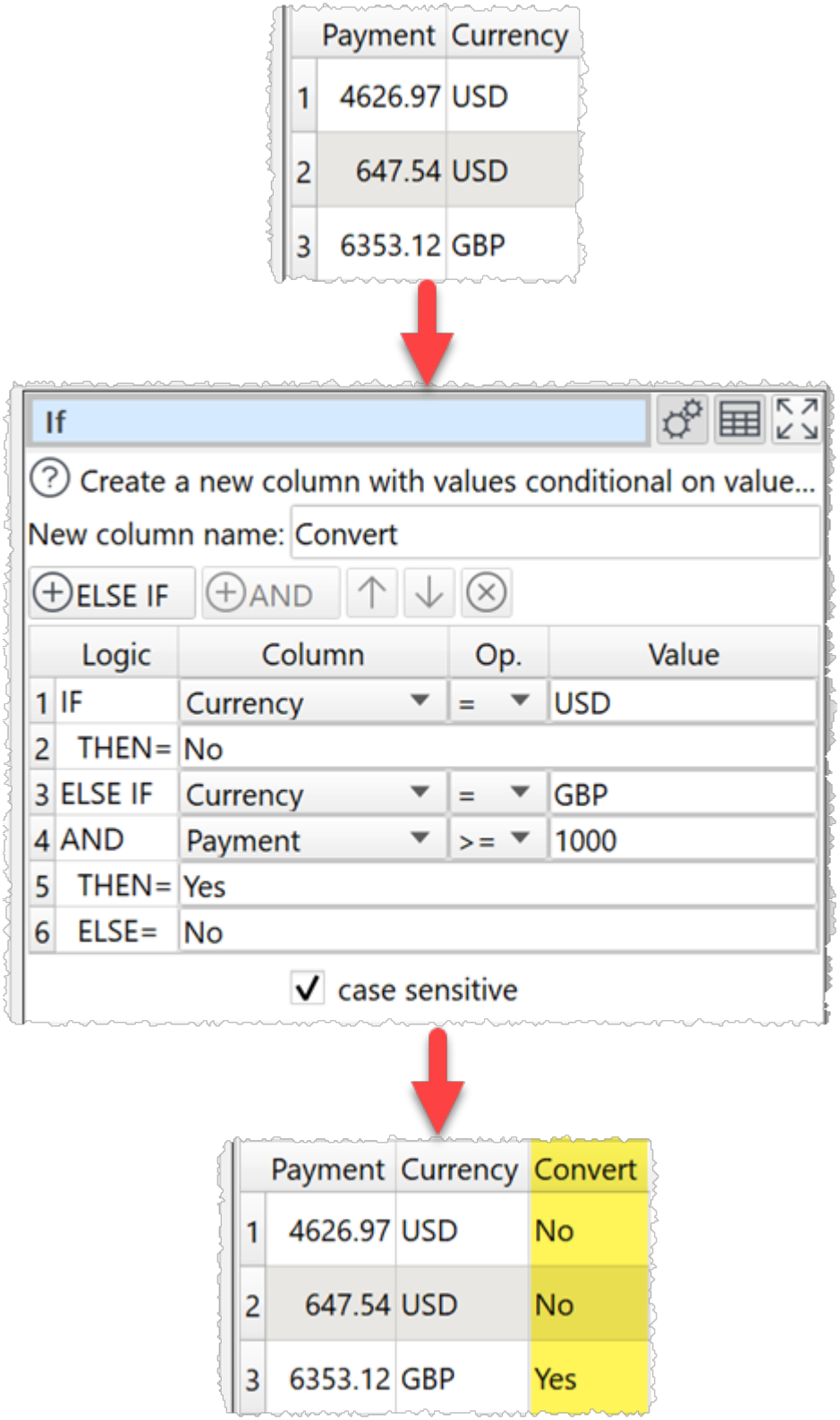
## Description

Sets the value of a new column based conditionally on values in one or more other columns.

## Examples

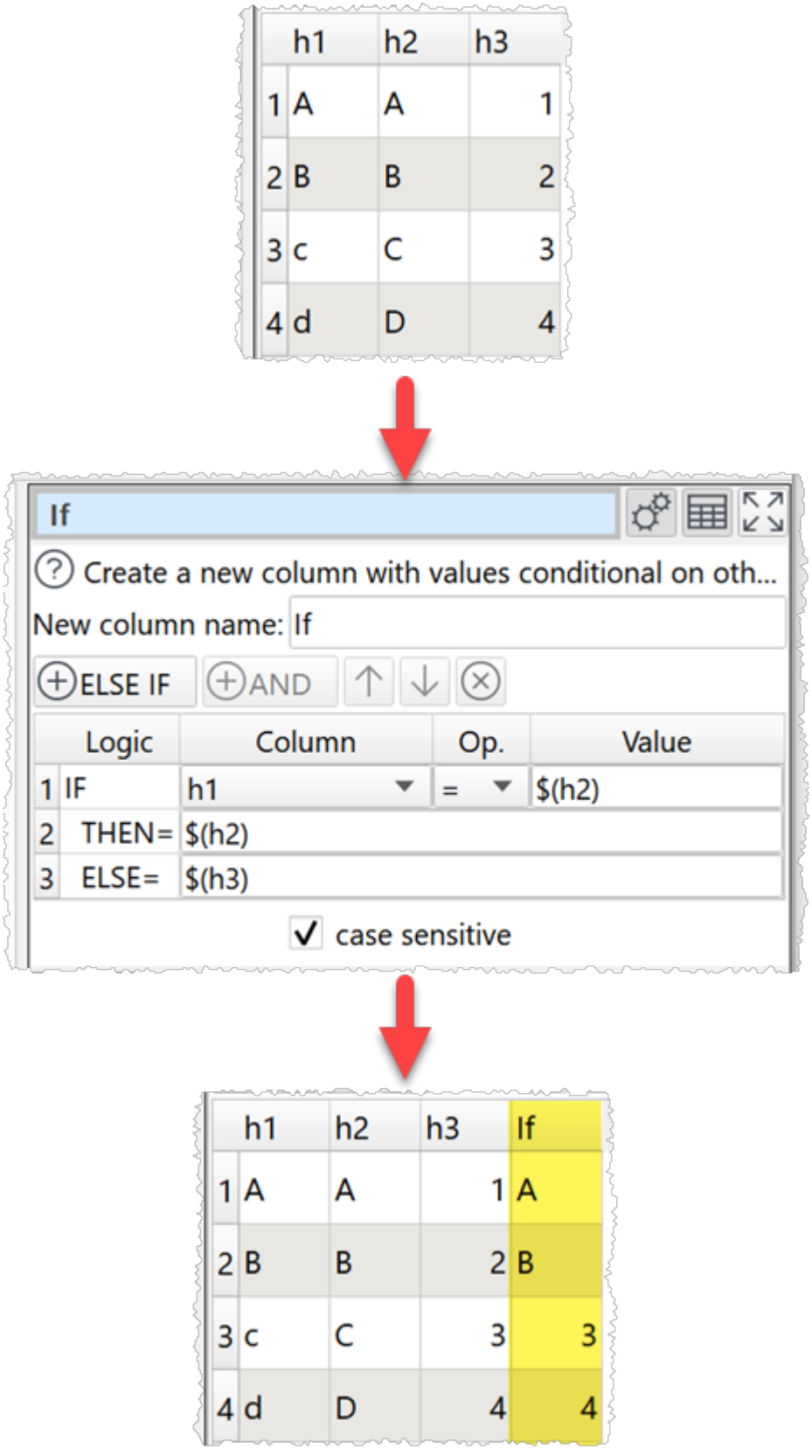
Add a new column based conditionally on 'Payment' and 'Currency' columns:


© 2023 Oryx Digital Ltd




If the 'h1' value is 1 use the value of the 'h2' column, otherwise use the value of 'h3' column (using column variables).:


*© 2023 Oryx Digital Ltd*





*© 2023 Oryx Digital Ltd*

## Inputs
One.

## Options
- Click the ⊕**ELSE IF** button to add a new IF/ELSE IF..THEN condition.
- Click the ⊕**AND**'button to add an AND to the selected IF/ELSE IF..THEN.
- Click the ⊗ button to delete the selected IF/ELSE IF..THEN/AND.
- The **Logic** column shows the type of row.
- Set **Column** to the column you wish to match.
- Set **Op.** to the comparison operator.
- Set **Value** to the value you wish to compare.
- Check **case sensitive** to use case sensitive matching for text.

## Notes
- The **IF**, **THEN** and **ELSE** values can use column variables, as in the second example above.
- You can simulate OR with multiple IF statements. For example:

```
IF x = 1 OR y = 2
   THEN 3
```

  Is equivalent to:

```
IF x = 1
   THEN 3
ELSE IF y = 2
   THEN 3
```

- Number, date and text values are treated differently for **Equal to**, **Greater than**, **Less than**, **Greater than equal, Less than equal** and **Not equal to** operations.
  - If both values are numeric, a numeric comparison will be carried out.
  - If both values match a supported date formats in Preferences, a date comparison will be carried out.
  - Otherwise the values will be treated as text.
  - For example, an empty value is considered less than 0, because they will be compared as text. So you might want to replace empty values with 0 or remove those rows with Filter, before comparing them.
- All values are treated as text for **Contains**, **Starts with**, **Ends with**, **Matches regex**, **Is not**, **Doesn't start with, Doesn't end with** and **Doesn't match regex** operations.
- Comparisons of text are whitespace sensitive. Cells with whitespace will not match **Is empty**. You can use Whitespace to remove whitespace before filtering and Replace to get of other unwanted characters (e.g. whitespace inside the text).
- See here for more details on Regular expressions (regex).


*© 2023 Oryx Digital Ltd*

## See also

- Lookup

### 2.3.23 Impute

## Description

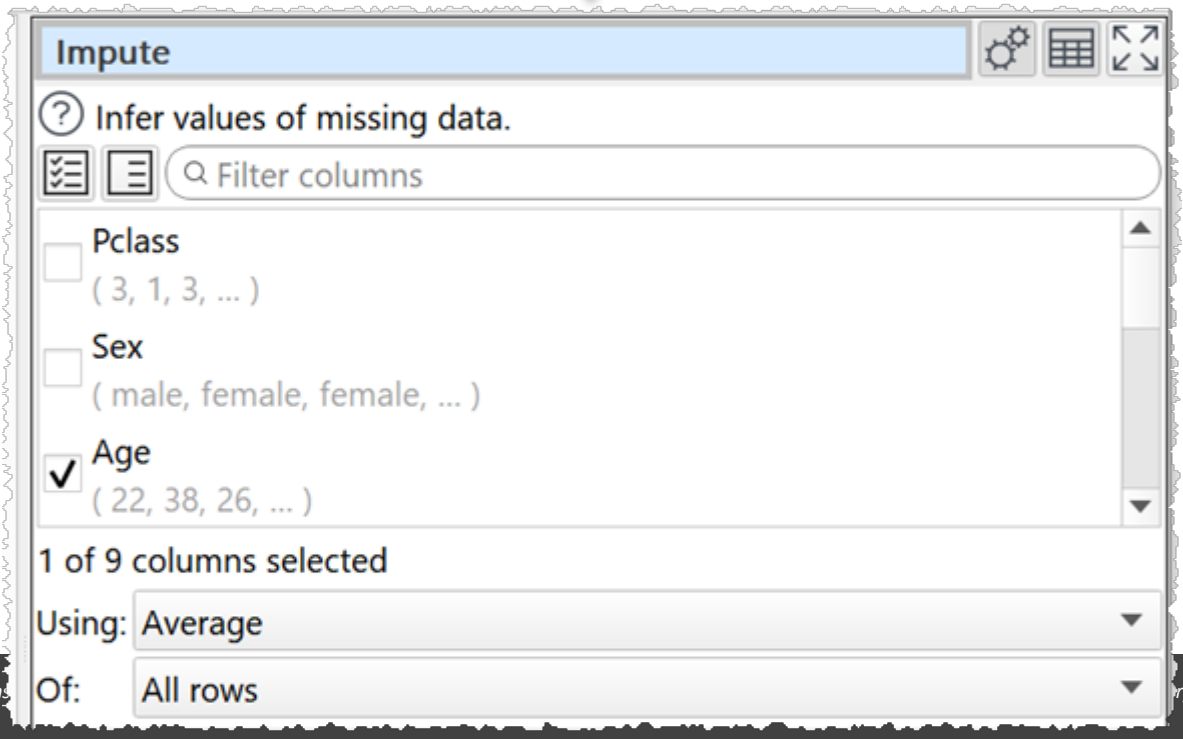
Infer values of missing data from other values in the same column.

## Examples

Impute the missing age of Titanic passengers based on the average age of all passengers:


© 2023 Oryx Digital Ltd

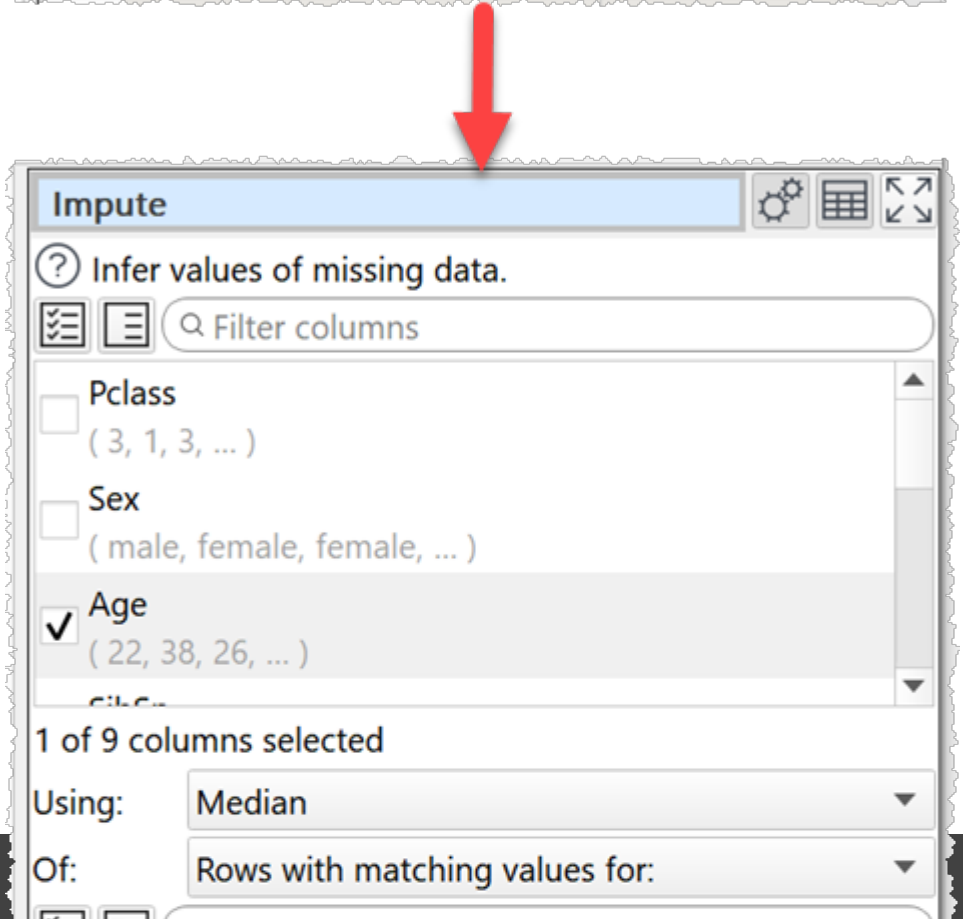
| | Pclass | Sex | Age | SibSp | Parch |
|---|---|---|---|---|---|
| 21 | 2 | male | 35 | 0 | 0 |
| 22 | 2 | male | 34 | 0 | 0 |
| 23 | 3 | female | 15 | 0 | 0 |
| 24 | 1 | male | 28 | 0 | 0 |
| 25 | 3 | female | 8 | 3 | 1 |
| 26 | 3 | female | 38 | 1 | 5 |
| 27 | 3 | male | | 0 | 0 |
| 28 | 1 | male | 19 | 3 | 2 |
| 29 | 3 | female | | 0 | 0 |
| 30 | 3 | male | | 0 | 0 |
| 31 | 1 | male | 40 | 0 | 0 |



Impute the missing age of Titanic passengers based on the median age of passengers with the same passenger class and sex:


*© 2023 Oryx Digital Ltd*

| | Pclass | Sex | Age | SibSp | Parch |
|---|---|---|---|---|---|
| 21 | 2 | male | 35 | 0 | 0 |
| 22 | 2 | male | 34 | 0 | 0 |
| 23 | 3 | female | 15 | 0 | 0 |
| 24 | 1 | male | 28 | 0 | 0 |
| 25 | 3 | female | 8 | 3 | 1 |
| 26 | 3 | female | 38 | 1 | 5 |
| 27 | 3 | male | | 0 | 0 |
| 28 | 1 | male | 19 | 3 | 2 |
| 29 | 3 | female | | 0 | 0 |
| 30 | 3 | male | | 0 | 0 |
| 31 | 1 | male | 40 | 0 | 0 |





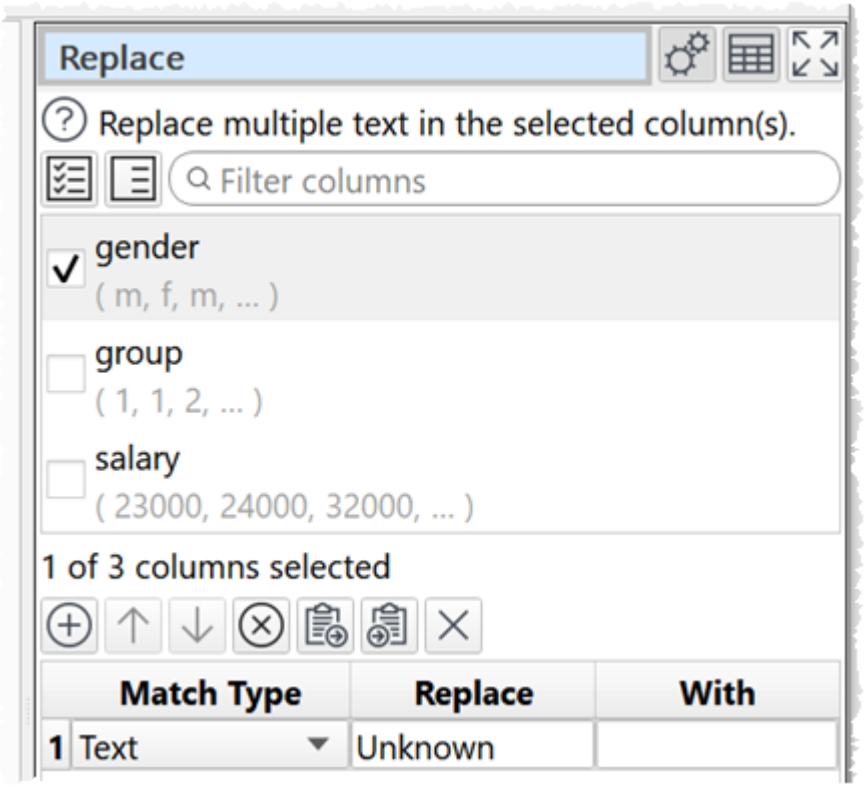
© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) which have empty values you wish to impute.
- Set **Using** to impute using the **Average** (mean), **Median** or **Mode** of non-empty data values in each column.
- Set **Of** to:
  - **All rows** to fill empty values in each column with the **Average**, **Median** or **Mode** of all non-empty values in the same column.
  - **All rows with matching values for** to fill empty values in each column with the **Average**, **Median** or **Mode** of all non-empty values in the same column where values match in the checked columns below.
- Set **Matching** to **Exact match** if every field has to match and **Best match** to use rows that have the most matching values (or all rows, if no matches). For example, if you are matching on passenger class and sex then:
  - **Exact match** will only use values from rows where the passenger class and sex both match.
  - **Best match** will use values from rows where the passenger class and sex both match, if available. Otherwise it will use values from rows where either the passenger class or sex match, if available. Otherwise it will use values from all rows.
- Check **case sensitive** to use case sensitive matching.

## Notes
- **Average** and **Median** only work on numerical data (non-numerical values are ignored).
- **Mode** treats all non-empty data values as text. Case and whitespace are taken into account.
- If **Mode** is selected and there are multiple values with the same maximum frequency, one will be picked at random.
- If you need to convert some values to empty before imputing, you can do this using a Replace transform.


*© 2023 Oryx Digital Ltd*




- You can use a Filter transform to remove rows with missing values instead of imputing values.

## See also

- Video: How to impute missing data
- Fill
- Interpolate

### 2.3.24 Insert

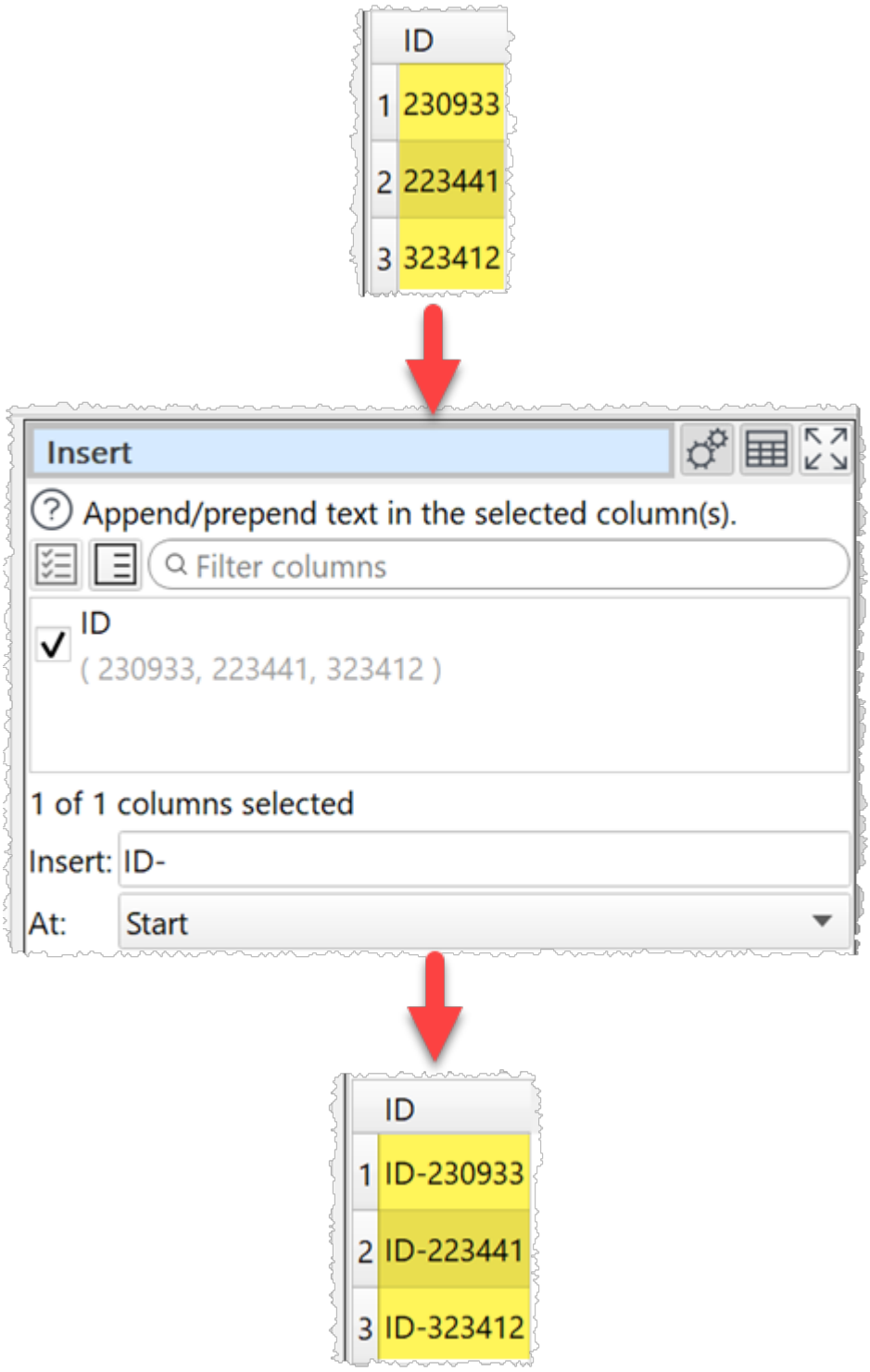
## Description

Append/prepend text to one or more columns.

## Example

Add 'ID-' to the front of a column of text:


© 2023 Oryx Digital Ltd




## Inputs

One.


*© 2023 Oryx Digital Ltd*

## Options

- Check the column(s) you wish to transform.
- In **Insert** put the text you want to insert. You can use a column variable.
- In **At** put the position you want the text inserted.

## Notes

- You can use Whitespace to remove whitespace before inserting.

## See also

- Pad
- Extract

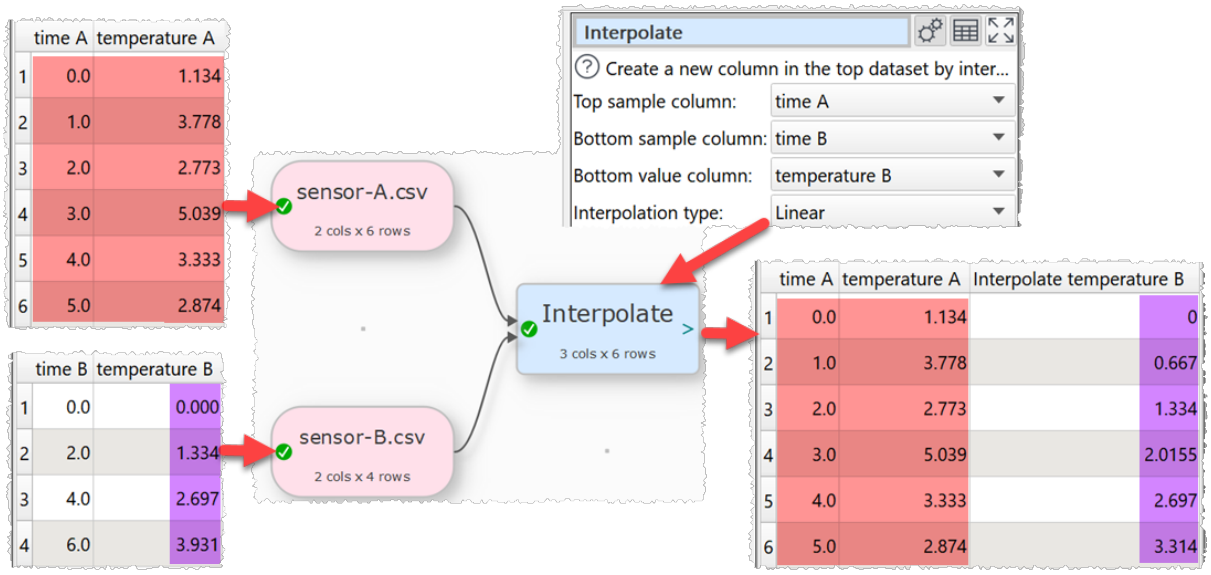
### 2.3.25 Interpolate

## Description

Interpolate values for a dataset based on numerical sample-value pairs in another dataset and puts the result in a new column.

## Example

Interpolate time and temperature datasets for sensors A and B with different sampling frequencies:



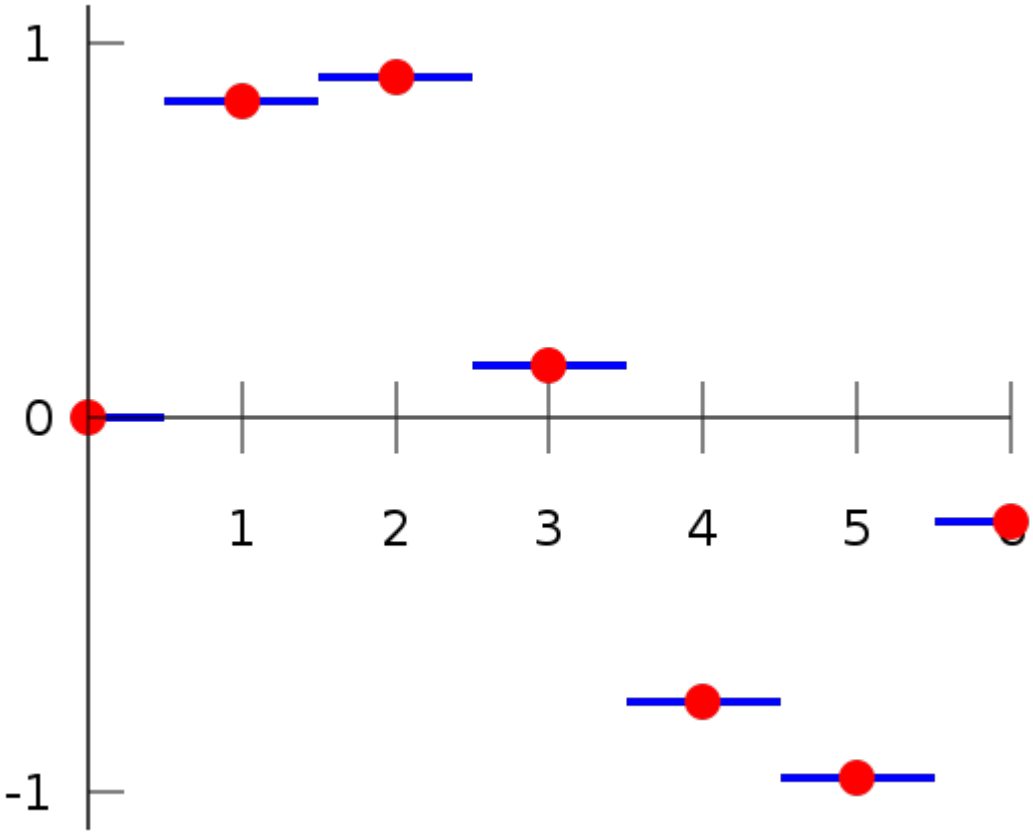

## Inputs

Two.

## Options

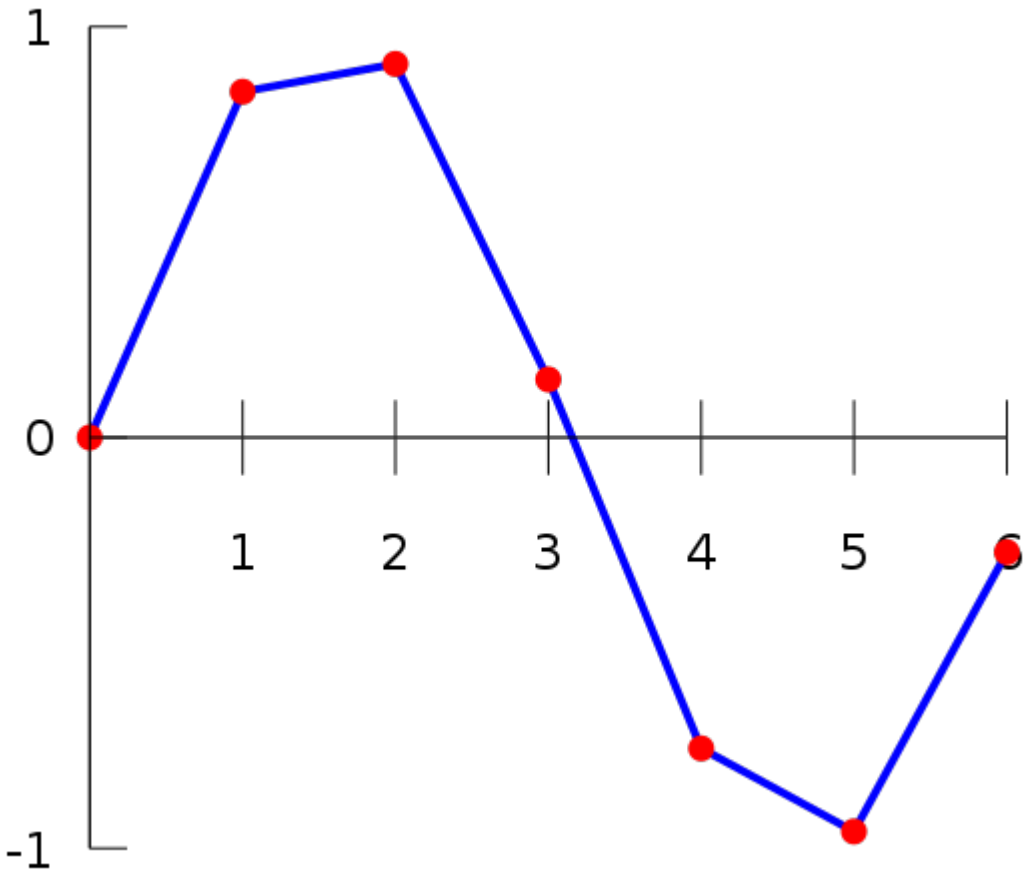

© 2023 Oryx Digital Ltd

- Place the dataset you want to modify as the top input and the dataset you want to sample values from as the bottom input.
- Select **Top sample column** for the column whose values you wish to sample.
- Select **Bottom sample column** for the column that matches the top sample column in the bottom dataset.
- Select **Bottom value column** for the column that contains the values.
- Set **Interpolation type** to the type of interpolation you wish to use.




*Piecewise interpolation (image from Wikipedia)*


© 2023 Oryx Digital Ltd




*Linear interpolation (image from Wikipedia)*

## Notes

- If your sample is below the first sample in the bottom dataset, the first value will be returned.
- If your sample is above the last sample in the bottom dataset, the last value will be returned.
- Easy Data Transform will try to guess sensible default values for **Top sample column**, **Bottom sample column** and **Bottom value column** based on column contents.
- If the first input has a header, this will be used for the output.
- The new column is added at the right end. You can change the column order with Reorder Cols and the column name with Rename Cols.
- Use Num Format to change the precision of the results.
- Messages are shown in the **Warnings** tab for non-numeric values in either dataset.
- Messages are shown in the **Info** tab for **Top sample column** outside the range of values in **Bottom sample column** .


© 2023 Oryx Digital Ltd

## See also

- Lookup
- Impute
- Join

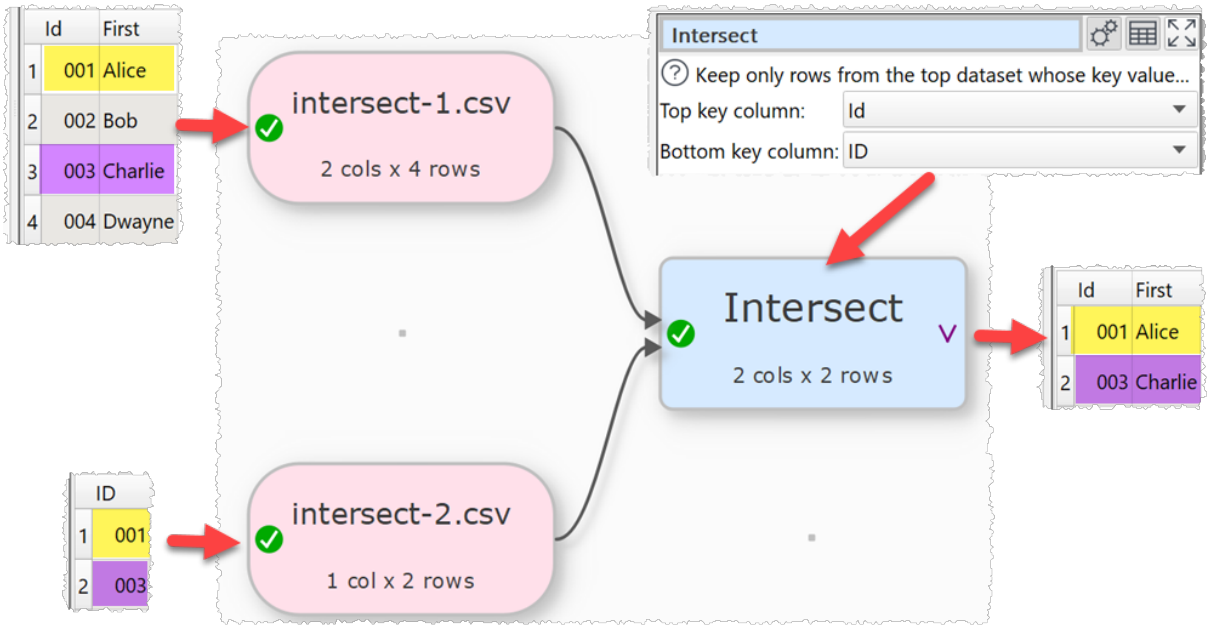
### 2.3.26 Intersect

## Description

Keep only rows from the top dataset with key values that are present in the lower dataset.

## Example

Keep rows in the top dataset that also have their ids in the bottom dataset:




## Inputs

Two.

## Options

- The output depends on the vertical (Y-axis) position of the inputs.
- Click **Explore Keys...** to compare key values in the 2 datasets.
- Select **Top key column** for the column you want to match in the top input dataset.
- Select **Bottom key column** for the column you want to match in the bottom input dataset.
- Check **case sensitive** to use case sensitive matching for keys.


*© 2023 Oryx Digital Ltd*

## Notes

- If there are 10k rows or less in both datasets, Easy Data Transform will try to guess sensible default values for **Top key column** and **Bottom key column** based on column header names and contents.
- If the first input has a header, this will be used for the output.
- All values are treated as text and comparisons are whitespace sensitive. You can use Whitespace to remove whitespace before the intersect.
- Does not remove duplicates. You can use Unique to do this.
- You can use Concat Cols to join several columns together (e.g. 'first name' and 'last name' columns) to form a key column.
- You can use Row Num to create a unique key column.

## See also

- Subtract

### 2.3.27 Javascript

## Description

Create a custom transform using Javascript (ECMAScript).

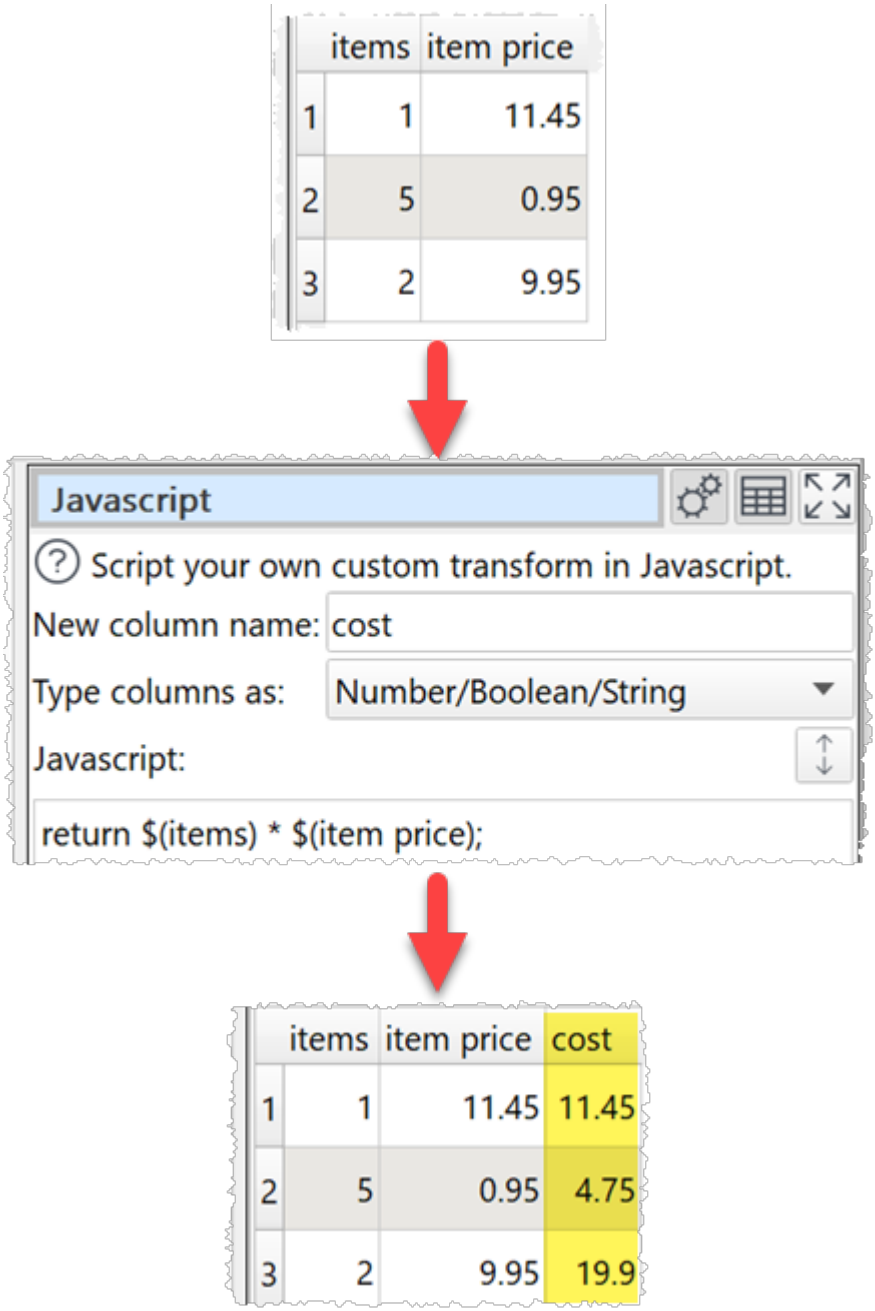
Easy Data Transform allows you to carry out a wide range of data transformations without programming. But sometimes you might need a specialist transformation that can't be done with the built-in transforms. For that you can use the **Javascript** transform. It allows you to write the body of a Javascript function, to calculate a value for each row in a new column. Existing column values can be used as variables.

Javascript is a fully-fledged programming language and can perform arbitrarily complex transforms. It can handle strings, numbers, dates and text.

## Examples

Multiply the value in column 'items' by the value in column 'item price':


© 2023 Oryx Digital Ltd




To concatenate 'last' and 'first' columns with a comma and a space:

```
return $(last) + ', ' + $(first);
```

To calculate the biggest of numeric columns 'v1' and 'v2':

```
return Math.max( $(v1), $(v2) );
```

To find the number of characters in column 'v1':


*© 2023 Oryx Digital Ltd*

```
return String( $(v1) ).length;
```

To determine whether phone numbers in the 'phone_num' column are valid using a regular expression:

```
const validPhoneNum = /^[\+]?[(]?[0-9]{3}[)]?[-\s\.]?[0-9]{3}[-\s\.]?
[0-9]{4,6}$/;
if ( validPhoneNum.test( $(phone_num) ) )
   return "valid";
else
   return "invalid";
```

To replace the last comma in the 'Location' column with a ';' using a regular expression:

```
return $(Location).replace(/(.*),([^ ]*)$/, '$1;$2');
```

To calculate the number of years difference between Javascript compatible dates in column 1 and column 2:

```
return new Date( $(1) ).getFullYear() - new
Date( $(2) ).getFullYear();
```

To calculate the number of milliseconds between a datetime in the 'datetime' column and 1st Jan 2000:

```
return new Date( $(date) ) - new Date( "2000-01-01T00:00" );
```

To calculate the number of whole days difference between a date in the 'created' column and today (negative for future dates):

```
return Math.floor( ( new Date() - new Date( $(created) ) ) /
( 1000*60*60*24 ) );
```

To use the value of the 'n' column if it is a number and 0 if it isn't:

```
if ( isNaN( $(n) ) )
   return 0;
else
   return $(n);
```

To reverse the text in the 'key' column:

```
var a = $(key);
var newString = "";
for (var i = a.length - 1; i >= 0; i--) {
   newString += a[i];
}
return newString;
```


© 2023 Oryx Digital Ltd

To convert meters to feet using a function:

```
function metersToFeet( x )
{
   return 3.28084 * x;
}
return metersToFeet( $(meters) );
```

## Inputs

One.

## Options

- Set **New column name** to the name of the new column you want to create.
- Set **Type column as** to:
  - **Number/Boolean/String** to pass:
    - A column that is numeric values as Javascript Number type.
    - A column that is `true` and `false` values as Javascript Boolean type.
    - Anything else as Javascript String type.
  - **String** to pass data all columns with Javascript String type.
- Enter your script into the **Javascript** field. The script should be the body of a Javascript function.
- Select a column from **Insert variable** to add that column variable into the **Javascript** field at the current cursor position.
- Click the **Evaluate** button to evaluate your Javascript expression over every row and show any errors (only available if **Run>Auto Run** is checked).

## Notes

- If **Run>Auto Run is** checked, the **Javascript** transform is calculated every time:
  - The **Evaluate** button is pressed.
  - The **Javascript** transform item is unselected in the **Center** pane and script changes have been made without the **Evaluate** button being clicked.
  - An item upstream of it changes.
- An empty string is passed as an empty Javascript string variable (""), not `null`. To test if the 'v' column has a blank or empty value:

```
if ( $(v).length === 0 )
{
   // empty cell
}
else if ( String( $(v) ).trim().length === 0 )
{
   // blank cell
}
```


© 2023 Oryx Digital Ltd

- Numeric values should use dot ('.') as the decimal separator and have no group separator. E.g. 1234.5 is valid, but 1,234.5 and 1.234,5 are not, regardless of the locale. You can use the Num Format and Replace transforms to put numeric data in the correct format before processing a **Javascript** transform.
- The new column is added at the right end. You can change the column order with Reorder Cols and the column name with Rename Cols.
- The Javascript Date() object evaluates to the number of milliseconds since 1 January 1970 UTC.
- Date values passed to Javascript Date() objects should be in ISO ('yyyy-mm-dd') format, e.g. '2020-01-31' (not '2020-1-31').
- If you want to carry out your transform across more than one dataset, you should merge them first, e.g. using Join.
- The Javascript transform is very versatile and quite fast. But is not as fast as built-in transforms. So we recommend you use built-in transforms where possible.
- Javascript running in Easy Data Transform is not 'sandboxed' and has the same privileges as the Easy Data Transform executable. However the Javascript does not have access to window(), XMLHttpRequest() or ActiveXObject(). So we aren't aware of any way that a bad actor could damage your system from Javascript sent in a .transform file.
- Javascript is far too big a topic to cover here. However there are many detailed resources online. If you are stuck contact support.
- If you only want to combine text from columns, use the simpler Substitute transform.
- Warnings are shown in the **Warnings** tab for:
  - Javascript syntax errors
  - Ambiguous column references
- For simple arithmetic operations on numbers and dates use the Calculate transform.

## See also

- Find the difference between dates/datetimes

### 2.3.28 Join

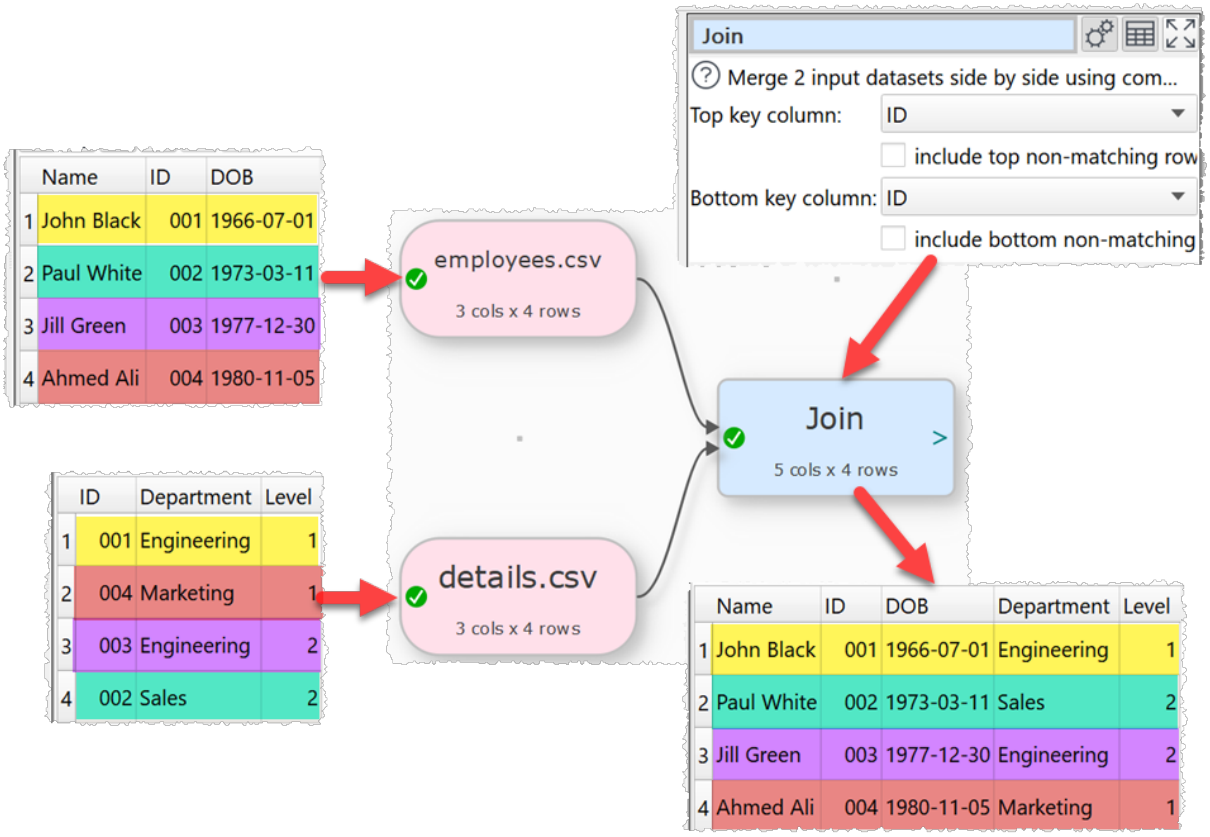
## Description

Merge two inputs based on common (key) columns, e.g. on an email address or id column present in both inputs. The bottom dataset (by vertical position) is joined to the right of the top dataset.

## Example

Join two datasets side-by-side using the 'ID' columns:


© 2023 Oryx Digital Ltd




## Inputs

Two.

## Options

- The output depends on the vertical (Y-axis) position of the inputs.
- Click **Explore Keys...** to compare key values in the 2 datasets.
- Select **Top key column** for the column you want to match in the top input dataset.
- Select **include top non-matching rows** if you want to include in the output any rows in the top input with no matching value in the bottom input.
- Select **Bottom key column** for the column you want to match in the bottom input dataset.
- Select **include bottom non-matching rows** if you want to include in the output any rows in the bottom input with no matching value in the top input.
- Check **detailed report** to show detailed information in the **Warnings** and **Info** tabs on duplicate keys in each dataset, overlap of keys in the 2 datasets and leading and trailing whitespace in keys. This slows down the join significantly.
- Check **case sensitive** to use case sensitive matching for keys.

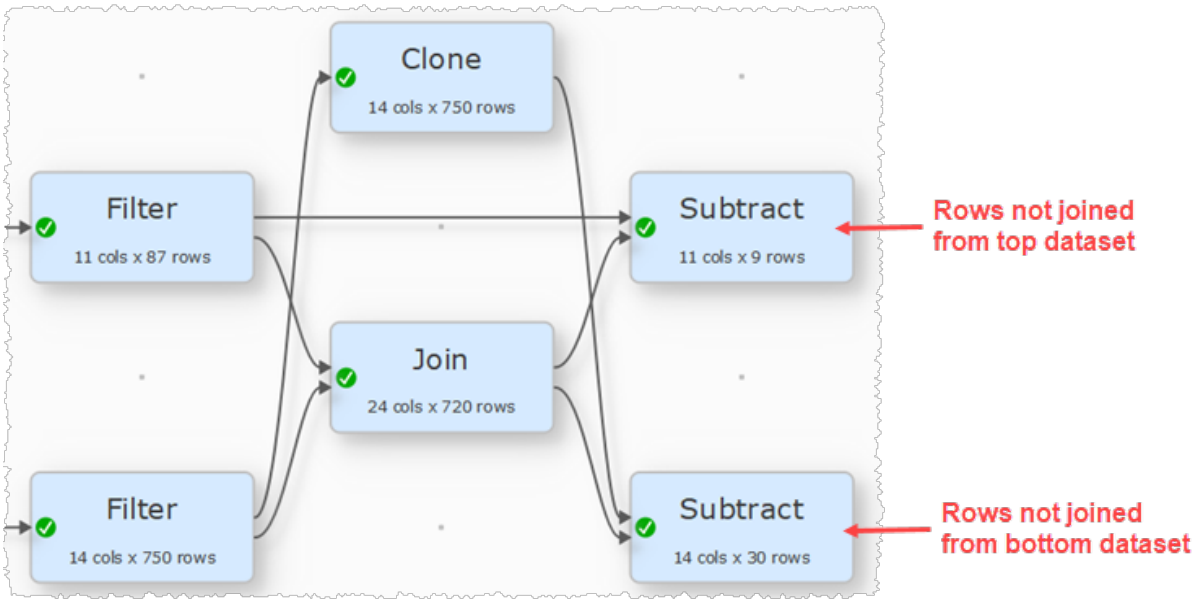

*© 2023 Oryx Digital Ltd*

| **include top non-matching rows** checked | **include bottom non-matching rows** checked | Also known as: |
|---|---|---|
| No | No | Inner join |
| No | Yes | Right outer join |
| Yes | No | Left outer join |
| Yes | Yes | Full outer join |

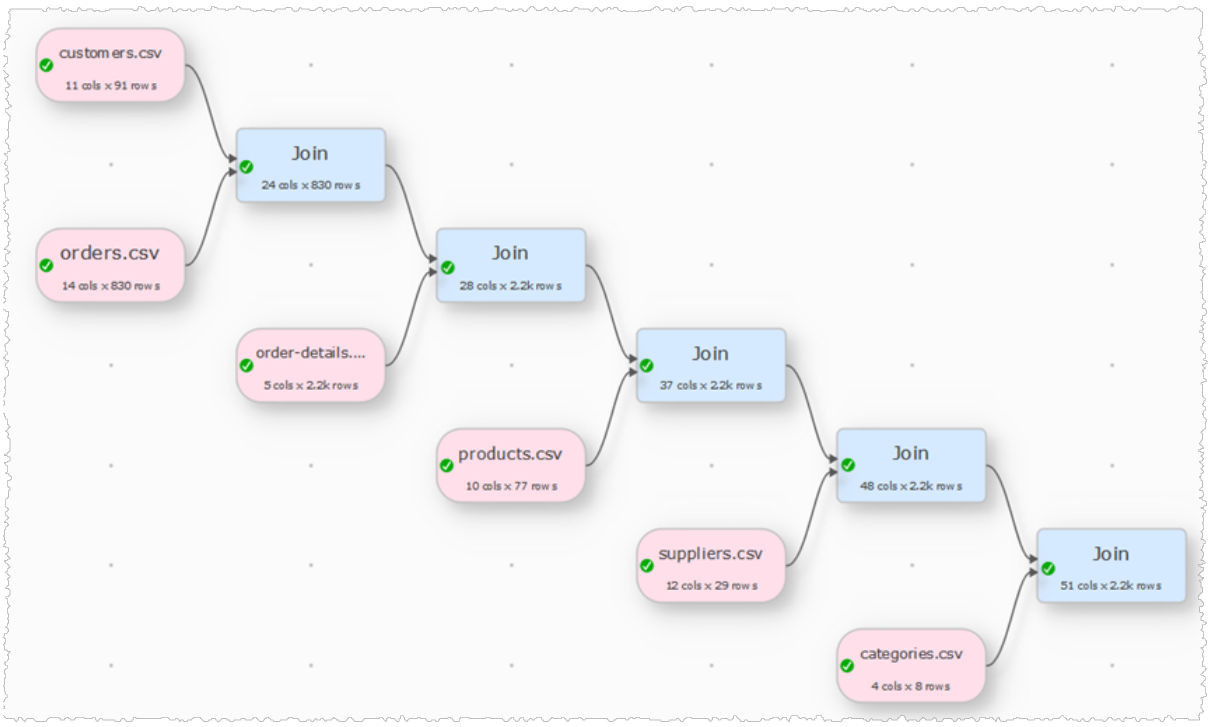
## Notes

- Join merges two datasets side-by-side (horizontally). To merge datasets one on top of the other (vertically) use Stack.
- If there are 10k rows or less in both datasets, Easy Data Transform will try to guess sensible default values for **Top key column** and **Bottom key column** based on column header names and contents.
- All values are treated as text. You can use Whitespace to remove whitespace before the join.
- If a key value occurs M times in the first dataset and N times in the second dataset, you will get M x N rows with this key value. You can use Unique to remove rows with duplicate key values.
- If you need more than one column for the key, use Concat Cols to create that column.
- Use Row Num to create a unique key column.
- Use Sort to sort the results.
- Use the Cross transform for cross joins.
- Messages are shown in the **Warnings** tab for keys that occur more than once in either dataset (duplicates).
- Messages are shown in the **Info** tab for keys that only occur in 1 of the 2 datasets (misses).
- Use Subtract to see rows not joined from the top or bottom dataset (the Clone is necessary to get the correct Y position of the bottom dataset for the **Subtract**).


*© 2023 Oryx Digital Ltd*




- Cascade multiple joins to join more than 2 datasets.




## See also

- Video: How to join Excel files
- Cross
- Lookup
- Interpolate
- Merge datasets


*© 2023 Oryx Digital Ltd*

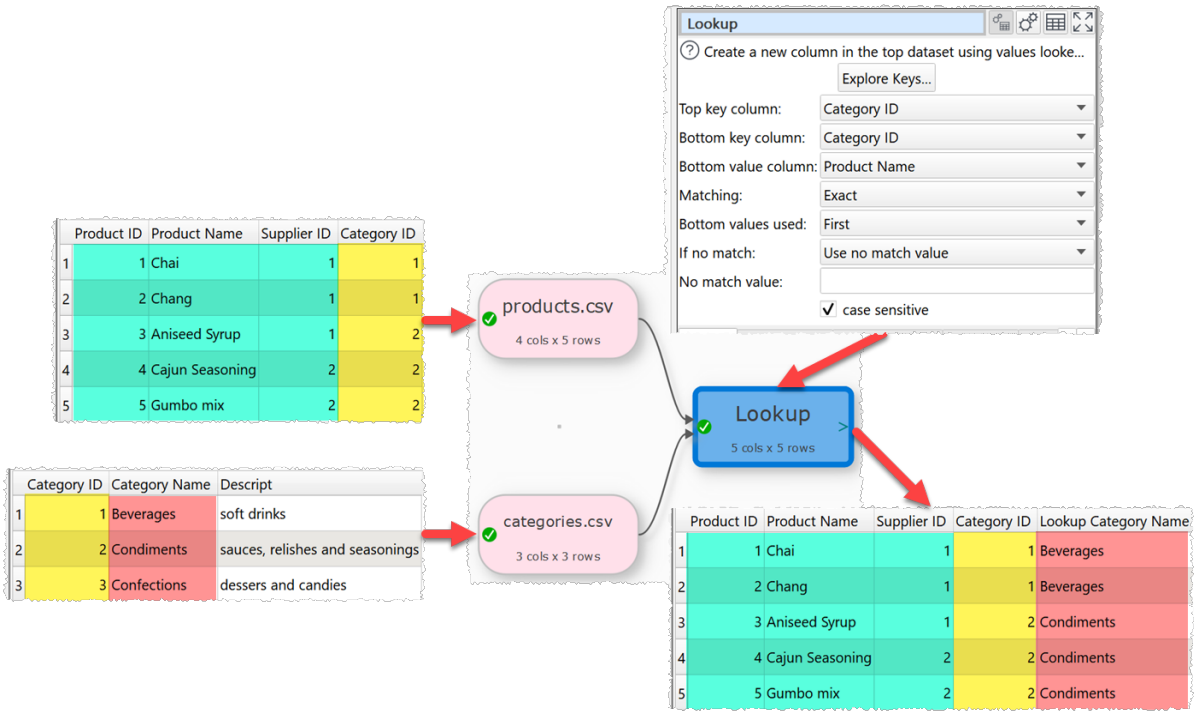
### 2.3.29 Lookup

## Description

Looks up the values of a column in the top input dataset in the bottom input dataset and puts the result in a new column.

## Examples

Lookup category name in a second dataset using 'Category ID':

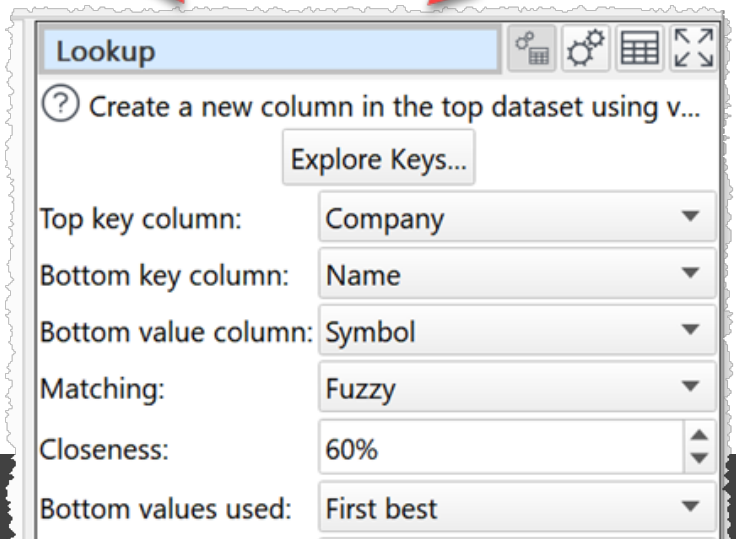



Fuzzy lookup of company stock 'Symbol' by 'Name':


*© 2023 Oryx Digital Ltd*

| | Company |
|---|---|
| 1 | Amazon |
| 2 | Bank of America |
| 3 | Home Depot |
| 4 | McDonalds |
| 5 | NVIDAI |
| 6 | Pepsi |

| | Name | Symbol |
|---|---|---|
| 3 | Amazon.com | AMZN |
| 4 | Bank of America | BAC |
| 11 | Home Depot (The) | HD |
| 17 | McDonald's | MCD |
| 21 | NVIDIA | NVDA |
| 24 | PepsiCo | PEP |





© 2023 Oryx Digital Ltd

## Inputs
Two.

## Options
- Place the dataset you want to modify as the top input and the dataset you want to lookup values from as the bottom input.
- Click **Explore Keys...** to compare key values in the 2 datasets.
- Select **Top key column** for the column whose values you wish to lookup.
- Select **Bottom key column** for the column that matches the lookup in the bottom dataset.
- Select **Bottom value column** for the column that contains the values.
- Set **Matching** to **Exact** to use exact matching to find duplicates and **Fuzzy** to match with some differences. Fuzzy matching is much slower.
- Set **Closeness** to how close a fuzzy match has to be to be considered a match. E.g. set it to 80% to make 2 values that are 80% the same a match. For **Fuzzy** matching only.
- Set **Bottom values used** to:
  - **First** if you want use the first match in **Bottom lookup column**. For **Exact** matching only.
  - **First best** if you want to use the best match. If there are multiple values that are equally good, use the first one. For **Fuzzy** matching only.
  - **All best** if you want to use the best matches. If there are multiple values that are equally good, use all of them. Duplicate values are only shown once. For **Fuzzy** matching only.
  - **All** if you want to use all matches. Duplicate values are only shown once.
- Set **If no match** to **Use not match value** or **Leave unchanged**, depending on what you want to do for values in **Top lookup column** that do not match in **Bottom lookup column**.
- Set **No match value** to the value you want to use for values in **Top lookup column** that do not match in **Bottom lookup column** when **If no match** is set to **Use no match value.**
- Check **case sensitive** to use case sensitive matching for keys.

## Notes
- Easy Data Transform will try to guess sensible default values for **Top lookup column, Bottom lookup column** and **Bottom value column**.
- **Bottom values used** is only important if there are duplicate values in the **Bottom lookup column**.
- If the top input has a header, this will be used for the output.
- If you want to see what bottom dataset keys are matched to each top dataset key, you can set **Bottom value column** to the same column as **Bottom key column**.
- All values are treated as text and comparisons are whitespace sensitive. You can use Whitespace to remove whitespace before the lookup.
- If you want to lookup values in multiple columns, use Concat Cols to join several columns together to form new columns.
- The new column is added at the right end. You can change the column order with Reorder Cols and the column name with Rename Cols.


*© 2023 Oryx Digital Ltd*

- Warnings are shown in the **Warnings** tab for:
  - Values that cannot be found in the bottom dataset (misses).
  - Values that occur more than once in the bottom dataset (duplicates), if **Bottom values used** is **First**.

## See also

- If
- Interpolate
- Join

### 2.3.30 Moving

## Description

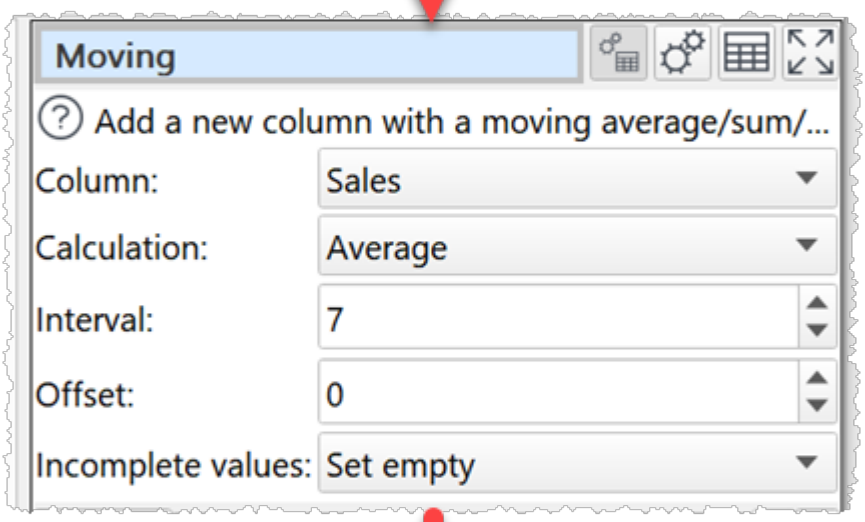
Add a new column with a moving average/median/sum/min/max of the selected column.

## Examples

Calculate a 7 day moving average for the sales column, leaving the first 6 days empty.


*© 2023 Oryx Digital Ltd*

| | Date | Sales |
|---|---|---|
| 1 | 2020-07-01 | 100 |
| 2 | 2020-07-02 | 120 |
| 3 | 2020-07-03 | 90 |
| 4 | 2020-07-04 | 145 |
| 5 | 2020-07-05 | 95 |
| 6 | 2020-07-06 | 75 |
| 7 | 2020-07-07 | 125 |
| 8 | 2020-07-08 | 130 |
| 9 | 2020-07-09 | 95 |



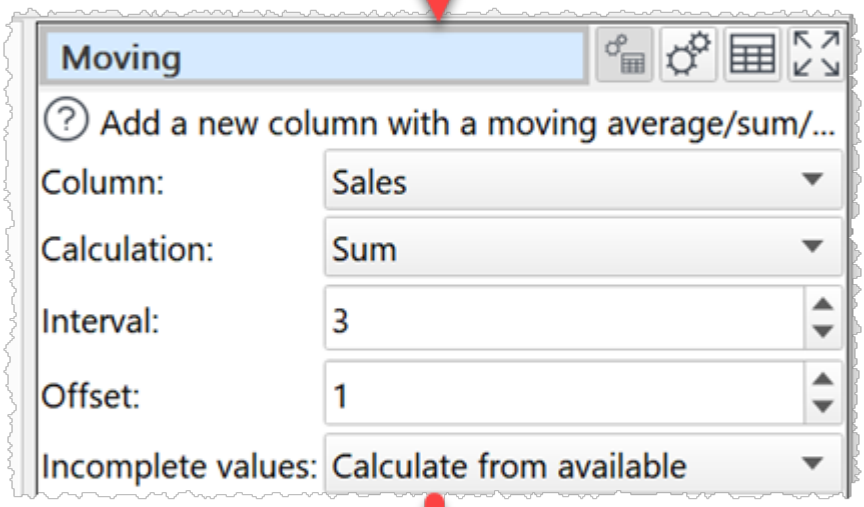

| | Date | Sales | Moving Average Sales |
|---|---|---|---|
| 1 | 2020-07-01 | 100 | |
| 2 | 2020-07-02 | 120 | |
| 3 | 2020-07-03 | 90 | |


© 2023 Oryx Digital Ltd

Calculate the sum of the current row and the rows above and below, where available, for the sales column.


*© 2023 Oryx Digital Ltd*

| | Date | Sales |
|---|---|---|
| 1 | 2020-07-01 | 100 |
| 2 | 2020-07-02 | 120 |
| 3 | 2020-07-03 | 90 |
| 4 | 2020-07-04 | 145 |
| 5 | 2020-07-05 | 95 |




| | Date | Sales | Moving Sum Sales |
|---|---|---|---|
| 1 | 2020-07-01 | 100 | 220 |
| 2 | 2020-07-02 | 120 | 310 |
| 3 | 2020-07-03 | 90 | 355 |
| 4 | 2020-07-04 | 145 | 330 |
| 5 | 2020-07-05 | 95 | 240 |


© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Select the **Column** you wish to calculate a moving average/median/sum/minimum/maximum for.
- Set **Calculation** to the statistic to be calculated.
  - **Average** shows the arithmetic mean of the values in the interval (Simple Moving Average).
  - **Median** shows the median of the values in the interval.
  - **Sum** show the sum of the values in the interval.
  - **Minimum** shows the smallest value in the interval.
  - **Maximum** shows the largest value in the interval.
- Set **Interval** to the number of rows to include in each moving average/median/sum/minimum/maximum value.
- Set **Offset** to the number of rows above (if negative) or below (if positive) the current row to end each interval at.
- Set **Incomplete values** according to what you want to show for rows without a full interval of data. For example the first 6 rows when **Interval** is 7 and **Offset** is 0.

## Notes
- Non-numeric values are ignored. E.g. a 7 day moving average that includes 2 non-numeric values will be shown as an average of the 5 numeric values.
- You will be warned of any non-numerical values in the selected column.

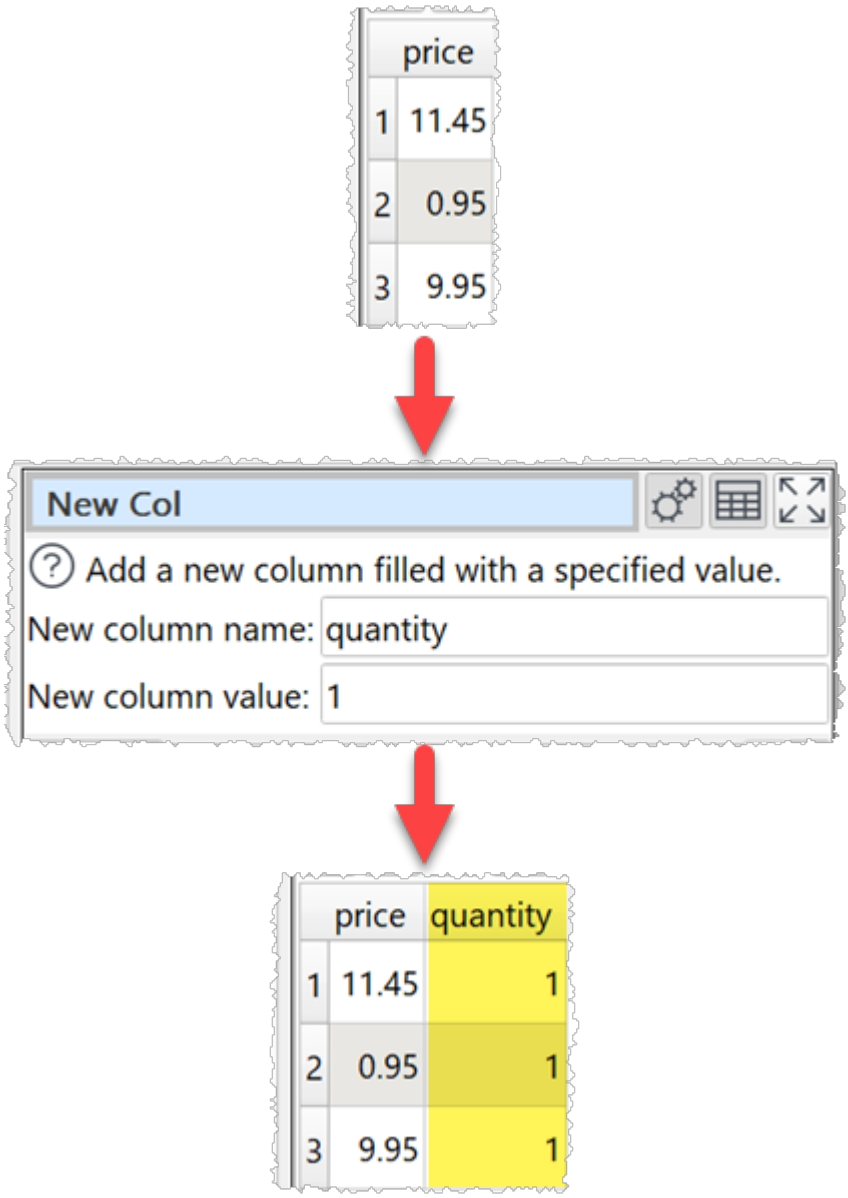
## See also
- Video: How to calculate moving averages
- Offset
- Stats

### 2.3.31 New Col

## Description
Adds a new column, filled with a given value.

## Example
Add a new column full of '1' values:


© 2023 Oryx Digital Ltd




## Inputs

One.

## Options

- Set **New column value** to the value for every cell of the new column. You can leave it blank for an empty column.

## Notes

- New columns are always added at the right end. You can change the column order with Reorder Cols and the column name with Rename Cols.


*© 2023 Oryx Digital Ltd*

### See also

- Copy Cols
- Remove Cols

## 2.3.32 New Rows

### Description

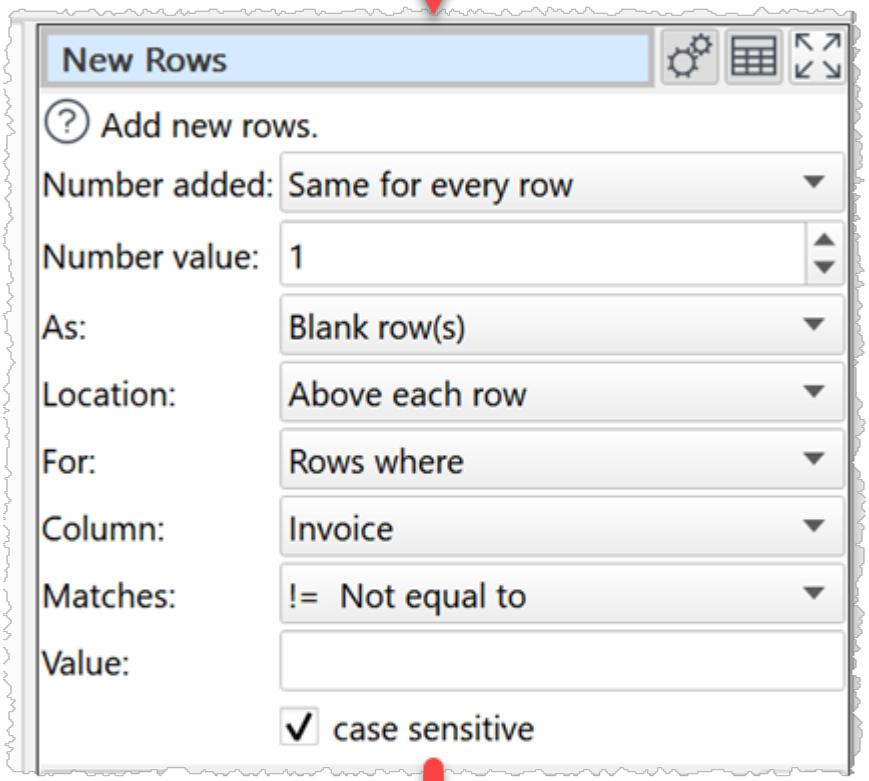
Add new rows.

### Examples

Add a blank row above every row with an invoice number:


© 2023 Oryx Digital Ltd

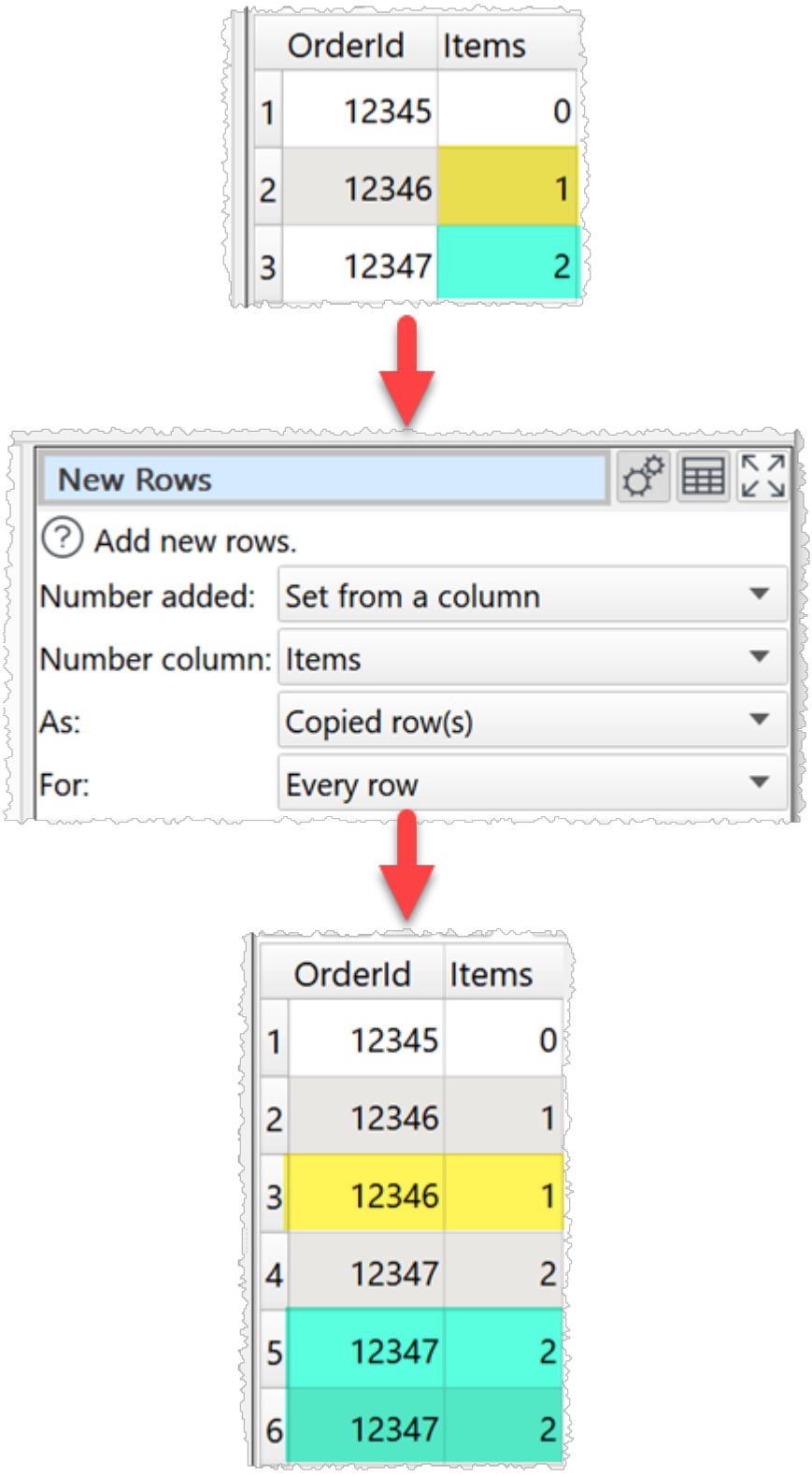
| | Invoice | Line Item | Value |
|---|---|---|---|
| 1 | 00456 | 1 | 150.00 |
| 2 | | 2 | 95.00 |
| 3 | | 3 | 50.0 |
| 4 | 00457 | 1 | 845.00 |
| 5 | | 2 | 12.50 |




| | Invoice | Line Item | Value |
|---|---|---|---|
| 1 | | | |
| 2 | 00456 | 1 | 150.00 |
| 3 | | 2 | 95.00 |
| 4 | | 3 | 50.0 |


© 2023 Oryx Digital Ltd

Copy each row the number of times in the 'Items' column:





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Set **Number added** according how many rows to add for each existing row.
  - Set **Number value** to the number of new rows to create for each existing row (available when **Number added** is **Same for every row**).
  - Set **Number column** to get the number of new rows to create for each existing row from a column (available when **Number added** is **Set from a column**).
- Set **As** to add either blank rows, copies of existing rows or user defined rows.
- Set **With value** to set the value used for each column of the new row (available when **As** is **User defined row(s)**).
- Set **Location** depending on whether you want new rows added above or below existing rows (available when **As** is **Blank row(s)** or **User defined row(s)**).
- Set **For** depending on whether you wish to add rows to all rows or only rows that meet certain conditions.

## Notes
- You can use Calculate or Javascript transforms to calculate the number for the **Number column.** E.g. to subtract 1 from the final number of copies wanted to give the number of new rows.

## See also
- Offset

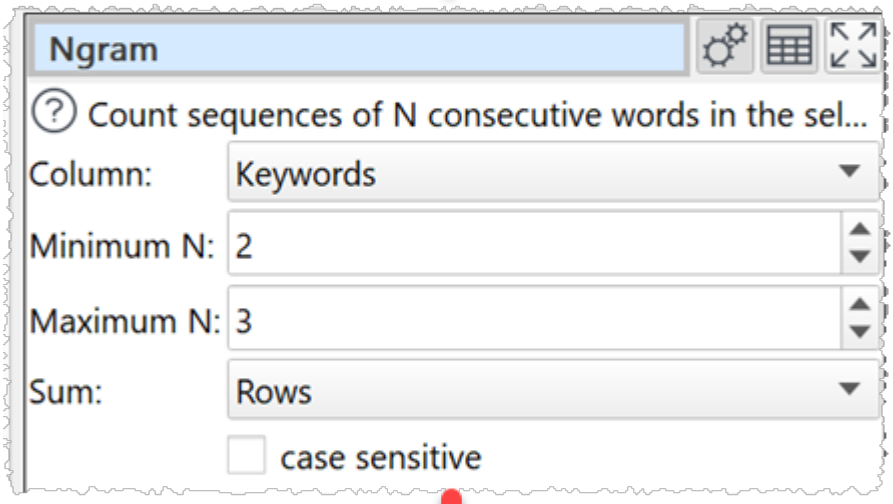
### 2.3.33 Ngram

## Description
Counts the number of times each sequence of consecutive words appear in the selected column.

## Examples
Find ngrams of 2 to 3 word length (bigrams and trigrams) in the 'Keywords' column:


*© 2023 Oryx Digital Ltd*

| | Keywords | Impressions |
|---|---|---|
| 1 | EDT | 47 |
| 2 | easy data transform | 1598 |
| 3 | Easy Data Transform | 964 |
| 4 | easy data transformer | 345 |
| 5 | data transformer | 19343 |
| 6 | EASY,DATA,TRANSFORM | 23 |



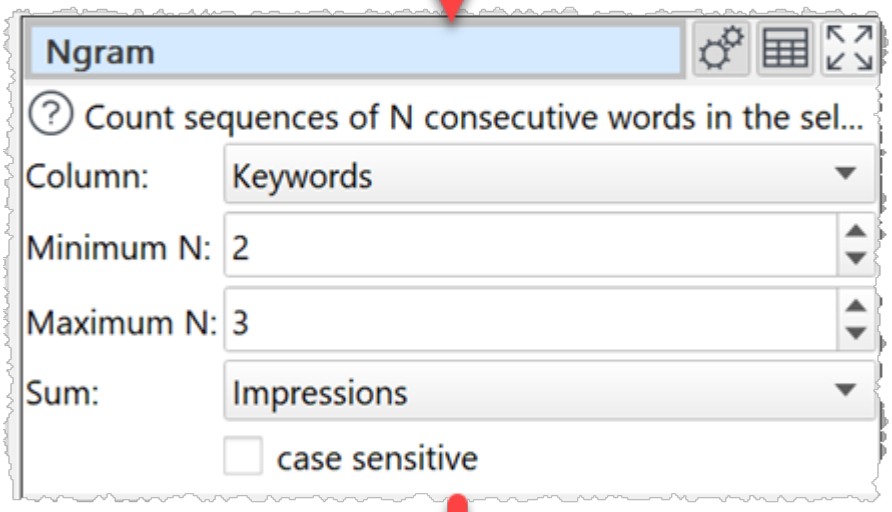

| | Keywords Ngrams | Words | Rows |
|---|---|---|---|
| 1 | easy data transform | 3 | 3 |
| 2 | easy data transformer | 3 | 1 |
| 3 | easy data | 2 | 4 |
| 4 | data transform | 2 | 3 |
| 5 | data transformer | 2 | 2 |


*© 2023 Oryx Digital Ltd*

Sum the impressions associated with each 2 to 3 word ngram:


*© 2023 Oryx Digital Ltd*

| | Keywords | Impressions |
|---|---|---|
| 1 | EDT | 47 |
| 2 | easy data transform | 1598 |
| 3 | Easy Data Transform | 964 |
| 4 | easy data transformer | 345 |
| 5 | data transformer | 19343 |
| 6 | EASY,DATA,TRANSFORM | 23 |




| | Keywords Ngrams | Words | Impressions |
|---|---|---|---|
| 1 | easy data transform | 3 | 2585 |
| 2 | easy data transformer | 3 | 345 |
| 3 | data transformer | 2 | 19688 |
| 4 | easy data | 2 | 2930 |
| 5 | data transform | 2 | 2585 |


*© 2023 Oryx Digital Ltd*

## Inputs
One.

## Options
- Select the **Column** you wish to analyze for ngrams.
- Set **Minimum N** to the minimum number of words in an ngram.
- Set **Maximum N** to the maximum number of words in an ngram.
- Set **Sum** to:
  - **Rows** to count the number of rows containing the ngram.
  - A column to sum numeric values in that column for all rows containing the ngram.
- Uncheck **case sensitive** to convert everything to lower case before counting ngrams.
- Check **drilldown** to allow double-clicking a row in the data table to drilldown to the rows that contributed to this row in the upstream data.

## Notes
- Words are made up of letters, digits and apostrophes ('). All other characters are treated as word separators.
- All letters are converted to lower case.
- If you have set **Sum** to a column then only numeric values in that column will be summed.
- The output sorted by number of words in the ngram, then the count and then the ngram. Use a Sort transform to sort it in a different order.
- Use a Replace transform before the **Ngram** transform to remove/replace any words or letters you don't wish to analyze.

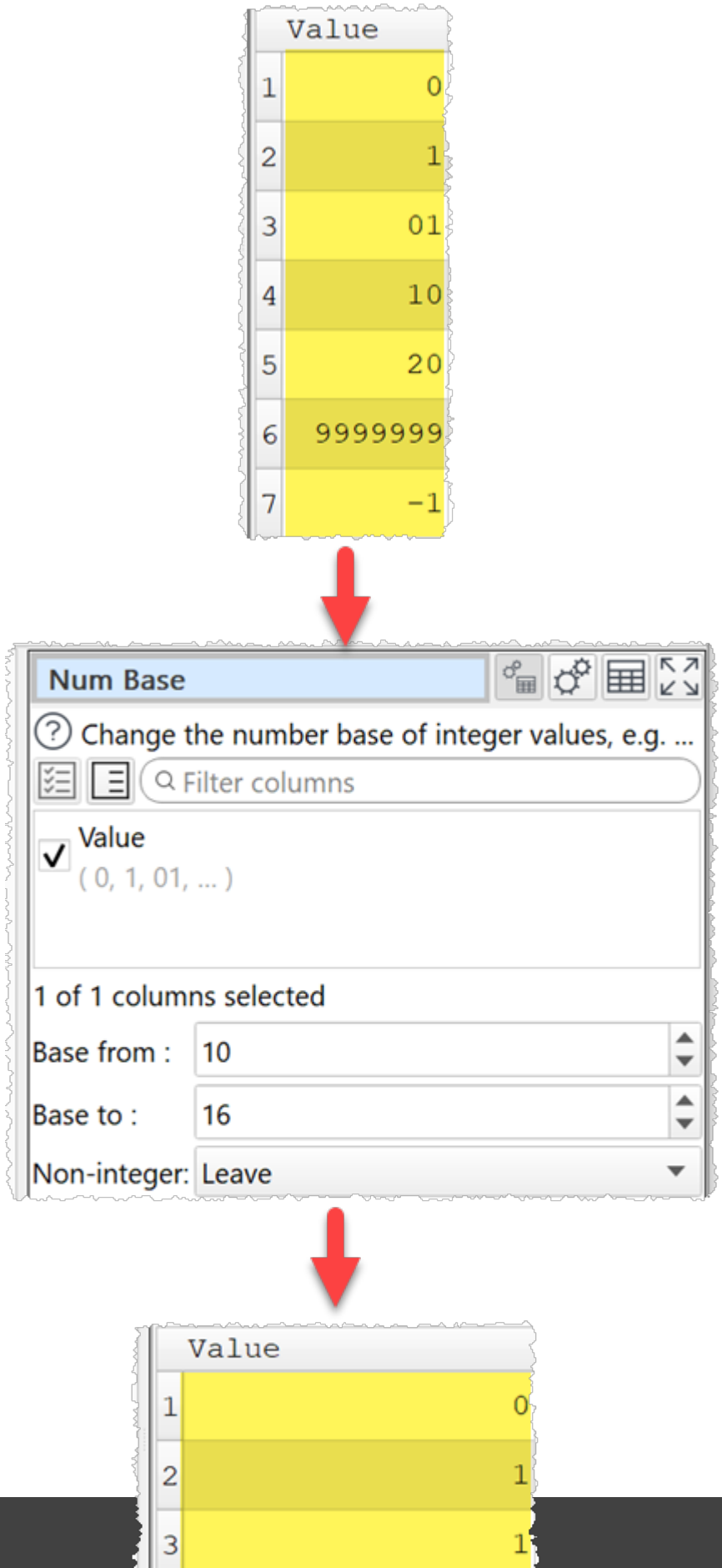
## See also
- Video: How to find NGrams in data for PPC, SEO and NPL
- Count
- Unique
- Total
- Pivot
- Stats
- Summary

### 2.3.34 Num Base

## Description
Change the number base of integer values, e.g. decimal to hexadecimal.

## Examples
Convert from decimal (base 10) to hexadecimal (base 16):

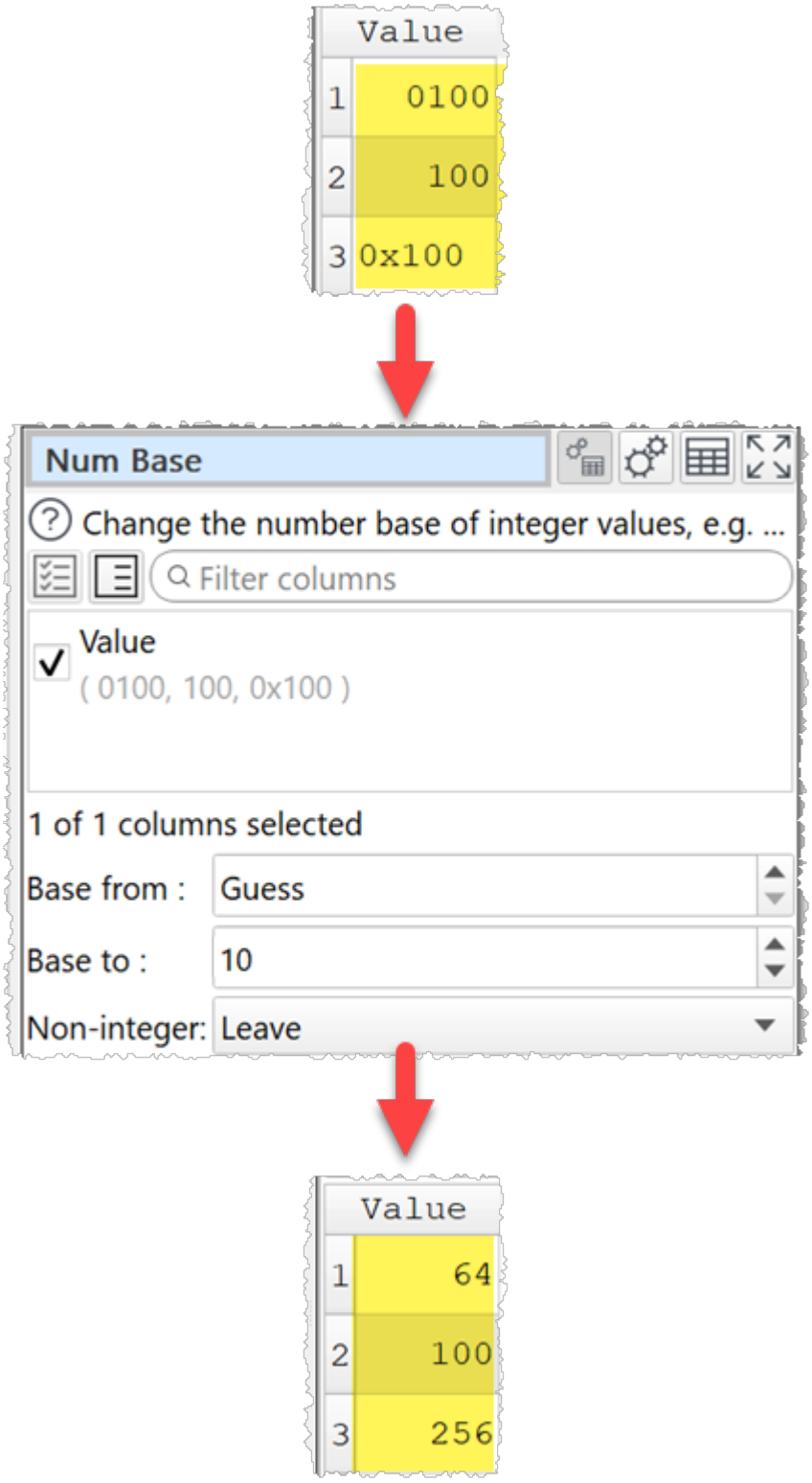

© 2023 Oryx Digital Ltd





© 2023 Oryx Digital Ltd

Convert from guessed base to decimal (base 10):





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) you wish to convert.
- Set **Base from** to the base of the current values. If you set it to **Guess**, it will guess the base for each value using C conventions:
  - Numbers starting with 0x (e.g. `0x123`) are assumed hexadecimal (base 16).
  - Numbers starting with 0 (e.g. `0123`) are assumed octal (base 8).
  - All other values (e.g. `123`) are decimal (base 10).
- Set **Base to** to the base wanted.

## Notes
- The values to be converted should be integers (whole values) without group separators (e.g. no commas or decimal points).
- If **Base from** and **Base to** are the same, no change is made.
- Negative numbers are represented using unsigned values (e.g. twos complement for binary) if **Base to** is anything other than 10.
- Warnings are shown in the **Warnings** tab for non-integer values.
- Use Copy Cols to copy columns before conversion.
- Use Num Format to change the format of numbers (e.g. remove group separators).
- You can use Insert to add characters to the front of values, e.g. to add '0x' to hexadecimal values.

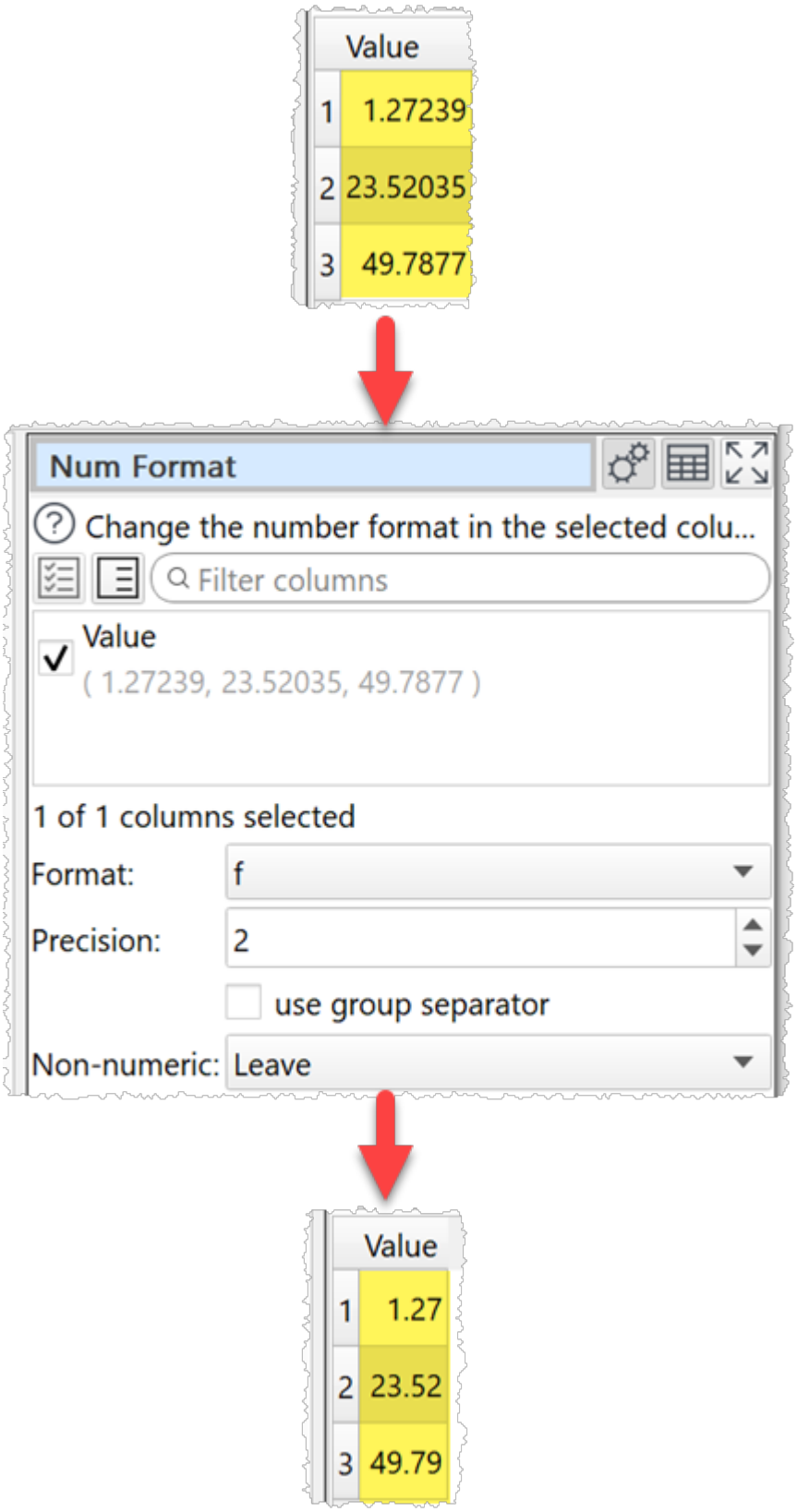
## See also
- Video: How to convert base
- Decode

### 2.3.35 Num Format

## Description
Change the number format in one or more columns.

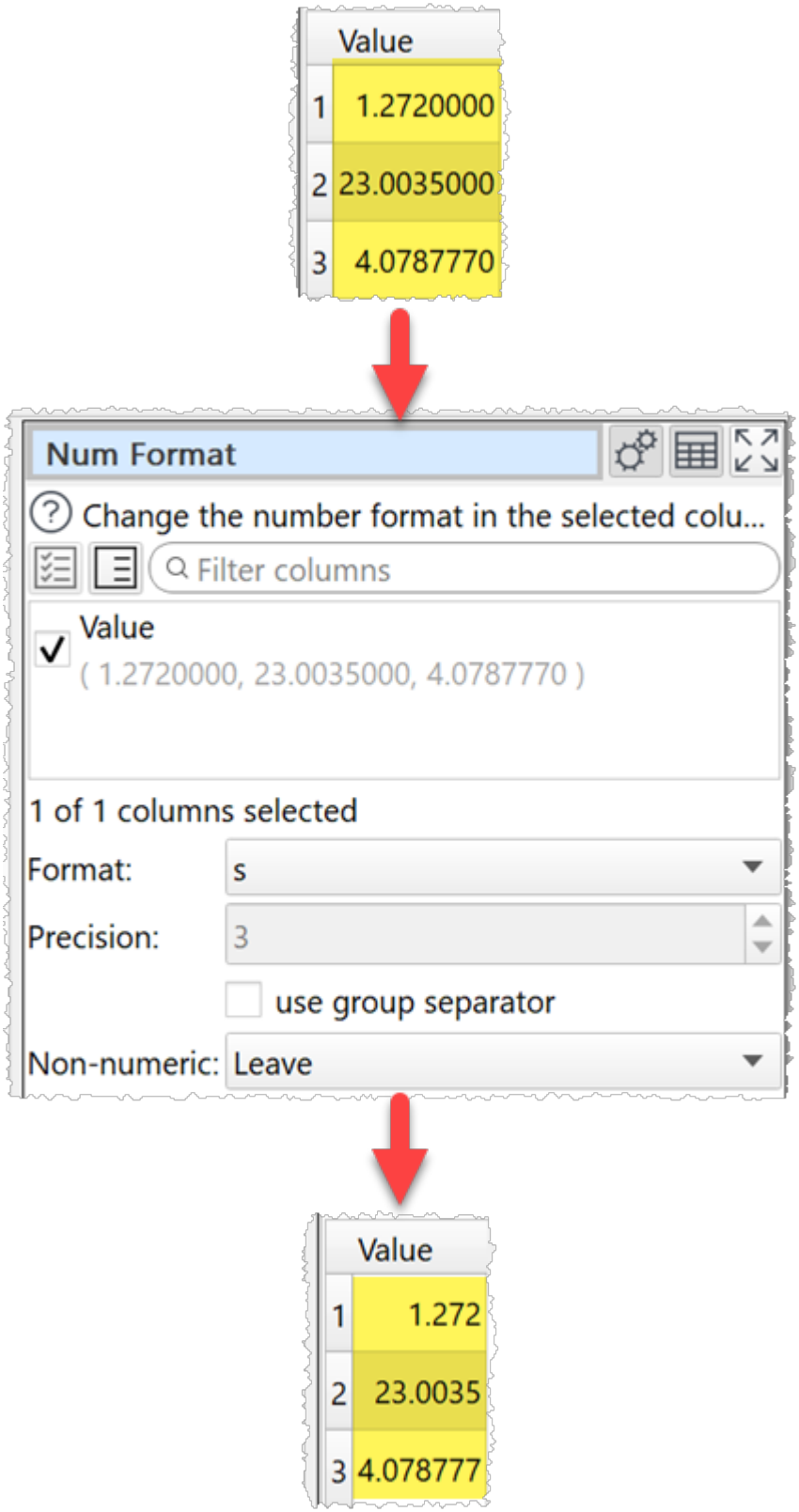
## Examples
Set numeric values to 2 decimal places:


© 2023 Oryx Digital Ltd




Set numeric to shortest accurate number:


*© 2023 Oryx Digital Ltd*





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) you wish to transform.
- Set **Format** to the new number format (see below).
- For the **e**, **E**, and **f** formats, **Precision** represents the number of digits after the decimal point. For the **g** and **G** formats, **Precision** represents the maximum number of significant digits (trailing zeros are omitted). For the **s** format **Precision** is ignored. The following number formats are supported:

| Format | Meaning |
|---|---|
| `e` | Format as [-]9.9e[+\|-]999.<br>E.g. `1234567.89` is shown as `1.235e+06`. |
| `E` | Format as [-]9.9E[+\|-]999.<br>E.g. `1234567.89` is shown as `1.235E+06`. |
| `f` | Format as [-]9.9.<br>E.g. `1234567.89` is shown as `1234567.89`. |
| `g` | Use e or f format, whichever is the most concise. |
| `G` | Use E or f format, whichever is the most concise. |
| `s` | The shortest accurate representation for the given number without exponents.<br>E.g. `1234567.00` is shown as `1234567`. |

- Check **use group separators** to include the group separators for your locale. E.g. to turn `1234567` to `1,234,567` for a UK or US locale.
- Set **Non-numeric** according to what you want to do with non-numeric values in transformed columns. NaN means Not a Number.

## Notes
- Numbers should be base 10 (decimal).
- The locale is used to decide how the number is represented (e.g. group and decimal separators).
- Non-numerical values are ignored.
- You can also use Extract and Pad to change the number of characters.
- Warnings are shown in the **Warnings** tab for non-numeric values.


*© 2023 Oryx Digital Ltd*

## See also

- DateTime Format

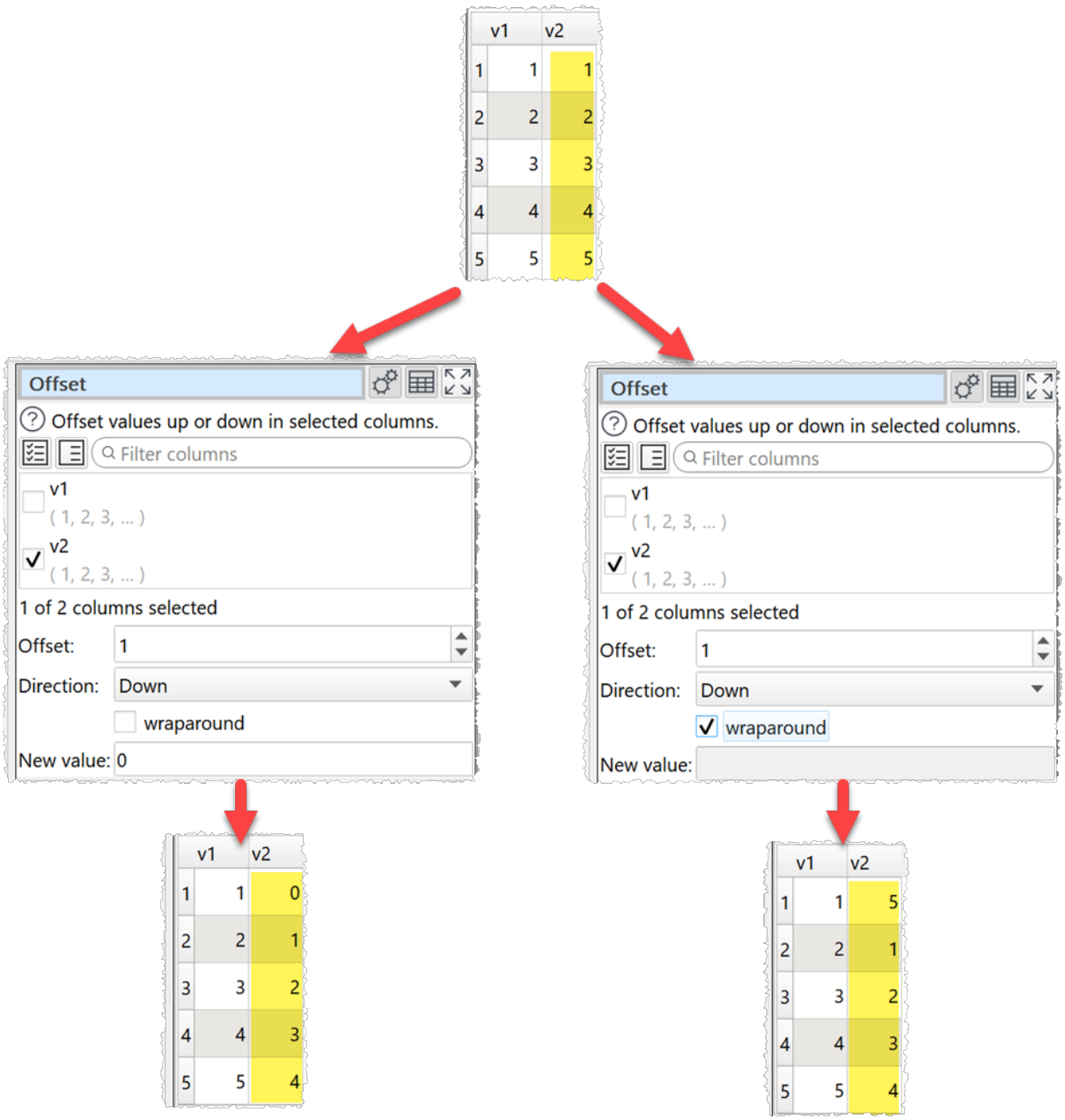
### 2.3.36 Offset

## Description

Moves values in selected columns up or down.

## Example

Move the 'v2' values down 1:





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) you wish to transform.
- Set **Offset** to the number of rows you wish to offset values by.
- Set **Direction** to the direction you want to offset in.
- Check **wraparound** if you want values pushed off the bottom of the table to be added to the top or values pushed off the top to be added to the bottom.
- Set **New value** to the value you wish to set for any cells newly created by the offset.

## Notes
- You can use Copy Cols to create copies of columns before you offset them.

## See also
- New Rows

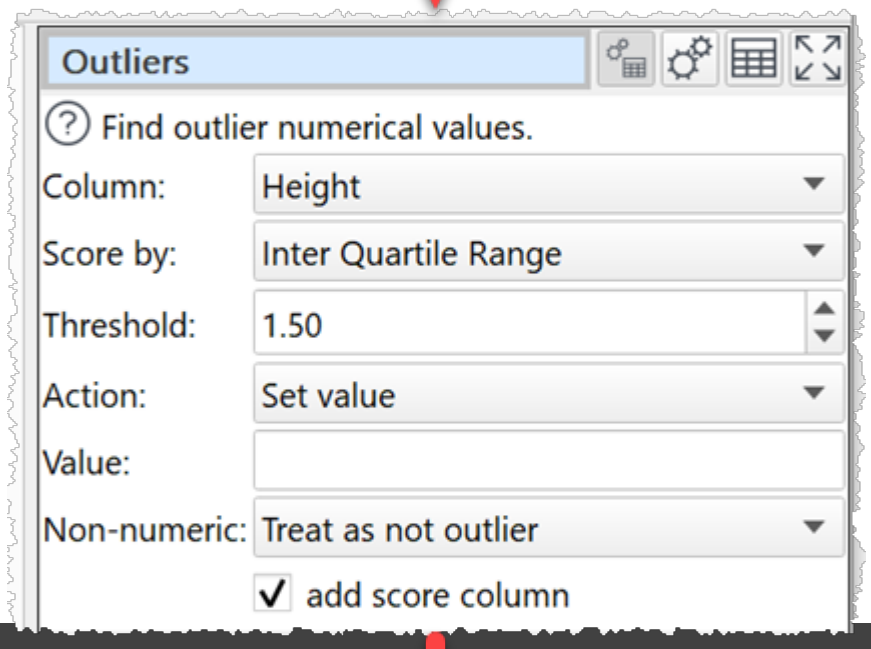
### 2.3.37 Outliers

## Description
Find outlier numerical values.

## Examples
Set outliers empty based on quartiles and add a column with the outlier score:


© 2023 Oryx Digital Ltd

| | Height |
|---|---|
| 1 | 0.00 |
| 2 | 1.59 |
| 3 | 1.71 |
| 4 | 1.71 |
| 5 | 1.73 |
| 6 | 1.73 |
| 7 | 1.77 |
| 8 | 1.84 |
| 9 | 1.89 |
| 10 | 1.95 |
| 11 | 2.10 |
| 12 | 5.00 |

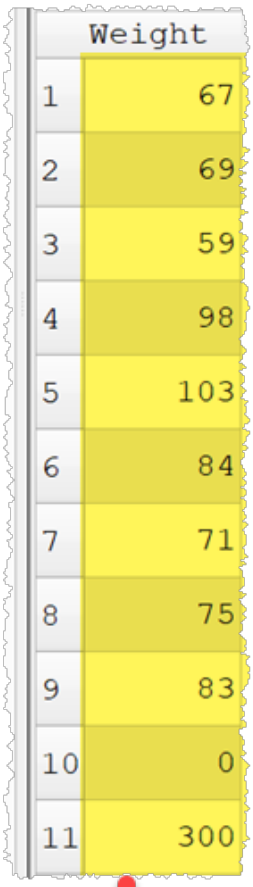




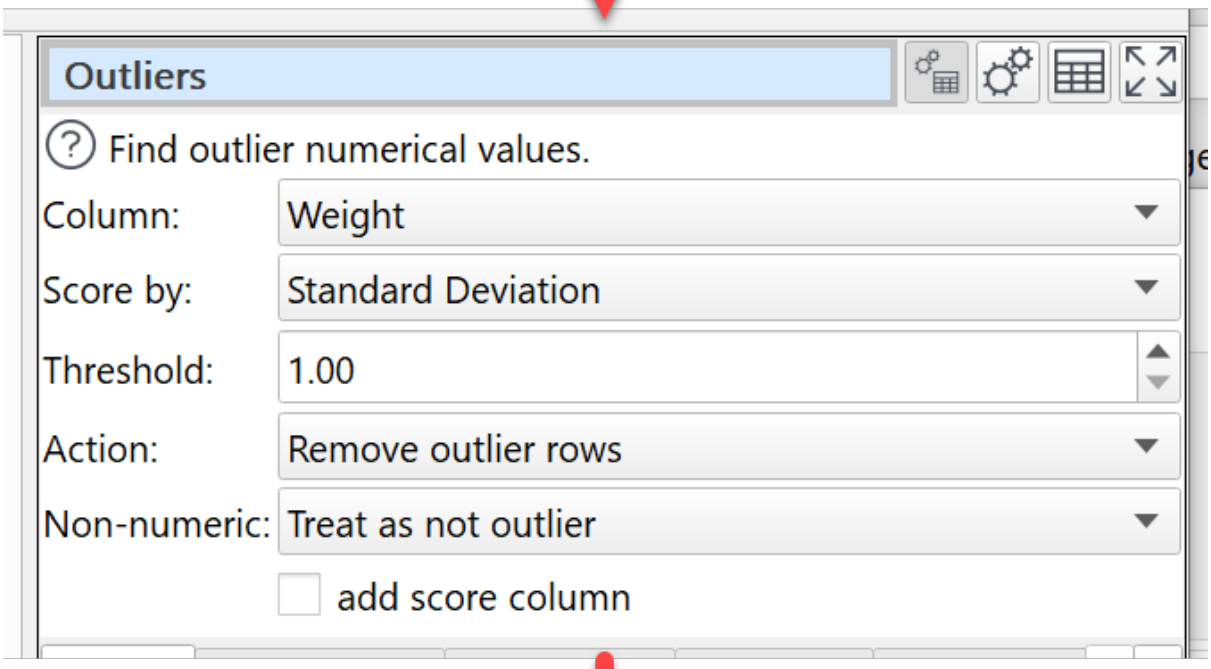
© 2023 Oryx Digital Ltd

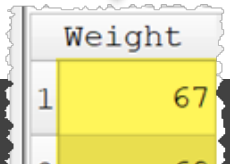
Remove rows with values that are more than 1 Standard Deviation from the mean:


*© 2023 Oryx Digital Ltd*



| | Weight |
|---|---|
| 1 | 67 |
| 2 | 69 |
| 3 | 59 |
| 4 | 98 |
| 5 | 103 |
| 6 | 84 |
| 7 | 71 |
| 8 | 75 |
| 9 | 83 |
| 10 | 0 |
| 11 | 300 |



| | Weight |
|---|---|
| 1 | 67 |


© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Select the column you wish to check for outliers.
- Set **Score by** depending on how you wish to score outliers.
  - **Inter Quartile Range** scores values on how far they are above Q3 or below Q1 as proportion of the Inter Quartile Range. For example:
    - A value 2 IQRs above Q3 scores 2.0.
    - A value 1 IQR below Q1 scores -1.0.
    - A value between Q3 and Q1 scores 0.0.
  - **Standard Deviation** scores values on how many deviations they are from the average (mean). This is also known as a Z-score. For example:
    - A value 2 Standard deviations above the average scores 2.0.
    - A value 1 Standard deviations below the average scores -1.0.
- Any score that is greater than **Threshold** or less than -**Threshold** is considered an outlier. E.g. if the threshold is 1.5 then values that scores more than 1.5 or less than -1.5 are considered outliers. The corresponding upper and lower 'fence' values are shown in the **Info** tab.
- Set **Action** to the action to take for outliers:
  - **Change to threshold**: Value greater than the upper fence value or less than the lower fence value are set to the nearest fence value. Non-numeric values are not changed, as it isn't clear which fence value to set them to.
  - **Remove outlier rows**: Rows with outlier values are removed.
  - **Keep outlier rows**: Only rows with outlier values are kept.
  - **Set value**: Outlier values are changed to the value provided in the **Value** field.
- Set **Non-numeric** depending on whether you want to treat non-numeric values as outliers.
- Check **add score column** if you want to add an extra column with the outlier scoring.

## Notes
- Warnings are shown in the **Warnings** tab for non-numeric values.
- Use Num Format to change the format of numbers (e.g. remove group separators).
- Use Impute to impute any missing values.
- Use Sort to sort by values in the score column.
- You can convert dates into numerical values (e.g milliseconds since 1970 or Julian day) using Calculate or Javascript.

See also:
Video: How to find outliers in data


© 2023 Oryx Digital Ltd

### 2.3.38 Pad

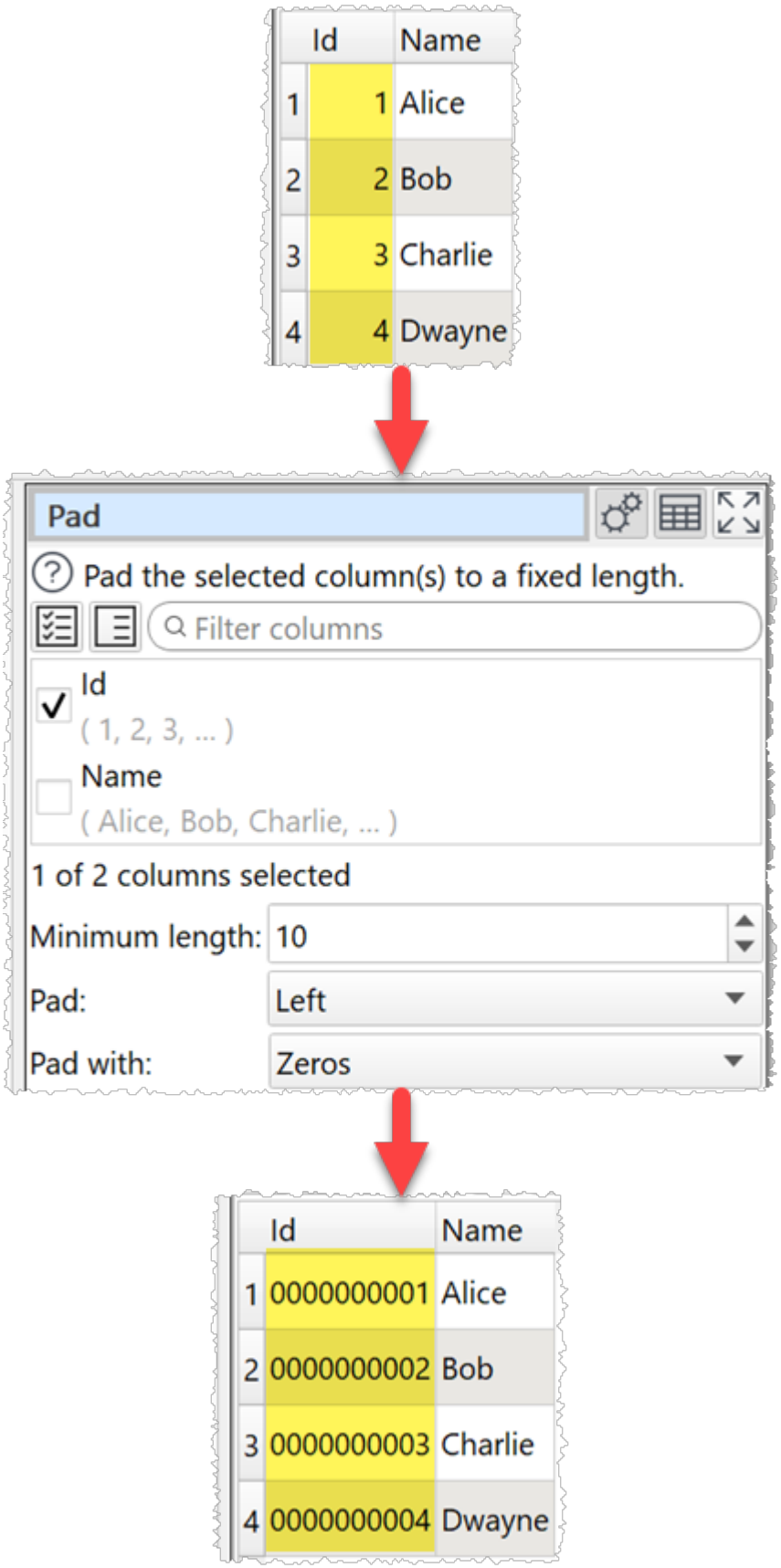
## Description

Pad text to a minimum length in one or more columns.

## Example

Pad the 'Id' column to 10 digits using zeros:


*© 2023 Oryx Digital Ltd*





*© 2023 Oryx Digital Ltd*

## Inputs
One.

## Options
- Check the column(s) you wish to transform.
- Set **Minimum length** to the length you want values in selected columns padded to. Values this length or longer are unaffected.
- Set **Pad** to **Left** or **Right** depending on where you want any padding characters added.
- Set **Pad with** to the character you want to pad with.

## Notes
- Whitespace is counted when calculating length. You can use Whitespace to remove whitespace before padding.

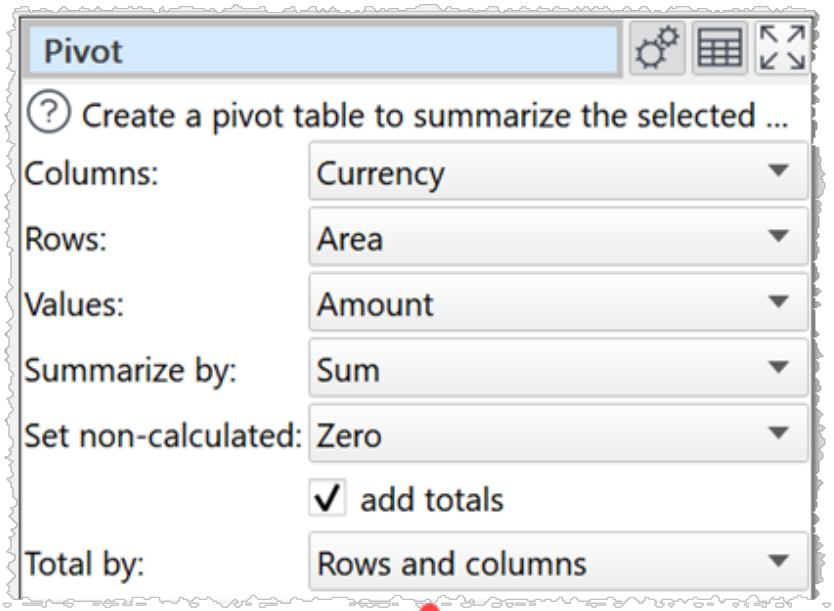
### 2.3.39 Pivot

## Description
Creates a pivot table to summarize values for one or two columns.

## Example
Pivot to sum 'Amount' by 'Area' and 'Currency':


*© 2023 Oryx Digital Ltd*

| | Area | Currency | Amount |
|---|---|---|---|
| 1 | Europe | EUR | 13937.00 |
| 2 | North America | USD | 22894.90 |
| 3 | Africa | ZAR | 12745.55 |
| 4 | North America | USD | 48356.95 |
| 5 | Europe | GBP | 1100.00 |
| 6 | Europe | EUR | 5905.80 |



| | Area | EUR | GBP | USD | ZAR | Grand Sum |
|---|---|---|---|---|---|---|
| 1 | Africa | 0 | 0 | 0 | 12745.55 | 12745.55 |
| 2 | Europe | 19842.8 | 1100 | 0 | 0 | 20942.8 |
| 3 | North America | 0 | 0 | 71251.85 | 0 | 71251.85 |
| 4 | Grand Sum | 19842.8 | 1100 | 71251.85 | 12745.55 | 104940.2 |


*© 2023 Oryx Digital Ltd*

## Inputs
One.

## Options
- Set **Columns** to the column values you want to use as columns in your pivot table.
- Set **Rows** to the column values you want to use as rows in your pivot table.
- Set **Values** to the column you wish to summarize.
- Set **Summarize by** to how you wish to summarize the values:
  - **Sum** show the sum of the values. Non-numeric and empty values are ignored.
  - **Minimum** shows the smallest value. Non-numeric and empty values are ignored.
  - **Maximum** shows the largest value. Non-numeric and empty values are ignored.
  - **Average** shows the arithmetic mean of the values. Non-numeric and empty values are ignored.
  - **Median** shows the median of numeric values in the column. Non-numeric and empty values are ignored.
  - **Mode** shows the mode of numeric values in the column. Non-numeric and empty values are ignored.
  - **Standard deviation** is the sample standard deviation (equivalent to Excel function stddev.s). Non-numeric and empty values are ignored.
  - **Count** shows the number of non-empty values. A value that contains whitespace or 0 is not considered empty.
  - **Count distinct** show the number of non-empty values. Duplicates are not counted. A value that contains whitespace or 0 is not considered empty.
- Set **Set non-calculated** depending on how you want to set cells not calculated by the pivot.
- Check **add totals** to add row and/or column totals.
- Set **Total by** depending on whether you want to total by **Rows and columns**, **Rows** or **Columns**. This is only available if **add totals** is checked and **Columns** and **Rows** are selected.
- Check **drilldown** to allow double-clicking a cell in the data table to drilldown to the rows that contributed to this value in the upstream data.

## Notes
- Column and row values are ordered alphabetically. You can change the order using Reorder Cols (for columns) and Sort (for rows).
- **Mode** may not work as expected with non-integer values, due to precision issues. You can fix this by using a Num Format transform first.
- Use Num Format to change the precision of the results.
- Use Scale to convert the results to percentages.
- To add a column of 1 values to **Sum** (e.g. to find out how many rows a value occurs in) use a New Col transform.


*© 2023 Oryx Digital Ltd*

### See also

- Count
- Stats
- Summary

## 2.3.40 Remove Cols

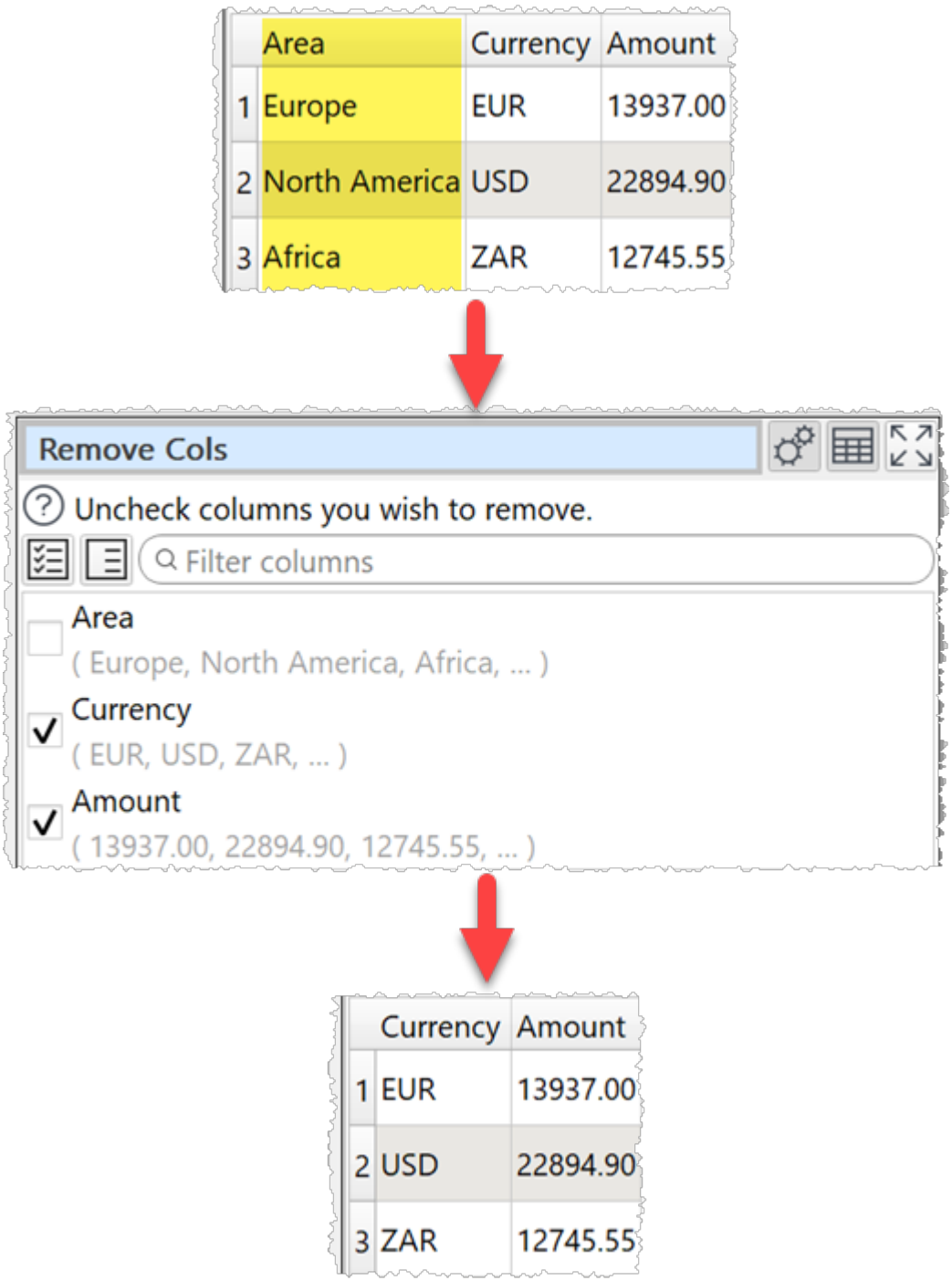
### Description

Removes columns.

### Example

Remove the 'Area' column:


© 2023 Oryx Digital Ltd




## Inputs

One.

## Options

- Uncheck the column(s) you wish to remove.


*© 2023 Oryx Digital Ltd*

## Notes

- The column will be removed from any dataset 'downstream'.

## See also

- New Col

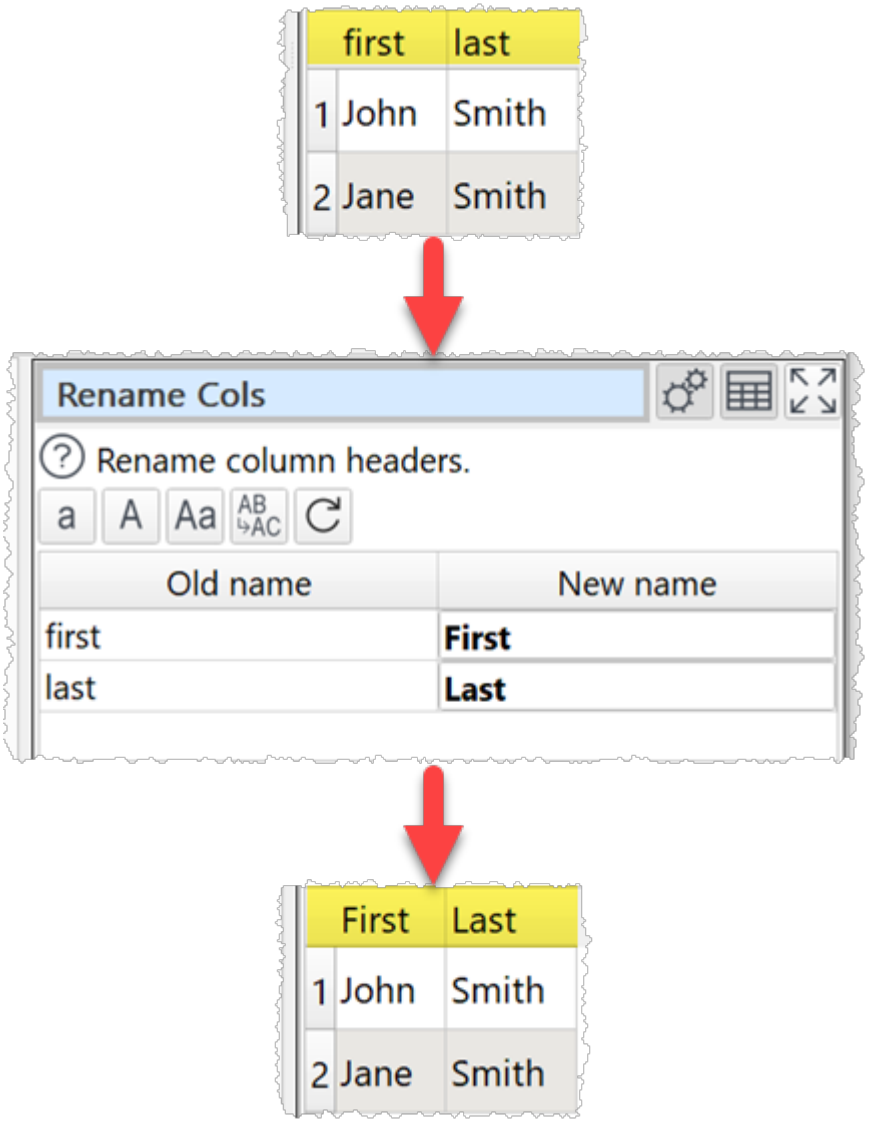
### 2.3.41 Rename Cols

## Description

Rename column headers.

## Example

Rename columns 'first' to 'First' and 'Last' to 'last':





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Change the column headers using the **New name** column.
- Click **Lower** to change all the names to lower case.
- Click **Upper** to change all the names to upper case.
- Click **Title** to change all the names to title case.
- Click **Replace** to replace name text using text, exact text or Regular expression matching.
- Click **Prefix/Suffix** to add text to the front or back of all the names.
- Click **Unique** to add '-2', '-3' etc to duplicate names, to make them unique.
- Click **Reset** to change all the names back to their original name.

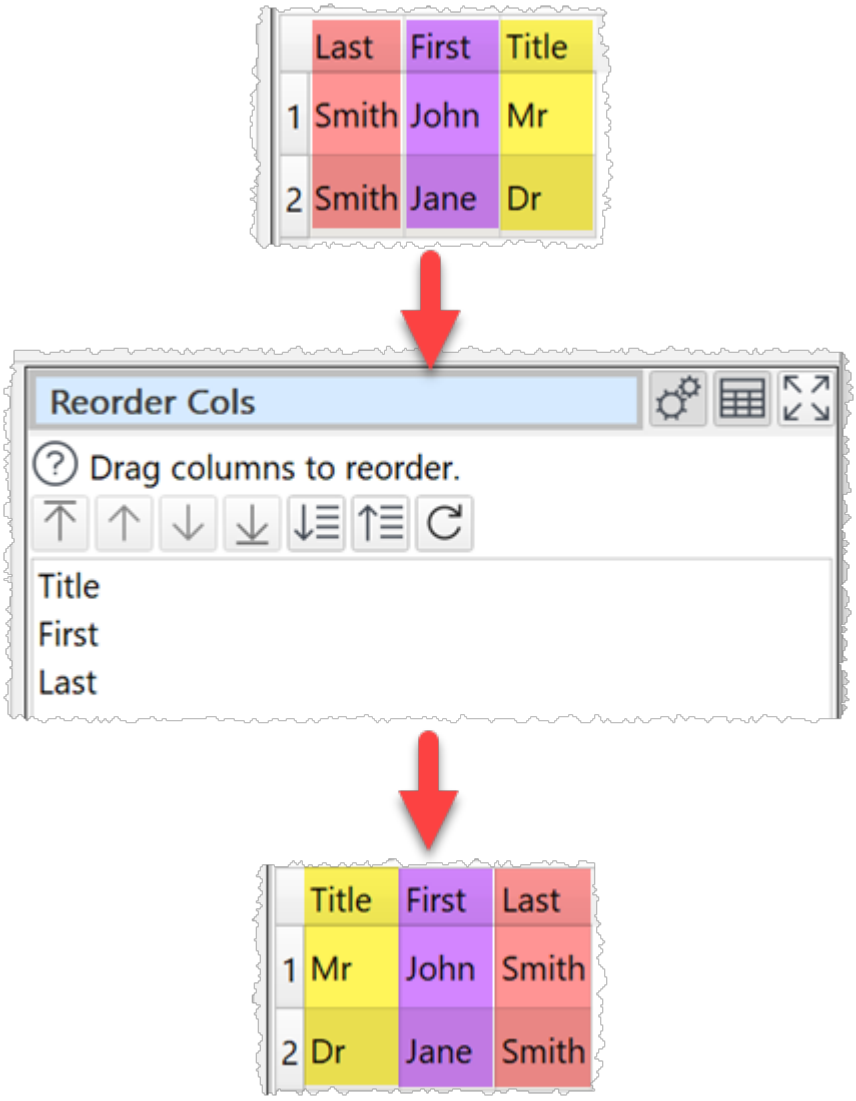
## Notes
- The names of column headers do not have to be unique.
- Warnings are shown in the **Info** tab for duplicate column names (case sensitive).

### 2.3.42 Reorder Cols

## Description
Reorder columns.

## Example
Reorder 'Last', 'First' and 'Title' columns:


© 2023 Oryx Digital Ltd




## Inputs

One.

## Options

Drag the columns into the desired order (left-most at the top).
Click the **Move first** button to move the selected columns to be first.
Click the **Move up** button to move the selected columns up one place.
Click the **Move down** button to move the selected columns down one place.
Click the **Move last** button to move the selected columns to be last.
Click the **Sort ascending** button to sort the columns in ascending order.
Click the **Sort descending** button to sort the columns in descending order.
Click the **Reset** button to restore the original column order.


*© 2023 Oryx Digital Ltd*

## Notes

- You can also rename columns with Rename Cols and remove unwanted columns with Remove Cols.
- Number, date and text values are treated differently for sorting purposes.

### 2.3.43 Replace

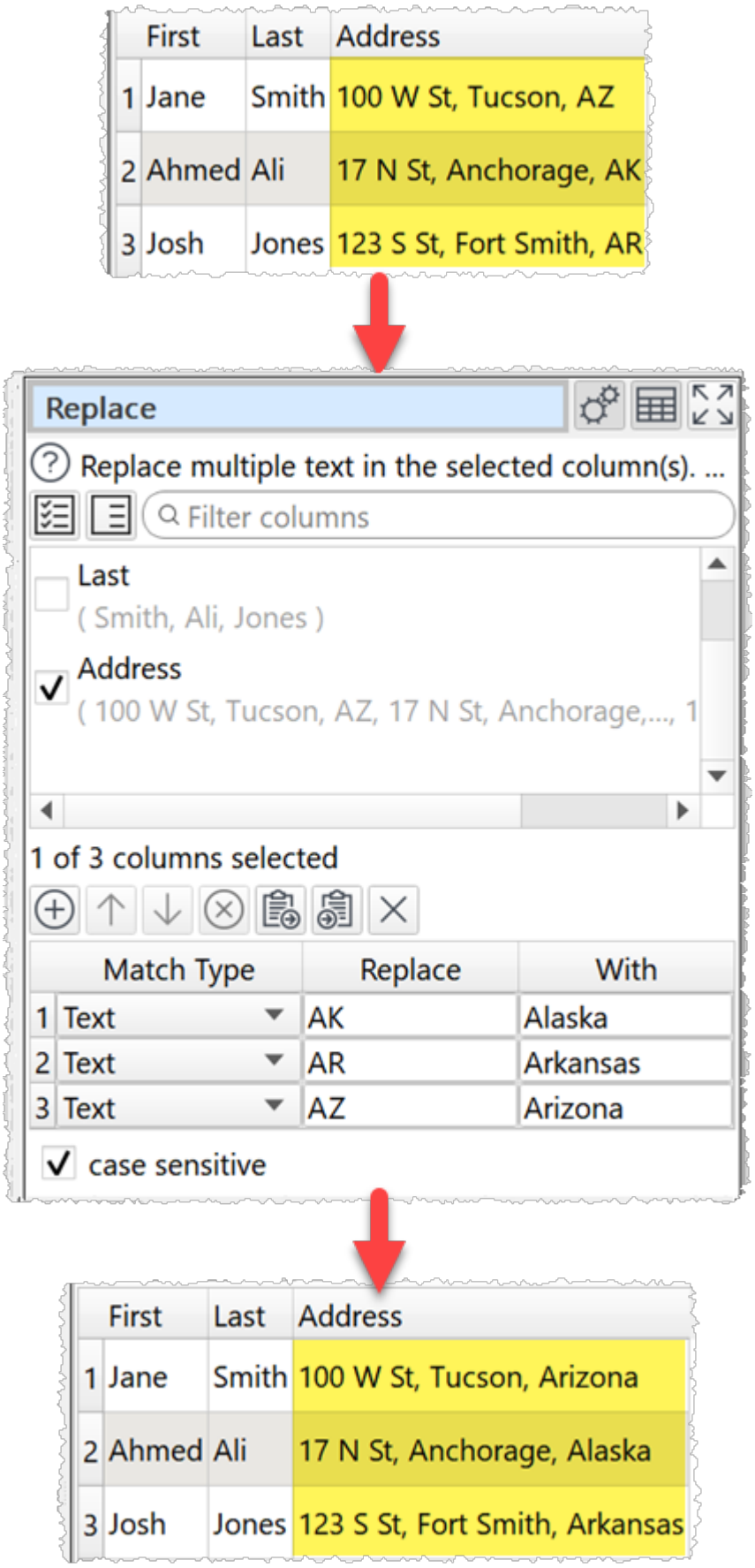
## Description

Replace text in one or more columns.

## Examples

Replace US state initials in the 'Address' column:


© 2023 Oryx Digital Ltd

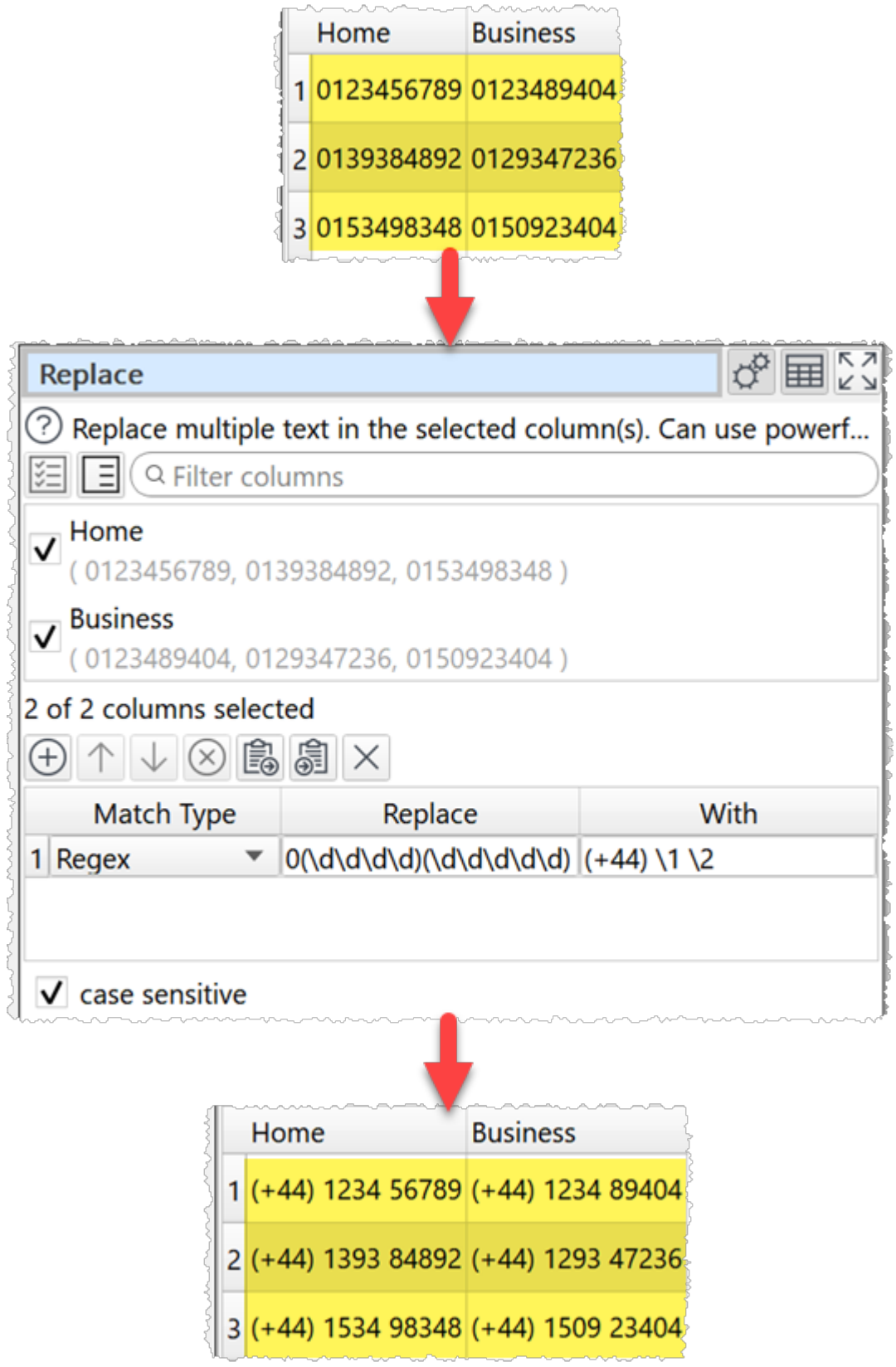




© 2023 Oryx Digital Ltd

Reformat phone numbers using a regular expression:

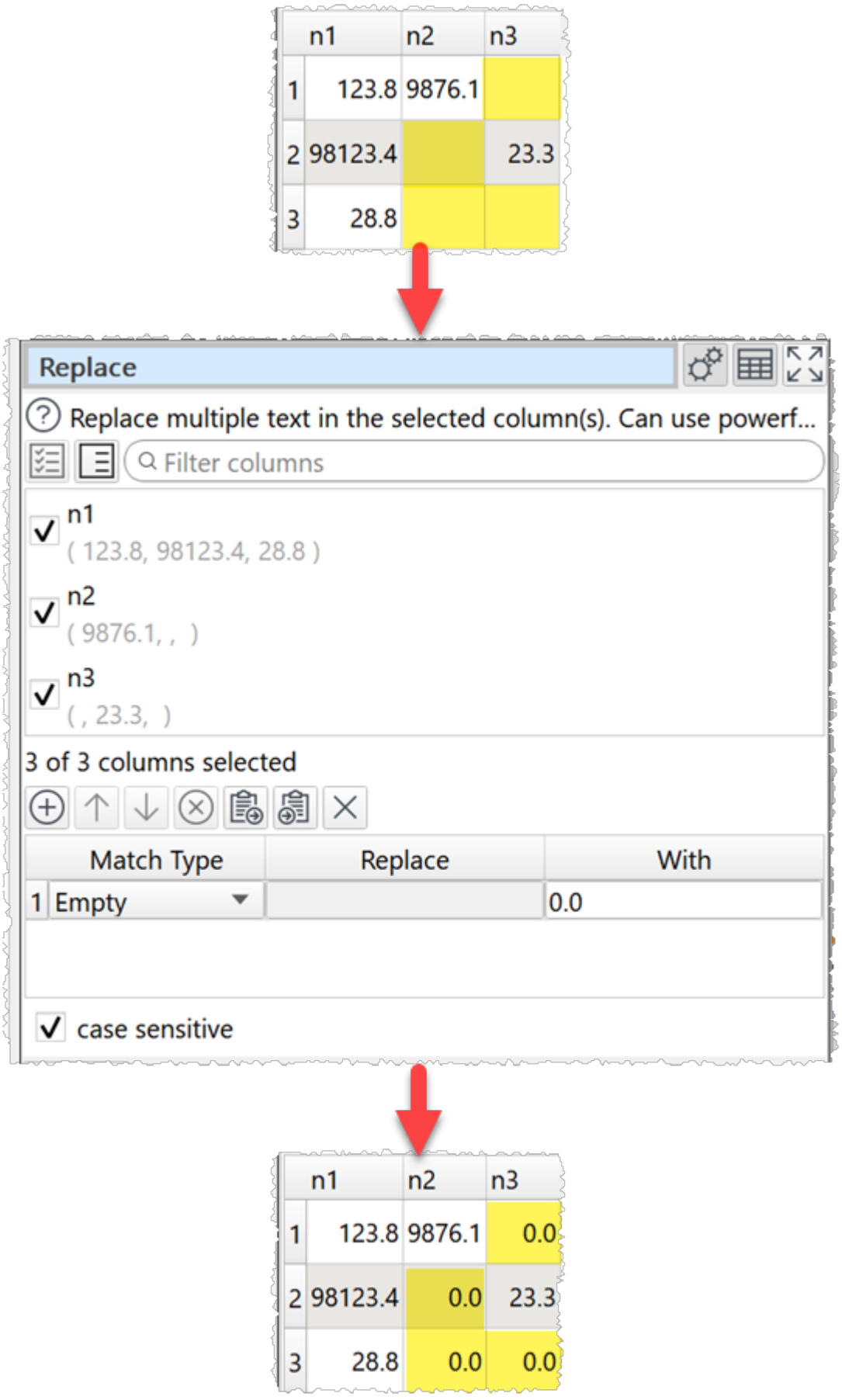

© 2023 Oryx Digital Ltd




Replace empty values:


© 2023 Oryx Digital Ltd

© 2023 Oryx Digital Ltd

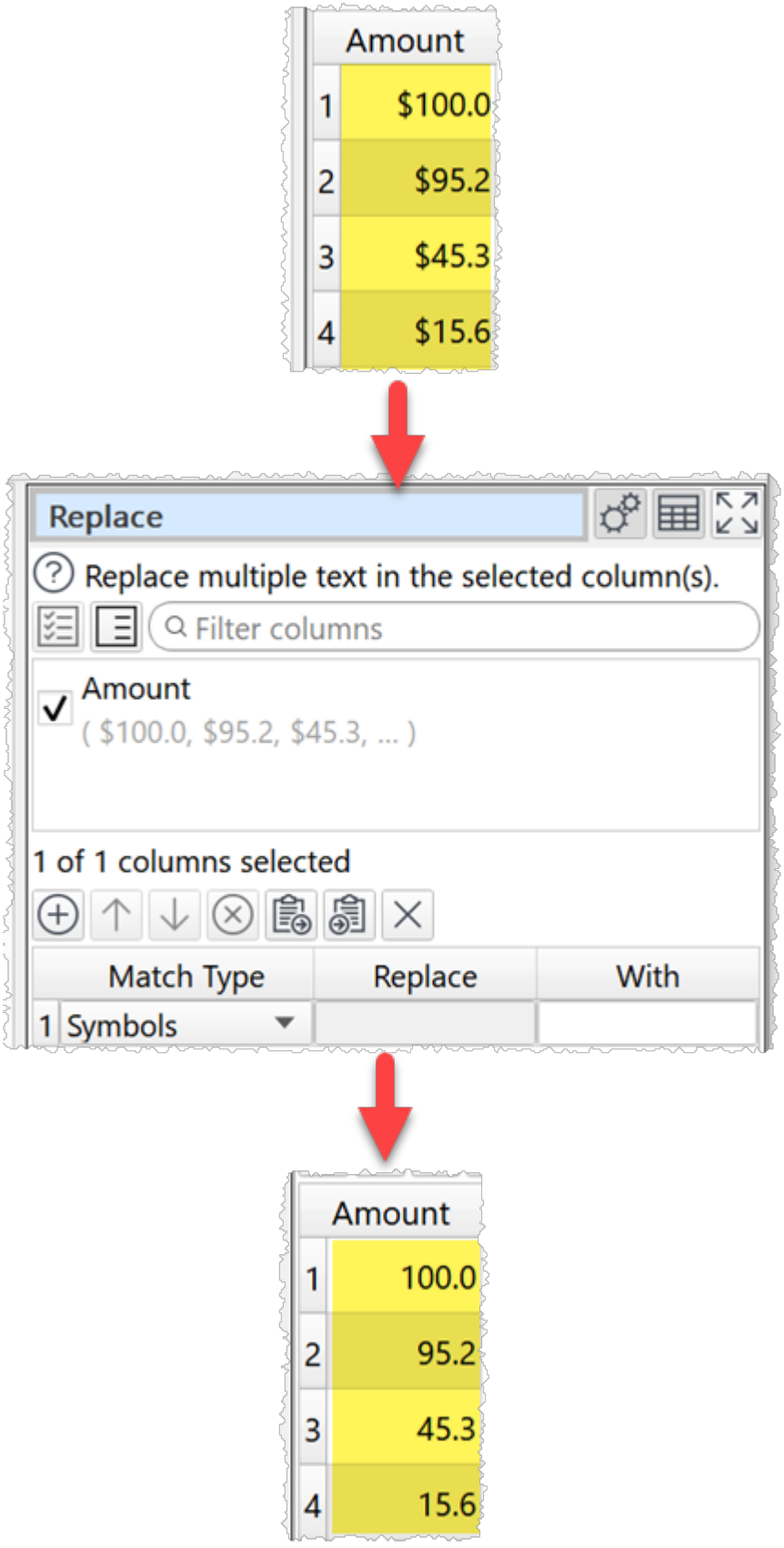




© 2023 Oryx Digital Ltd

Remove symbols:


© 2023 Oryx Digital Ltd





© 2023 Oryx Digital Ltd

## Inputs

One.

## Options

- Check the column(s) you wish to transform.
- Click the ⊕ button to add a new filter criteria.
- Click the ↑ button to move the selected terms up.
- Click the ↓ button to move the selected terms down.
- Click the ⊗ button to delete the selected filter terms.
- Click the button to copy all terms to the clipboard.
- Click the button to paste terms from the clipboard.
- Click the ✕ button to clear all terms.
- Add the terms you wish to replace and how you wish to replace them.
  - Choose the **Match type** as:
    - **Text** to replace matches contained in the text.
    - **Exact text** to replace the text only if it matches exactly (whitespace sensitive).
    - **Regex** to match using a Regular expression.
    - **Empty** to match empty cells (cells with whitespace are not empty). The **Replace** value is ignored.
    - **Letters** to match letters, e.g. 'a'. The **Replace** value is ignored.
    - **Digits** to match numeric digits, e.g. '1'. The **Replace** value is ignored.
    - **Punctuation** to match punctuation characters e.g. ','. The **Replace** value is ignored.
    - **Symbols** to match symbol characters e.g. '$'. The **Replace** value is ignored.
    - **Space** to match white space characters. The **Replace** value is ignored.
  - In **Replace** put the text you want to replace for **Text**, **Exact text** or **Regex** matching. You can use a column variable.
  - In **With** put the text you want to replace it with. You can use a column variable.
- Check **case sensitive** to use case sensitive matching for **Text**, **Exact text** or **Regex** matching.

## Notes

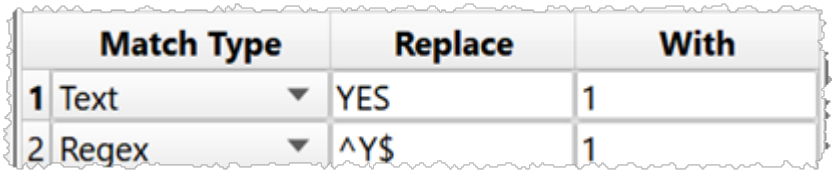
- If you can to keep the original column use Copy Cols to copy the column first.
- Comparisons are whitespace sensitive. You can use Whitespace to remove whitespace before replacing.
- If you are pasting in replacement terms, the format expected is comma delimited text with 3 columns.
  - The first column is the 0-based index of the **Match Type** drop-down list.
  - The second column is the text **Replace**.
  - The third column is the text **With**.

For example, pasting in:


*© 2023 Oryx Digital Ltd*

```
0,YES,1
2,^Y$,1
```

Will add:



| | Match Type | Replace | With |
|---|---|---|---|
| 1 | Text | YES | 1 |
| 2 | Regex | ^Y$ | 1 |

You can also try copying terms to the clipboard and pasting into a text file to see the format. If you only paste in 1 column of text, it will be assumed to be a list of **Replace**. You can, of course, use Easy Data Transform to create the appropriate filter terms and paste them into memory.

## See also

- Insert
- Substitute

### 2.3.44 Row Num

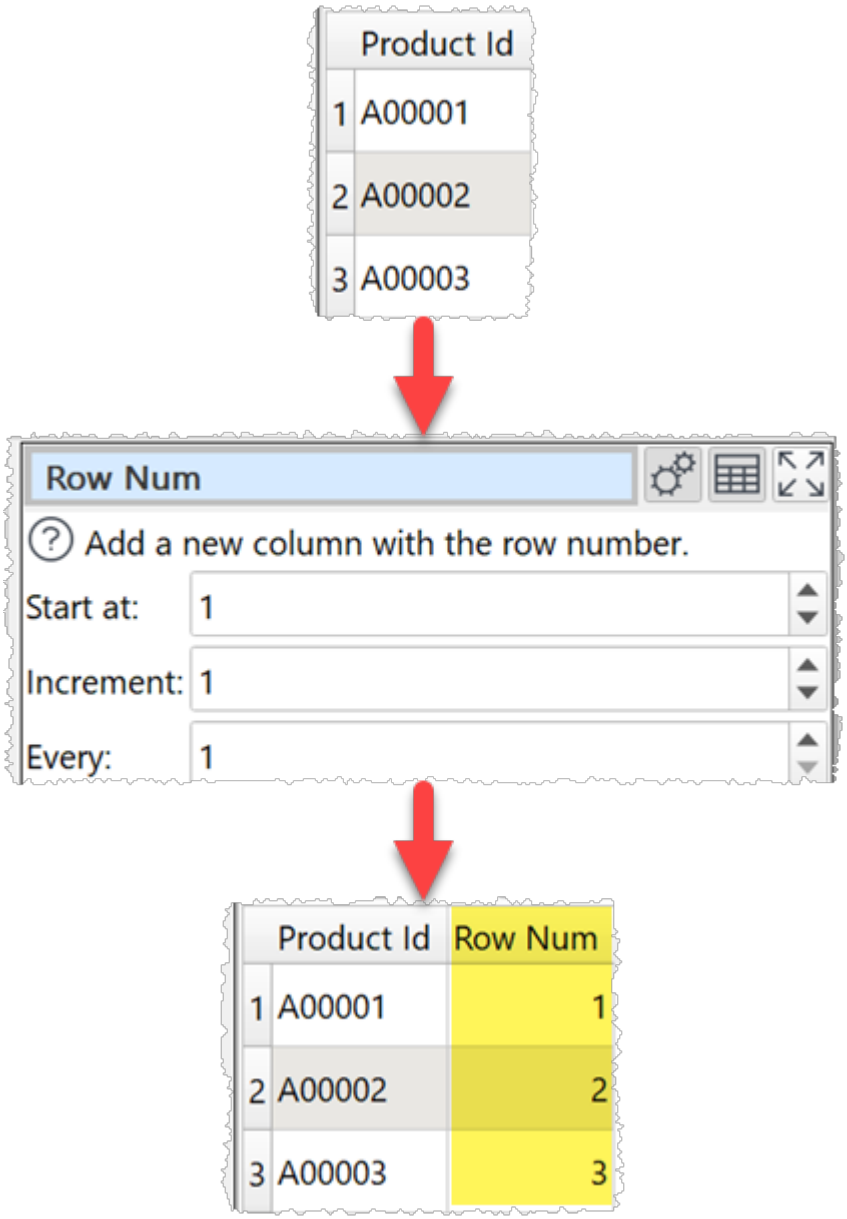
## Description

Add a new column that contains the row number.

## Example

Add a row number starting at 1:


© 2023 Oryx Digital Ltd




## Inputs

One.

## Options

- Set **Start at** to the number you want to use for the first row.
- Set **Increment** to the amount you wish to increment by.
- set **Every** to how often to apply the increment (e.g. set to 5 to increment once every 5 rows).


*© 2023 Oryx Digital Ltd*

## Notes

- The new column is added at the right end. You can change the column order with Reorder Cols and the column name with Rename Cols.
- You can Pad the row number with zeros to turn it into a column of unique Ids with a fixed length.
- You can use **Row Num** with a Calculate transform using operation **MSecsToDateTime** to create a sequence of dates.
- You can use **Row Num** with Calculate transform using operation **Modulus** and an If transform to create a sequence of repeating text.

### 2.3.45 Sample

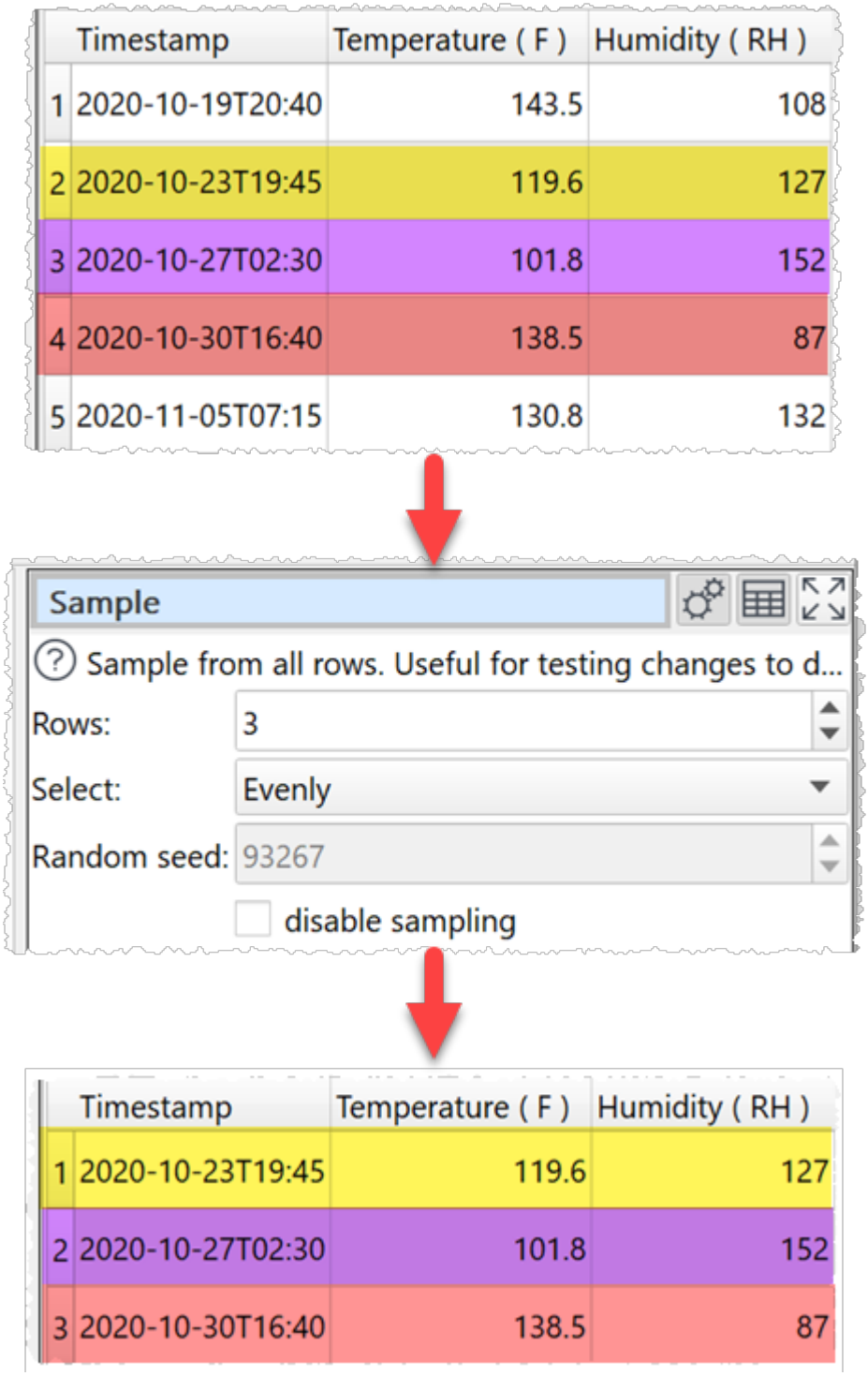
## Description

Selects a subset of rows from the input.

## Example

Sample 3 rows from 5:


*© 2023 Oryx Digital Ltd*




## Inputs

One.


*© 2023 Oryx Digital Ltd*

## Options

- Set **Rows** to the number of rows you want to output. If it is the same or greater than the number of rows in the input, then the input will be unaffected.
- Set **Select** depending on how you want the rows sampled.
- Check **disable sampling** to turn off sampling. If sampling is disabled, the transform does nothing.

## Notes

- If you set **Select** to **Randomly (for quality)** or **Randomly (for speed)**, rows will be selected using a pseudo-random number generator, initialized with **Random seed**. Changing the seed will change the rows sampled.
- With the same **Random seed** and the same **Select** value you should get the same random sampling, even across Windows and Mac versions of Easy Data Transform.
- **Random (for speed)** is significantly faster than **Randomly (for quality)**, but the results may be less random.
- If you are transforming a large dataset, then you can use **Sample** to test a small subset.
- If you need to do something more complex than **Sample** can handle (e.g. keep only rows 500 to 1000) then use Slice or Row Num followed by a Filter. For the most complex cases use **Row Num**, followed by Javascript, followed by a **Filter**. E.g. this **Javascript** function returns 1 for every 10th row between 1000 and 2000 and 0 otherwise:

```
return $(Row Num) >= 1000 & $(Row Num) <= 2000 & $(Row Num) % 10 ==
0;
```

### 2.3.46 Scale

## Description

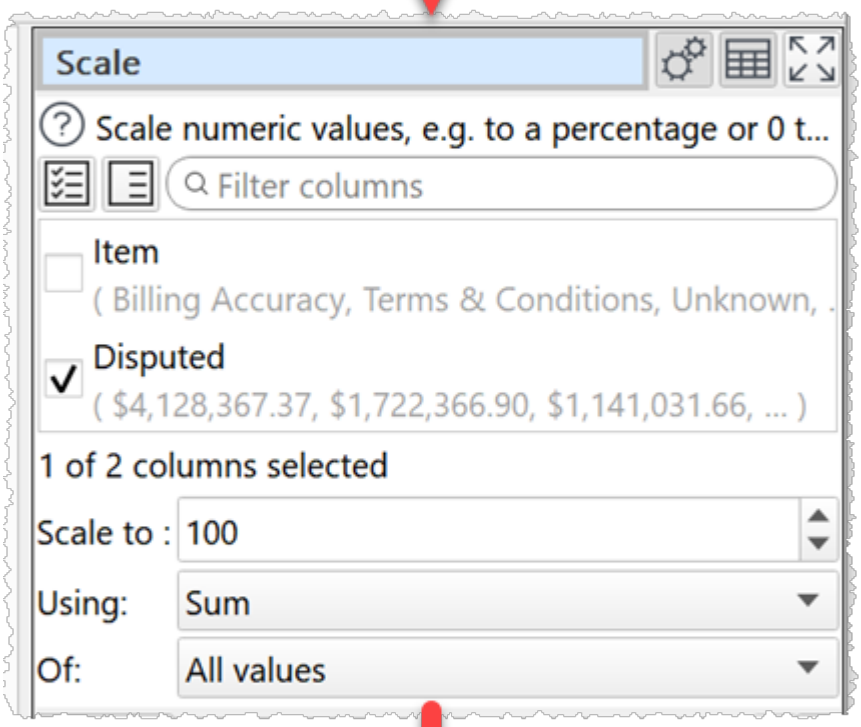
Scale (normalize) numeric values, e.g. to a percentage or 0 to 1.

## Examples

Scale a single column to a percentage:


© 2023 Oryx Digital Ltd

| | Item | Disputed |
|---|---|---|
| 1 | Billing Accuracy | $4,128,367.37 |
| 2 | Terms & Conditions | $1,722,366.90 |
| 3 | Unknown | $1,141,031.66 |
| 4 | Delivery & Installation | $1,091,424.82 |
| 5 | Service Delivery | $238,914.81 |
| 6 | Technical Issues | $53,816.33 |



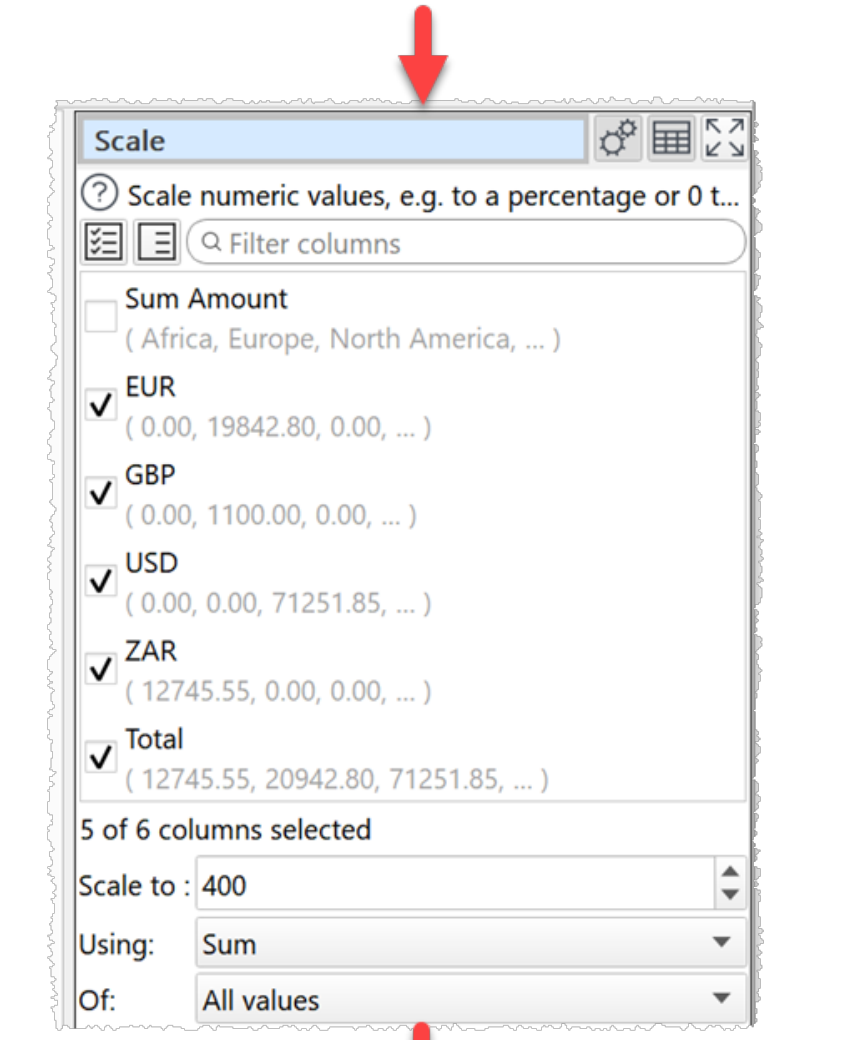

| | Item | Disputed |
|---|---|---|
| 1 | Billing Accuracy | 49.2885132433 |
| 2 | Terms & Conditions | 20.5633113897 |
| 3 | Unknown | 13.6227590823 |
| 4 | Delivery & Installation | 13.0305037981 |


© 2023 Oryx Digital Ltd

Scale a table with row, column and grand totals to a percentage (scale to 400 as each value is also added to the row total, column total and grand total):


*© 2023 Oryx Digital Ltd*

| | Sum Amount | EUR | GBP | USD | ZAR | Total |
|---|---|---|---|---|---|---|
| 1 | Africa | 0.00 | 0.00 | 0.00 | 12745.55 | 12745.55 |
| 2 | Europe | 19842.80 | 1100.00 | 0.00 | 0.00 | 20942.80 |
| 3 | North America | 0.00 | 0.00 | 71251.85 | 0.00 | 71251.85 |
| 4 | Total | 19842.80 | 1100.00 | 71251.85 | 12745.55 | 104940.20 |



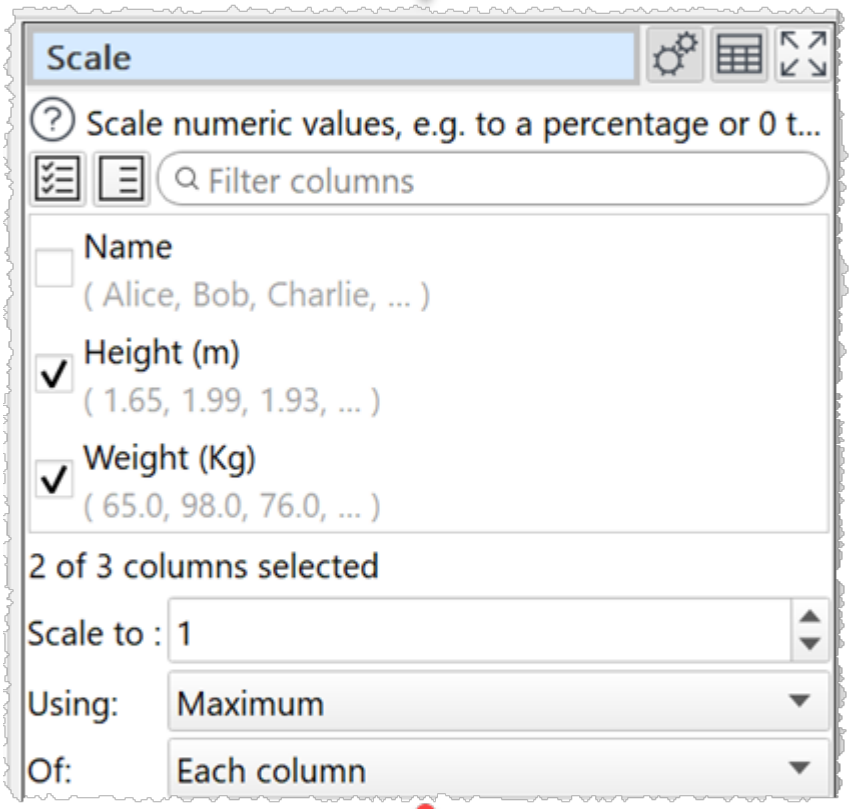

| | Sum Amount | EUR | GBP | USD | ZAR | Total |
|---|---|---|---|---|---|---|
| 1 | Africa | 0 | 0 | 0 | 12.1455362197 | 12.1455362197 |
| 2 | Europe | 18.9086737018 | 1.0482160316 | 0 | 0 | 19.9568897334 |
| 3 | North America | 0 | 0 | 67.8975740469 | 0 | 67.8975740469 |
| 4 | Total | 18.9086737018 | 1.0482160316 | 67.8975740469 | 12.1455362197 | 100 |


*© 2023 Oryx Digital Ltd*

Scale heights and weights so that the largest in each column is 1 and the smallest is 0.


*© 2023 Oryx Digital Ltd*

| | Name | Height (m) | Weight (Kg) |
|---|---|---|---|
| 1 | Alice | 1.65 | 65.0 |
| 2 | Bob | 1.99 | 98.0 |
| 3 | Charlie | 1.93 | 76.0 |
| 4 | Doug | 2.02 | 89.5 |
| 5 | Erin | 1.89 | 76.3 |



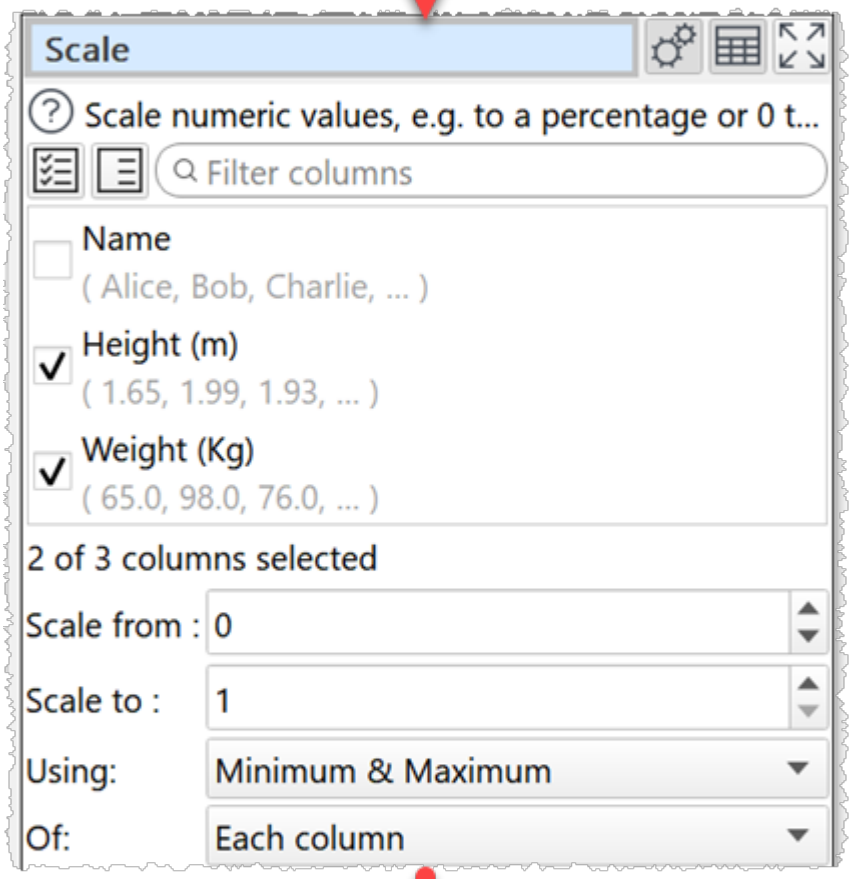

| | Name | Height (m) | Weight (Kg) |
|---|---|---|---|
| 1 | Alice | 0.8168316832 | 0.6632653061 |
| 2 | Bob | 0.9851485149 | 1 |
| 3 | Charlie | 0.9554455446 | 0.7755102041 |


© 2023 Oryx Digital Ltd

Scale heights and weights so that the largest in each column is 1 and the smallest is 0.


*© 2023 Oryx Digital Ltd*

| | Name | Height (m) | Weight (Kg) |
|---|---|---|---|
| 1 | Alice | 1.65 | 65.0 |
| 2 | Bob | 1.99 | 98.0 |
| 3 | Charlie | 1.93 | 76.0 |
| 4 | Doug | 2.02 | 89.5 |
| 5 | Erin | 1.89 | 76.3 |




| | Name | Height (m) | Weight (Kg) |
|---|---|---|---|
| 1 | Alice | 0 | 0 |
| 2 | Bob | 0.9189189189 | 1 |
| 3 | Charlie | 0.7567567568 | 0.3333333333 |


© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Set **Columns** to the columns whose values you want to scale.
- set **Scale from** to the value you want to scale the minimum value to. E.g. 0 if you want the minimum value scaled to 0. This is only available when **Using** is set to **Minimum and Maximum**.
- Set **Scale to** to the value you want to scale the maximum or sum value to. E.g. 100 for a percentage.
- Set **Using** depending on whether you want to use the **Sum**, **Maximum** or **Minimum and Maximum** values for scaling.
- Set **Of** depending on whether you want to scale for **Each column**, **Each row** or for **All values**.

## Notes
- Whether values are interpreted as numbers depends on locale.
- Non-numeric values (including empty values) are ignored.
- To replace empty values with zeros use the Replace transform.
- To modify the numerical precision of the results use the Num Format transform.
- To add a '%' at the end of values use the Insert transform.
- To copy columns before using **Scale** use Copy Cols .
- Warnings are shown in the **Warnings** tab for non-numeric values.

See also:
- Video: How to normalize data

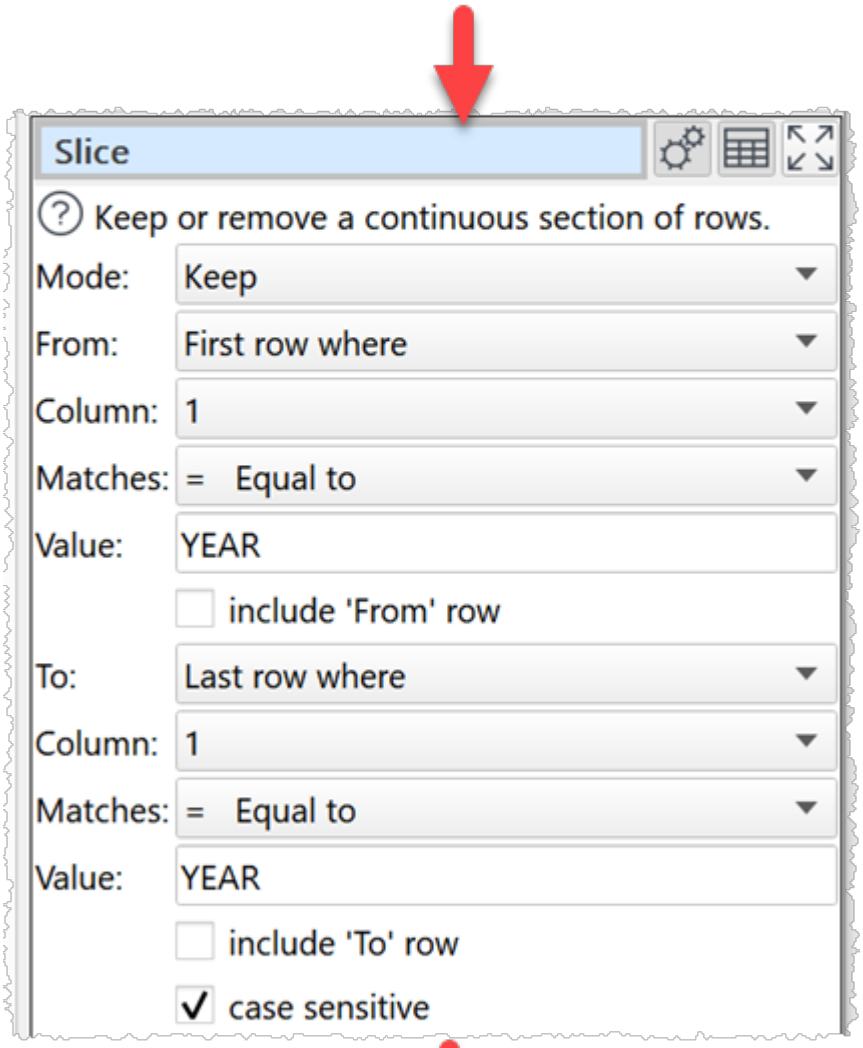
### 2.3.47 Slice

## Description
Keep or remove a continuous section of rows.

## Example
Keep only the records between the 'YEAR' rows:


© 2023 Oryx Digital Ltd

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | YEAR | JAN | FEB | MAR |
| 2 | 1980 | 256.9 | 238.5 | 275.4 |
| 3 | 1981 | 248.9 | 224.9 | 259.1 |
| 4 | YEAR | JAN | FEB | MAR |
| 5 | 1982 | 258.6 | 230.0 | 280.4 |
| 6 | 1983 | 275.9 | 234.9 | 251.9 |




| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1980 | 256.9 | 238.5 | 275.4 |


© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Set **Mode** depending on whether you want to keep or remove the slice.
- Set **From** according to how you want to choose the first row of the slice.
- Set **Column**, **Matches** and **Value** to match the first row. You can use a column variable for either **Value**.
- Uncheck **Include 'From' row** if you don't want to include the first row in the slice.
- Set **To** according to how you want to choose the last row of the slice.
- Set **Column**, **Matches** and **Value** to match the last row.
- Uncheck **Include 'To' row** if you don't want to include the last row in the slice.
- Check **case sensitive** to use case sensitive matching for text.

## Notes
- If there is no match for the **From** or **To** row:
  - **Mode**=**Keep** will remove all rows
  - **Mode**=**Remove** will keep all rows
- Number, date and text values are treated differently for **Equal to**, **Greater than**, **Less than**, **Greater than equal, Less than equal** and **Not equal to** operations.
  - If both values are numeric, a numeric comparison will be carried out.
  - If both values match a supported date formats in Preferences, a date comparison will be carried out.
  - Otherwise the values will be treated as text.
  - For example, an empty value is considered less than 0, because they will be compared as text. So you might want to replace empty values with 0 or remove those rows with Filter, before comparing them.
- All values are treated as text for **Contains**, **Starts with**, **Ends with**, **Matches regex**, **Is not**, **Doesn't start with, Doesn't end with** and **Doesn't match regex** operations.
- Comparisons of text are whitespace sensitive. Cells with whitespace will not match **Is empty**. You can use Whitespace to remove whitespace before filtering and Replace to get of other unwanted characters (e.g. whitespace inside the text).
- See here for more details on Regular expressions (regex).
- If you want to keep or remove a selection of rows by row number, use a Row Num transform followed by a Filter transform.

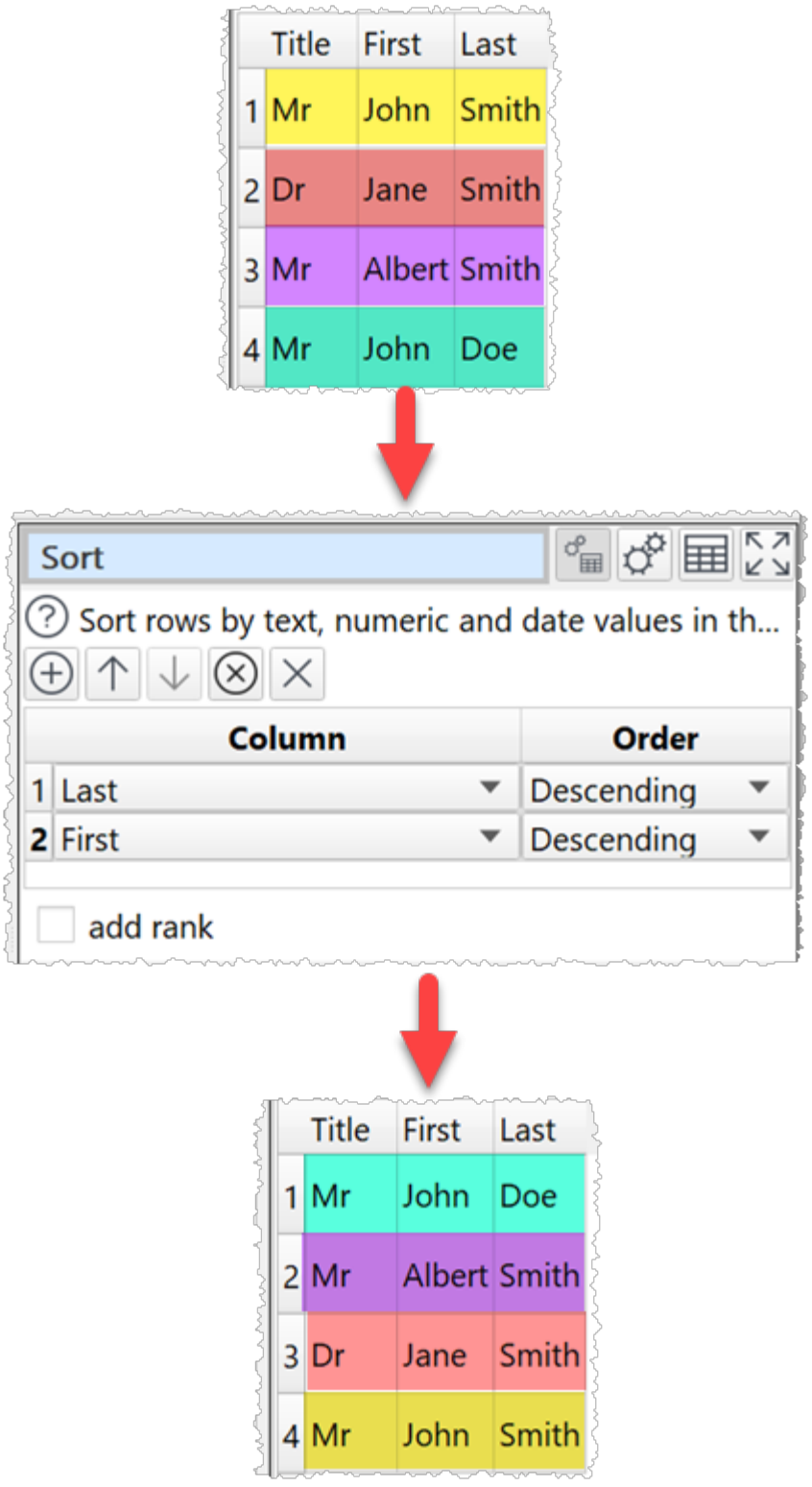
### 2.3.48 Sort

## Description
Sorts rows by one or more columns.


© 2023 Oryx Digital Ltd

## Examples

Sort by 'Last' and then First' columns:



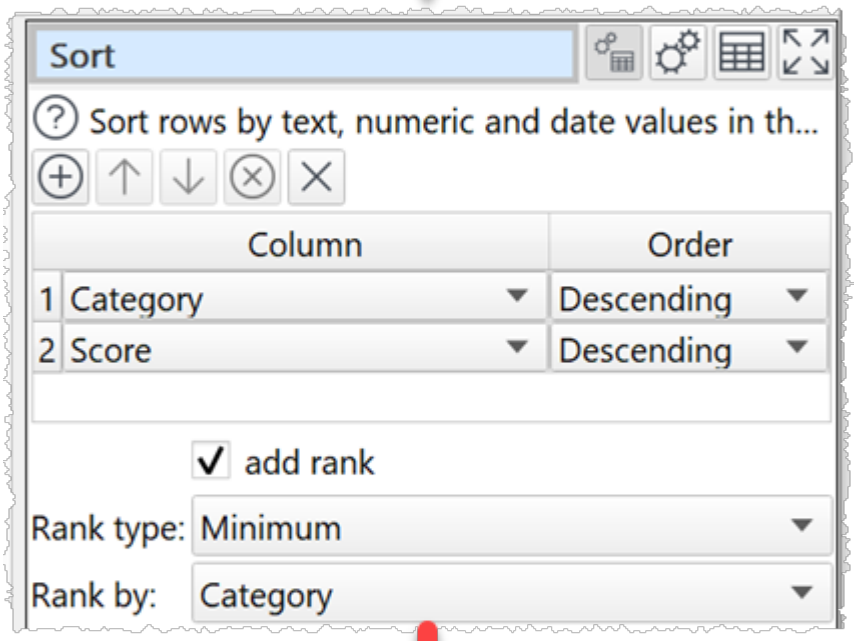


© 2023 Oryx Digital Ltd

Sort by 'Category' and then 'Score' columns. Rank for each distinct 'Category' value:


*© 2023 Oryx Digital Ltd*

| | Name | Category | Score |
|---|---|---|---|
| 1 | Andy | Novice | 122 |
| 2 | Bill | Veteran | 120 |
| 3 | Charles | Novice | 123 |
| 4 | Dennis | Veteran | 131 |
| 5 | Eric | Veteran | 120 |
| 6 | Fred | Veteran | 123 |




| | Name | Category | Score | Rank |
|---|---|---|---|---|
| 1 | Dennis | Veteran | 131 | 1 |
| 2 | Fred | Veteran | 123 | 2 |
| 3 | Bill | Veteran | 120 | 3 |
| 4 | Eric | Veteran | 120 | 3 |
| 5 | Charles | Novice | 123 | 1 |


© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Click the ⊕ button to add a new sort level.
- Click the ↑ button to move the selected sort levels up.
- Click the ↓ button to move the selected sort levels down.
- Click the ⊗ button to delete the selected sort level(s).
- Click the ✕ button to clear all sort levels.
- Set **Column** to the column you want to sort by.
- Set **Order** depending on whether you want to sort this column **Ascending** or **Descending**.
- Check **add rank** to add a ranking column
  - Set **Rank type** according to how you wish to do the ranking:
    - **Minimum**: Uses the minimum rank of each group.
    - **Average**: Uses the average rank of each group.
    - **Maximum**: Uses the maximum rank of each group.
    - **Dense**: Each successive rank is only incremented by 1. There are no gaps.
    - **Cumulative**: Shows the proportion of rows (0 to 1) with the same or lower numbered rank.

For example:

| | Sorted score |
|---|---|
| 1 | 131 |
| 2 | 123 |
| 3 | 123 |
| 4 | 123 |
| 5 | 120 |
| 6 | 120 |

Gives the following rankings, depending on **Rank type**:


© 2023 Oryx Digital Ltd

| | Rank type | | | | |
|---|---|---|---|---|---|
| **Sorted score** | **Minimum** | **Average** | **Maximum** | **Dense** | **Cumulative** |
| **131** | 1 | 1 | 1 | 1 | 0.167 |
| **123** | 2 | 3 | 4 | 2 | 0.667 |
| **123** | 2 | 3 | 4 | 2 | 0.667 |
| **123** | 2 | 3 | 4 | 2 | 0.667 |
| **120** | 5 | 5.5 | 6 | 3 | 1 |
| **120** | 5 | 5.5 | 6 | 3 | 1 |

- Set **Rank by** to rank separately for each distinct value in this column. Leave as `<None selected>` if you only want a single ranking for all rows.

## Notes

- If you add multiple levels, it will sort by level 1 then level 1 values that are the same will be sorted by level 2 etc.
- Number, date and text values are treated differently for sorting purposes.
- Any values that can be converted to numbers will be treated as numbers.
- Any values that match the supported date formats in Preferences will be treated as dates.
- Comparisons of text are case and whitespace sensitive. You can use Case to change the case and Whitespace to remove whitespace before sorting.
- If you want to rank rows without losing the original order, use Row Num to add a row number before the **Sort**, and then use another **Sort** on the **Row Num** column to return to the original order.
- Unsorted columns can be sorted into any order and this may vary between different released of Easy Data Transform and Windows and Mac versions.

See also:

- Video: How to rank data

### 2.3.49 Split Col

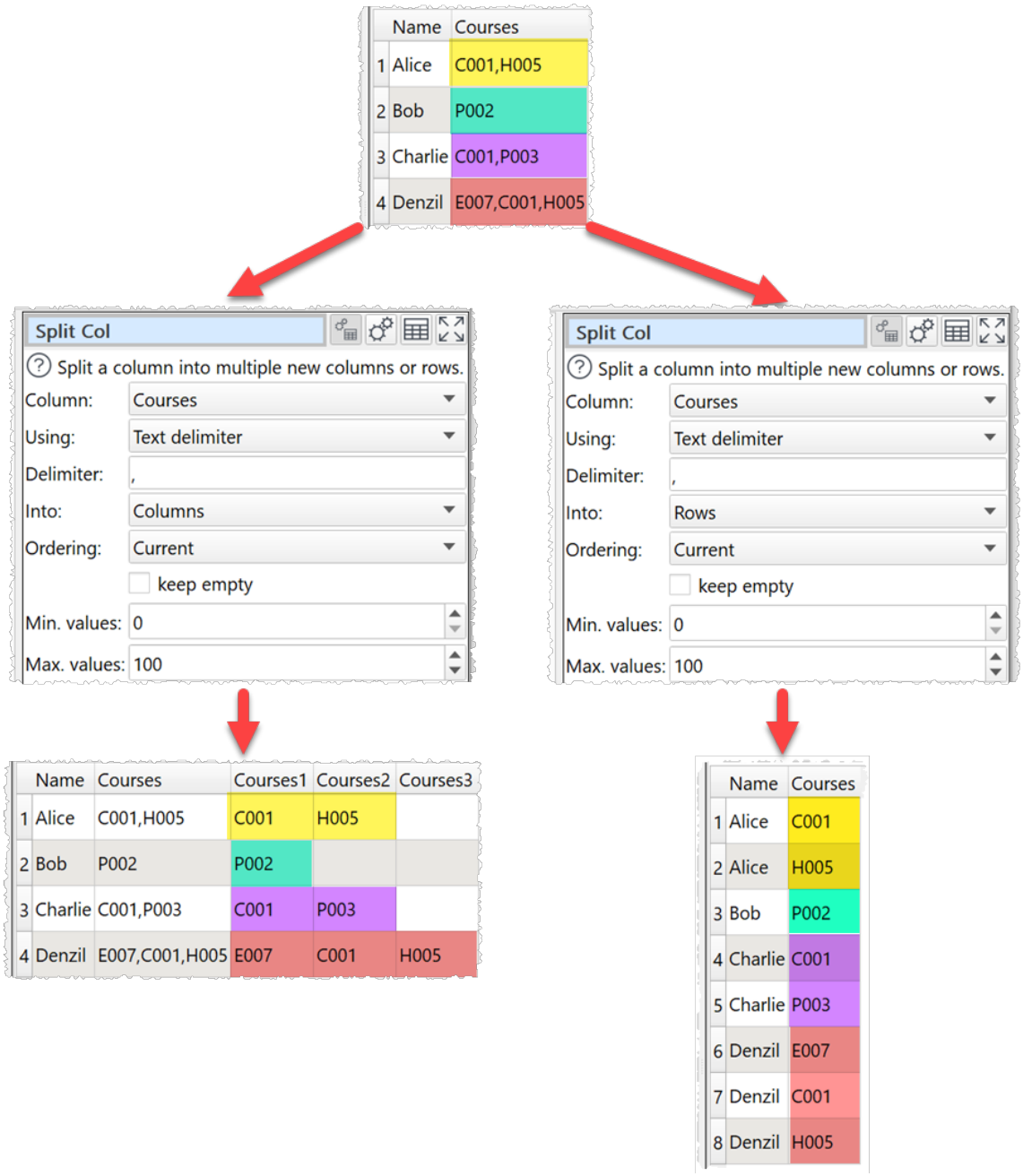
## Description

Split a column into multiple new columns or rows.


© 2023 Oryx Digital Ltd

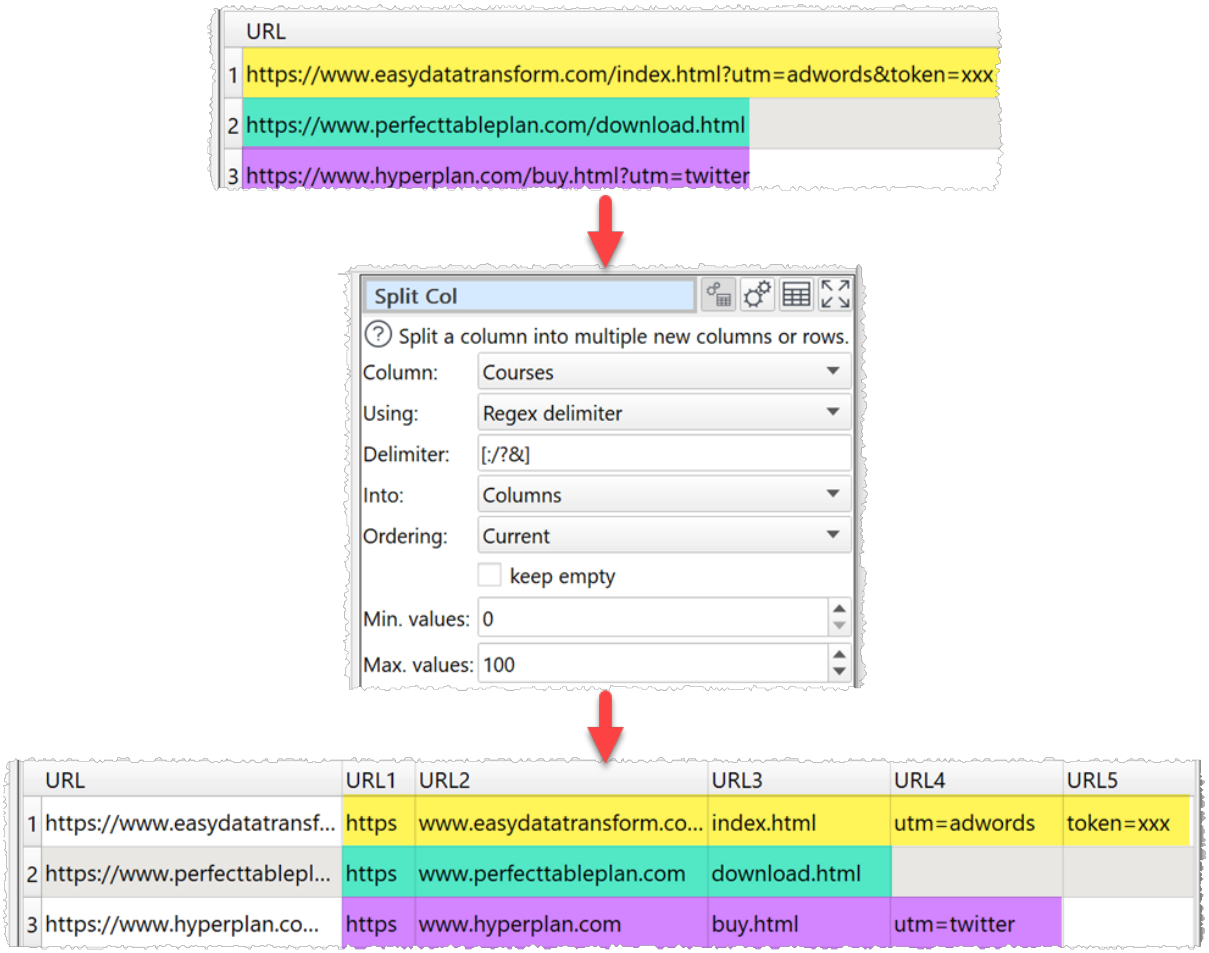
## Examples

Split the 'Courses' column into multiple columns or rows:



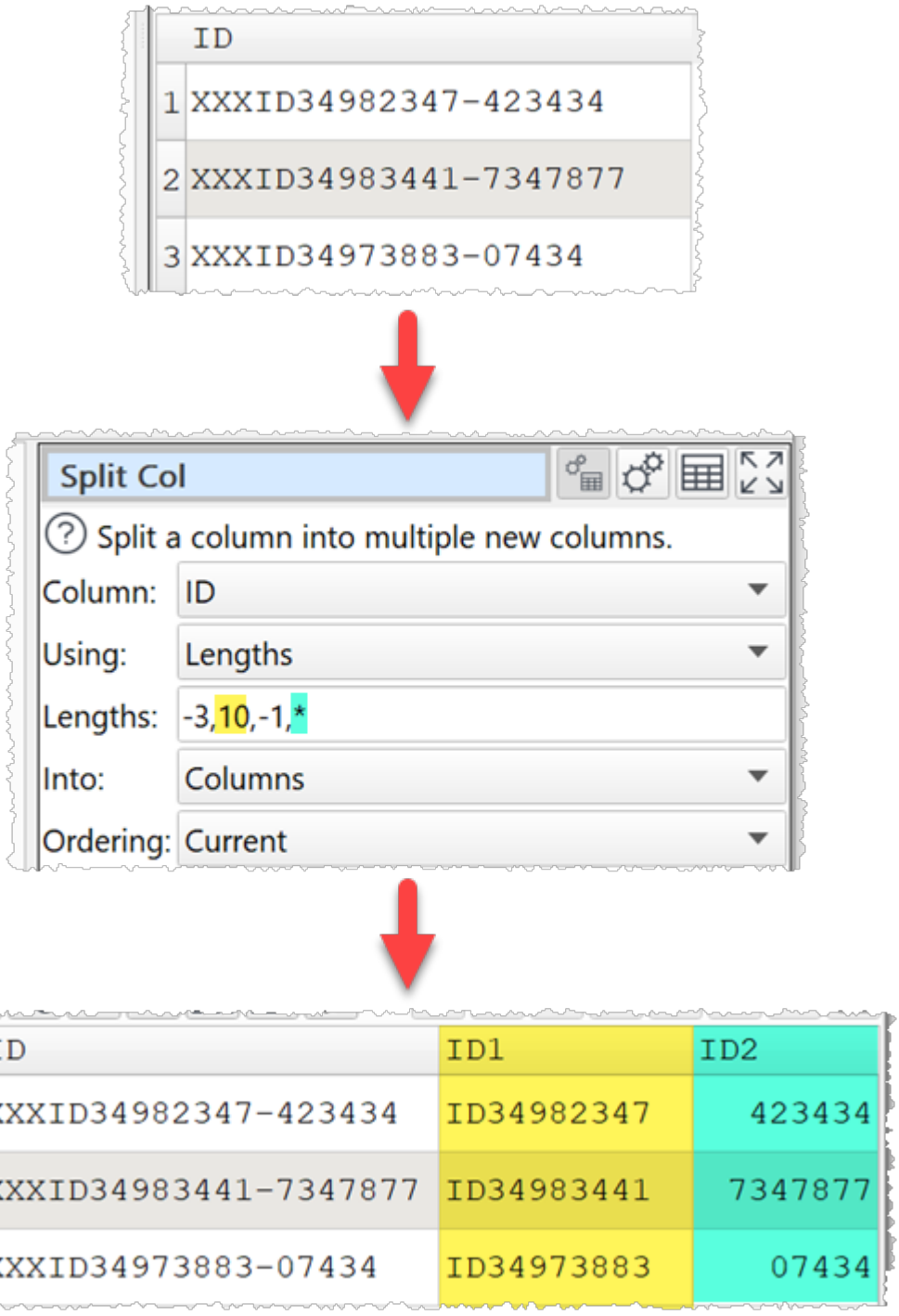

Split the 'URL' column into multiple columns using colon, slash, question mark or ampersand:


*© 2023 Oryx Digital Ltd*




Split characters 4 to 13 and 15 onward into new columns:


*© 2023 Oryx Digital Ltd*




## Inputs

One.

## Options

- Select the **Column** you wish to split.


*© 2023 Oryx Digital Ltd*

- Set **Using** depending on whether you wish to split using a **Text delimiter** (usually a single character), a **Regex delimiter** or **Lengths** of each column.
- Set the **Delimiter** for splitting the column. E.g. , (comma). For **Text delimiter** or **Regex delimiter** only.
- Set **Lengths** as a comma separated list of the number of characters in each new column. Put - in front of a value to ignore it. Use * as a wildcard to mean 'the rest of the value'. Only 1 * can be used. For **Lengths** only. For example:
  - `3,5,4` will create 3 columns with characters 1-3, 4-8 and 8-11
  - `-3,5,-4,*` will create 2 columns with characters 4-8 and 12 onward
  - `-*,3` will create 1 column with the last 3 characters
  - `2,-*,2` will create 2 columns with the first 2 and last 2 characters
- Set **Into** to:
  - **Columns** to create a new column for each split value.
  - **Rows** to create a new row for each split value (the original row is removed).
- Set **Ordering** depending on how you want to order values after splitting.
- Check **keep empty** if you wish to keep empty values (e.g. honor delimiters with nothing in between). For **Text delimiter** or **Regex delimiter** only.
- set **Min. values** to the minimum number of columns/rows you wish to split each value into. For **Text delimiter** or **Regex delimiter** only.
- set **Max. values** to the maximum number of columns/rows you wish to split each value into (ignored if less than minimum). For **Text delimiter** or **Regex delimiter** only.

## Notes

- To split a column by carriage returns or tabs set **Using** to **Regex delimiter** and set **Delimiter** to `\n` or `\t`, respectively.
- If **Into** is **Columns**, new columns are added at the right end. You can change the column order with Reorder Cols.
- If there is a header, the header of the new column is based on the original header. You can change the column name with Rename Cols.
- Set **Min. values** to the same as **Max. values** to always create the same number of columns.
- The opposite of **Split Col** with **Into** as **Columns** is Concat Cols.
- The opposite of **Split Col** with **Into** as **Rows** is Unique.

## See also

- Split Rows

### 2.3.50 Split Rows

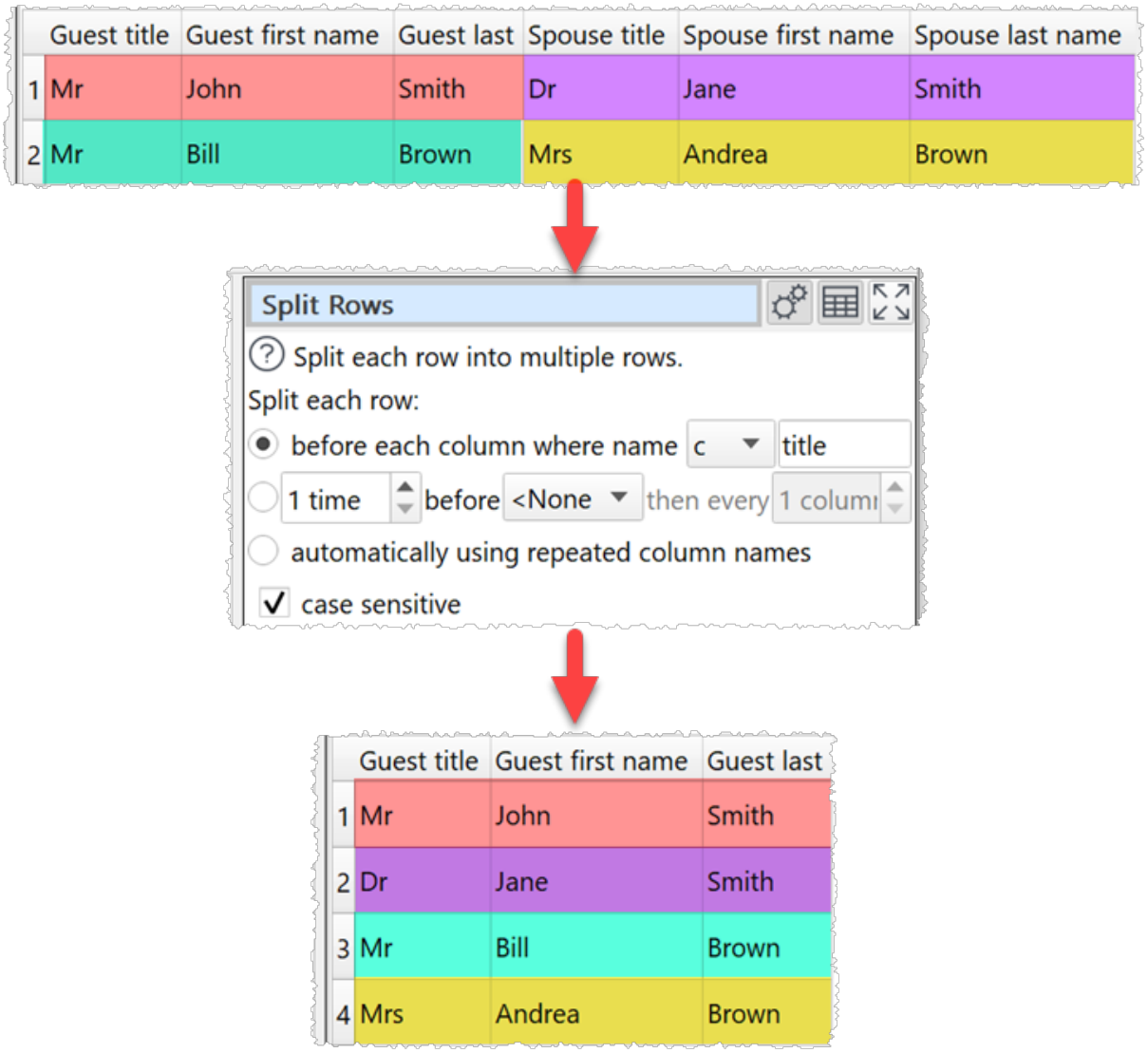
## Description

Split each row into multiple rows.


© 2023 Oryx Digital Ltd

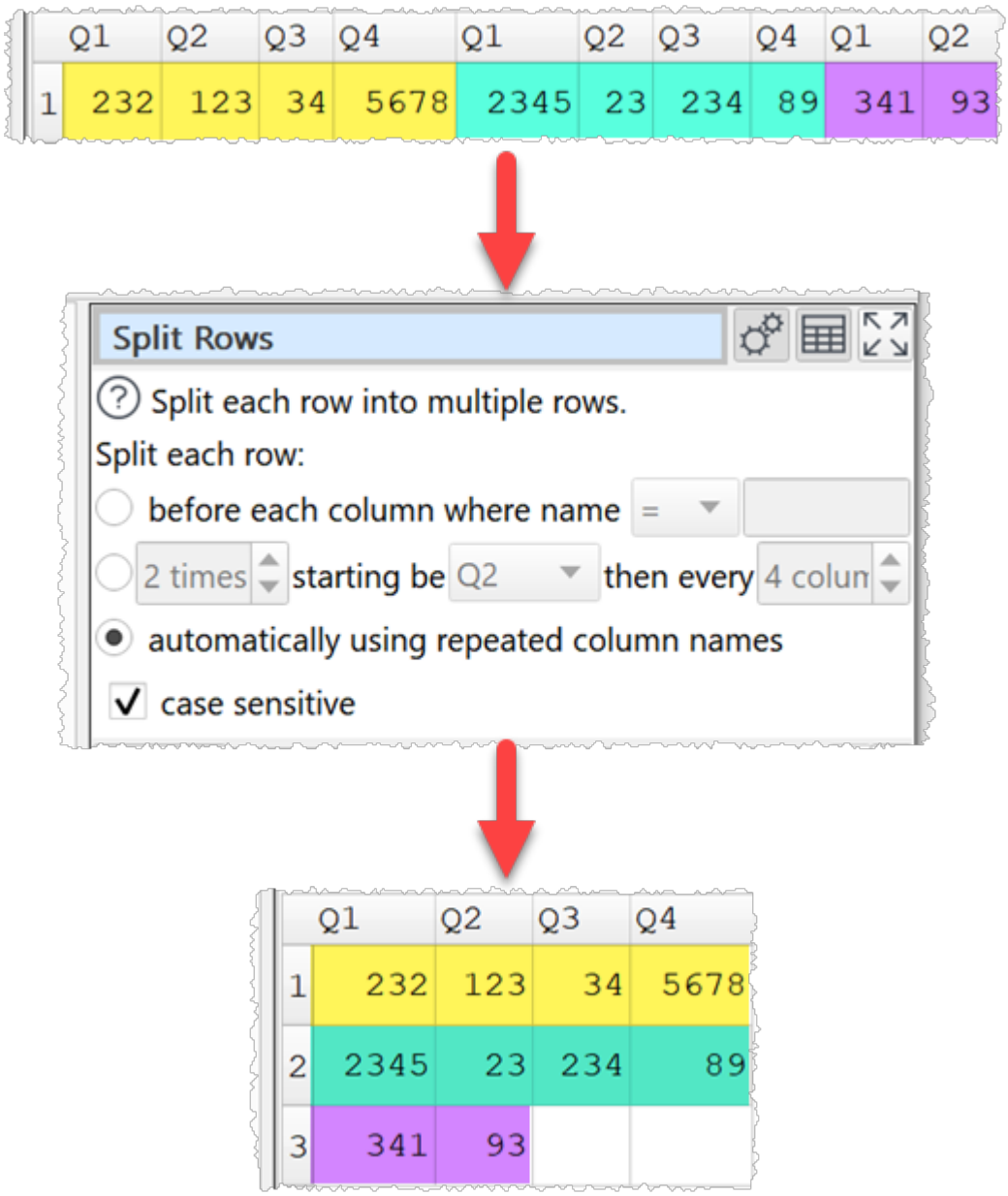
## Examples

Split rows before each column containing 'title':




Split rows using repeated column names:


© 2023 Oryx Digital Ltd




## Inputs

One.

## Options

- Split by column name:
  - Each row will be split before each column that matches the criteria.
- Split a fixed number of times:
  - Define the number of **times** you want to split each row, **starting before** which column and **then every** N columns.
  - The split is added before the designated columns.


*© 2023 Oryx Digital Ltd*

    - Splits after the last column are ignored. So you can set **times** to a large number if you don't know how many columns there will be.
- Split automatically:
    - Find the first column name that appears more than once and split each row before each column with that name.
- Check **case sensitive** to use case sensitive matching for column names.

## Notes

- Comparisons of column names are whitespace sensitive.
- Use the keyboard `Up` and `Down` arrow keys to move the focus between the 'radio' buttons.
- Splits added before the first column are ignored, as there is already a split there.
- Use New Col or Rename Cols if you need to add additional columns or rename columns before splitting rows.
- The opposite of **Split Rows** is Concat Rows.

## See also

- Gather
- Split Col

### 2.3.51 Spread

## Description

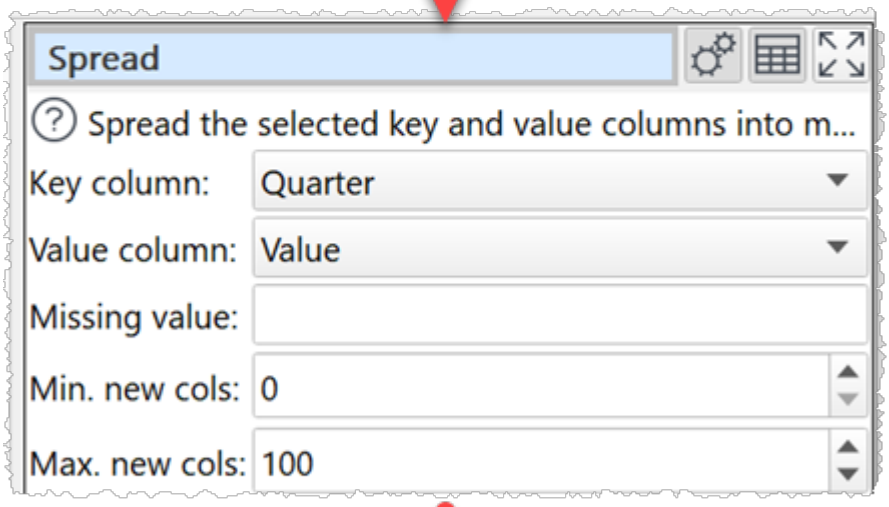
Spread a column into multiple new columns. Also called wide pivot or crosstab.

## Example

Spread 'Quarter' and 'Amount' columns to multiple columns:


© 2023 Oryx Digital Ltd

| | salesman | area | Quarter | Value |
|---|---|---|---|---|
| 1 | Alice | North | Q1 | 11.3 |
| 2 | Alice | North | Q2 | 89.3 |
| 3 | Alice | North | Q3 | 44.3 |
| 4 | Alice | North | Q4 | 18 |
| 5 | Bob | East | Q1 | 4.5 |
| 6 | Bob | East | Q2 | 7.9 |
| 7 | Bob | East | Q3 | 8 |
| 8 | Bob | East | Q4 | 3.3 |




| | salesman | area | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|---|---|
| 1 | Alice | North | 11.3 | 89.3 | 44.3 | 18 |
| 2 | Bob | East | 4.5 | 7.9 | 8 | 3.3 |


© 2023 Oryx Digital Ltd

## Inputs
One.

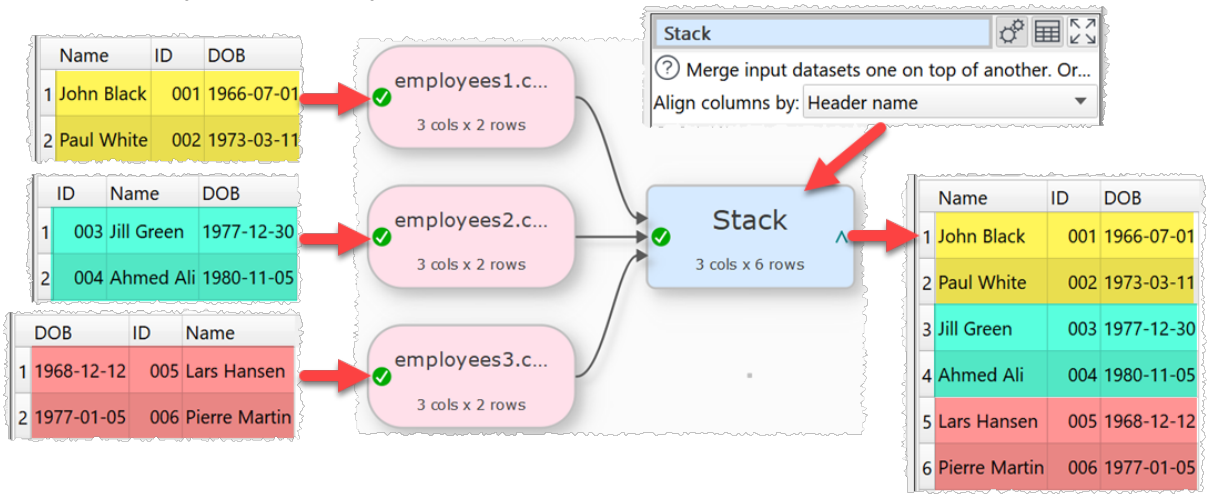
## Options
- Select the **Key column** and **Value column** you wish to spread.
- **Missing values** is used for values missing from the input dataset.
- set **Min. new cols** to the minimum number of new columns you wish to add, compared to the original dataset.
- set **Max. new cols** to the maximum number of new columns you wish to add, compared to the original dataset. Ignored if less than minimum.

## Notes
- If there are rows that are duplicates, apart from the value column, this will cause errors.
- New columns are added at the right end. You can change the column order with Reorder Cols.
- You can merge the new columns into a single column with Concat Cols.
- The opposite of Spread is Gather.

### 2.3.52 Stack

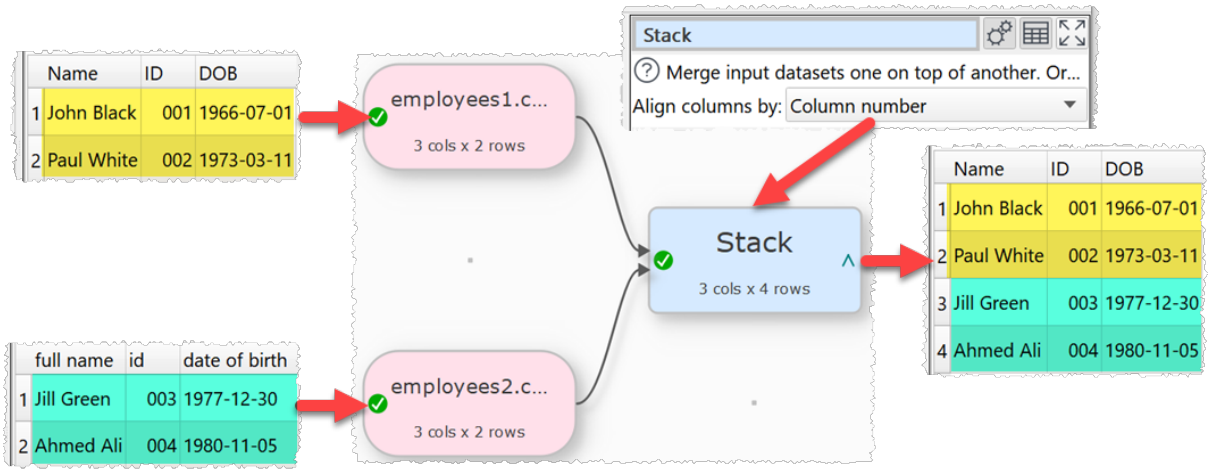
## Description
Merge rows from inputs, one on top of the other.

## Examples
Stack 3 employee datasets by header name:

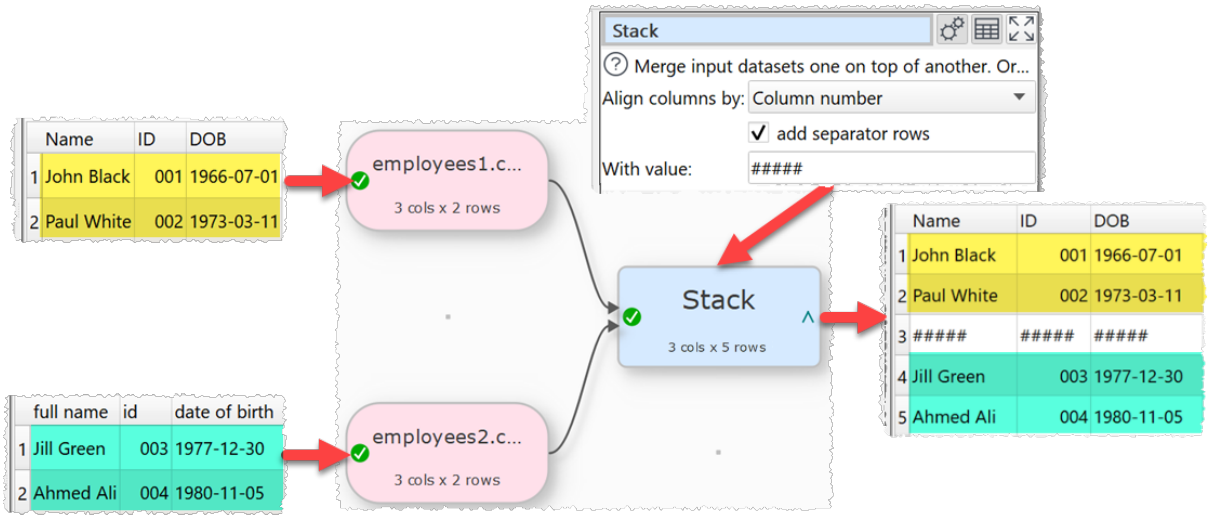



Stack 2 employee datasets by column number:


© 2023 Oryx Digital Ltd




Stack 2 employee datasets with a separator row:




## Inputs

One or more.

## Options

- Select **Align columns by** to **Header name** if you want line up column values by header name (e.g. the 'id' column in input 1 with the 'id' column in input 2) and **Column number** to align by the column number (e.g. the first column of input 1 with the first column of input 2). The headers will be matched in order:
  - case and whitespace sensitive (e.g. 'ID' to 'ID'); then
  - case insensitive and whitespace sensitive (e.g. 'Id' to 'ID'); then
  - case sensitive and whitespace insensitive (e.g. 'ID' to ' ID '); then
  - case and whitespace insensitive (e.g. 'Id' to ' ID ')
- Check **use only top dataset columns** to ignore columns not in the first dataset.


*© 2023 Oryx Digital Ltd*

- Check **add separator rows** to extra rows to separate the inputs. The value entered for **With value** is used for each column of the separator rows.

## Notes

- The stacking order depends on the vertical (Y-axis) position of the inputs.
- Stack merges datasets one on top of the other (vertically). To merge datasets side-by-side (horizontally) use Join.
- If you align by **Column number** the header of the first input is used.
- Messages are shown in the **Warnings** tab for:
  - duplicated column names within an input dataset when **Align columns by** is set to **Header name**
  - columns stacked under columns with different names when **Align columns by** is set to **Column number**
  - differing numbers of columns between input datasets
- It is sometimes useful to **Stack** a single dataset, so you can add more inputs later.

## See also

- Cross
- Join
- Merge datasets

### 2.3.53 Stamp

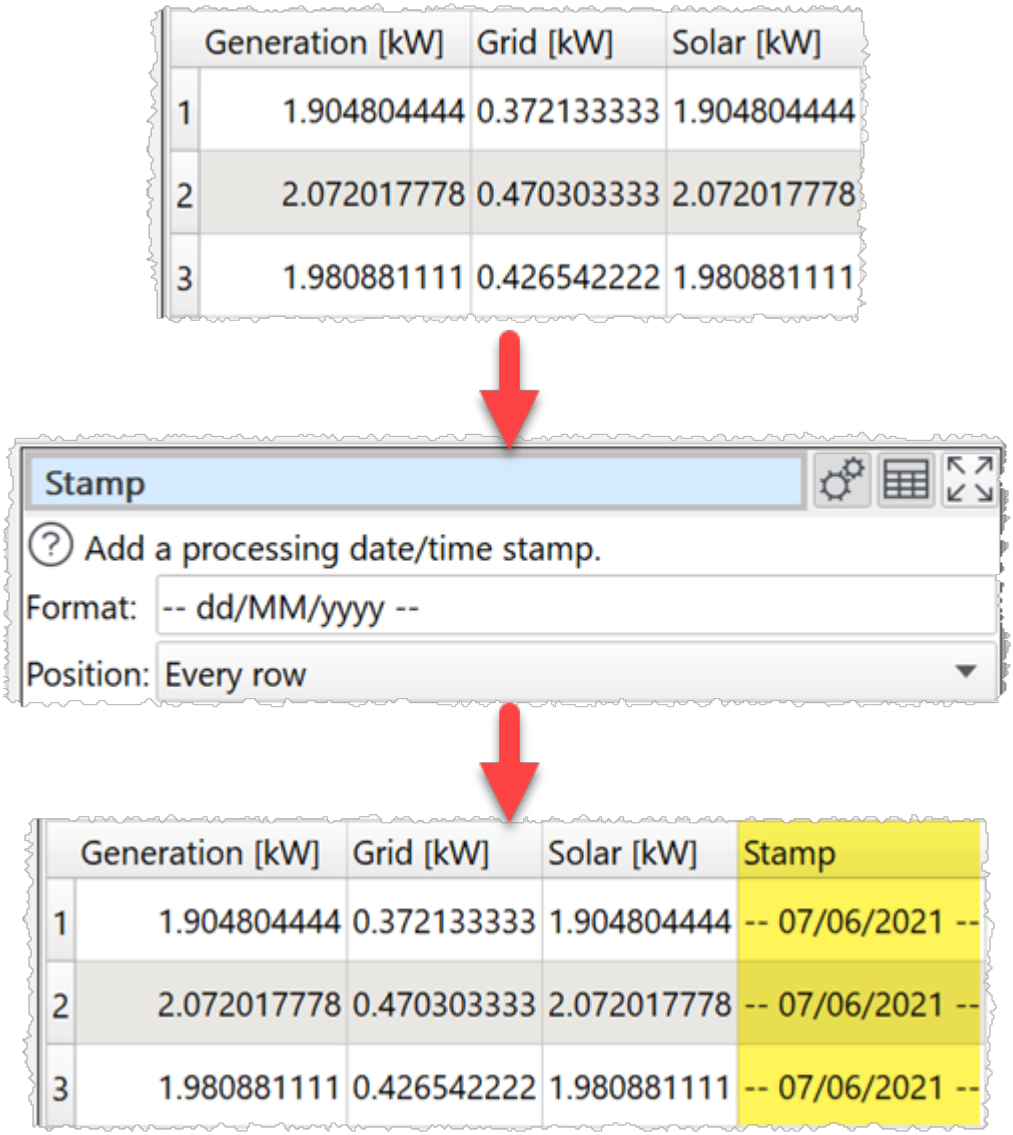
## Description

Adds a stamp with time, date and other information as a new row or a new column.

## Examples

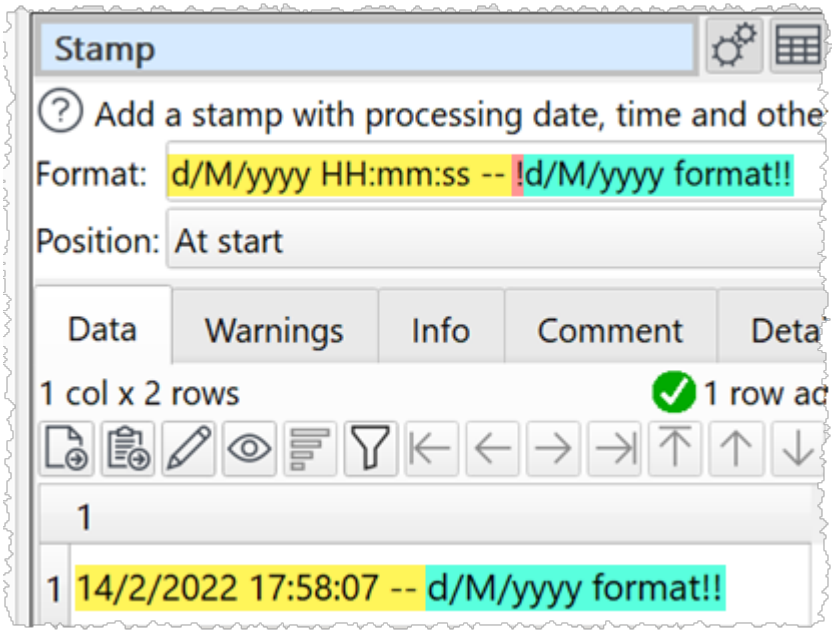
Add the current date to the end of each row:


© 2023 Oryx Digital Ltd




Add the current date and time to the top of a dataset with a comment:


© 2023 Oryx Digital Ltd




## Inputs

One.

## Options

- Supply the stamp format in **Format** (see below).

| Format | Meaning |
|---|---|
| `d` | The day as number without a leading zero (1 to 31) |
| `dd` | The day as number with a leading zero (01 to 31) |
| `ddd` | The abbreviated localized day name (e.g. 'Mon' to 'Sun'). Uses the locale to localize the name. |
| `dddd` | The long localized day name (e.g. 'Monday' to 'Sunday'). Uses the locale to localize the name. |
| `M` | The month as number without a leading zero (1 to 12). |
| `MM` | The month as number with a leading zero (01 to 12) |


*© 2023 Oryx Digital Ltd*

| Format | Meaning |
|---|---|
| MMM | The abbreviated localized month name (e.g. 'Jan' to 'Dec'). Uses the locale to localize the name. |
| MMMM | The long localized month name (e.g. 'January' to 'December'). Uses the locale to localize the name. |
| yy | The year as two digit number (00 to 99). |
| yyyy | The year as four digit number. If the year is negative, a minus sign is prepended in addition. |
| h | The hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display). |
| hh | The hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display). |
| H | The hour without a leading zero (0 to 23, even with AM/PM display). |
| HH | The hour with a leading zero (00 to 23, even with AM/PM display). |
| m | The minute without a leading zero (0 to 59). |
| mm | The minute with a leading zero (00 to 59). |
| s | The whole second without a leading zero (0 to 59). |
| ss | The whole second with a leading zero where applicable (00 to 59). |
| z | The fractional part of the second, to go after a decimal point, without trailing zeroes (0 to 999). Thus "s.z" reports the seconds to full available (millisecond) precision without trailing zeroes. |
| AP or A | The fractional part of the second, to millisecond precision, including trailing. |


© 2023 Oryx Digital Ltd

| Format | Meaning |
|---|---|
| `ap or a` | Use am/pm display. a/ap will be replaced by either "am" or "pm". |
| `t` | The timezone (for example "CEST"). |
| `$(c)` | The computer name. |
| `$(f)` | The path of the .transform file, e.g. `C:\Users\andy\Desktop\mytransform.transform`. |
| `$(u)` | The name of the user (from the `USER` or `USERNAME` environment variable). |
| `!` | Anything after the first `!` (exclamation) character won't be substituted. |

- Select from **Position** whether you want the stamp row added to the start or end of the dataset or to every row in a new column.

## Notes

- If you add the stamp to **Every Row** you can move the column using Reorder Cols.
- The **Format** text is case-sensitive.

## See also

- Meta information
- Stack

### 2.3.54 Stats

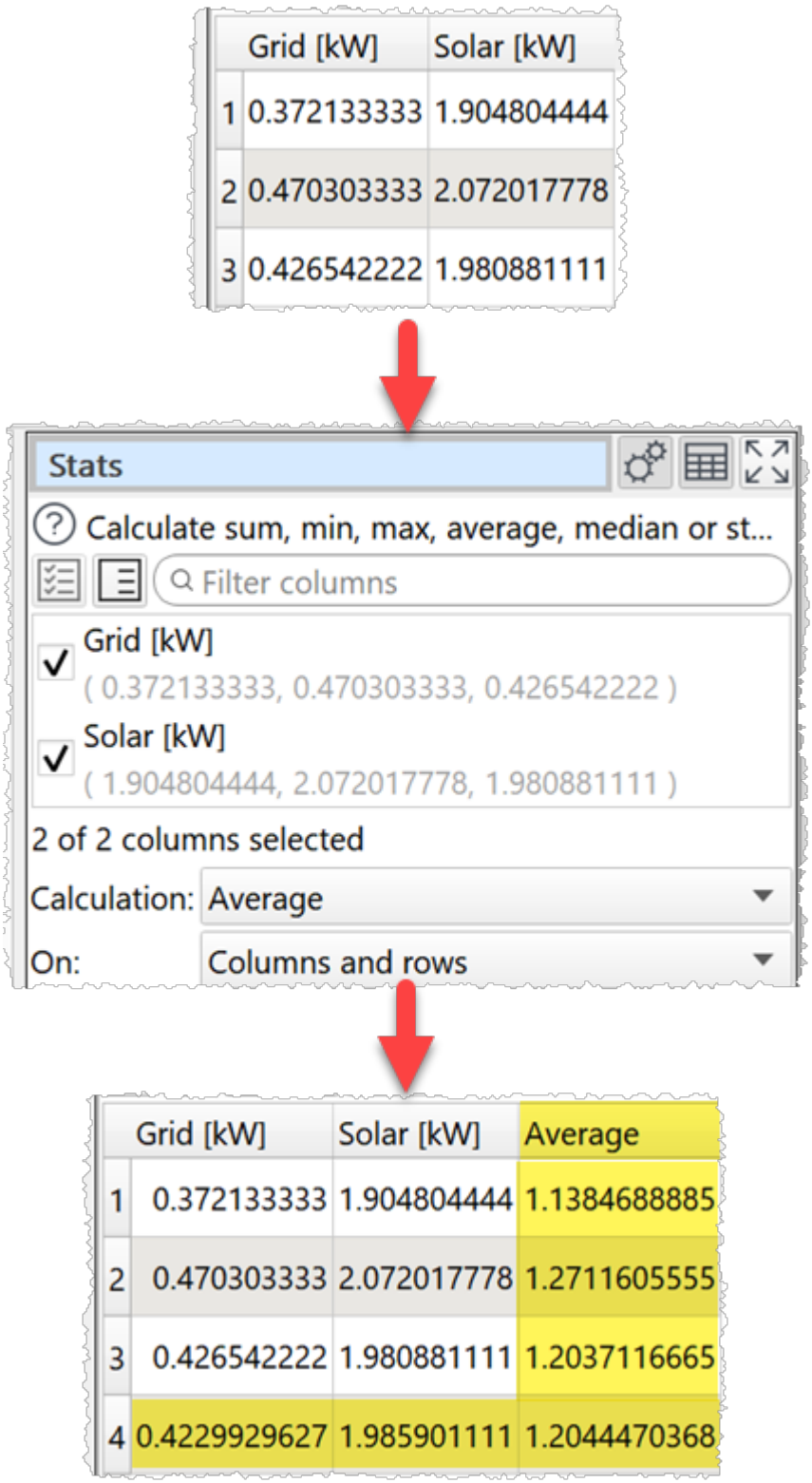
## Description

Calculates statistics by column and/or row in one or more selected columns.

## Example

Calculate column and row averages:


© 2023 Oryx Digital Ltd




## Inputs

One.


*© 2023 Oryx Digital Ltd*

## Options

- Check the column(s) you wish to calculate stats for.
- Set **Calculation** to the statistic you want to calculate:
  - **Sum** show the sum of the values.
  - **Minimum** shows the smallest value.
  - **Maximum** shows the largest value.
  - **Average** shows the arithmetic mean of the values.
  - **Median** shows the median of numeric values in the column.
  - **Mode** shows the mode of numeric values in the column.
  - **Standard deviation** is the sample standard deviation (equivalent to Excel function stddev.s).
  - **Variance** is the square of the sample standard deviation.
  - **Q1** is the first quartile (calculated using Method 1 here).
  - **Q3** is the third quartile (calculated using Method 1 here).
  - **IQR** is the Inter Quartile Range.
  - **Skew** is the alternative Pearson Mode Skewness (3 * (Mean – Median) / Standard Deviation ).
- Set **On** depending on whether you wish to calculate the statistics for columns, rows or both.
  - If **On** is set to **Columns** an extra row with the results is added to the bottom.
  - If **On** is set to **Rows** an extra column with the results is added to the right.
  - If **On** is set to **Columns and rows** an extra row with the results is added to the bottom and extra column with the results is added to the right. The bottom right cell contains the calculation across all values.

## Notes

- Non-numerical and empty values are ignored.
- **Mode** may not work as expected with non-integer values, due to precision issues. You can fix this by using a Num Format transform first.
- Use Num Format to change the precision of the results.
- Use Scale to convert the results to percentages.
- If you want to compute the sum, minimum, maximum or average for each distinct value in 1 or more columns you can use Unique.

## See also

- Count
- Pivot
- Summary

### 2.3.55 Substitute

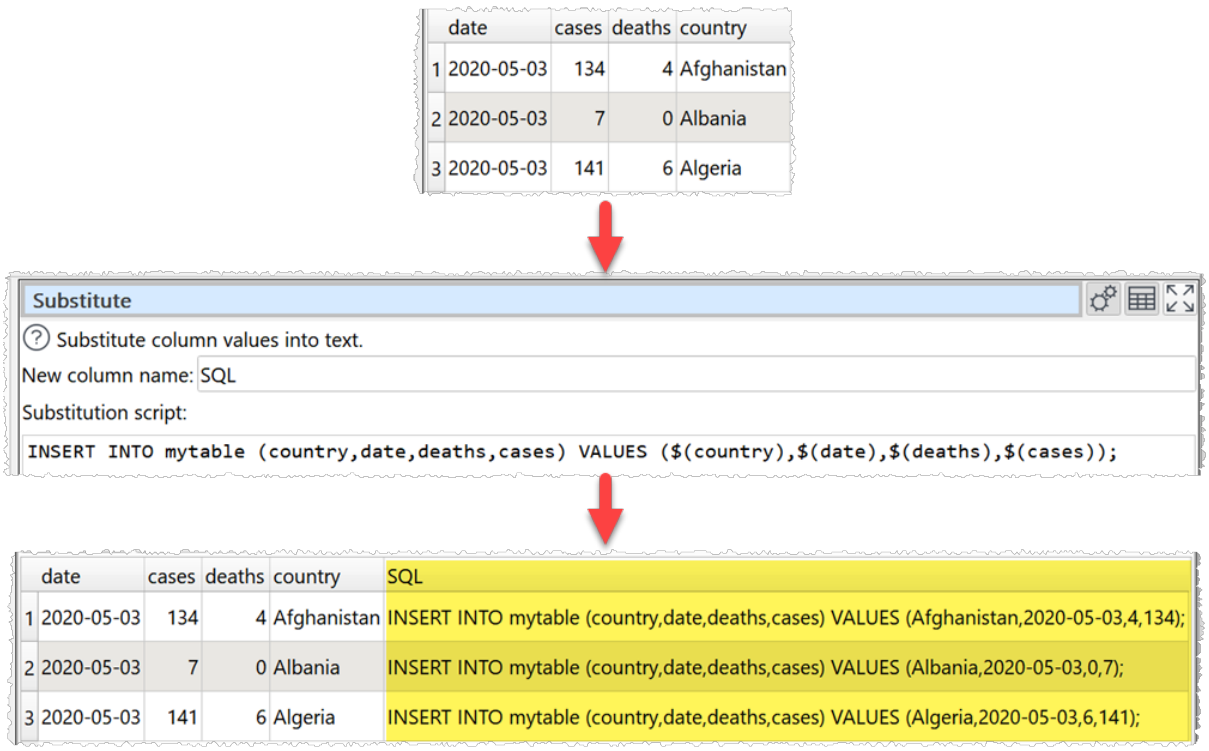
## Description

Substitute column values into text.


© 2023 Oryx Digital Ltd

## Example

Create an SQL statement to insert 'date', 'cases', 'deaths' and 'country' values into 'mytable':




| | date | cases | deaths | country |
|---|---|---|---|---|
| 1 | 2020-05-03 | 134 | 4 | Afghanistan |
| 2 | 2020-05-03 | 7 | 0 | Albania |
| 3 | 2020-05-03 | 141 | 6 | Algeria |

| | date | cases | deaths | country | SQL |
|---|---|---|---|---|---|
| 1 | 2020-05-03 | 134 | 4 | Afghanistan | INSERT INTO mytable (country,date,deaths,cases) VALUES (Afghanistan,2020-05-03,4,134); |
| 2 | 2020-05-03 | 7 | 0 | Albania | INSERT INTO mytable (country,date,deaths,cases) VALUES (Albania,2020-05-03,0,7); |
| 3 | 2020-05-03 | 141 | 6 | Algeria | INSERT INTO mytable (country,date,deaths,cases) VALUES (Algeria,2020-05-03,6,141); |

## Inputs

One.

## Options

- Enter your substitution script into the **Substitution script** field.
- Select a column from **Insert variable** to add that column variable into the **Substitution script** field at the current cursor position.
- Click the **Evaluate** button to evaluate your script over every row and show any errors (only available if **Run>Auto Run** is checked).

## Notes

- The new column is added at the right end. You can change the column order with Reorder Cols and the column name with Rename Cols.
- If you want to carry out your transform across more than one dataset, you should Join them first.
- If you need to do something more complex than this transform allows, try the Javascript transform.


*© 2023 Oryx Digital Ltd*

- Warnings are shown in the **Warnings** tab for ambiguous column references.

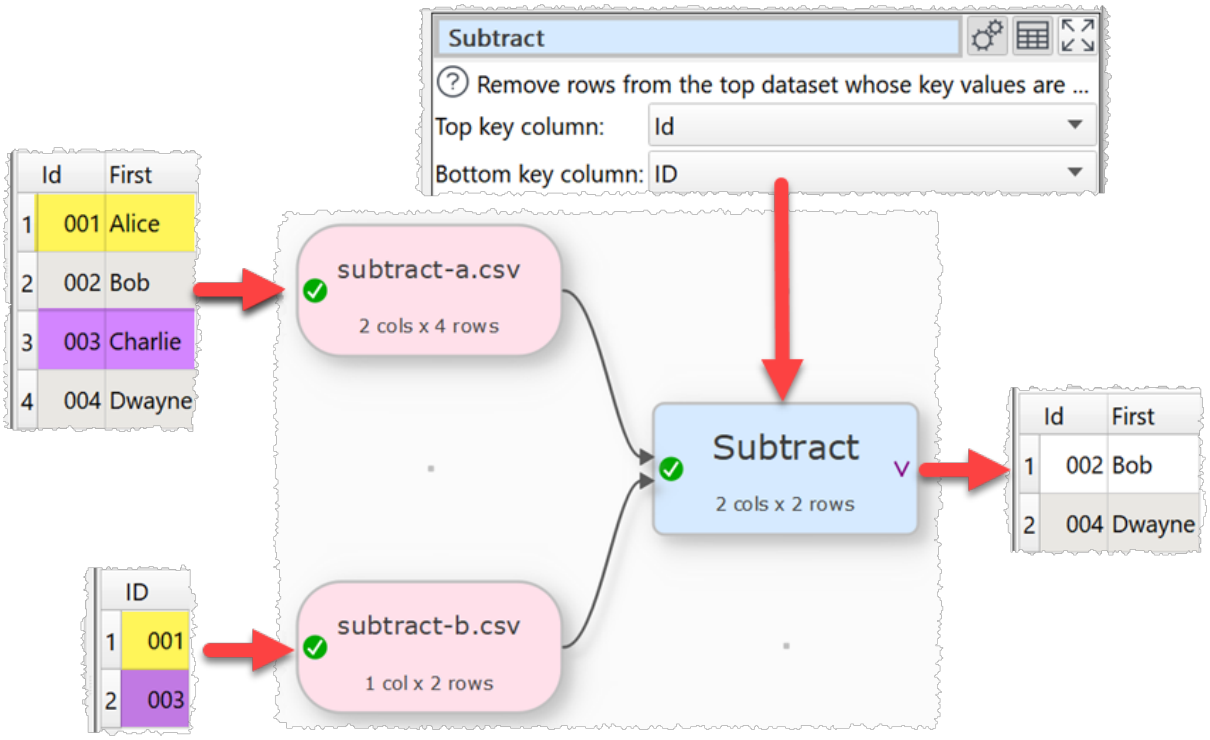
### 2.3.56 Subtract

## Description

Remove rows from the top dataset with key values that are present in the lower dataset.

## Example

Subtract from the top dataset all the records with ids in the bottom dataset:




## Inputs

Two.

## Options

- The output depends on the vertical (Y-axis) position of the inputs.
- Click **Explore Keys...** to compare key values in the 2 datasets.
- Select **Top key column** for the column you want to match in the top input dataset.
- Select **Bottom key column** for the column you want to match in the bottom input dataset.
- Check **case sensitive** to use case sensitive matching for keys.


*© 2023 Oryx Digital Ltd*

## Notes

- If there are 10k rows or less in both datasets, Easy Data Transform will try to guess sensible default values for **Top key column** and **Bottom key column** based on column header names and contents.
- If the first input has a header, this will be used for the output.
- All values are treated as text and comparisons are whitespace sensitive. You can use Whitespace to remove whitespace before the subtract.
- All rows in the top dataset with a given key value are removed if that key value occurs 1 or more times in the bottom dataset.
- You can use Concat Cols to join several columns together (e.g. 'first name' and 'last name' columns) to form a key column.
- You can use Row Num to create a unique key column.

## See also

- Intersect

### 2.3.57 Summary

## Description

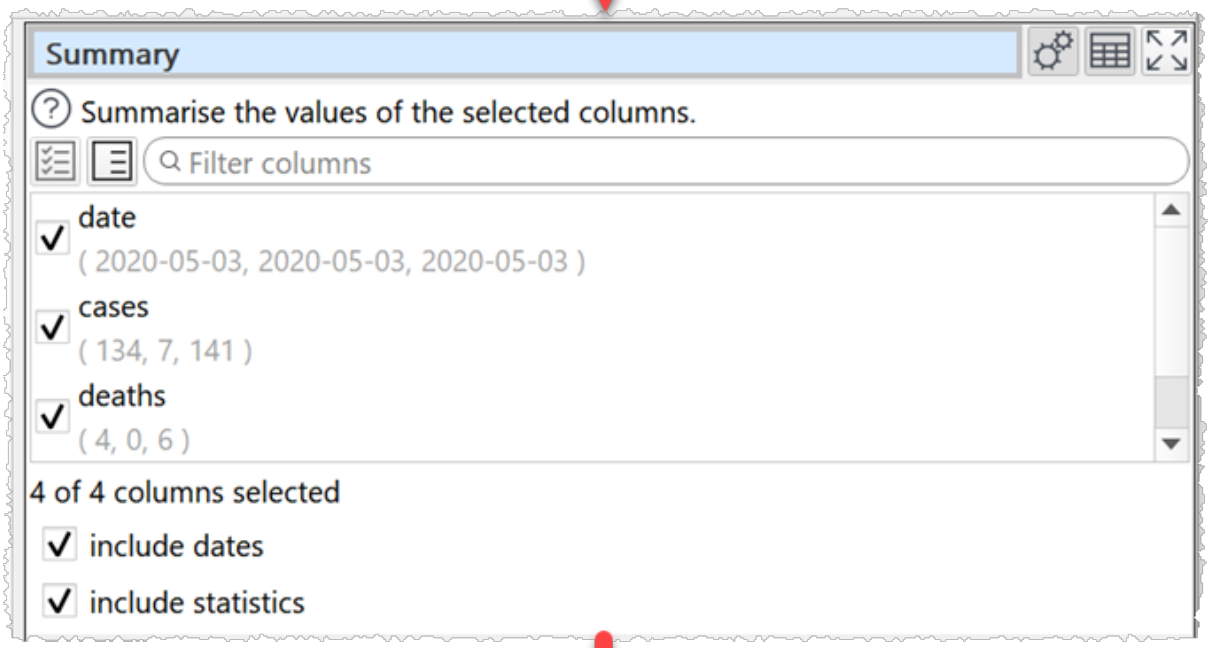
Summarize the values in the selected columns.

## Example

Create a summary, including dates and statistics:


© 2023 Oryx Digital Ltd

| | date | cases | deaths | country |
|---|---|---|---|---|
| 1 | 2020-05-03 | 134 | 4 | Afghanistan |
| 2 | 2020-05-03 | 7 | 0 | Albania |
| 3 | 2020-05-03 | 141 | 6 | Algeria |



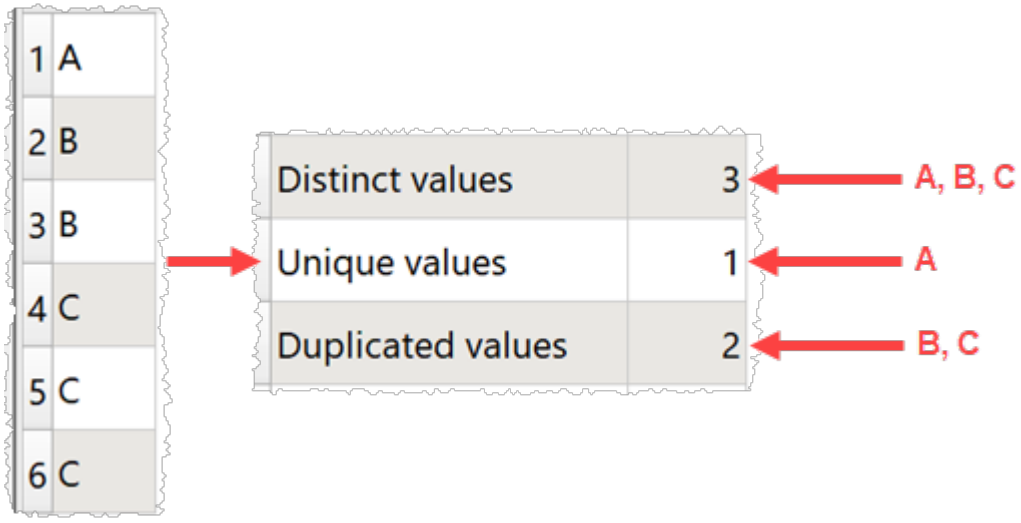

| | Metric | date | cases | deaths | country |
|---|---|---|---|---|---|
| 1 | Empty values | 0 | 0 | 0 | 0 |
| 2 | Non-empty values | 3 | 3 | 3 | 3 |
| 3 | Numeric values | 0 | 3 | 3 | 0 |
| 4 | Integer values | 0 | 3 | 3 | 0 |
| 5 | Date values | 3 | 0 | 0 | 0 |
| 6 | Text values | 0 | 0 | 0 | 3 |
| 7 | Distinct values | 1 | 3 | 3 | 3 |
| 8 | Unique values | 0 | 3 | 3 | 3 |
| 9 | Duplicated values | 1 | 0 | 0 | 0 |
| 10 | Min length | 10 | 1 | 1 | 7 |

## Inputs
One.

## Options
- Select the **Columns** you wish to summarize.
- Check **include dates** if you wish to check for date values using supported date formats. This can be slow for large datasets.
- Check **include statistics** if you wish to add the sum, average, median and standard deviation for numeric values.

## Notes
- Whether values are interpreted as number or dates depends on **Supported date formats** and locale.
- **Empty values** is the number of values in the column that are completely empty. Values with whitespace do not count as empty.
- **Non-empty values** is the number of values in the column that are not completely empty. Values with whitespace do not count as empty.
- **Numeric values** is the number of numeric of values in the column that can be interpreted as a number.
- **Integer values** is the number of numeric of values in the column that can be interpreted as integers (whole numbers). "1.0" (depending on your locale), "0", "-1" and "1e3" are considered integers.
- **Real values** is the number of numeric values in the column that can be interpreted as reals (floating point numbers). "1.2345" (depending on your locale) and "1e-3" are considered reals.
- **Boolean values** is the number of values in the column that can be interpreted as booleans (true/false). "TRUE", "True" and "false" are considered booleans.
- **Date values** is the number of values in the column that can be interpreted as a date. Only shown if **check for dates** is checked.
- **Text values** is the number of values in the column that cannot be interpreted as empty, numeric or date.
- **Distinct values** is the number of different values in the column. Date and numeric values are treated as text (e.g. '7' is treated as different to '7.0' and '1/1/2020' is treated as different to '01/01/2020'). Comparison between values is sensitive to case and whitespace.
- **Unique values** is the number of values that only occur once in the column. Date and numeric values are treated as text. Comparison between values is sensitive to case and whitespace.
- **Duplicated values** is the number of values that occur more than once in the column. Date and numeric values are treated as text. Comparison between values is sensitive to case and whitespace.


*© 2023 Oryx Digital Ltd*




- **Min length** is the minimum number of characters of non-empty values in the column. Whitespace is counted. Date and numeric values are treated as text.
- **Max length** is the maximum number of characters of values in the column. Whitespace is counted. Date and numeric values are treated as text.
- **Average length** is the arithmetic mean of the number of characters of non-empty values in the column. Whitespace is counted. Date and numeric values are treated as text.
- **Min numeric** is the minimum numeric value in the column. Non-numeric and empty values are ignored.
- **Max numeric** is the maximum numeric value in the column. Non-numeric and empty values are ignored.
- **Range numeric** is **Max numeric - Min numeric.**
- **Negative numeric** is the number of negative numeric values in the column. 0 is not considered negative here.
- **Zero numeric** is the number of numeric zero values in the column.
- **Positive numeric** is the number of positive numeric values in the column. 0 is not considered positive here.
- **Sum numeric** is the sum of numeric values in the column. Non-numeric and empty values are ignored. Only shown if **include statistics** is checked.
- **Average numeric** is the arithmetic mean of numeric values in the column. Non-numeric and empty values are ignored. Only shown if **include statistics** is checked.
- **Median numeric** is the median of numeric values in the column. Non-numeric and empty values are ignored. Only shown if **include statistics** is checked.
- **Mode numeric** is the mode of numeric values in the column. Non-numeric and empty values are ignored. Only shown if **include statistics** is checked.
- **Stddev numeric** is the sample standard deviation (equivalent to Excel function stddev.s) of numeric values in the column. Non-numeric and empty values are ignored. Only shown if **include statistics** is checked.
- **Variance numeric** is the square of the sample standard deviation. Only shown if **include statistics** is checked.


*© 2023 Oryx Digital Ltd*

- **Q1 numeric** is the first quartile (calculated using Method 1 here). Only shown if **include statistics** is checked.
- **Q3 numeric** is the third quartile (calculated using Method 1 here). Only shown if **include statistics** is checked.
- **IQR numeric** is the Inter Quartile Range. Only shown if **include statistics** is checked.
- **Skew numeric** is the alternative Pearson Mode Skewness (3 * (Mean – Median) / Standard Deviation ). Only shown if **include statistics** is checked.
- **Min date** is the minimum date value in the column. Only shown if **include dates** is checked.
- **Max date** is the maximum date value in the column. Only shown if **include dates** is checked.
- **Most frequent** lists the most common text in the column. Empty values are not counted. Date and numeric values are treated as text. Comparison between values is sensitive to case and whitespace.

## Notes

- If your dataset is large, you might want to Sample it first.
- Use Num Format to change the precision of numerical results.
- You can use Whitespace to remove any whitespace at the start or end of values before **Summary**.
- If you wish to have a row displayed per column you can Transpose the table first.
- **Mode** may not work as expected with non-integer values, due to precision issues. You can fix this by using a Num Format transform first.

## See also

- Count
- Ngram
- Pivot
- Stats

### 2.3.58 Total

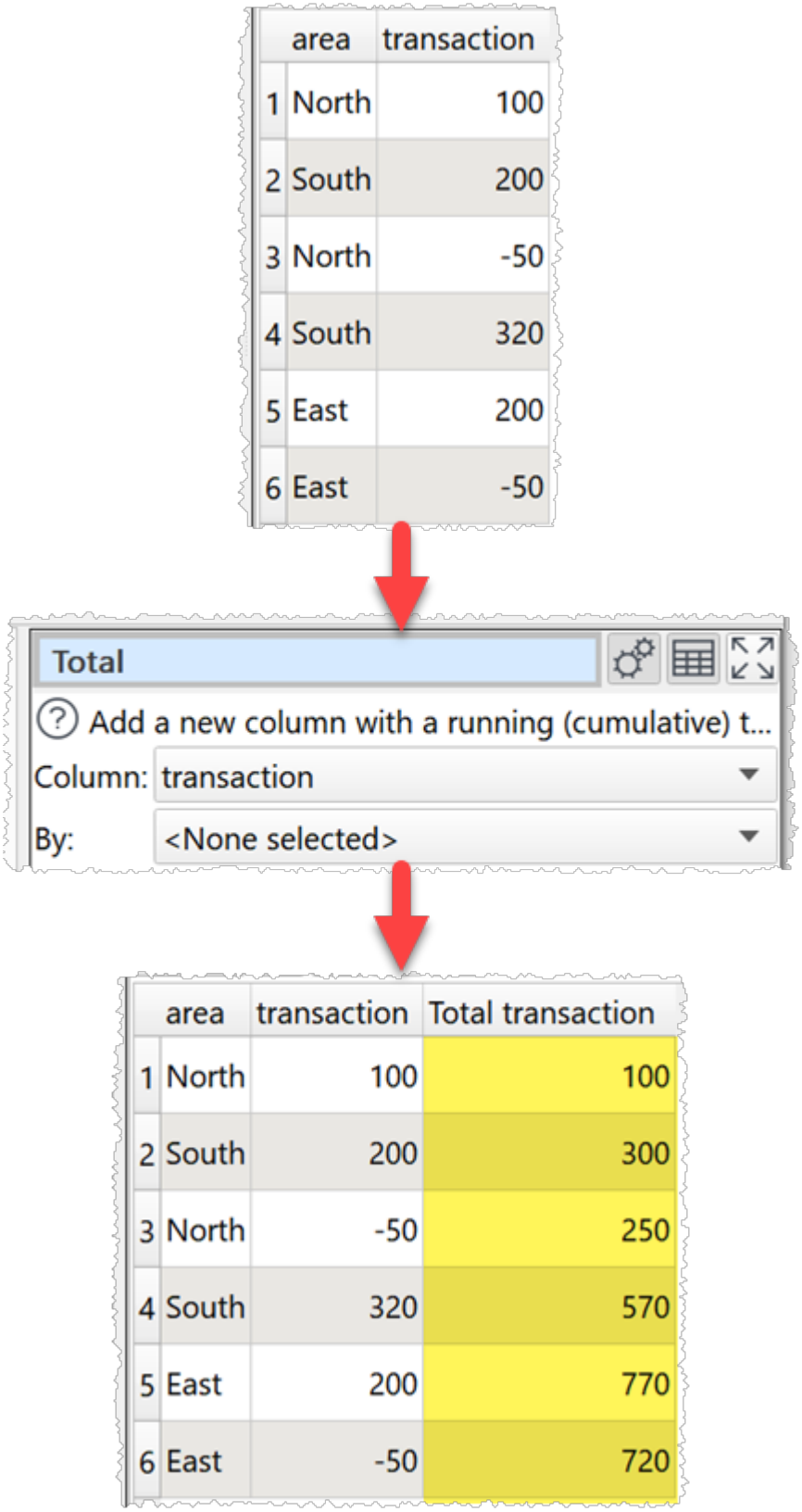
## Description

Add a new column with a running (cumulative) total of the selected column.

## Examples

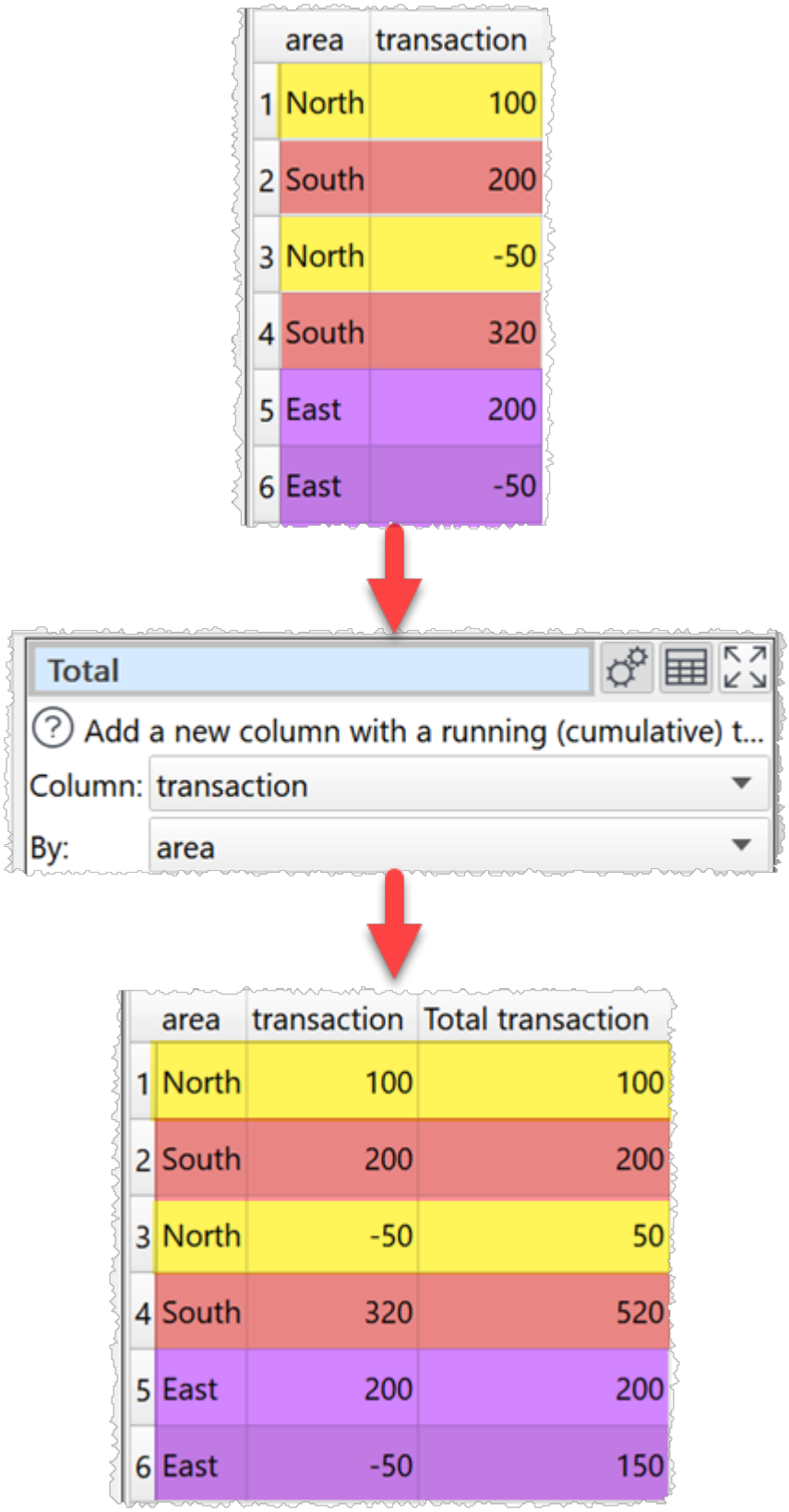
Total the 'transaction' column:


© 2023 Oryx Digital Ltd




Total the 'transaction' column separately for each 'area' value:


*© 2023 Oryx Digital Ltd*





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Set **Column** to the column you want to total.
- Set **By** to a column if you want a separate running total per value in that column. Leave as **<None selected>** if you only want a single running total.

## Notes
- Non-numerical values are ignored.
- The **By** column takes account of case and whitespace in values.
- The new column is added at the right end. You can change the column order with Reorder Cols and the column name with Rename Cols.

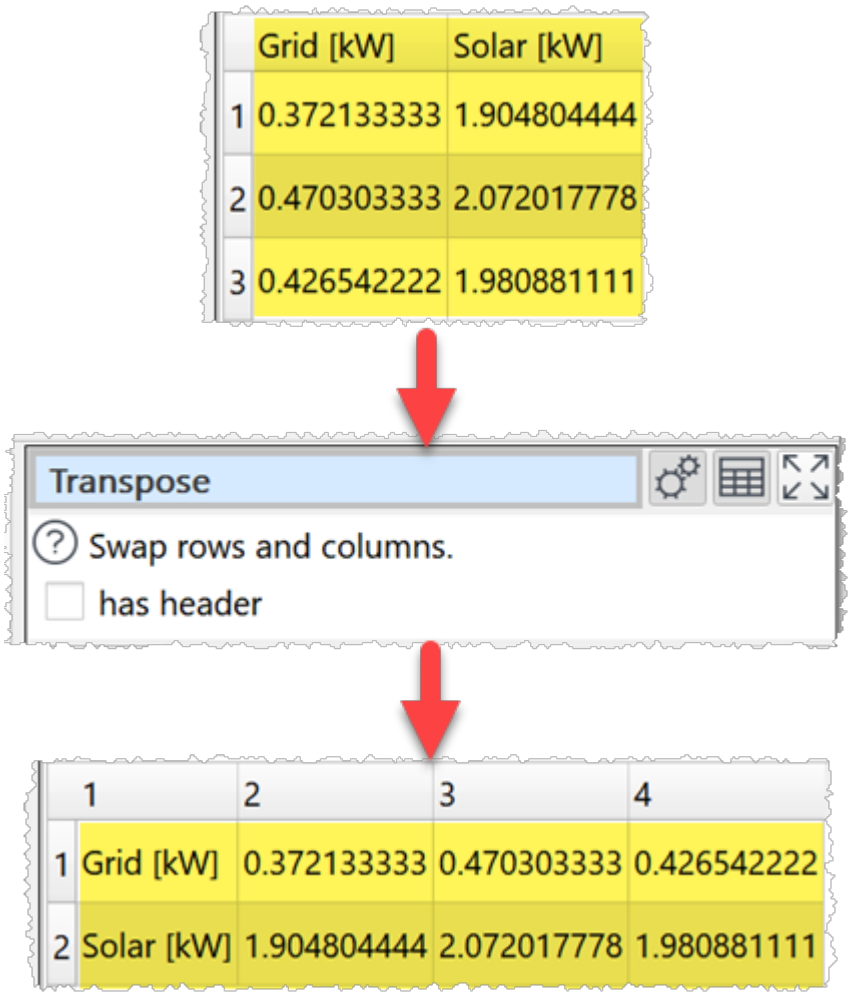
## See also
- Count
- Pivot
- Stats

### 2.3.59 Transpose

## Description
Swap (rotate) rows and columns, so that each row becomes a column and each column becomes a row.

## Example
Swap rows and columns:


*© 2023 Oryx Digital Ltd*




## Inputs

One.

## Options

- Check **has header** to make the new first row into a header.

## Notes

- If the input dataset has a header, it will become the new first column. Use Remove Cols to remove it.

### 2.3.60 Unfill

## Description

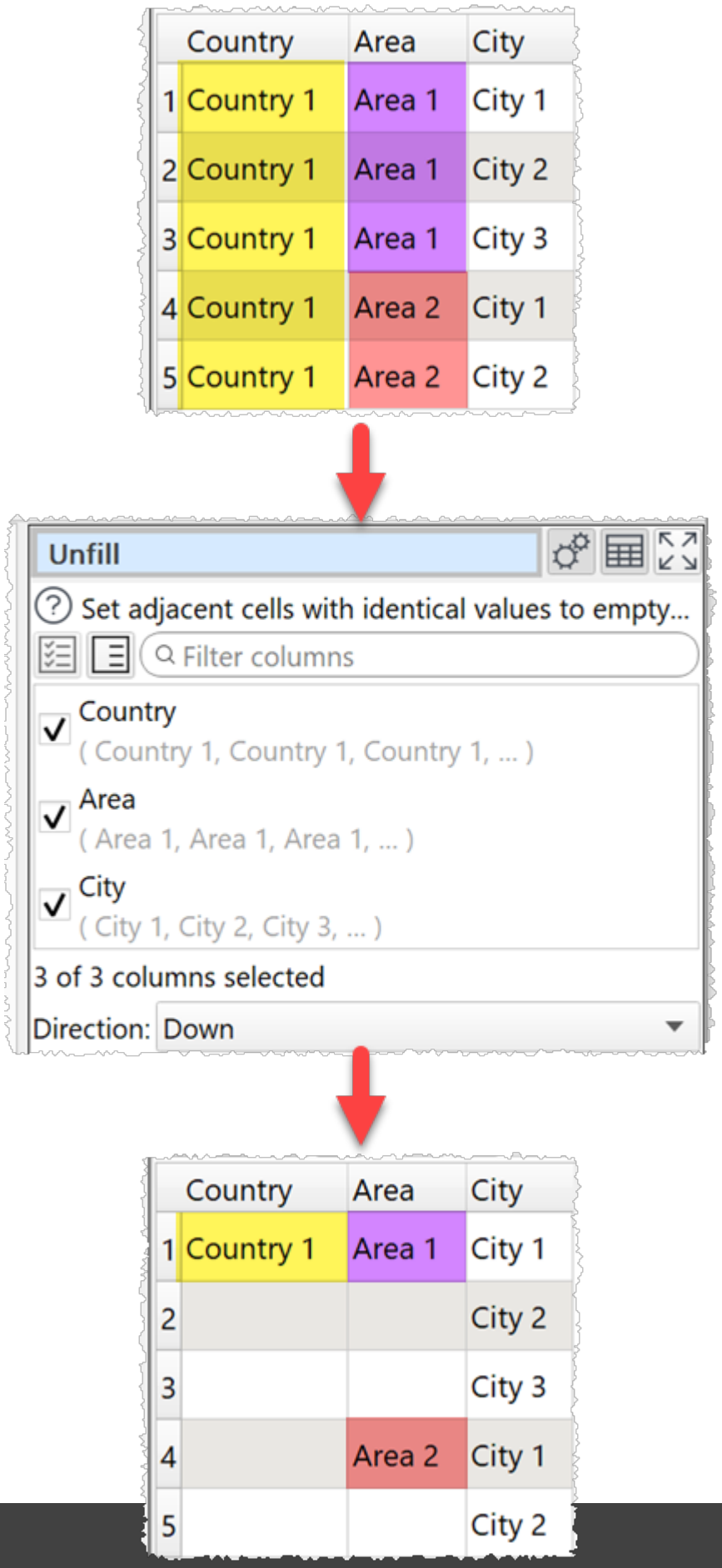
Set adjacent cells with identical values to empty in selected columns.


*© 2023 Oryx Digital Ltd*

### Example

Unfill down:


© 2023 Oryx Digital Ltd





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) you wish to unfill.
- Select **Direction** depending on the direction you wish to unfill to.

## See also
- Fill

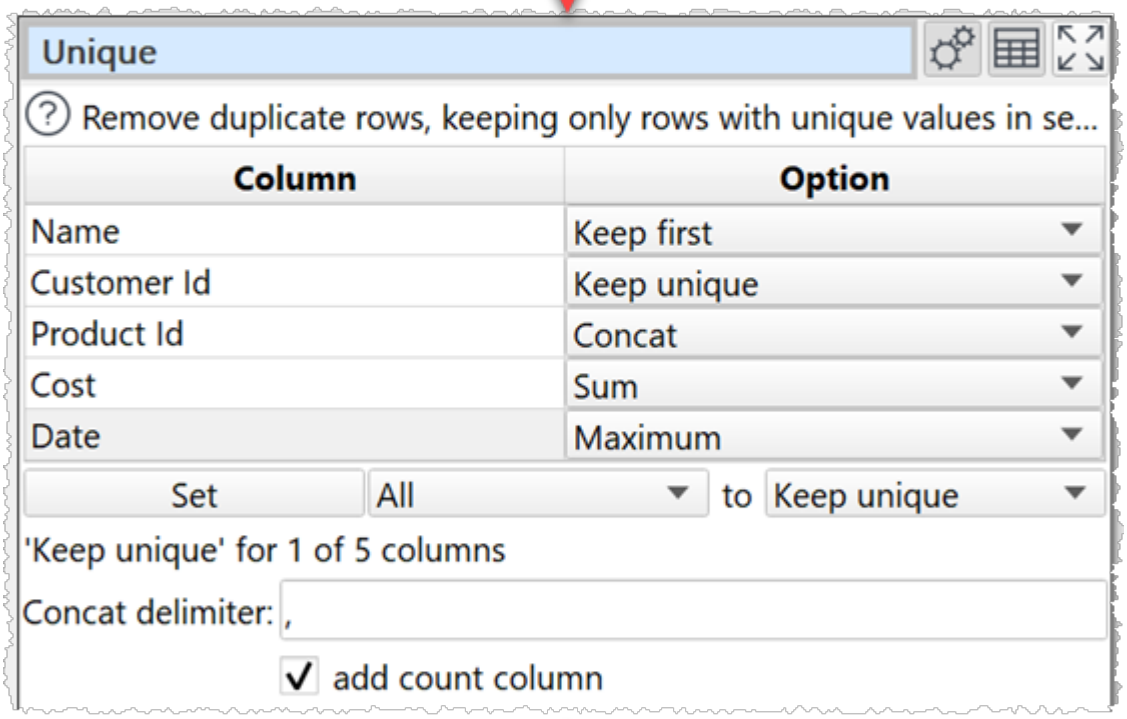
### 2.3.61 Unique

## Description
Remove duplicate rows based on keeping only unique values in selected columns

## Example
If you have a dataset of orders and you want to:
- keep one row per unique Customer Id
- keep the first listed Name for each Customer Id
- concatenate Product Ids for each Customer Id, delimited by a comma
- sum the Costs for each Customer Id
- keep the latest Date for each Customer Id
- add a Count column showing how many rows in the input correspond to each row in the output


*© 2023 Oryx Digital Ltd*

| | Name | Customer Id | Product Id | Cost | Date |
|---|---|---|---|---|---|
| 1 | Alice Anderson | C018930 | 13574 | 29.95 | 01/10/2020 |
| 2 | Bob Brown | C018917 | 89456 | 10.55 | 01/10/2020 |
| 3 | Charlie Jones | C017783 | 96352 | 19.95 | 02/10/2020 |
| 4 | Robert Brown | C018917 | 98526 | 10.00 | 02/10/2020 |
| 5 | Charles Jones | C017783 | 38746 | 25.00 | 03/10/2020 |




| | Name | Customer Id | Product Id | Cost | Date | Count |
|---|---|---|---|---|---|---|
| 1 | Alice Anderson | C018930 | 13574 | 29.95 | 01/10/2020 | 1 |
| 2 | Bob Brown | C018917 | 89456,98526 | 20.55 | 02/10/2020 | 2 |
| 3 | Charlie Jones | C017783 | 96352,38746 | 44.95 | 03/10/2020 | 2 |


© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Set an **Option** for each column:
  - Only 1 row is kept where all the **Keep unique** columns have the same value.
  - **Keep first** keeps first value in the current sort order.
  - **Keep last** keeps the last value in the current sort order.
  - **Sum** sums any numerical values. Blank values are ignored.
  - **Maximum** keeps the maximum numerical or date value.
  - **Minimum** keeps the minimum numerical or date value.
  - **Average** takes the average (mean) of any numerical values. Blank values are ignored.
  - **Concat** to concatenate values. Duplicate values are kept. All values are treated as text.
  - **Concat unique** to concatenate values. Duplicate values are ignored. All values are treated as text.
- Use the **Set** button to quickly set the option value for multiple columns.
- Set **Concat delimiter** if you want to add a delimiter between **Concat** or **Concat unique** values
- Check **add count column** to add a column showing how many rows in the input dataset created each unique row.
- Check **drilldown** to allow double-clicking a row in the data table to drilldown to the rows that contributed to this row in the upstream data.

## Notes
- Rows are considered duplicates if they have exactly the same value in all the columns set to **Keep unique**. Comparisons are case and whitespace sensitive. You can use Case and Whitespace to change case and whitespace before deduping.
- If no columns are set to **Keep unique** the transform won't do anything.
- If you are using **Keep first** or **Keep last** the sort order is important. You can use Sort to change the sort order before deduping.
- The Dedupe transform is a less powerful (but simpler and faster) alternative to Unique.
- You can use Split Col **Into** rows to perform the opposite function to Unique.
- The opposite Unique is Split Col with **Into** as **Rows**

## See also
- Dedupe a dataset

### 2.3.62 Units

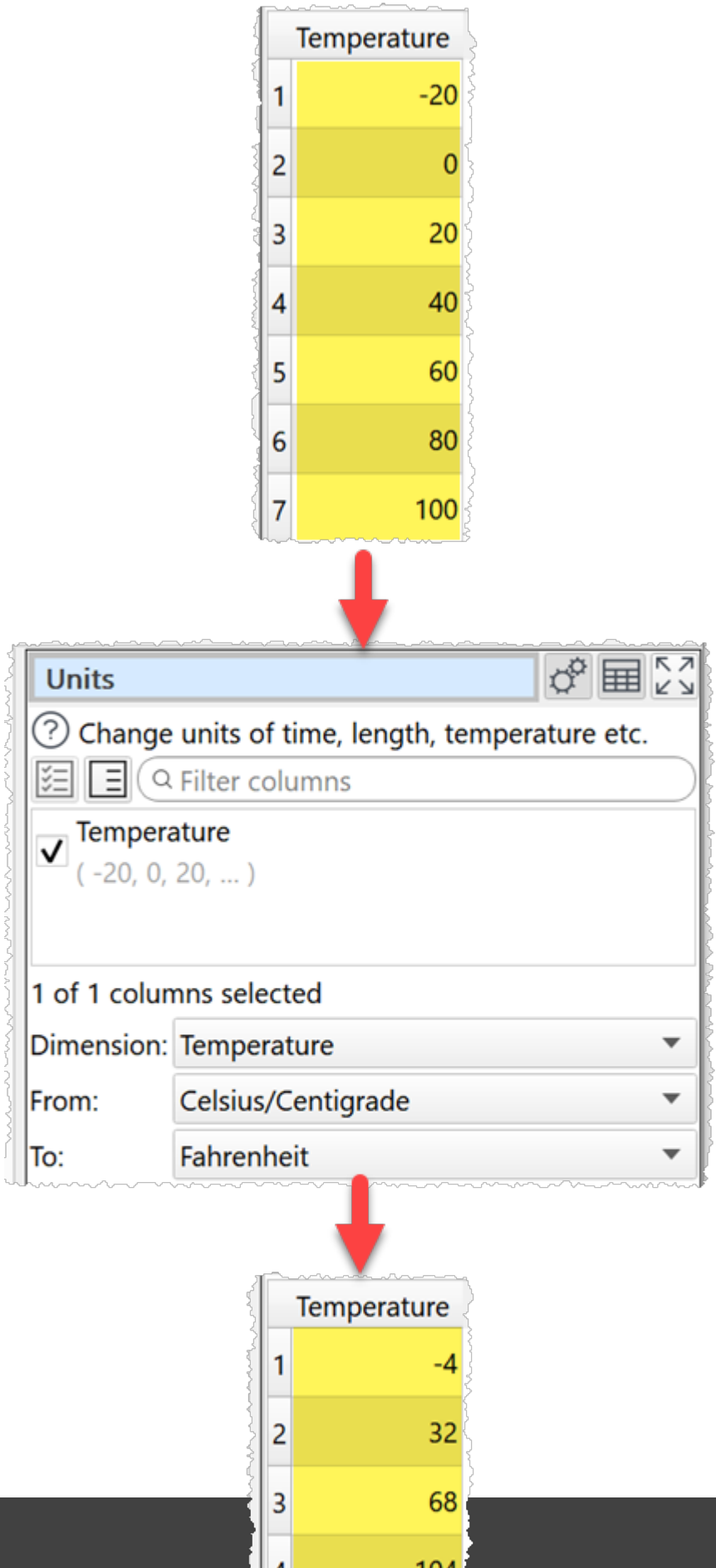
## Description
Change units of time, length, temperature etc.


© 2023 Oryx Digital Ltd

### Example

Convert the 'Temperature' column from Celsius to Fahrenheit:


*© 2023 Oryx Digital Ltd*





© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) you wish to transform.
- Set **Dimension** to the dimensions of the units you wish to convert.
- Set **From** to the units you wish to convert from.
- Set **To** to the units you wish to convert to.

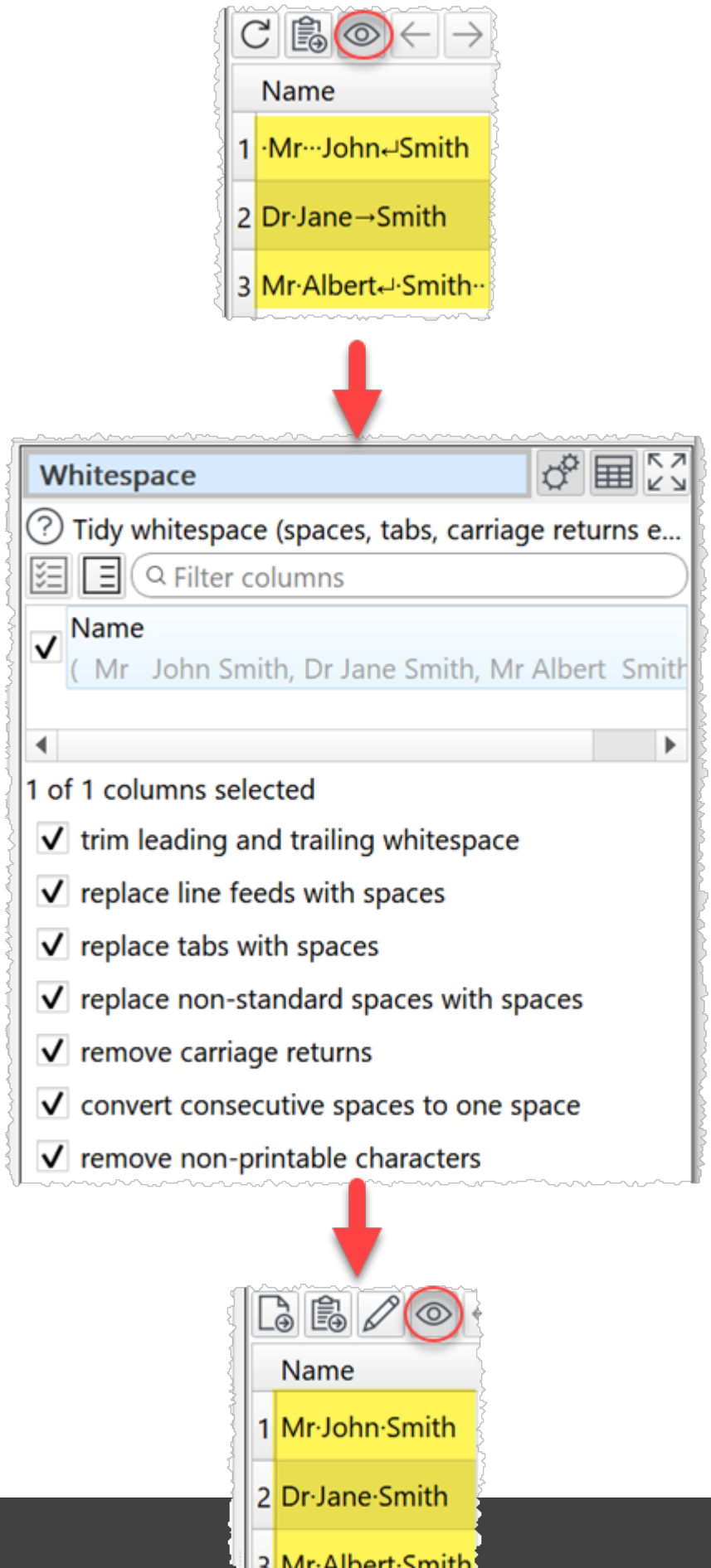
## Notes
- The locale is used to decide how the number is represented (e.g. decimal separators).
- Unit disambiguation:
  - **Years** are 365.25 days.
  - **Calories** are thermal/small/gram (not nutritional) Calories.
  - **Miles** are International/Statute Miles.
  - **Nautical miles** are International Nautical Miles.
  - **Gallons (US)** are liquid gallons.
  - **Pints (US)** are liquid pints.
- Warnings are shown in the **Warnings** tab for non-numeric values.
- Non-numerical and empty values are not changed.
- Use Copy Cols to copy columns before you transform them.
- Use Num Format to change the number format of the results.
- Use Extract or Split Col to remove any unit symbols from the value before conversion, e.g. to change "0.0 Kg" to "0.0".
- If you want to convert units not covered here you can do it using the Javascript transform.

### 2.3.63 Whitespace

## Description
Tidy whitespace (spaces, tabs, carriage returns etc) in the selected column(s).

## Example
Tidy tabs, carriage returns and repeated spaces:


© 2023 Oryx Digital Ltd



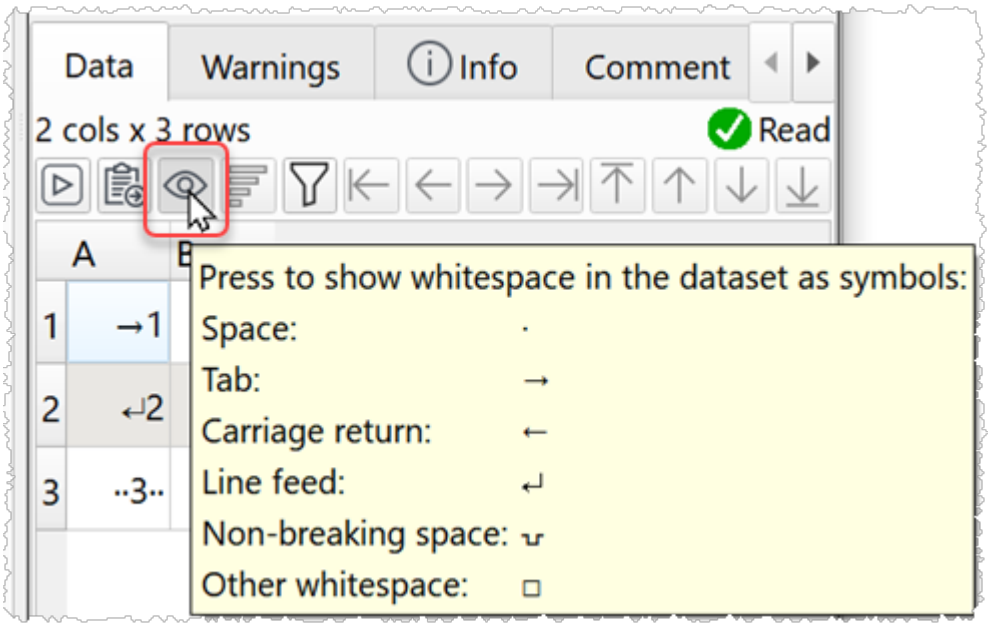


© 2023 Oryx Digital Ltd

## Inputs
One.

## Options
- Check the column(s) you wish to transform.
- Check **trim leading and trailing whitespace** to remove whitespace characters, such as space and tab.
- Check **replace line feeds with spaces** to replace LF (\n) characters with spaces.
- Check **replace tabs with spaces** to replace tab (\t) characters with spaces.
- Check **replace non-standard spaces with spaces** to replace non-standard spaces (such as non-breaking space, thin space etc) with spaces.
- Check **remove carriage returns** to remove CR (\r) characters.
- Check **convert consecutive spaces to one space** to replace 2 or more consecutive spaces with a single space.
- Check **remove non-printable characters** to remove characters of Unicode type Other_*. This include ASCII codes 0 to 31, such as tab, line feed, carriage return, bell and backspace. It does not remove spaces.

## Notes
- The operations are carried out in top to bottom order, e.g. **Replace line feeds with spaces** is carried out before **Convert consecutive spaces to one space**.
- Click the eye icon in the **Right** pane to show whitespace in the data.





© 2023 Oryx Digital Ltd

### See also

- Case

## 2.4 Output

### 2.4.1 Output data

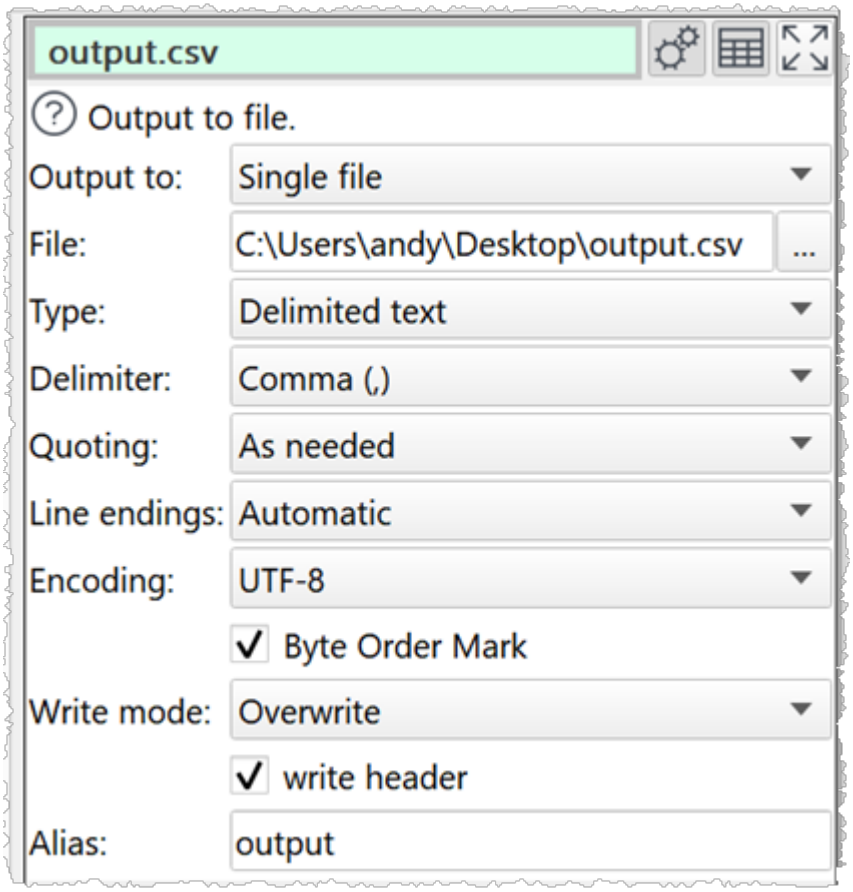
Once you have finished transforming your data you can output it in the following formats:

- CSV
- Excel
- JSON
- HTML
- Markdown
- Plain text
- TSV
- vCard
- XML
- YAML

To create an output, select 1 input and/or transform item in the Center pane and then click the **To File** button at the bottom of the Left pane. You can choose the file type in the **Save as type** drop-down list of the **Output** window.

You can select the output item in the **Center** pane and change any options related to the output in the Right pane.


*© 2023 Oryx Digital Ltd*




Set **Output to** depending on whether you want to output the dataset to a single file or multiple files.

Set **File** to the location of the file you want to output (only available when **Output to** set to **Single file**). The location can use file name variables to dynamically change output file names based on input file names.

Set **Files column** to the column containing the location of the files you want to output to (only available when **Output to** set to **Multiple files**).

- This column is not output.
- The location can be an absolute location (e.g. `c:\users\andy\output.csv`) or a location relative to the .transform file location (e.g. `results\output.csv`).
- The location can use file name variables to dynamically change output file names based on input file names.
- Folders output to must already exist.
- Empty or invalid locations (e.g. file names with illegal characters) are ignored.
- Check **Confirm files** if you want to manually confirm before writing to output files (ignored for command line processing).


*© 2023 Oryx Digital Ltd*

If you are writing to an Excel file the output will be written to a sheet called 'Easy Data Transform' by default. You can change this by adding the sheet name inside [], e.g. `myfile.xlsx[mysheet]`.

Set **Type** to the file type. The default type will be set according to the file extension and the settings in the **Output Extensions** tab of the Preferences window.

Set **Value types** depending on how you want to set JSON types (only available for JSON files).

- **Automatic** to let Easy Data Transform decide the JSON type depending on the contents of each column.
- **String** to set the JSON type for all columns to String
- **Manual** to choose the JSON type of each column as:
  - Automatic (based on column contents).
  - Boolean.
  - Number.
  - String.

If you output a column as Boolean or Number any values not of that type will be output as null.

Set **Delimiter** to the delimiter you wish to use (only available for delimited text files, such as CSV and TSV).

Set **Quoting** depending how you want to use quoting in the output (only available for delimited text files, such as CSV and TSV).

- **As needed** to add quotes (") around values that contain the delimiter character, quote characters  or carriage returns.
- **Always** to add " quotes around every value.
- **Never** to never add " quotes around values (delimiters and carriage returns with values are replaced with spaces).

Set **Line endings** to the control characters you wish to use to denote line endings (only available for text files).

- **Automatic** to choose the standard line ending for the current operating system.
- **Windows (CRLF)** to choose the standard Windows line ending, carriage return+line feed (`\r\n`).
- **Mac\Unix (LF)** to choose the standard macOS/Unix line ending, line feed (`\n`).
- **Old Mac (CR)** to choose the standard classic (pre mac OS X) Mac line ending, carriage return (`\n`).

Set **Encoding** to the text encoding you wish to use (only available for text files).


*© 2023 Oryx Digital Ltd*

Set **Formatting** depending on how you want to set the Excel cell formatting (only available for Excel files).

- **Automatic** to let Easy Data Transform decide cell formatting depending on the contents of each column.
- **General** to set the cell format for all columns to 'General'.
- **Manual** to choose the cell format of each column as:
  - Automatic (based on column contents).
  - Boolean (expects `true` or `false`, not case-sensitive)
  - Date (expects a date format in Preferences).
  - General.
  - Number (expects a real or integer number, e.g. `123` or `123.456`)
  - Text.
  - Formula (expects starting with =).
  - Time (expects `hh:mm:ss` or `hh:mm`, e.g. `13:59:01` or `13:59`).
  - Hyperlink (expects a hyperlink, e.g. `https://www.easydatatransform.com`).

Set **Byte Order Mark** checked to write a Unicode Byte Order Mark to the file (only available for UTF encodings and ignored when appending to an existing file).

Set **Escape special characters** checked to escape HTML special characters (for example < and >) so they are rendered correctly in the HTML (only available for HTML files). Uncheck it to write 'raw' HTML. Note that you can also set **Change** to **To escaped HTML** to escape selected columns in the Decode transform.

Check **remove empty** to recursively remove any nodes that have an empty value and no children (only available for JSON and XML files). For example:

| | A | A.B | A.B.C |
|---|---|---|---|
| 1 | | | |
| 2 | | 1 | |
| 3 | | | 1 |

Is output to XML with **remove empty** checked as:

```
<root>
  <record>
    <A>
      <B>1</B>
    </A>
  </record>
  <record>
```

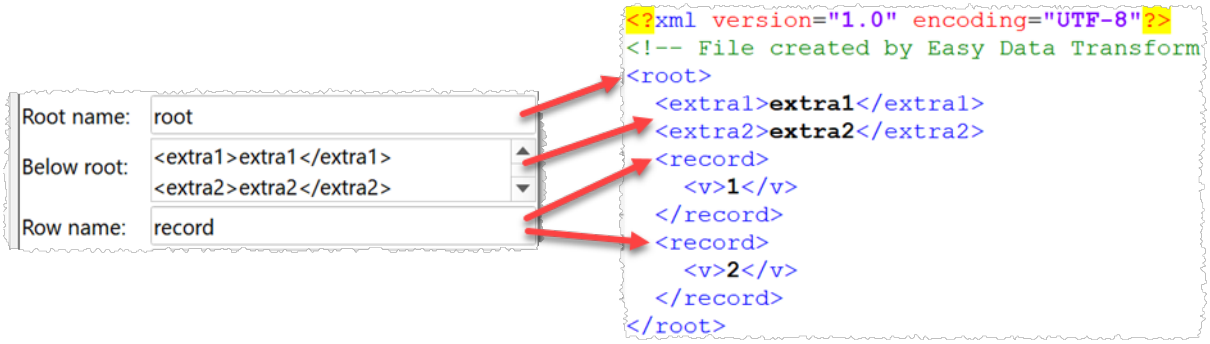

© 2023 Oryx Digital Ltd

```
    <A>
      <B>
        <C>1</C>
      </B>
    </A>
  </record>
</root>
```

Is output to XML with **remove empty** unchecked as:

```
<root>
  <record>
    <A>
      <B>
        <C></C>
      </B>
    </A>
  </record>
  <record>
    <A>
      <B value="1">
        <C></C>
      </B>
    </A>
  </record>
  <record>
    <A>
      <B>
        <C>1</C>
      </B>
    </A>
  </record>
</root>
```

For XML output you can set **Root name**, **Below root** and **Row name** depending on the name you want to use for the root and row records and any extra information you want to add below the root.




Use **Write mode** to determine how existing files are treated:


© 2023 Oryx Digital Ltd

| **Write mode** for Excel files | **File exists with named sheet** | **File exists without named sheet** | **File does not exist** |
|---|---|---|---|
| **Overwrite / File** | Overwrite named sheet, delete all other sheets | Add named sheet, delete all other sheets | Create file with only named sheet |
| **Overwrite / Sheet** | Overwrite named sheet* | Add named sheet | Create file with only named sheet |
| **Append** | Append to named sheet | Add named sheet | Create file with only named sheet |
| **New** | Do nothing | Do nothing | Create file with only named sheet |
| **Disabled** | Do nothing | Do nothing | Do nothing |

*Overwriting a named sheet is slower than creating a new file and can cause the file size to increase each time (due to string indexing by Excel). If this becomes a problem, you can delete the file and re-output it. Because of this you should use **Overwrite / File** if the file will only have 1 sheet.

| **Write mode** for non-Excel files | **File exists** | **File does not exist** |
|---|---|---|
| **Overwrite** | Overwrite file | Create file |
| **Append** | Append to file | Create file |
| **New** | Do nothing | Create file |
| **Disabled** | Do nothing | Do nothing |

Check **write header** to write the header (only available for Excel, Delimited text, HTML and Markdown files). If **Write mode** is **Append** you can check **if empty** to only write the header if the file (sheet for Excel files) is currently empty or does not exist (and not for subsequent appends).

Set **File comment** to any comment you wish to output to the file (only available for HTML, Markdown, XML and YAML files). This can include file name variables such as `{date}`. Leave it empty to output no comment.

Use **Alias** to identify the file for batch processing.


© 2023 Oryx Digital Ltd

## 2.5 File formats

### 2.5.1 File formats

Easy Data Transform supports the following data file formats:

| Format | Input | Output |
|---|---|---|
| Delimited text (including CSV and TSV) | Yes | Yes |
| Excel XLSX/XLS | Yes | Yes |
| Fixed width | Yes | No |
| JSON | Yes | Yes |
| HTML | No | Yes |
| Markdown | No | Yes |
| Plain text | Yes | Yes |
| vCard | Yes | Yes |
| XML | Yes | Yes |
| YAML | No | Yes |

You can manage the default file type for different file extensions in the **Input Extensions** and **Output Extensions** tabs of the Preferences window.

### 2.5.2 CSV format

Easy Data Transform can input from and output to CSV format files. Default file extension ".csv".

CSV (Comma Separated Value) format is commonly used for exchanging tabular data between programs.

CSV is a type of delimited text file format. The column delimiter is usually a comma, but not always. The row delimiter is line feed, carriage return or carriage return+line feed.

Easy Data Transform supports the following column delimiters:
- Comma (,)
- Semi-colon (;)
- Colon (:)


© 2023 Oryx Digital Ltd

- Pipe (|)
- Caret (^)
- Tab (\t) (used for TSV format)
- Space ( )
- Custom (choose your own)

For files padded to fixed column widths with space see Fixed width format. For file with no column delimiters see Plain text format.

To output to a CSV file you should typically set **Delimiter** to **Comma (,)** and **Quoting** to **As needed** in the **Right** pane.

For example:

| | CategoryID | CategoryName | Description | In stock |
|---|---|---|---|---|
| 1 | 1 | Beverages | Soft drinks, coffees & teas | true |
| 2 | 2 | Condiments | Sweet and savory sauces | false |
| 3 | 3 | Confections | Candies and sweet breads | true |

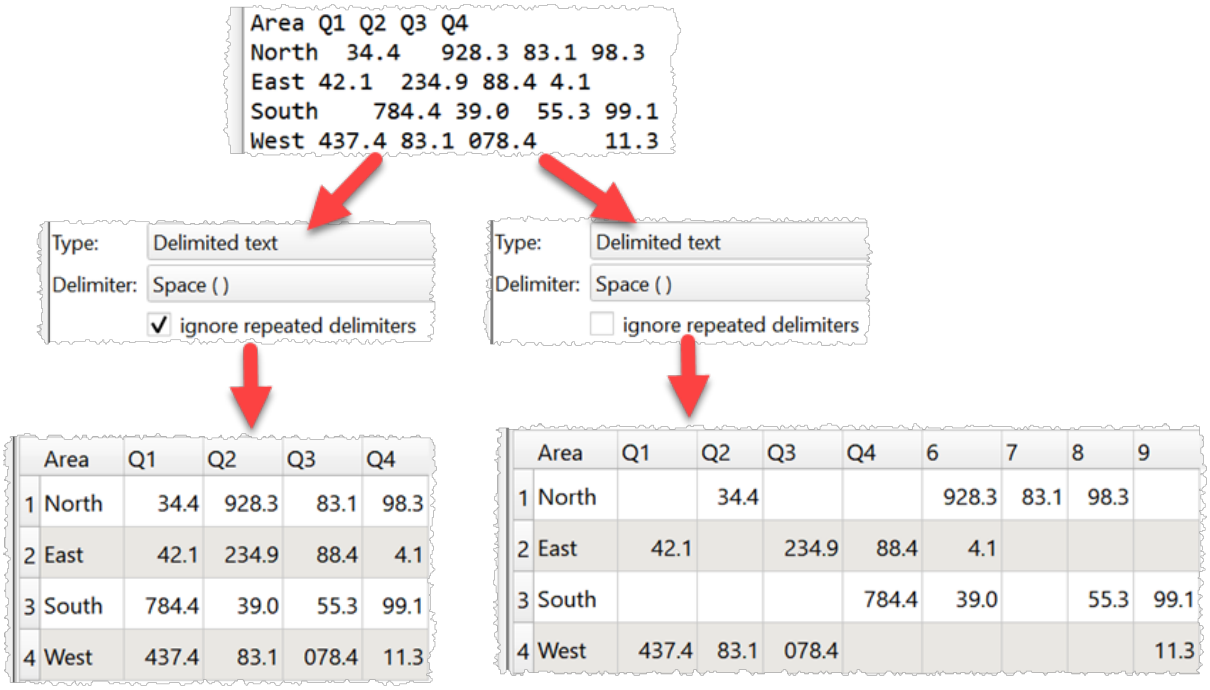
Is output with **Delimiter** set to **Comma (,)** and **Quoting** set to **As needed** as:

```
CategoryID,CategoryName,Description,In stock
1,Beverages,"Soft drinks, coffees & teas",true
2,Condiments,Sweet and savory sauces,false
3,Confections,Candies and sweet breads,true
```

Note that " quotes have been used to enclose values that contain delimiter characters.

Check **ignore repeated delimiters** if you want to treat 2 or more consecutive delimiters as a single delimiter. For example, this might be useful if values are separated by a variable number of spaces:


*© 2023 Oryx Digital Ltd*

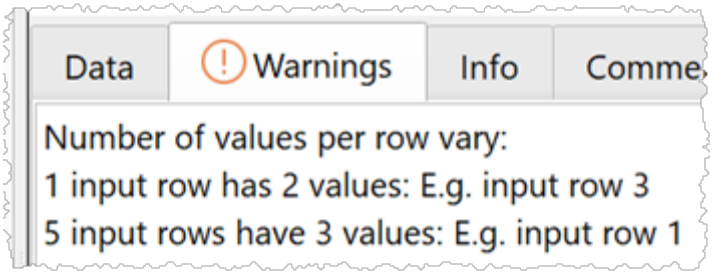



| | Area | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|---|
| 1 | North | 34.4 | 928.3 | 83.1 | 98.3 |
| 2 | East | 42.1 | 234.9 | 88.4 | 4.1 |
| 3 | South | 784.4 | 39.0 | 55.3 | 99.1 |
| 4 | West | 437.4 | 83.1 | 078.4 | 11.3 |

| | Area | Q1 | Q2 | Q3 | Q4 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | North | | 34.4 | | | 928.3 | 83.1 | 98.3 | |
| 2 | East | 42.1 | | 234.9 | 88.4 | 4.1 | | | |
| 3 | South | | | | 784.4 | 39.0 | | 55.3 | 99.1 |
| 4 | West | 437.4 | 83.1 | 078.4 | | | | | 11.3 |

Set **Quoting** to **Always** if you want to quote every value. Set **Quoting** to **Never** if you don't want to quote values (delimiters and carriage returns within values will be changed to spaces).

You will be warned when inputting a CSV format file that has a different number of values per row. This may be a sign that there are problems with the data (e.g. incorrect quoting).




If you want to input a file with a different delimiter to those listed, then:

1. Input it with a delimiter unlikely to be in the text, e.g. caret (^), to input it as a single column
2. Use the Split Col transform to transform it into multiple columns using the appropriate delimiter.

Many CSV file are not well formed. For example, they have unescaped quotes. As the CSV format is not well-defined, badly formed CSV files can be interpreted in more than one way. Easy Data Transform will do the best it can in these circumstances.

See also:


*© 2023 Oryx Digital Ltd*

- Video: How to convert CSV to JSON
- Video: How to convert CSV to Markdown
- Video: How to convert CSV to tab delimited
- Video: How to convert CSV to vCard
- Video: How to convert CSV to XML
- Video: How to convert Excel to pipe delimited
- Video: How to convert fixed column width to CSV or Excel
- Video: How to convert JSON to CSV
- Video: How to convert tab delimited to CSV
- Video: How to convert XML to CSV
- Video: How to split CSV into multiple files
- TSV format

### 2.5.3 Excel format

Easy Data Transform can input from and output to Excel ".xlsx" and ".xls" format files, even if you don't have Excel installed.

Excel format is the native format of the Microsoft Excel spreadsheet application. It is commonly used for exchanging tabular data.

You can specify the sheet name when inputting or outputting Excel files using square brackets, for example:
- `MySpreadsheet.xlsx[Sheet1]` means the sheet named `Sheet1` of file `MySpreadsheet.xlsx` (the sheet name is not case sensitive).
- `MySpreadsheet.xlsx[1]` means sheet named `1` of file `MySpreadsheet.xlsx` (if it exists) or else the first sheet.
- `MySpreadsheet.xlsx` means the first sheet of file `MySpreadsheet.xlsx`.

Hyperlinks are read in from Excel in the form `anchor|hyperlink`.

You can specify how to format columns when outputting to Excel using the **Formatting** option.

Please note the following limitations of Excel:
- Excel .xlsx files are limited to 1,048,576 rows and 16,384 columns.
- Excel .xls files are limited to 65,536 rows and 256 columns.
- Date values are input using ISO date format YYYY-MM-DD, regardless of your locale.
- Datetime values are input using ISO date format YYYY-MM-DDTHH:MM:SS, regardless of your locale.
- The following characters are not allowed in sheet names: `\ / * [ ] : ?`
- Writing an Excel file is significantly slower than writing a CSV file for the same dataset.


© 2023 Oryx Digital Ltd

See also:

- Video: How to combine columns in Excel
- Video: How to convert Excel to JSON
- Video: How to convert Excel to pipe delimited
- Video: How to convert Excel to XML
- Video: How to convert Excel to YAML
- Video: How to convert fixed column width to CSV or Excel
- Video: How to join Excel files
- Video: How to convert JSON to Excel
- Video: How to convert XML to Excel
- Video: How to remove Excel duplicates
- Video: How to split a column in Excel

### 2.5.4 Fixed width format

Easy Data Transform can input from fixed width format files, also known as fixed column width format. Default file extension ".txt".

Fixed width format is used for exchanging tabular data between programs. It is often associated with legacy systems, but is also used for large files where performance is an issue (e.g. bioinformatics).

In fixed width format each column has a fixed width in characters. There is no column delimiter. Spaces are typically used as padding to make up the column width. The row delimiter is line feed, carriage return or carriage return+line feed. For example:

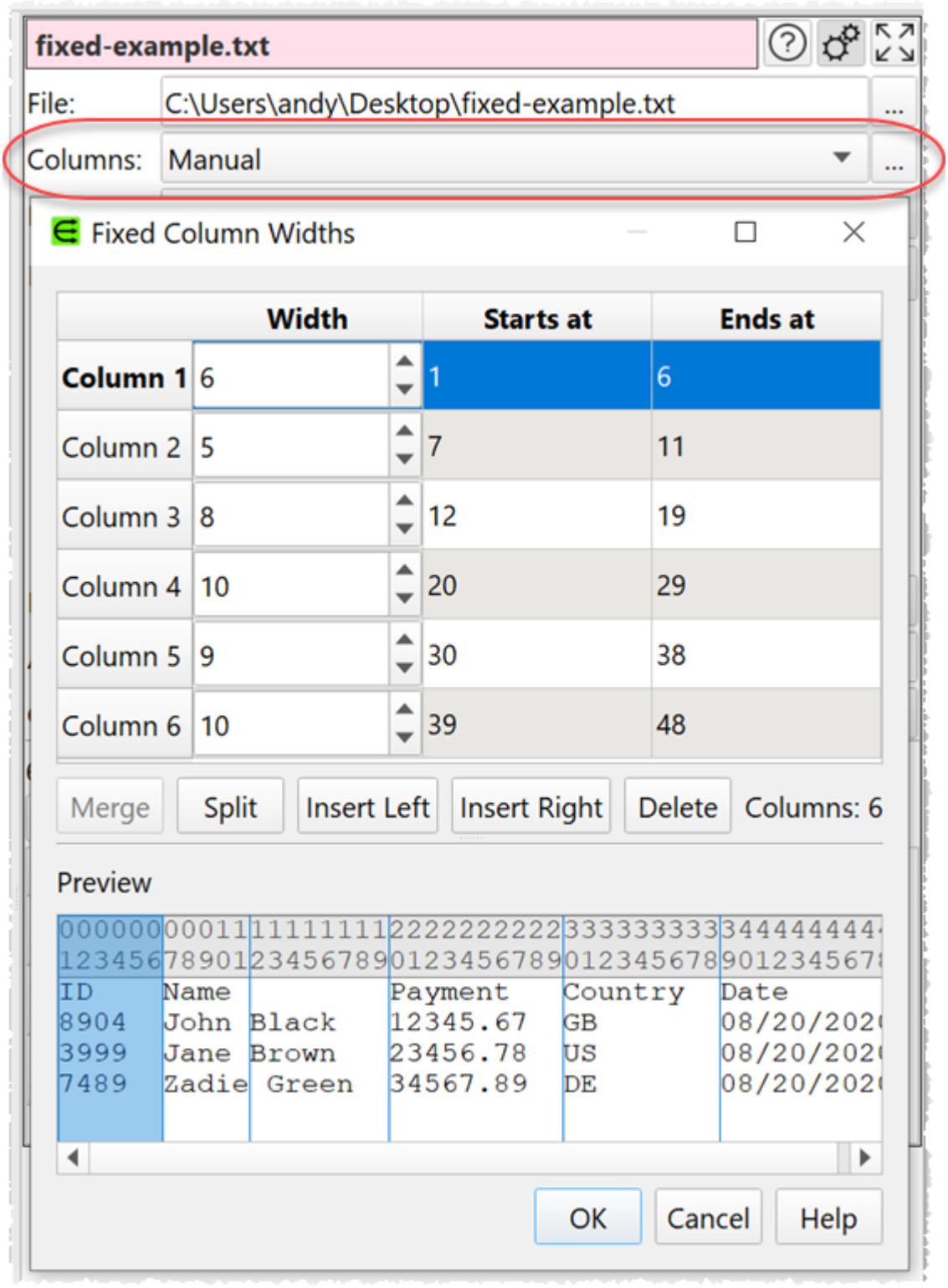
```
ID     Name          Payment    Country   Date
8904   John Black    12345.67   GB        08/20/2020
3999   Jane Brown    23456.78   US        08/20/2020
7489   Zadie Green   34567.89   DE        08/20/2020
```

Is input as:

| | ID | Name | Payment | Country | Date |
|---|---|---|---|---|---|
| 1 | 8904 | John Black | 12345.67 | GB | 08/20/2020 |
| 2 | 3999 | Jane Brown | 23456.78 | US | 08/20/2020 |
| 3 | 7489 | Zadie Green | 34567.89 | DE | 08/20/2020 |


© 2023 Oryx Digital Ltd

Easy Data Transform will analyze the data and guess the column layout if you set **Columns** in the **Right** pane to **Automatic**. Or you can choose the column widths by setting **Columns** to **Manual**. Click the '...' button to edit the manual column widths.





*© 2023 Oryx Digital Ltd*

The current column boundaries are shown on the first few rows in the **Preview**. The horizontal offset of each character is shown in gray at the top. The currently selected columns are highlighted. Click on a column in the **Preview** to select it in the table, or vice versa.

You can change the column widths using the **Width** column of the table.

Select 2 or more adjacent columns and click **Merge** to merge them into 1 column. Click then Shift+click in either the table or the preview to select multiple adjacent columns.

Select 1 column with a **Width** > 1 and click **Split** to split into 2 columns.

Select 1 or more adjacent columns and click **Insert Left** or **Insert Right** to add a new column with width 1 to the left or right of the selected columns.

Select 1 or more adjacent columns and click **Delete** delete the selected columns.

Click **OK** to save your changes and **Cancel** to discard them.

Unwanted columns and rows in the dataset can be removed after input using the Remove Cols and Filter transforms.

See also:
- Video: How to convert fixed column width to CSV or Excel

### 2.5.5 JSON format

Easy Data Transform can input from and output to JSON format files. Default file extension ".json".

JSON (Javascript Object Notation) format is commonly used for exchanging data between programs. JSON data is generally expected to be in UTF8 encoding without a Byte Order Mark (BOM).

For example:

| | CategoryID | CategoryName | Description | In stock |
|---|---|---|---|---|
| 1 | 1 | Beverages | Soft drinks, coffees & teas | true |
| 2 | 2 | Condiments | Sweet and savory sauces | false |
| 3 | 3 | Confections | Candies and sweet breads | true |


© 2023 Oryx Digital Ltd

Is equivalent to:

```
[
  {
    "CategoryID": 1,
    "CategoryName": "Beverages",
    "Description": "Soft drinks, coffees & teas",
    "In stock": true
  },
  {
    "CategoryID": 2,
    "CategoryName": "Condiments",
    "Description": "Sweet and savory sauces",
    "In stock": false
  },
  {
    "CategoryID": 3,
    "CategoryName": "Confections",
    "Description": "Candies and sweet breads",
    "In stock": true
  }
]
```

The dot ('.') character is used in the column header to show nesting. For example:

| | name | carb | cholesterol | fiber | minerals.ca | minerals.fe | protein | sodium | vitamins.a | vitamins.c |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Avocado Dip | 2 | 5 | 0 | 0 | 0 | 1 | 210 | 0 | 0 |

Is equivalent to:

```
[
  {
    "name": "Avocado Dip",
    "carb": 2,
    "cholesterol": 5,
    "fiber": 0,
    "minerals": {
      "ca": 0,
      "fe": 0
    },
    "protein": 1,
    "sodium": 210,
    "vitamins": {
      "a": 0,
      "c": 0
    }
  }
]
```

Any dots in JSON names are converted to hyphens ('-') on input.


© 2023 Oryx Digital Ltd

JSON arrays can be input in either long or wide **Format**. For example:

```
[
   {
      "name": "1",
      "values": [ "a", "b" ]
   },
   {
      "name": "2",
      "values": [ "c", "d" ]
   }
]
```

Input as **Long (more rows):**

| | name | values |
|---|---|---|
| 1 | 1 | a |
| 2 | 1 | b |
| 3 | 2 | c |
| 4 | 2 | d |

Input as **Wide (more columns)**:

| | name | values.0 | values.1 |
|---|---|---|---|
| 1 | 1 | a | b |
| 2 | 2 | c | d |

If children of the top level object are named, then they are flattened slightly different to avoid creating thousands or millions of columns. For example:

```
{
      "record 1":{
            "value": 1
      },
      "record 2":{
            "value": 2
      },
      "record 3":{
            "value": 3
      }
```


© 2023 Oryx Digital Ltd

```
}
```

Is flattened into:

| | _ObjectName_ | value |
|---|---|---|
| 1 | record 1 | 1 |
| 2 | record 2 | 2 |
| 3 | record 3 | 3 |

An input JSON format file is expected to have either:
- a single top-level array or object; or
- a single JSON object on each line (see jsonlines.org).
- a record separator character followed by a single JSON object on each line (see RFC7464).

You can specify the JSON type for each columns when outputting to JSON using the **Value types** option. Any values that do not match the type selected are output as a JSON null value.

See also:
- Video: How to convert CSV to JSON
- Video: How to convert Excel to JSON
- Video: How to convert JSON to CSV
- Video: How to convert JSON to Excel

### 2.5.6 HTML format

Easy Data Transform can output to tables in HTML format files. Default file extension ".html".

HTML (HyperText Markup Language) format is commonly used for creating web pages. If you don't need the data to take up a whole page, you can just copy the `<table>` to `</table>` part of the output.

For example:

| | CategoryID | CategoryName | Description | In stock |
|---|---|---|---|---|
| 1 | 1 | Beverages | Soft drinks, coffees & teas | true |
| 2 | 2 | Condiments | Sweet and savory sauces | false |
| 3 | 3 | Confections | Candies and sweet breads | true |


© 2023 Oryx Digital Ltd

Is output as:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8"/>
    <title>C:\Users\andyb\Desktop\output.html</title>
    <style>table,td,th{border:1px solid black;text-
align:left;vertical-align:top;border-spacing:0px;border-
color:gray;font-family:Verdana,sans-serif;}th{background-
color:#E0E0E0;}td,th{padding:5px;}</style>
  </head>
  <body>
    <table>
      <tbody>
        <tr>
          <th>CategoryID</th>
          <th>CategoryName</th>
          <th>Description</th>
          <th>In stock</th>
        </tr>
        <tr>
          <td>1</td>
          <td>Beverages</td>
          <td>Soft drinks, coffees &amp;amp; teas</td>
          <td>true</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Condiments</td>
          <td>Sweet and savory sauces</td>
          <td>false</td>
        </tr>
        <tr>
          <td>3</td>
          <td>Confections</td>
          <td>Candies and sweet breads</td>
          <td>true</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

Usually you will want to 'escape' special HTML characters in dataset values when output to HTML format. For example to convert < to `<`, so it can be correctly displayed in the HTML. However, if you wish to write 'raw' HTML, then you can uncheck **Escape special characters** in the **Right** pane. Note that you can also set **Change** to **To escaped HTML** to escape selected columns in the Decode transform.


© 2023 Oryx Digital Ltd

### 2.5.7 Markdown format

Easy Data Transform can output to tables in Markdown format files. Default file extension ".md".

Markdown format is commonly used as a human-friendly markup language, which can be automatically translated to HTML.

For example:

| | CategoryID | CategoryName | Description | In stock |
|---|---|---|---|---|
| 1 | 1 | Beverages | Soft drinks, coffees & teas | true |
| 2 | 2 | Condiments | Sweet and savory sauces | false |
| 3 | 3 | Confections | Candies and sweet breads | true |

Is output as:

```
| CategoryID | CategoryName | Description                 | In stock
|
|------------|--------------|-----------------------------|----------
|
| 1          | Beverages    | Soft drinks, coffees & teas | true
|
| 2          | Condiments   | Sweet and savory sauces     | false
|
| 3          | Confections  | Candies and sweet breads    | true
|
```

You can also use Markdown when you need a plain text version of your data, for example in a code comment.

Note that not all Markdown implementations support tables. If your implementation does not support tables, you may need to output to HTML instead.

See also:
- Video: How to convert CSV to Markdown

### 2.5.8 Plain text format

Easy Data Transform can input from and output to plain text format files. Default file extension is ".log" for input and ".log" or ".txt" for output.

Plain text format is used for semi-structured data, such as computer generated log files.

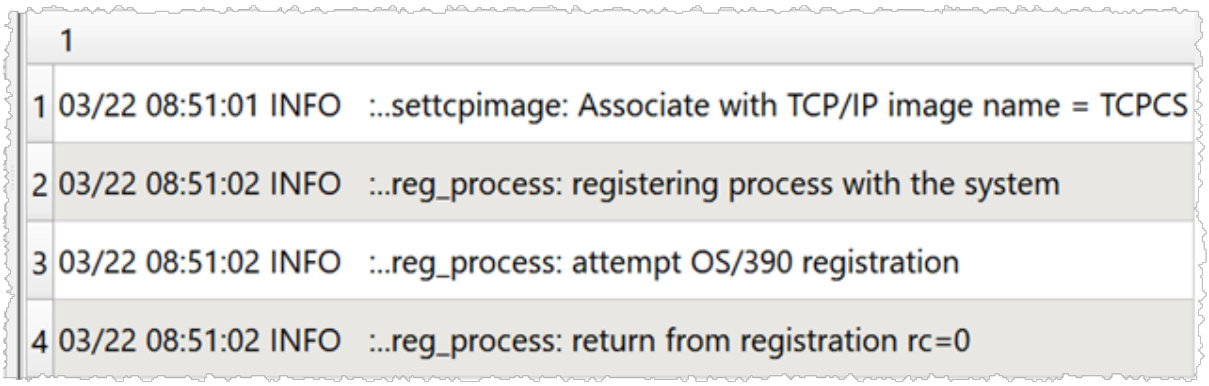

© 2023 Oryx Digital Ltd

When inputting plain text the dataset is read into a single column. The row delimiter is line feed, carriage return or carriage return+line feed.

For example:

```
03/22 08:51:01 INFO   :..settcpimage: Associate with TCP/IP image name = TCPCS
03/22 08:51:02 INFO   :..reg_process: registering process with the system
03/22 08:51:02 INFO   :..reg_process: attempt OS/390 registration
03/22 08:51:02 INFO   :..reg_process: return from registration rc=0
```

Is input in plain text format as:



| | 1 |
|---|---|
| 1 | 03/22 08:51:01 INFO :..settcpimage: Associate with TCP/IP image name = TCPCS |
| 2 | 03/22 08:51:02 INFO :..reg_process: registering process with the system |
| 3 | 03/22 08:51:02 INFO :..reg_process: attempt OS/390 registration |
| 4 | 03/22 08:51:02 INFO :..reg_process: return from registration rc=0 |

When outputting plain text:

- Each row ends with a line feed, carriage return or carriage return+line feed, as selected by the user.
- Typically plain text datasets only have 1 column. But if there is more than one column, the column values will be concatenated without a delimiter. If you want a delimiter use Concat Cols and Remove Cols to create a single column that contains the appropriate delimiters, or output as delimited text.
- Values are not escaped.

### 2.5.9 TSV format

Easy Data Transform can input from and output to TSV format files. Default file extension ".tsv".

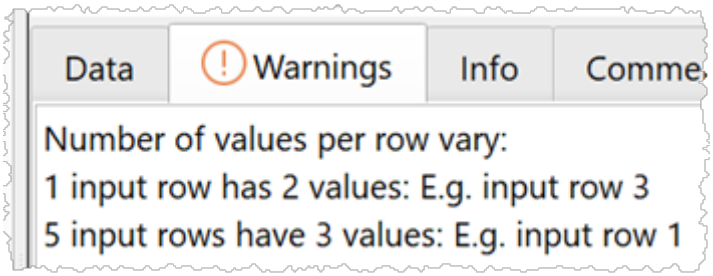
TSV (Tab Separated Value) format is commonly used for exchanging tabular data between programs.

TSV is a type of delimited text file format where values are separated by tab characters and rows by line feed, carriage return or carriage return+line feed. Tabs and carriage return are not allowed within data values, so there is no need for quoting or escaping delimiters, as with CSV files. This means that TSV files are generally a bit more compact and faster to read and write than CSV files.


© 2023 Oryx Digital Ltd

To input a TSV file you should set **Type** to **Delimited text** and **Delimiter** to **Tab (\t)** in the **Right** pane. You will usually also want to uncheck **ignore repeated delimiters** and set **Quoting** to **Automatic** (which is the same as **Unquoted** for tab delimiters).

You will be warned when inputting a TSV format file that has a different number of values per row. This may be a sign that there are problems with the data.




To output a TSV file you should set **Delimiter** to **Tab (\t)**. You will usually also want to set **Quoting** to **Never** in the **Right** pane. If you have a tab character or carriage return within a value, then Easy Data Transform will then convert it to a space on output.

For example:

| | CategoryID | CategoryName | Description | In stock |
|---|---|---|---|---|
| 1 | 1 | Beverages | Soft drinks, coffees & teas | true |
| 2 | 2 | Condiments | Sweet and savory sauces | false |
| 3 | 3 | Confections | Candies and sweet breads | true |

Is output with **Delimiter** set to **Tab (\t)** and **Quoting** set to **Never** as:

```
CategoryID→CategoryName→Description→In stock
1→Beverages→Soft drinks, coffees & teas→true
2→Condiments→Sweet and savory sauces→false
3→Confections→Candies and sweet breads→true
```

See also:

- Video: How to convert tab delimited to CSV
- Video: How to convert CSV to tab delimited


© 2023 Oryx Digital Ltd

### 2.5.10 vCard format

Easy Data Transform can input from and output to vCard format files. Default file extension ".vcf".

VCard format is commonly used as way of exchanging contact details between programs.

Note that you need to change the column header names to the values expected by vCard (using the Rename Cols transform). For example vCard expects the full name column to be named `FN`.

Choose an appropriate text encoding. You should probably uncheck **Byte Order Mark** if you are encoding in a UTF format, such as **UTF-8**.

For example:

| | N | FN | ORG | TEL;TYPE=WORK,VOICE | ADR;TYPE=WORK,PREF |
|---|---|---|---|---|---|
| 1 | Gump;Forrest;;Mr.; | Forrest Gump | Bubba Gump Shrimp Co. | (111) 555-1212 | 100 Waters Edge;Baytown; |

Is equivalent to:

```
BEGIN:VCARD
VERSION:3.0
N:Gump;Forrest;;Mr.;
FN:Forrest Gump
ORG:Bubba Gump Shrimp Co.
TEL;TYPE=WORK,VOICE:(111) 555-1212
ADR;TYPE=WORK,PREF:100 Waters Edge;Baytown;LA;30314;United States of
America
END:VCARD
```

See also:

- Video: How to convert CSV to vCard

### 2.5.11 XML format

Easy Data Transform can input from and output to XML format files. Default file extension ".xml".

XML (Extensible Markup Language) format is commonly used for exchanging data between programs.

For example:


© 2023 Oryx Digital Ltd

| | CategoryID | CategoryName | Description | In stock |
|---|---|---|---|---|
| 1 | 1 | Beverages | Soft drinks, coffees & teas | true |
| 2 | 2 | Condiments | Sweet and savory sauces | false |
| 3 | 3 | Confections | Candies and sweet breads | true |

Is equivalent to:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <record>
    <CategoryID>1</CategoryID>
    <CategoryName>Beverages</CategoryName>
    <Description>Soft drinks, coffees &amp;amp; teas</Description>
    <In-stock>true</In-stock>
  </record>
  <record>
    <CategoryID>2</CategoryID>
    <CategoryName>Condiments</CategoryName>
    <Description>Sweet and savory sauces</Description>
    <In-stock>false</In-stock>
  </record>
  <record>
    <CategoryID>3</CategoryID>
    <CategoryName>Confections</CategoryName>
    <Description>Candies and sweet breads</Description>
    <In-stock>true</In-stock>
  </record>
</root>
```

The dot ('.') character is used in the column header to show nesting. For example:

| | name | carb | cholesterol | fiber | minerals.ca | minerals.fe | protein | sodium | vitamins.a | vitamins.c |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Avocado Dip | 2 | 5 | 0 | 0 | 0 | 1 | 210 | 0 | 0 |

Is equivalent to:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <record>
    <name>Avocado Dip</name>
    <carb>2</carb>
    <cholesterol>5</cholesterol>
    <fiber>0</fiber>
    <protein>1</protein>
    <sodium>210</sodium>
    <minerals>
```


© 2023 Oryx Digital Ltd

```
      <ca>0</ca>
      <fe>0</fe>
    </minerals>
    <vitamins>
      <a>0</a>
      <c>0</c>
    </vitamins>
  </record>
</root>
```

Note that the columns may be input in a different order to the nodes in the XML. You can use Reorder Cols or Stack transforms to change the ordering. But note that during XML output nodes with no children are always ordered before nodes with children, regardless of the column order.

Any dots in XML element names are converted to hyphens ('-') on input.

The underscore ('_') character is used at the start of a column header name to identify it as an XML attribute. For example:

| | _name | _carb | _cholesterol | _fiber | minerals.ca | minerals.fe | _protein | _sodium | vitamins.a | vitamins.c |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Avocado Dip | 2 | 5 | 0 | 0 | 0 | 1 | 210 | 0 | 0 |

Is equivalent to:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <record carb="2" cholesterol="5" fiber="0" name="Avocado Dip"
protein="1" sodium="210">
    <minerals>
      <ca>0</ca>
      <fe>0</fe>
    </minerals>
    <vitamins>
      <a>0</a>
      <c>0</c>
    </vitamins>
  </record>
</root>
```

Repeated XML values can be input in either long or wide **Format**. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<ITEMS>
  <ITEM>
    <PARAM name="a" value="1"/>
    <PARAM name="b" value="2"/>
  </ITEM>
</ITEMS>
```


© 2023 Oryx Digital Ltd

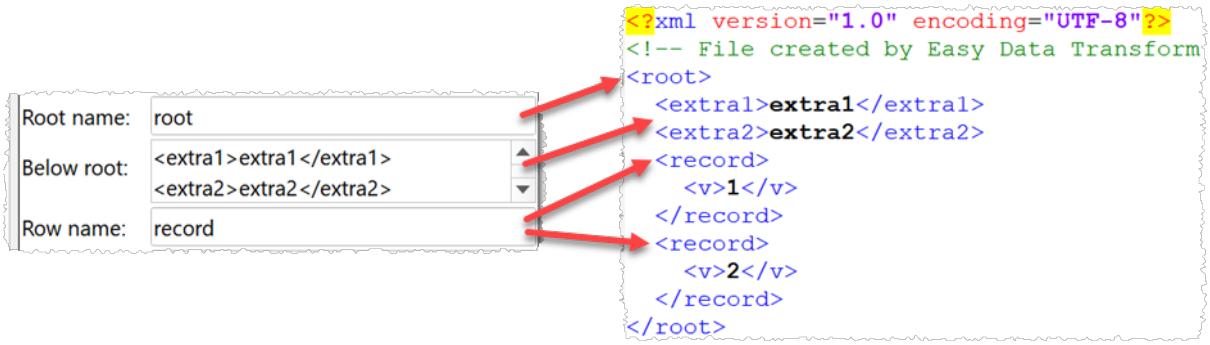
Input as **Long (more rows):**

| | PARAM._name | PARAM._value |
|---|---|---|
| 1 | b | 2 |
| 2 | a | 1 |

Input as **Wide (more columns)**:

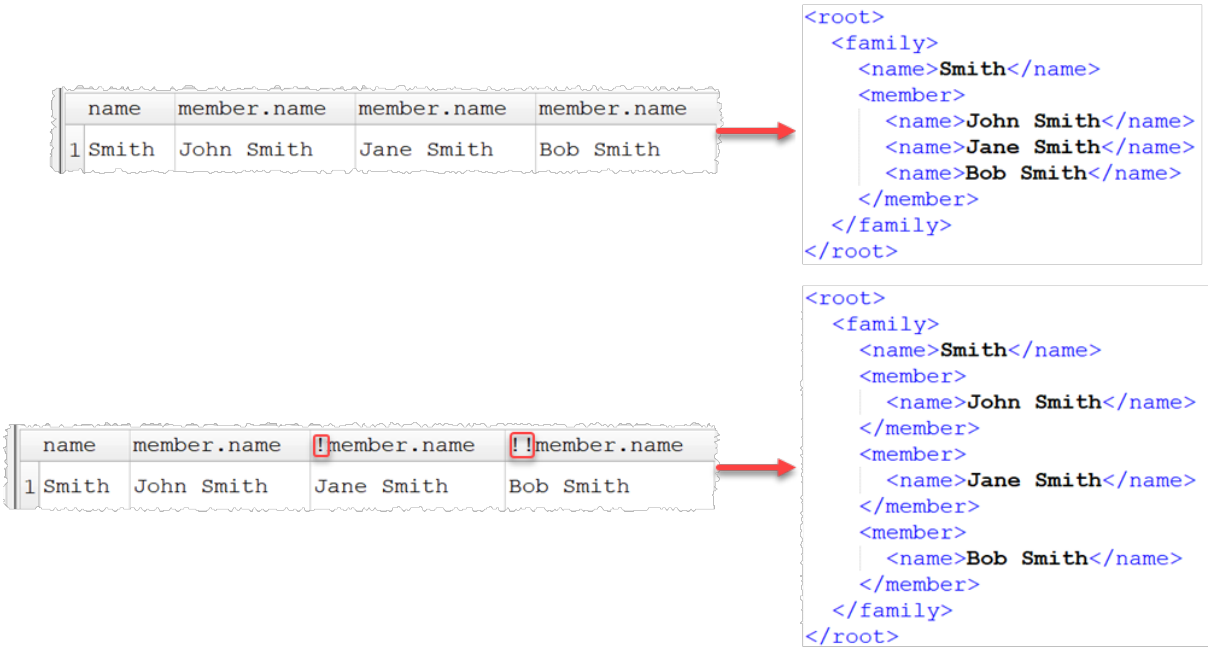
| | PARAM._value | PARAM._name | PARAM._value.1 | PARAM._name.1 |
|---|---|---|---|---|
| 1 | 1 | a | 2 | b |

You can set the names for the root and record tags and add extra XML below the root.




Exclamation marks at the start of column names are removed from name tags when output to XML. This can be useful when you want duplicate tag names with the same parent.

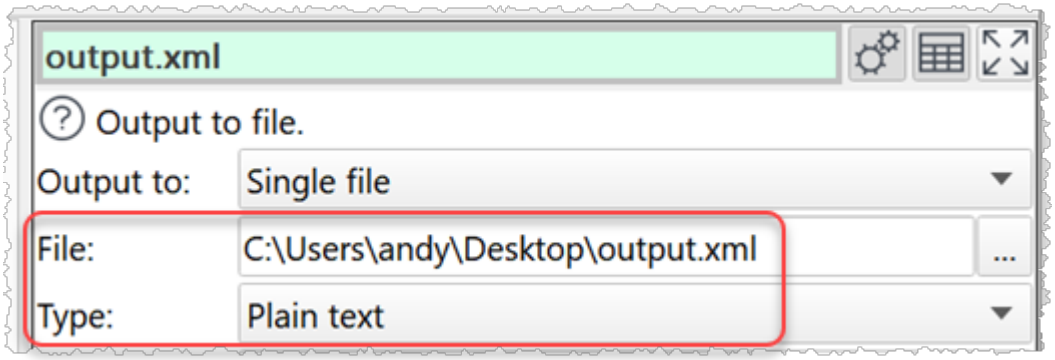

© 2023 Oryx Digital Ltd




You are responsible for ensuring that the names of XML nodes and attributes are valid (e.g. start with a letter or underscore and do not contain spaces).

If you need really fine grained control over the XML output, you can:

- output the XML to a file
- read the file back in as plain text
- apply transforms
- write it back out as plain text




See also:

- Video: How to convert CSV to XML
- Video: How to convert Excel to XML
- Video: How to convert XML to CSV
- Video: How to convert XML to Excel


© 2023 Oryx Digital Ltd

### 2.5.12 YAML format

Easy Data Transform can output to YAML format files. Default file extension ".yaml".

YAML (YAML Ain't Markup Language) format is commonly used for exchanging data between programs and for configuration files.

For example:

| | CategoryID | CategoryName | Description | In stock |
|---|---|---|---|---|
| 1 | 1 | Beverages | Soft drinks, coffees & teas | true |
| 2 | 2 | Condiments | Sweet and savory sauces | false |
| 3 | 3 | Confections | Candies and sweet breads | true |

Is output as:

```
---
-
  CategoryID: 1
  CategoryName: Beverages
  Description: Soft drinks, coffees & teas
  In stock: true
-
  CategoryID: 2
  CategoryName: Condiments
  Description: Sweet and savory sauces
  In stock: false
-
  CategoryID: 3
  CategoryName: Confections
  Description: Candies and sweet breads
  In stock: true
```

See also:

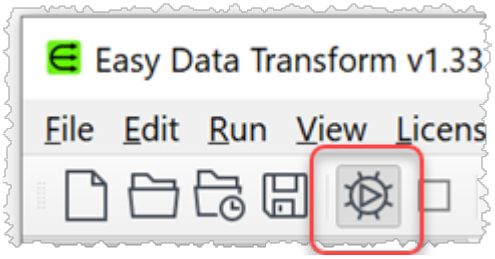
- Video: How to convert Excel to YAML

## 2.6 Processing

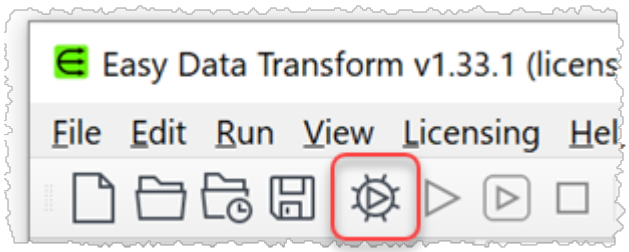
You can run processing of your data in one of two modes:

- Automatic mode: Processing is run automatically as soon as you have stopped making changes for the amount of time specified in the Preferences window.


© 2023 Oryx Digital Ltd




- Manual mode: You control when processing is run.



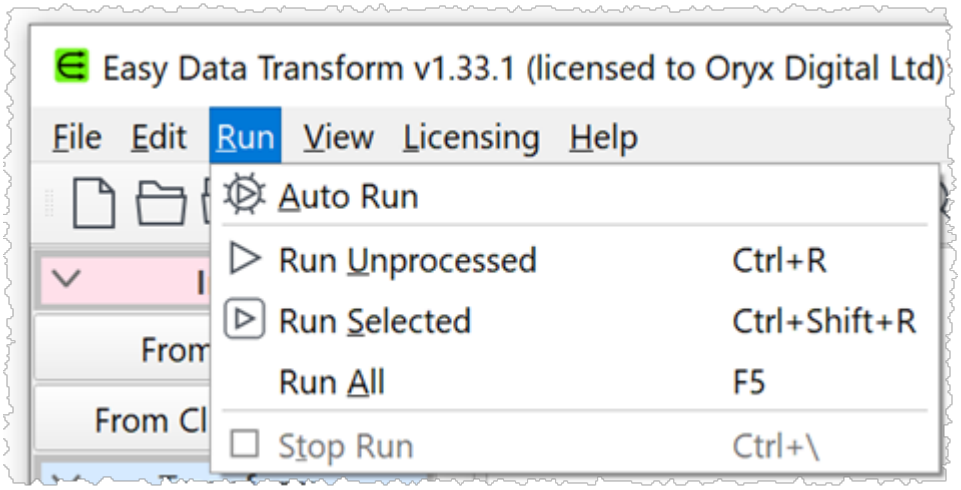

You can switch between modes by toggling **Run>Auto Run**, or the equivalent tool bar button.

Typically automatic mode is better for small to medium datasets, but manual mode may be a better choice for large datasets.

There are 3 options for running processing in manual mode:

- **Run>Run Unprocessed**: Run processing on any items that are not already processed.
- **Run>Run Selected**: Run processing on one or more selected items, even if they are already processed.
- **Run>Run All**: Run processing on all items, even if they are already processed. This is provided for completeness, but you should usually use one of the above 2 options, for efficiency.


*© 2023 Oryx Digital Ltd*




You can also click the **Run** button in the **Right** pane, which is equivalent to **Run>Run Selected**. Note that **Run**, for a clipboard input, applies the options to data previously imported. If you want to re-import from the clipboard in automatic mode, then you have to click the **Import from clipboard** button.

You can stop processing with **Run>Stop Run**, or the equivalent tool bar button. If you stop processing in automatic mode it will switch to manual mode.

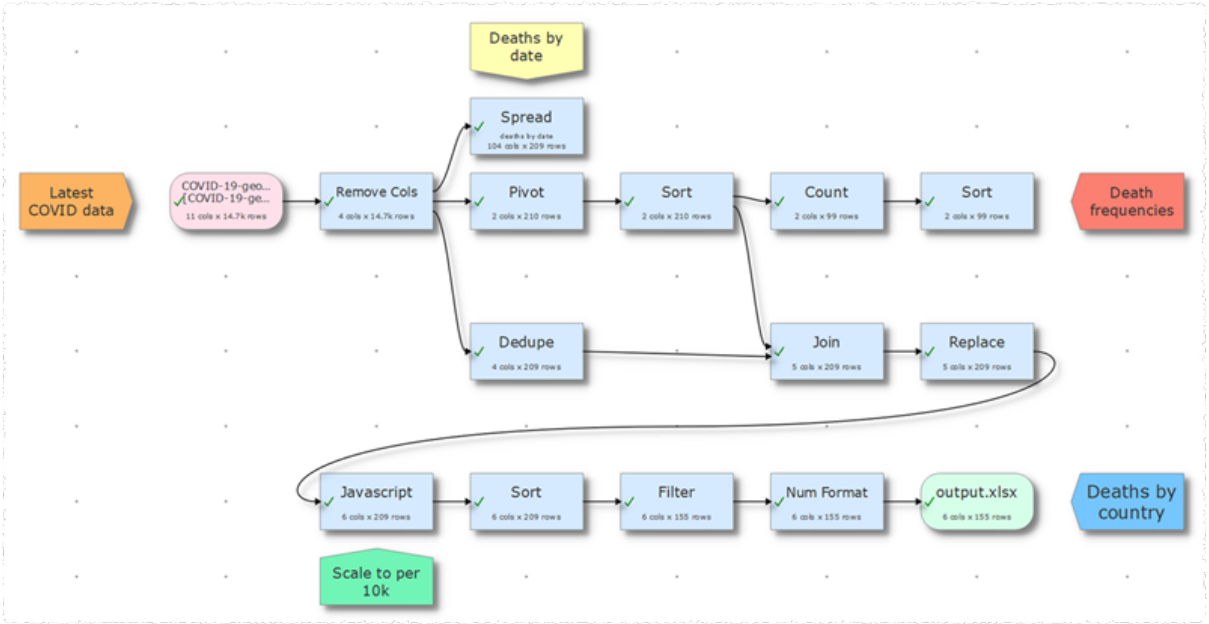
When an item is processed, all it's upstream items will be processed first. You can only change the options for an item in the **Right** pane if all it's upstream inputs and transforms are processed.

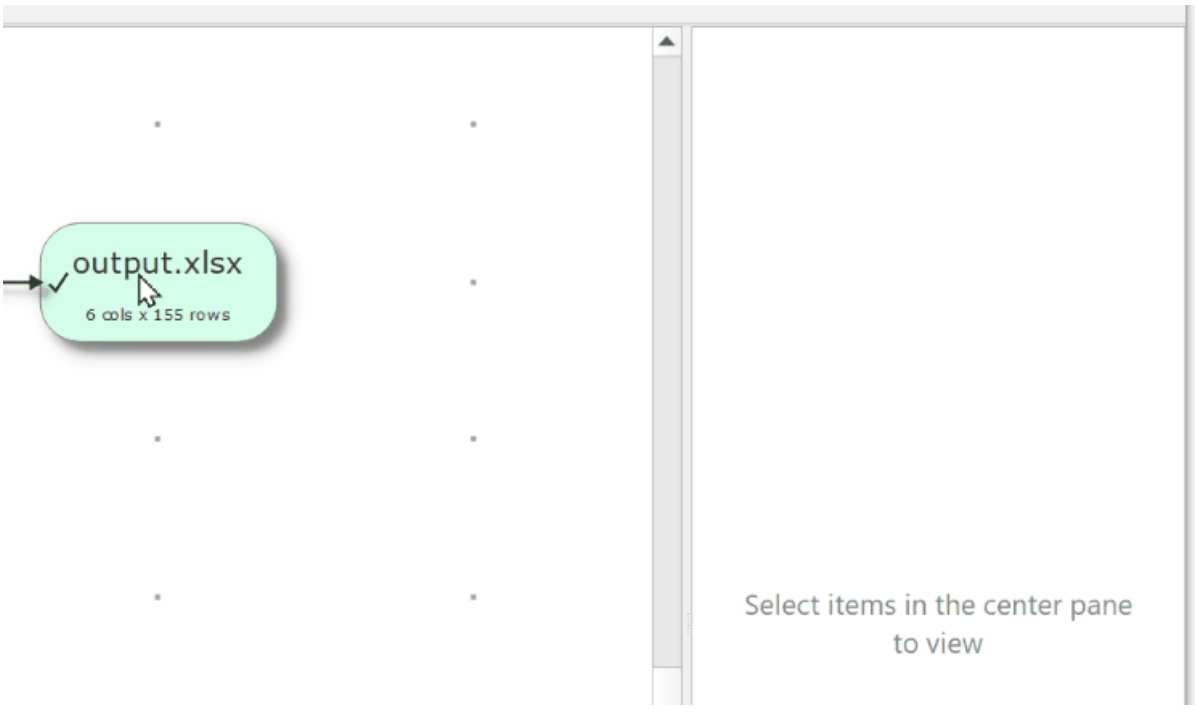
## 2.7 Comments

You can manage the complexity of large .transform files adding comment text to:

- The **Comment** field of Note items. These are added to the center pane by right clicking and selecting **Note** from the menu. You can also set an **Arrow** direction and a **Color**. If you right click on an existing item, the Note item will be placed pointing to that item, if there is space. The comment will be shown as a tooltip when you hover over the Note item.

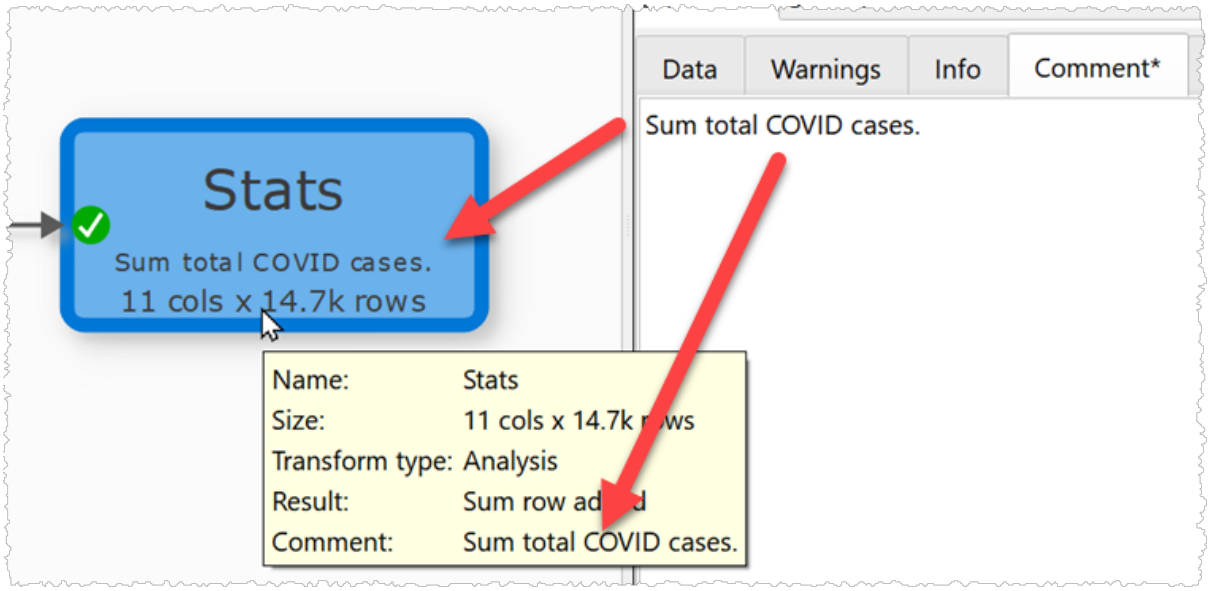

*© 2023 Oryx Digital Ltd*




- The **Comment** tab of Input, Transform or Output items. Check **View>Show Comments** to show these comments in the center pane. The comment will be shown as a tooltip when you hover over the item.


*© 2023 Oryx Digital Ltd*



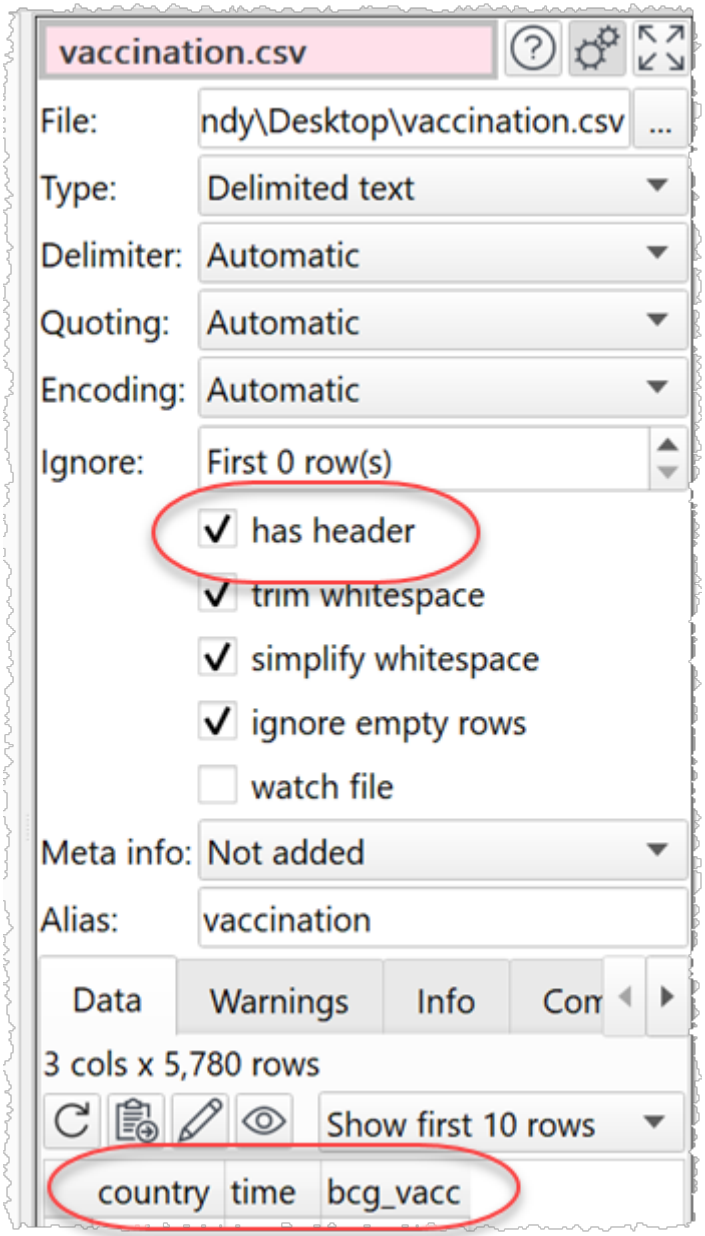

- In the **Notes** window that is displayed when you select **File>Notes...** from the main menu.

You can also group related items in the center pane to show that they are related.

## 2.8 Headers

If the first row of an input is a header (i.e. one that describes the columns below) check **has header** for that input in the **Right** pane.


*© 2023 Oryx Digital Ltd*




When you first read in a dataset Easy Data Transform will make a guess about whether the first row is a header (it will assume it is a header if it contains no dates or numbers).

You can move 1 or more dataset rows to the header using the Header transform.

## 2.9 Connections

When you select an input or transform item and add a transform or output item, connections are added automatically.

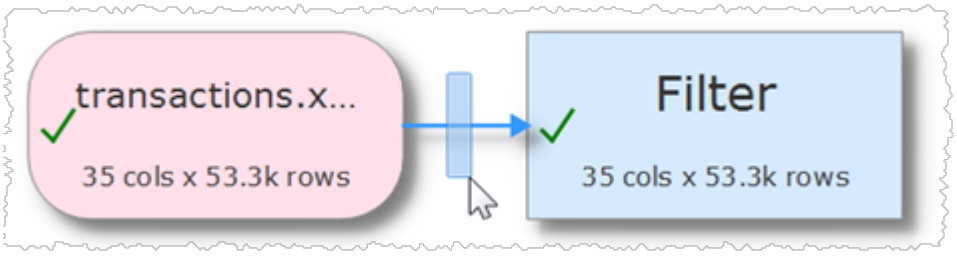
### To select a connection

To select a connection either:

- Click on the connection; or


© 2023 Oryx Digital Ltd

- Click and drag a box over any part of the connection. This may be easier than clicking the connection when you are zoomed back.




## To delete a connection

To delete a connection:

- Select the connection.
- Select **Edit>Delete** (or click the **Delete** tool bar button).

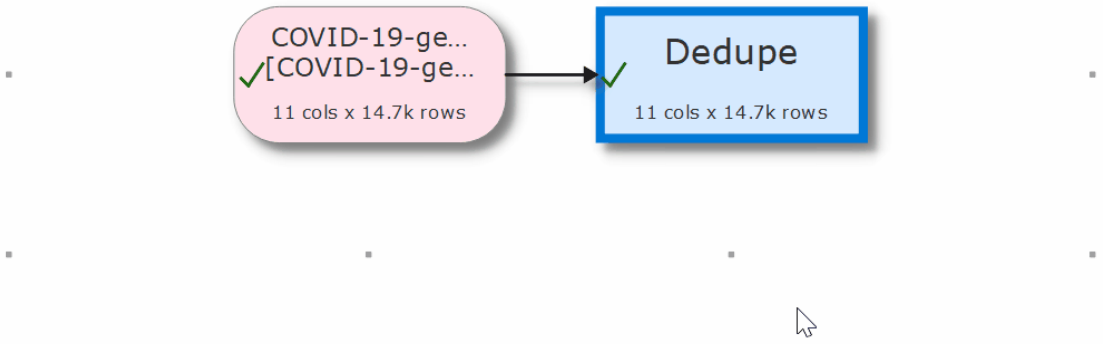
Note that deleting a connection may unset column related parameters downstream, so should generally be avoided where possible.

- If you want to change an input file, do it by selecting the input and clicking on '...' in the **Right** pane, rather than disconnecting the input and connecting a new one.
- If you want to add a new transform between 2 already connected items, you can do it without disconnecting (see below).

## To add a transform to a connection

To add a transform between two already connected items:

- Select the connection.
- Choose the new transform from the **Left** pane or using the right click menu.



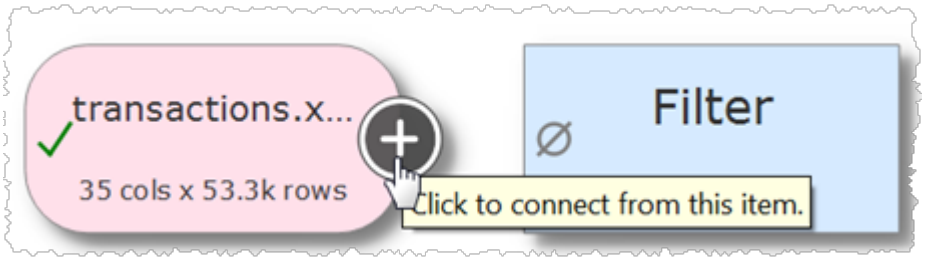

## To add a connection

To add a new connection between two existing items:

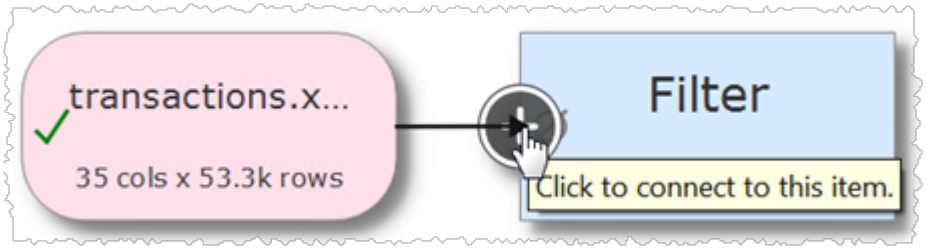
- Hover over the start item.


© 2023 Oryx Digital Ltd

- Click the '+' that appears.




- Hover over the end item
- Click the '+' that appears.




Press the 'Esc' key or click away from an item to cancel adding the connection.

Note that the '+' will only appear if an additional connection is allowed. For example you can't:

- Create a loop.
- Connect more than once from a transform.
- Connect more than once to an output.

## 2.10 Text

Whitespace (such as Space and Tab characters) and case are always significant, unless stated otherwise.

You can remove leading and trailing whitespace by checking **trim whitespace** in the Input or using the Whitespace transform.

You can change the case using the Case transform.

You can concatenate and split columns of text using the Concat Cols and Split Col transforms.

## 2.11 Dates

Set **Supported Date Formats** in the Preferences window according to how you want to interpret values as dates.


© 2023 Oryx Digital Ltd

| Format | Meaning |
|---|---|
| d | The day as number without a leading zero (1 to 31) |
| dd | The day as number with a leading zero (01 to 31) |
| ddd | The abbreviated localized day name (e.g. 'Mon' to 'Sun'). Uses the locale to localize the name. |
| dddd | The long localized day name (e.g. 'Monday' to 'Sunday'). Uses the locale to localize the name. |
| M | The month as number without a leading zero (1 to 12). |
| MM | The month as number with a leading zero (01 to 12) |
| MMM | The abbreviated localized month name (e.g. 'Jan' to 'Dec'). Uses the locale to localize the name. |
| MMMM | The long localized month name (e.g. 'January' to 'December'). Uses the locale to localize the name. |
| yy | The year as two digit number (00 to 99). |
| yyyy | The year as four digit number. If the year is negative, a minus sign is prepended in addition. |

For example:
- To support a date such as 31/1/2019 add a supported date format: d/M/yyyy
- To support a date such as 1-31-19 add a supported date format: M-d-yy

List the date formats in order of preference, with the most likely to be used first.

Dates with only two year digits, are treated as a date between 1900 and 1999. E.g. "31/1/19" is interpreted in d-M-yy format as 31st January 1919.


© 2023 Oryx Digital Ltd

Values that are in a recognized date format will be treated as dates in transforms such as: Calculate, Compare, Filter, If and Sort. Supporting large numbers of date formats will slow down these transforms.

You can change the format of dates using the DateTime Format transform.

You can do date calculations, such as calculating the number of days between 2 dates, using the Calculate or Javascript transform.

You can convert between ISO standard datetimes and milliseconds since 1st Jan 1970 using the Calculate transform.

You can convert an ISO standard datetimes (e.g. 2020-10-16T01:51) into separate date and time columns by using the Split Cols transform on the "T" delimiter.

## 2.12 Numbers

Set the locale according to how you want to interpret values as numbers. For example, if locale is set to US or UK then "123.45" is a number and "123,45" isn't, and vice versa if locale is Germany or France.

An integer is a whole number. E.g. "1.0" or "1,0" (depending on your locale), "0", "-1" and "1e3" are considered integers.

"NaN" means 'not a number'. It can be returned by some calculations, for example using **Calculate** to raise 1000 to the power 1000 will overflow numerically and return "NaN".

You can change the format of numbers using the Num Format transform.

You can do numeric calculations, such as multiplying 2 columns, using the Calculate or Javascript transforms.

You can change the base of integers using the Num Base transform.

## 2.13 Booleans

A boolean value can be `true` or `false`.

| Canonical value | Allowed values | Examples |
|---|---|---|
| `true` | `true` (case insensitive) or a non-zero numeric value | `true`<br>`TRUE`<br>`True`<br>`1` |


*© 2023 Oryx Digital Ltd*

| Canonical value | Allowed values | Examples |
|---|---|---|
| | | `1.0`<br>`-1`<br>`0.1`<br>`999` |
| `false` | `false` (case insensitive) or a zero numeric value | `false`<br>`FALSE`<br>`False`<br>`0`<br>`0.0` |

## 2.14 Locale

You can set the **Locale** in the Preferences window.

The locale language and country setting affects how some numbers and dates are interpreted and displayed. Consequently it may have an affect on some transforms, for example the DateTime format and Num format transforms.

This shows using the DateTime format transform with **Format to** set to `ddd MMMM d yyyy` for different locales:

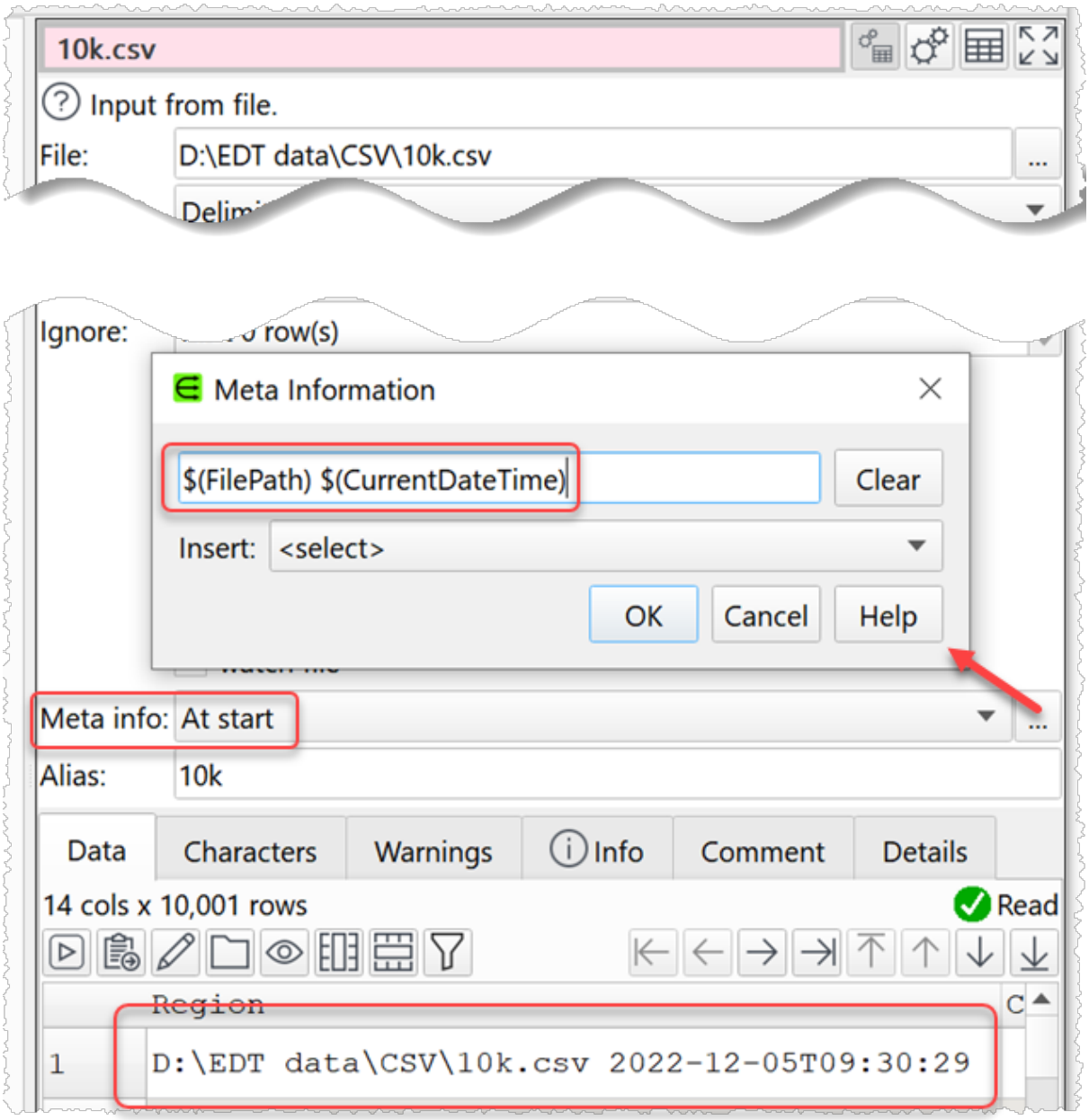
| Locale | Output |
|---|---|
| `English/United States` | `Mon May 21 2001` |
| `English/United Kingdom` | `Mon May 21 2001` |
| `German/Germany` | `Mo. Mai 21 2001` |
| `French/France` | `lun. mai 21 2001` |
| `Spanish/Spain` | `lun. mayo 21 2001` |
| `Italian/Italy` | `lun maggio 21 2001` |
| `Portuguese/Portugal` | `segunda maio 21 2001` |
| `Chinese/China` | `周一 五月 21 2001` |
| `Japanese/Japan` | `月 5月 21 2001` |

You will be warned if you **File>Open** a `.transform` file with a different locale to the one you have set.


© 2023 Oryx Digital Ltd

## 2.15 Meta Information

You can add meta information to input data using the **Meta info** field in the **Right** pane when you select an input item.




Set it to **At start**, **At end** or **Every row**, depending on where you want the meta information to appear. Then click on the ... button to edit which information you wish to show. The following placeholders are substituted by their actual values at the time of input.


*© 2023 Oryx Digital Ltd*

| Meta Information | Description | Example |
|---|---|---|
| $(ComputerName) | The name of the computer. | MyComputer |
| $(CurrentDate) | The current date, in ISO format. | 2020-08-18 |
| $(CurrentDateTime) | The current datetime, in ISO format. | 2020-08-18T18:00:00 |
| $(DataColumns) | The number of columns in the dataset (not including meta data). | 10 |
| $(DataRows) | The number of row in the dataset (not including meta data). | 10,000 |
| $(DataValues) | The number of columns x rows in the dataset (not including meta data). | 100,000 |
| $(FileCreatedDate) | The date the file was created, in ISO format. Only available for file input. | 2020-08-18 |
| $(FileCreatedDateTime) | The datetime the file was created, in ISO format. Only available for file input. | 2020-08-18T18:00:00 |
| $(FileName) | The name of the file, including it's extension. Only available for file input. | myfile.csv |
| $(FilePath) | The full path (location) of the file. Only available for file input. | C:\users\andy\Documents\myfile.csv |
| $(FileSheetName) | The name of the sheet. Only available for Excel files. | Sheet1 |
| $(FileSizeBytes) | The size of the file in bytes. Only available for file input. | 1,234,567 |
| $(FileUpdatedDate) | The date the file was last updated, in ISO format. Only available for file input. | 2020-08-18 |


© 2023 Oryx Digital Ltd

| Meta Information | Description | Example |
|---|---|---|
| `$(FileUpdatedDateTime)` | The datetime the file was last updated, in ISO format. Only available for file input. | `2020-08-18T18:00:00` |
| `$(UserName)` | The name of the user (from the USER or USERNAME environment variable). | `Andy` |

See also:
- Stamp

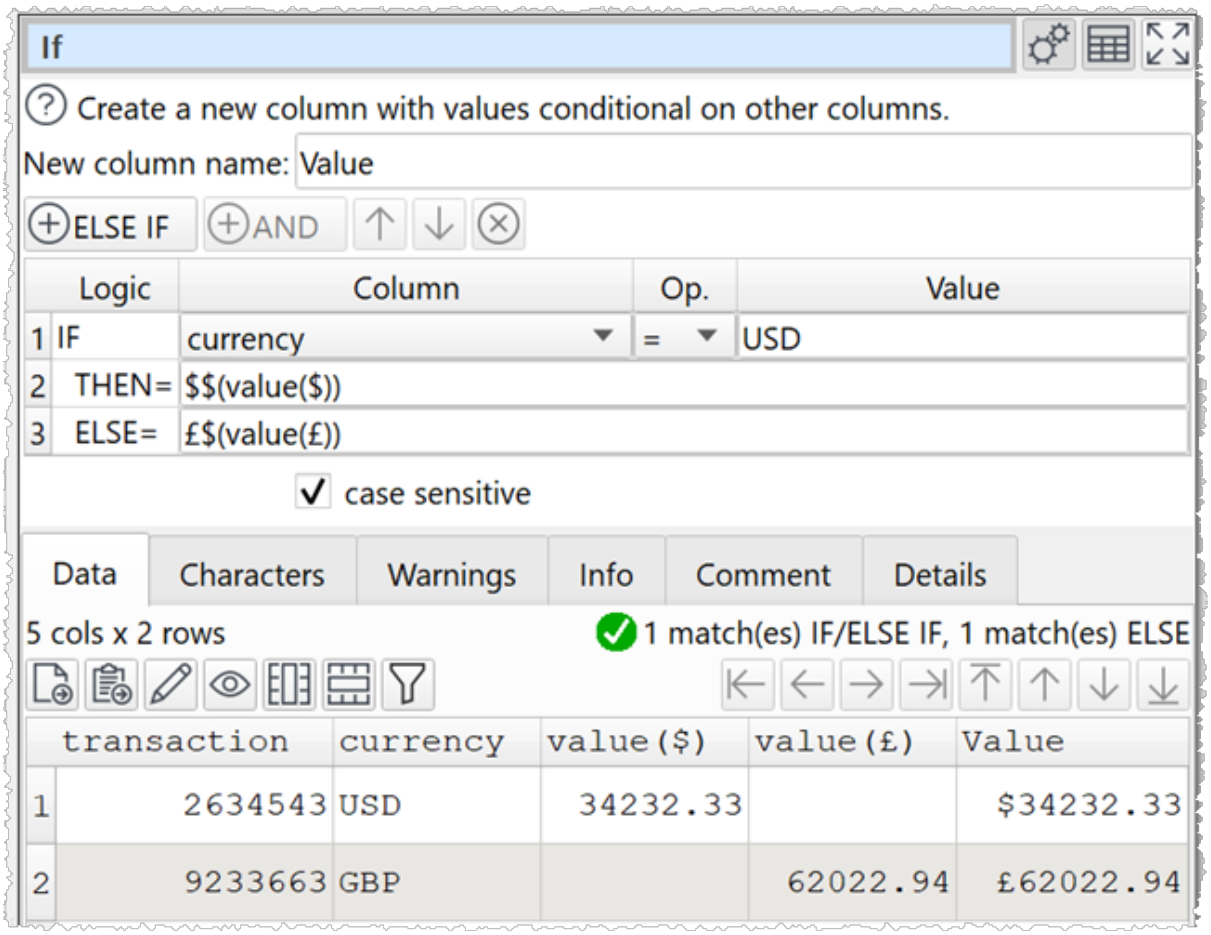
## 2.16 Column variables

Transforms such as If, Filter, Insert, Replace and Substitute allow you to use the values of columns on the same row using column variables. Column values can be referenced either:

- By column header name, e.g. `$(item cost)` for the 'item cost' column; or
- By column index, e.g. `$(1)` for the first column.

Notes:
- The column name is case sensitive. E.g. column 'A' is referenced by variable `$(A)` and column 'a' is referenced by variable `$(a)`.
- Whitespace at the start or end of the column name is ignored. E.g. column ' A ' is referenced by variable `$(A)`.
- Whitespace within the column variable is important. E.g. column 'AB' is not referenced by variables `$(A B)`, `$ (AB)` `$( AB)` or `$(AB )`.
- If multiple columns have the same name, the first column from the left with that name will be used.
- Reference by name takes priority over reference by index. For example, if there is a column named "1" then `$(1)` will refer to that rather than the first column.
- Column variables are replaced as text. For example if column 'A' has value '2' and column 'B' has value '3', then `$(A)*$(B)` evaluates as '2*3' (not '6').
- Column names can contain `$`, `(` and `)` characters. For example:

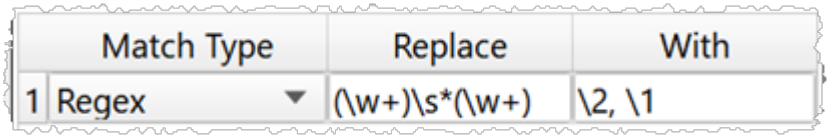

© 2023 Oryx Digital Ltd




## 2.17 Regular expressions

Easy Data Transform allows the use of regular expressions in the Compare, Extract, Filter, If, Replace, Slice, Split Col and Split Rows transforms. It is also available as part of the Javascript language in the Javascript transform.

Regular expressions are a powerful way to match patterns in text (including text representation of dates and numbers). For example, you can use a regular expression in the Replace transform to swap first and last names:

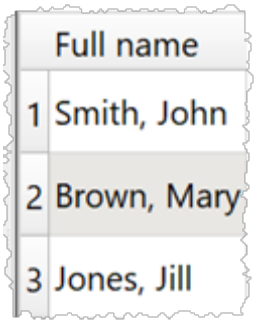



Turns:


© 2023 Oryx Digital Ltd

| | Full name |
|---|---|
| 1 | John Smith |
| 2 | Mary Brown |
| 3 | Jill Jones |

Into:



| | Full name |
|---|---|
| 1 | Smith, John |
| 2 | Brown, Mary |
| 3 | Jones, Jill |

Regular expressions are far too big a topic to cover here. However there are many detailed resources online, such as www.regular-expressions.info and regexr.com.

## 2.18 Fuzzy matching

Easy Data Transform supports the use of fuzzy matching in various transforms, including the Cluster, Dedupe , If, Filter and Lookup transforms.

Fuzzy matching allows you to match 2 values that are similar, but not identical. For example if you set fuzzy match closeness to 80% then 2 values that are 80% the same or better are considered a match.

For example, doing a fuzzy match to this value:

```
100 avenue street, townsville, ohio
```

Gives:

| Value | Fuzzy closeness |
|---|---|
| `100 avenue street, townsville, ohio` | `100%` |
| `100 avnue street, townsville, ohio` | `98%` |


© 2023 Oryx Digital Ltd

| Value | Fuzzy closeness |
|---|---|
| 100 avenue street townsville ohio | 95% |
| 100 avenue st., townsville, ohio | 89% |
| 100 avenue st, townsville | 72% |
| 100 av. st., citysville, texas | 52% |
| townsville, ohio | 46% |
| 742 evergreen terrace, springfield, oregon | 36% |

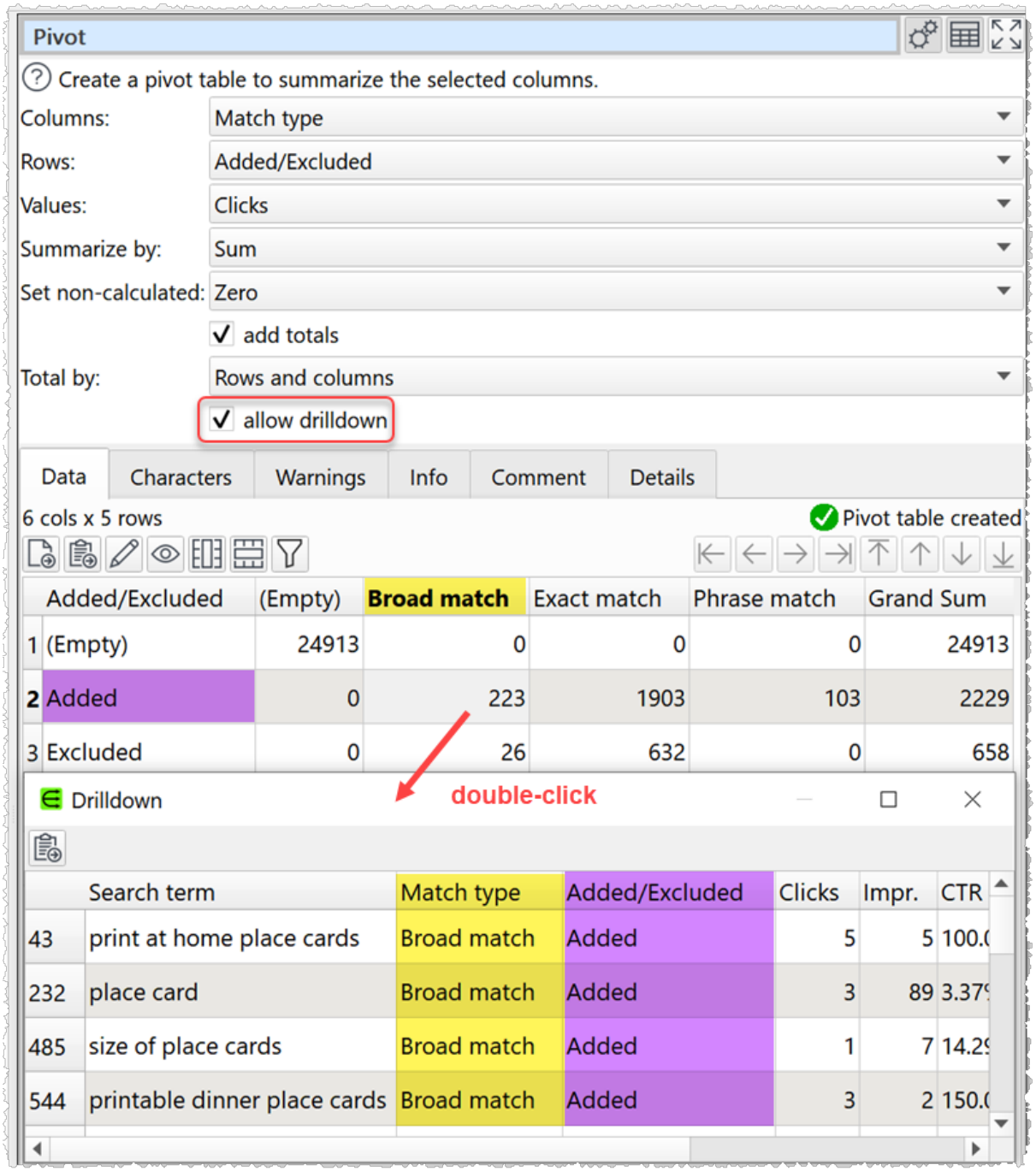
Fuzzy matching treats everything as a string and takes account of whitespace. It can optionally take account of case.

Fuzzy matching is significantly slower than exact matching. The closeness score is based on Levenshtein distance.

## 2.19 Drilldown

If you check **allow drilldown** for a transform, double-clicking a cell in the **Right** pane data table shows the rows in the upstream dataset that this cell/row was derived from. For example you can double-click a cell in a pivot table to drilldown into the rows it was calculated from:


© 2023 Oryx Digital Ltd




The following transforms allow drilldown:

- Count
- Dedupe
- Ngram
- Pivot
- Unique

Note that enabling drilldown slows down processing and requires more memory.

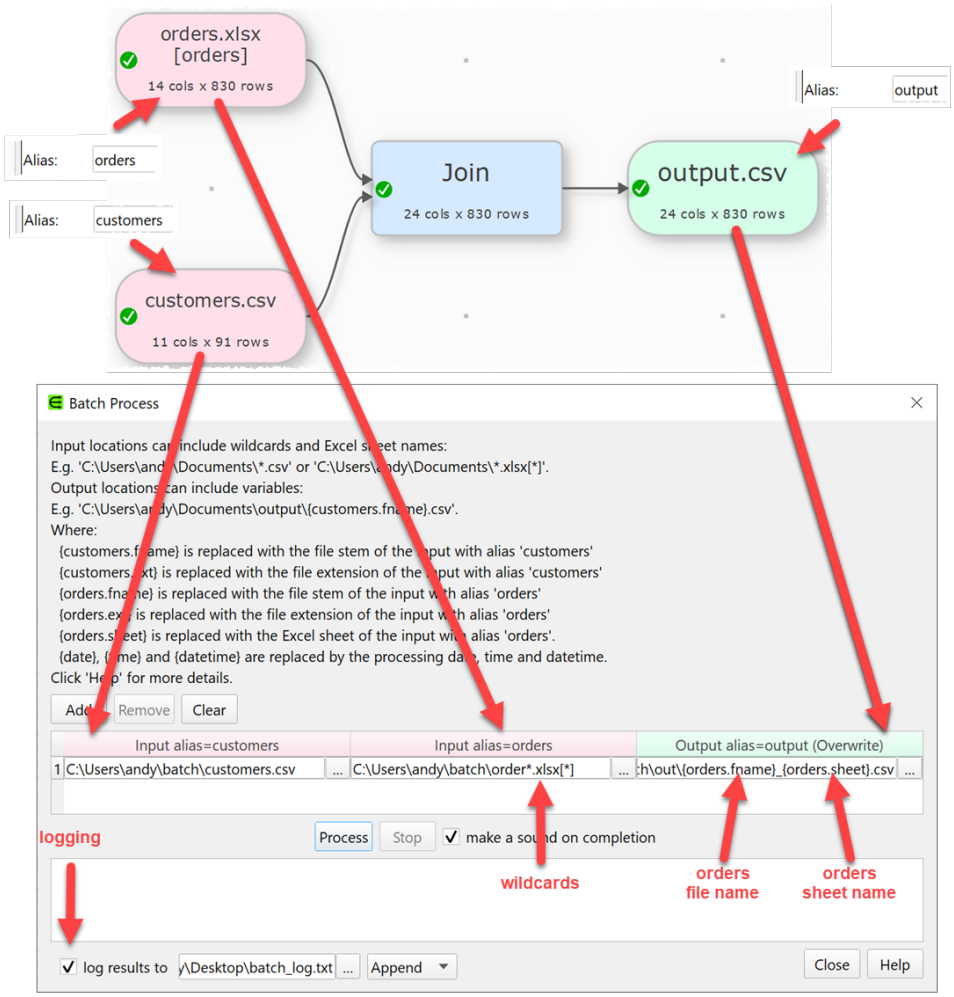

*© 2023 Oryx Digital Ltd*

See also:

- Video: How to drilldown into data

## 2.20 Batch processing

To apply the current `.transform` file to multiple input files/sheets select **File>Batch Process...** . The **Batch Process** window will appear with a column for each input item and a column for each output item. The **Alias** for each item is displayed in the column header.




Note:

- All input and output items must have an alias.
- An output item can't have the same alias as another output or input item.
- Disconnected input and output items are not shown.
- Output items with **Write mode**=Disabled are not shown.

Click **Add** to add a new processing row.


© 2023 Oryx Digital Ltd

Click **Remove** to remove the selected processing row(s).

Click **Clear** to remove all processing rows.

Check **clear 'Append' outputs before first append** if you want to clear output files you are appending to before the first append.

Check **log results to** if you want to log the results to a text file. Set the file location and set the write mode to **Overwrite** or **Append**. You can use `{datetime}`, `{date}` and `{time}` in the file location.

In the input column(s) you can use * and ? wildcards for file name stems, file extensions and Excel sheet names. E.g.:

| Input | Description |
|---|---|
| `C:\Users\andy\Documents\*.csv` | All files with extension .csv in the Documents folder |
| `C:\Users\andy\Documents\d?.csv` | All the files with name 'd' plus a single character in the Documents folder |
| `C:\Users\andy\Documents\data.xlsx[*]` | All the sheets in data.xlsx in the Documents folder |
| `C:\Users\andy\Documents\*.xlsx[data*]` | All the sheets beginning with 'data' in all the .xslx files in the Documents folder |

Note:

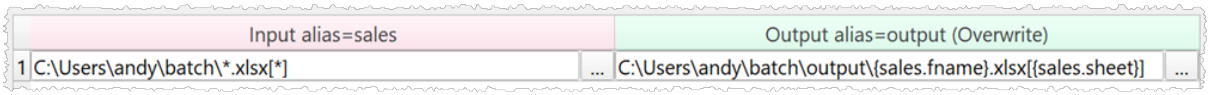
- If there is more than 1 input column on a row that specifies multiple files or sheets, then an output will be created for each possible permutation of input files/sheets in the row. E.g. 3 input files from column 1 x 4 sheets from column 2 = 12 outputs to process.
- Excel sheet names are not case sensitive.
- You cannot use wildcards for folder names.
- Batch processing will ignore files in sub-folders. Add them as extra rows.
- All the files input to an input item or output from an output item should be the same file type as the original.

In the output column(s) you can use the file name variables to dynamically change output file names based on input file names.

You can write to multiple sheets of an Excel file.


*© 2023 Oryx Digital Ltd*

This example shows using a wildcard for the input file name and sheet and file name variables to set the output file name based on the input file with alias `sales`:




But make sure to set the corresponding output **Write mode=**Overwrite/Sheet.

Click **Process** to start processing the rows.

Click **Stop** to stop processing the rows.

Check **make a sound on completion** if you want to play a sound when all processing is finished.

Click **Close** to close the window.

Note:
- Whether an output file is created, overwritten or appended to depends on the **Write mode** of the output item.
- We recommend you output to a different folder than the one the input files are in.

See also:
- Video: Batch processing
- Batch processing examples
- Command line arguments

## 2.21 Command line arguments

Easy Data Transform accepts the following command line arguments:

| Argument | Description |
|---|---|
| `<file name>` | The .transform file to open at start-up. |
| `-cli` | Close the application once any processing on the opened file is complete. |
| `-log <location>` | Appends the command line output to the file at `<location>`. `-cli` must also be set. |
| `-file <alias>=<location>` | Sets the input or output file with the given alias to the location (path) specified. Input Excel files should include the sheet name, e.g. `file.xlsx[sheet]`. Output Excel files |


© 2023 Oryx Digital Ltd

| Argument | Description |
|---|---|
| | may optionally include a sheet name. The file type should be the same as the original. |
| `-new_window` | Don't load the last opened `.transform` file, even if **open previous file at start-up** is checked in **Preferences**. |
| `-verbose` | Output additional information to the terminal. Useful for debugging problems. |

This allows you to run Easy Data Transform from the Windows Command Prompt or a .bat file. For example:

To run `C:\Users\andy\Documents\myfile 1.transform` with the output with alias `output1` output instead to `C:\Users\andy\Documents\data1.csv`:

```
"C:\Program Files (x86)\EasyDataTransform_v1\EasyDataTransform.exe"
"C:\Users\andy\Documents\myfile 1.transform" -file "output1=C:
\Users\andy\Documents\data1.csv" -cli -verbose
```

To run `C:\Users\andy\Documents\myfile2.transform` with the input with alias `input1` input instead from sheet `sheet1` of `C:\Users\andy\Documents\data 2.xlsx`:

```
"C:\Program Files (x86)\EasyDataTransform_v1\EasyDataTransform.exe"
"C:\Users\andy\Documents\myfile2.transform" -file "input1=C:
\Users\andy\Documents\data 2.xlsx[sheet1]" -cli -verbose
```

Use wildcards to transform multiple files in a folder:

```
forfiles /p C:\Users\andy\Desktop\cli /m in*.csv /c "cmd /c \"C:
\Program Files (x86)\EasyDataTransform_v1\EasyDataTransform.exe\" C:
\Users\andy\Desktop\cli\cli.transform -file \"cli-in=@file\" -file
\"cli-out=out-@fname.csv\" -cli -verbose"
```

Use file name variables to dynamically change output file names based on input file names.

Put quotes (") around any arguments with spaces (as shown in the examples above).

If you are using wildcards, then it is usually a good idea to output to a different folder to the one that you are inputting from.

Select **File>Command Line...** to show sample command line text which you can copy and modify. It will also warn you of potential issues.

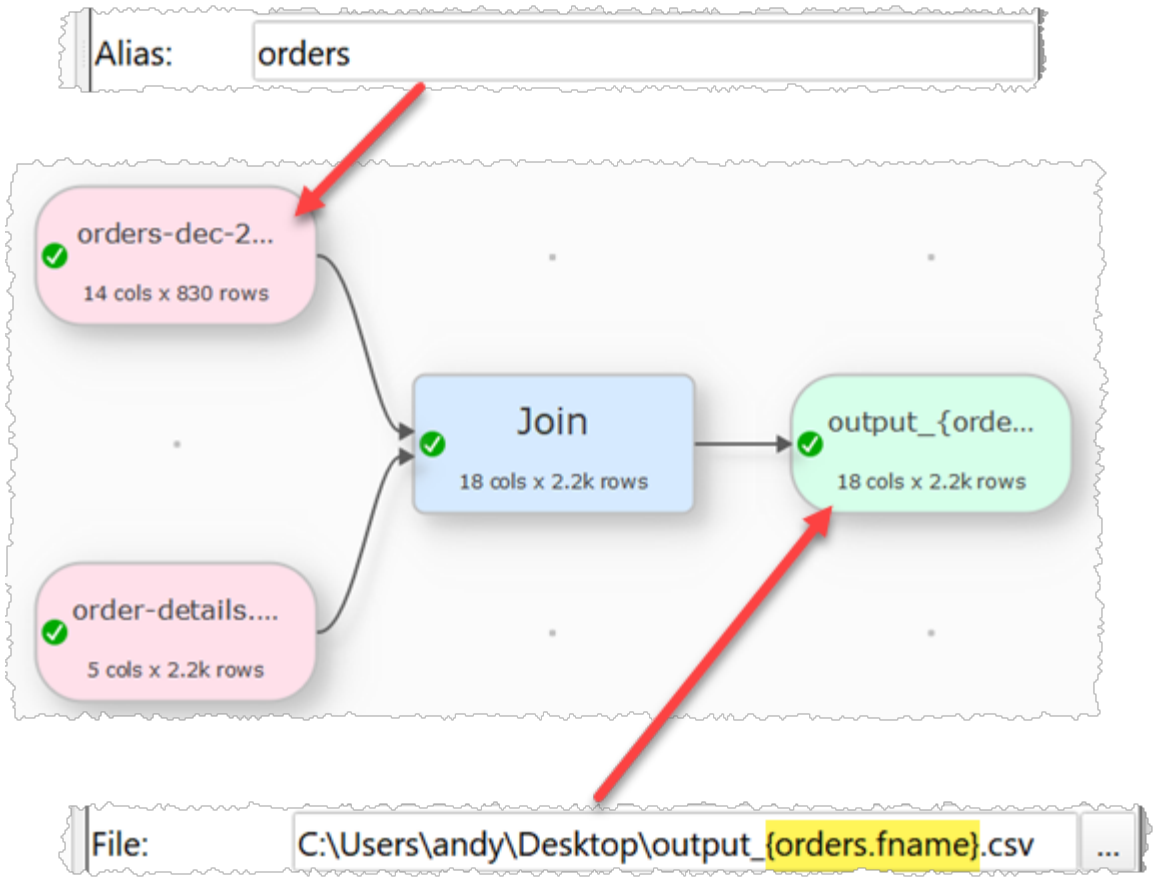

*© 2023 Oryx Digital Ltd*

To run on a schedule, call the command line interface (or a script file that calls the command line interface) from a scheduling program, such as Windows Task Scheduler.

See also:
- Batch processing
- Run jobs on a schedule

## 2.22 File name variables

You can dynamically change the name of an output file based on input file names using file name variables. For example, if the input file with alias `orders` has file name `orders_dec_2022.csv`, then setting the output filename to `output_{orders.fname}.csv` will write to `output_orders_dec_2022.csv`.




You can also include dates and times in your output file names.

| Output variables | Meaning | Example |
|---|---|---|


© 2023 Oryx Digital Ltd

| | | |
|---|---|---|
| {<input alias>.fname} | The name of the input file being processed in the column with the corresponding alias. | If the input with alias orders is using file C:\Users\andy\Documents\orders_2020.csv then {orders.fname} is replaced with orders_2020. |
| {<input alias>.ext} | The extension of the input file being processed in the column with the corresponding alias. Does not include the '.'. | If the input with alias orders is using file C:\Users\andy\Documents\orders_2020.csv then {orders.ext} is replaced with value csv. |
| {<input alias>.sheet} | The sheet of the input file being processed in the column with the corresponding alias (Excel only). | If the input with alias orders is using file C:\Users\andy\Documents\orders_2020.xlsx with sheet Sheet1 then {orders.sheet} is replaced with value Sheet1. |
| {<input alias>} | The name of the input file (plus an underscore and the sheet name, if an Excel file) being processed in the column with the corresponding alias. | If the input with alias orders is using file C:\Users\andy\Documents\orders_2020.csv then {orders} is replaced with value orders_2020.<br>If the input with alias orders is using file C:\Users\andy\Documents\orders_2020.xlsx with sheet Sheet1 then {orders} is replaced with value orders_2020_Sheet1. |
| {date} | Date processing was carried out in year_month_day format. | 2020_04_18 |
| {time} | Time processing was carried out in | 15_21_56_599 |


© 2023 Oryx Digital Ltd

| | | |
|---|---|---|
| | hours_minutes_seconds_milliseconds format. | |
| {datetime} | Date/Time processing was carried out in year_month_day_hours_minutes_seconds_milliseconds format | 2020_04_18_15_21_56_599 |
| {version} | The Easy Data Transform version number | v1_39_0 |

File name variables can be used for output file names set in the **Right** pane, in batch processing or through command line arguments.

## 2.23 .transform files

.transforms file are stored in a simple XML format. So you can edit them with a standard text editor. However we recommend you make a copy first.

The results of transformations are not stored in the .transform file, and are recalculated whenever you **File>Open...** the file.

The contents of Input and Output files are not stored in the .transform file, only their locations. These locations are stored as 'absolute' locations, so you can move the .transform file without changing the locations of the Input and Output files.

If you open a .transform file in a different location from that in which it was saved and it can't find Input and Output files at the expected location it will look for them in the same location relative to the old .transform file. This allows you to easily move .transform files to different locations and computers if you keep the Input and Output files in the same relative location (e.g. in the same folder as the .transform file). This even works between Windows and Mac (and vice versa),

Example:

- `mytransform.transform` is in `C:\Users\andy\Documents\` on Windows and uses Input file `MyData.csv` in sub-folder `MyData` (`C:\Users\andy\Documents\Data\MyData.csv`).
- `mytransform.transform` is moved to `/Users/Bob/Documents/EDT` on a Mac.
- When `mytransform.transform` is opened it will look for `MyData.csv` in `/Users/andy/Documents/Data`.
- If it can't find that it will look for `MyData.csv` in sub-folder `MyData` (`/Users/Bob/Documents/EDT/Data/MyData.csv`).


© 2023 Oryx Digital Ltd

If you paste in data **From Clipboard** this is stored in the .transform file. We don't recommend you do this for large datasets as XML is not very efficient for storing large amounts of data.

## 2.24 Keyboard shortcuts

Using keyboard shortcuts can improve your productivity. If you are using Easy Data Transform a lot we suggest you find the time to learn at least some of them. The following keyboard shortcuts are available for the Windows version of Easy Data Transform:

| Key | Shortcut | Action |
|---|---|---|
| A | `Ctrl+A` | Select all in **Center** pane. |
| B | `Ctrl+B` | Show the **Batch Process** window. |
| D | `Ctrl+D` | Duplicate the selected branch in the **Center** pane. |
| I | `Alt+I` | Input From File. |
| | `Alt+Shift+I` | Input From Clipboard. |
| K | `Ctrl+K` | Search transforms by keyword. |
| L | `Ctrl+L` | Toggle **Log** pane. |
| N | `Ctrl+N` | New .transform file. |
| O | `Ctrl+O` | Open .transform file. |
| | `Ctrl+Shift+O` | Show the **Open Recent** window. |
| | `Alt+O` | Output To File. |
| R | `Ctrl+R` | Run unprocessed items [4]. |
| | `Ctrl+Shift+R` | Run selected items [4]. |
| S | `Ctrl+S` | Save .transform file. |
| T | `Ctrl+T` | Toggle two screen mode on/off |
| U | `Ctrl+U` | Toggle user interface between light and dark themes (if available). |


*© 2023 Oryx Digital Ltd*

| Key | Shortcut | Action |
|---|---|---|
| Y | `Ctrl+Y` | Redo |
| Z | `Ctrl+Z` | Undo |
| Del | `Del` | Delete selected item(s) in **Center** pane. |
| , | `Ctrl+,` | Show **Preferences** window. |
| = | `Ctrl+=` | Zoom **Center** pane so all items fit. |
| + | `Ctrl++` | Zoom **Center** pane in. |
| - | `Ctrl+-` | Zoom **Center** pane out. |
| \ | `Ctrl+\` | Stop processing. |
| Left arrow | `Ctrl+Left arrow` | Move **Center** pane selection from item to highest[1] item that inputs to it. |
| | `Alt+Left arrow` | Move keyboard focus to **Center** pane. |
| Right arrow | `Ctrl+Right arrow` | Move **Center** pane selection from item to highest[1] item that it outputs to. |
| | `Alt+Right arrow` | Move keyboard focus to **Right** pane. |
| Up arrow | `Ctrl+Up arrow` | Move **Center** pane selection from item to highest[1] sibling[3]. |
| Down arrow | `Ctrl+Down arrow` | Move **Center** pane selection from item to lowest[2] sibling[3]. |
| 1...9 | `Ctrl+1 ... Ctrl+9` | Select input item 1 to 9 (based on height in **Center** pane). |
| | `Ctrl+Shift+1 ... Ctrl+Shift+9` | Select output item 1 to 9 (based on height in **Center** |


*© 2023 Oryx Digital Ltd*

| Key | Shortcut | Action |
|---|---|---|
| | | pane). |
| F1 | `F1` | Show help. |
| F5 | `F5` | (Re)process all. |
| F8 | `F8` | Show options and data in the **Right** pane. |
| F9 | `F9` | Show only options in the **Right** pane. |
| F10 | `F10` | Show only data in the **Right** pane. |
| F11 | `F11` | Toggle setting **Right** pane item to fullscreen. Only works if 1 item in **Right** pane. |

[1] Highest=nearest the top of the **Center** pane.
[2] Lowest=nearest the bottom of the **Center** pane.
[3] Two items are considered siblings if they have inputs from the same item(s) or they both have no inputs.
[4] If **Run>Auto Run** is unchecked.

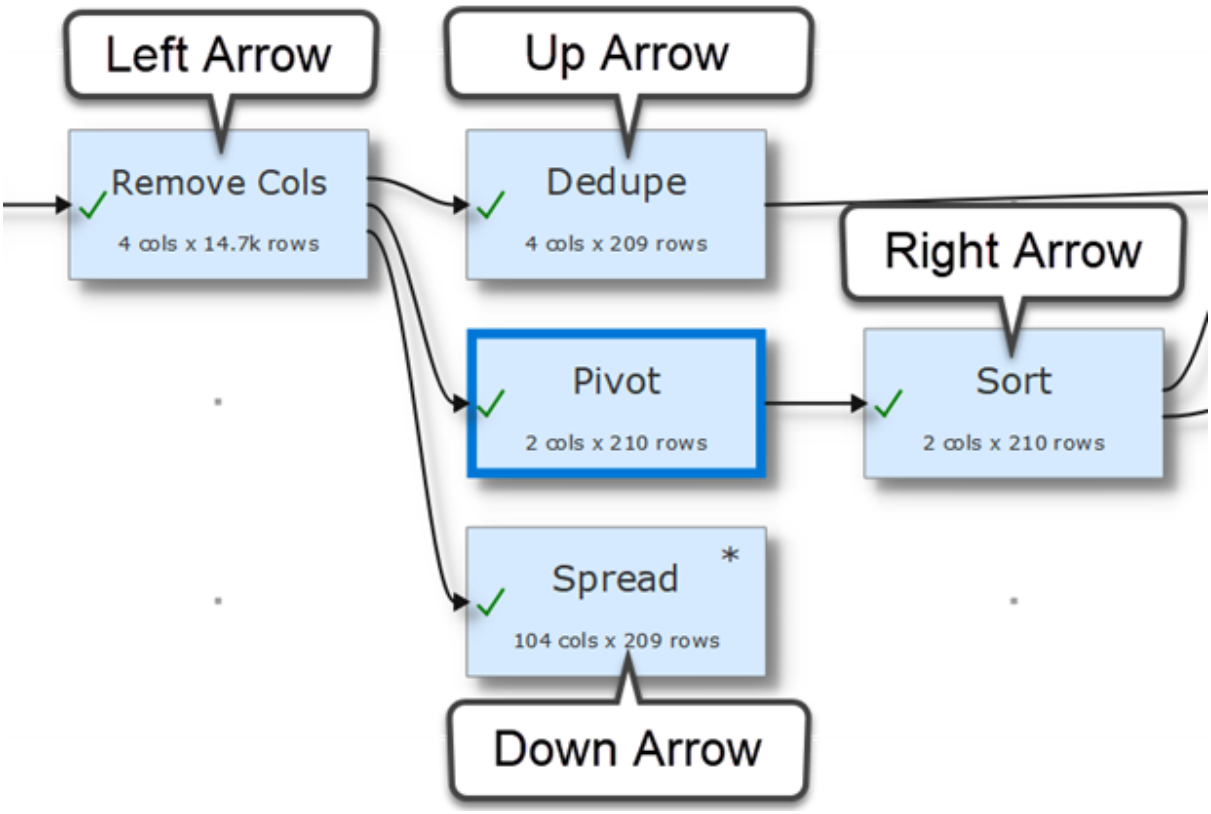
You can also use the keyboard to add transforms in the **Center** pane. Just select the item(s) you want to add the transform to and start typing the name. Only eligible transform that contain the typed letters will be displayed (spaces are ignored).

For example, to add the **Rename Cols** transform an existing Input item:
- select the input items
- type `ren`
- press the `Return` key

If you want to see a list of all the transform names, press the `Space` key before you start typing. You can use the `Del` or `Backspace` key to undo letters typed.

You can quickly change selection in the **Center** pane using arrow keys with the `Ctrl` key.


*© 2023 Oryx Digital Ltd*

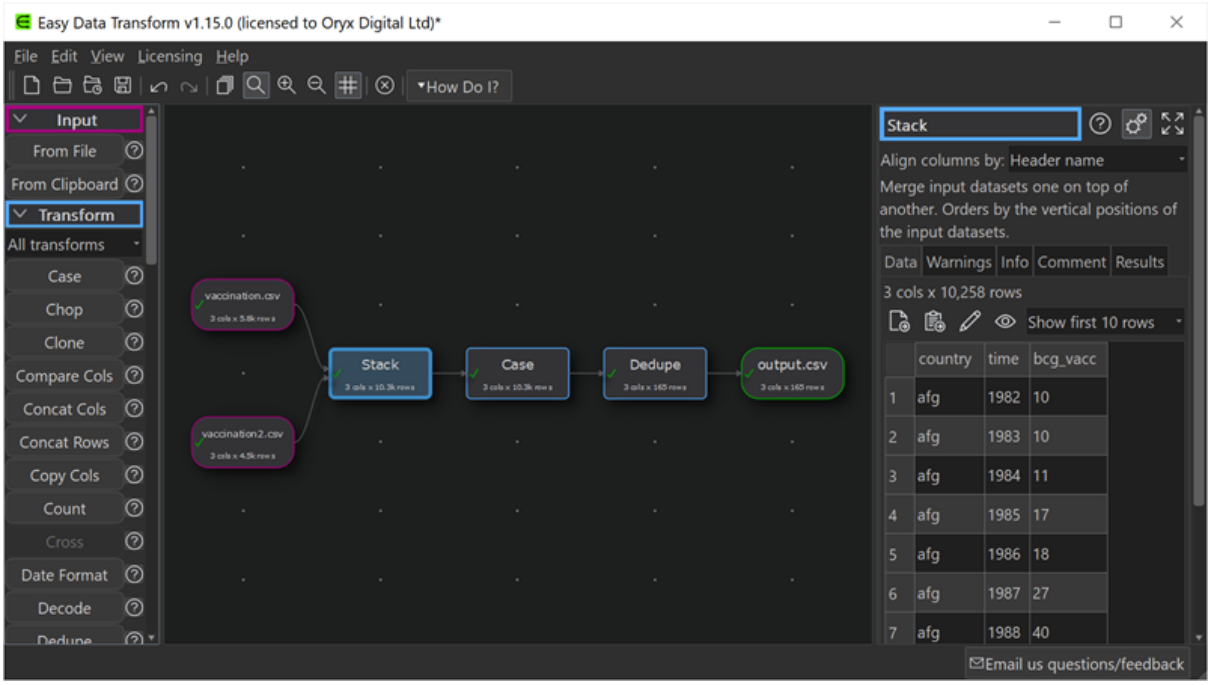



If you are zoomed in you can scroll the Center pane by pressing the `Shift` key and dragging the canvas.

## 2.25 Dark mode

Select **View>Toggle UI Theme** (or the corresponding status bar button) to swap between light and dark user interface themes. You can also change the **Center** color scheme in the **Colors** tab of the **Preferences** window.


*© 2023 Oryx Digital Ltd*





*© 2023 Oryx Digital Ltd*

# How do I?

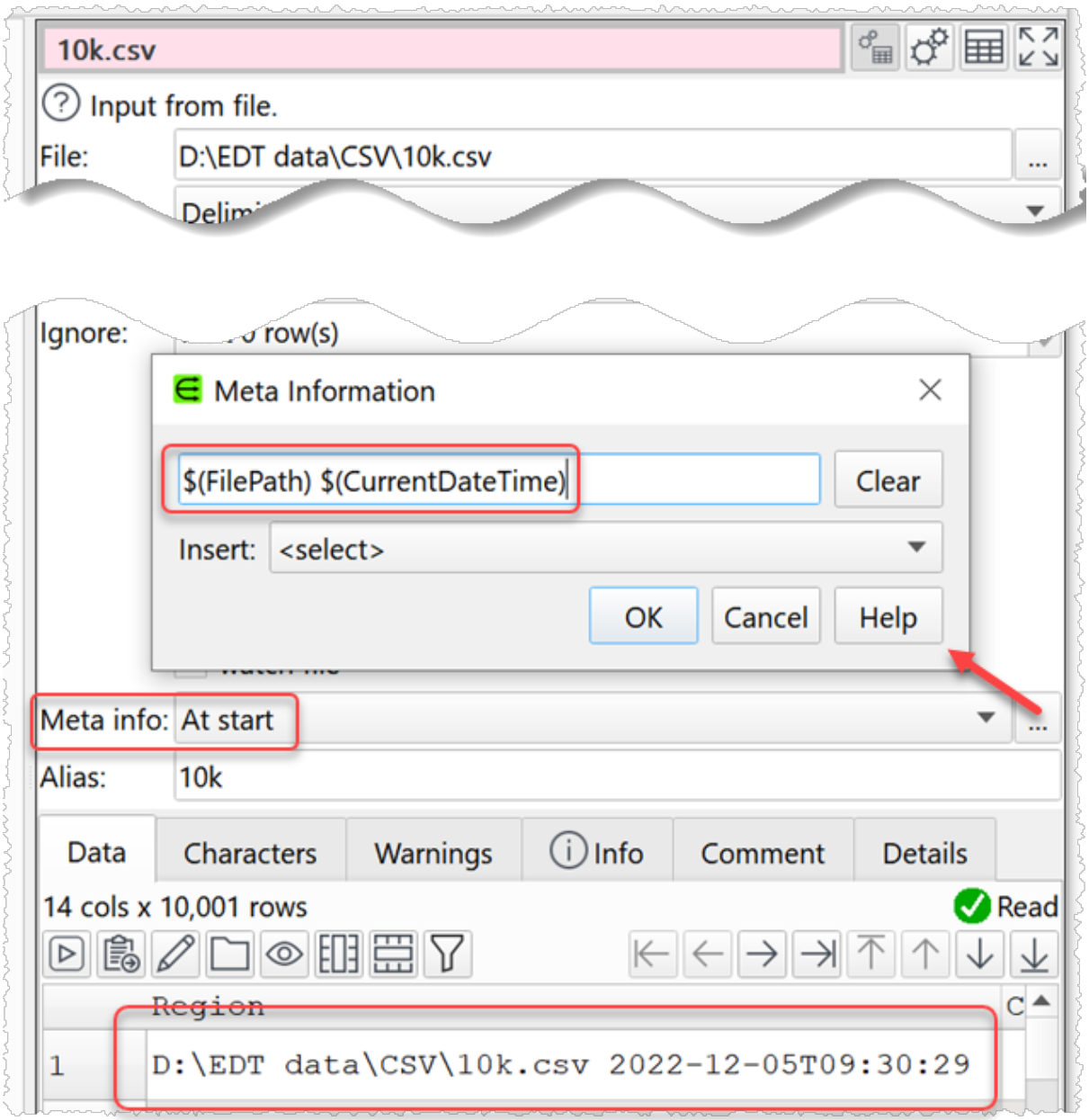
# 3 How do I?

## 3.1 Add a transform between existing items

To add a new transform between existing items (e.g. between 2 already connected transforms) see connections.

## 3.2 Add metadata to a dataset

You can add meta data to an input dataset by setting the **Meta info** field. See meta information for more information.





© 2023 Oryx Digital Ltd

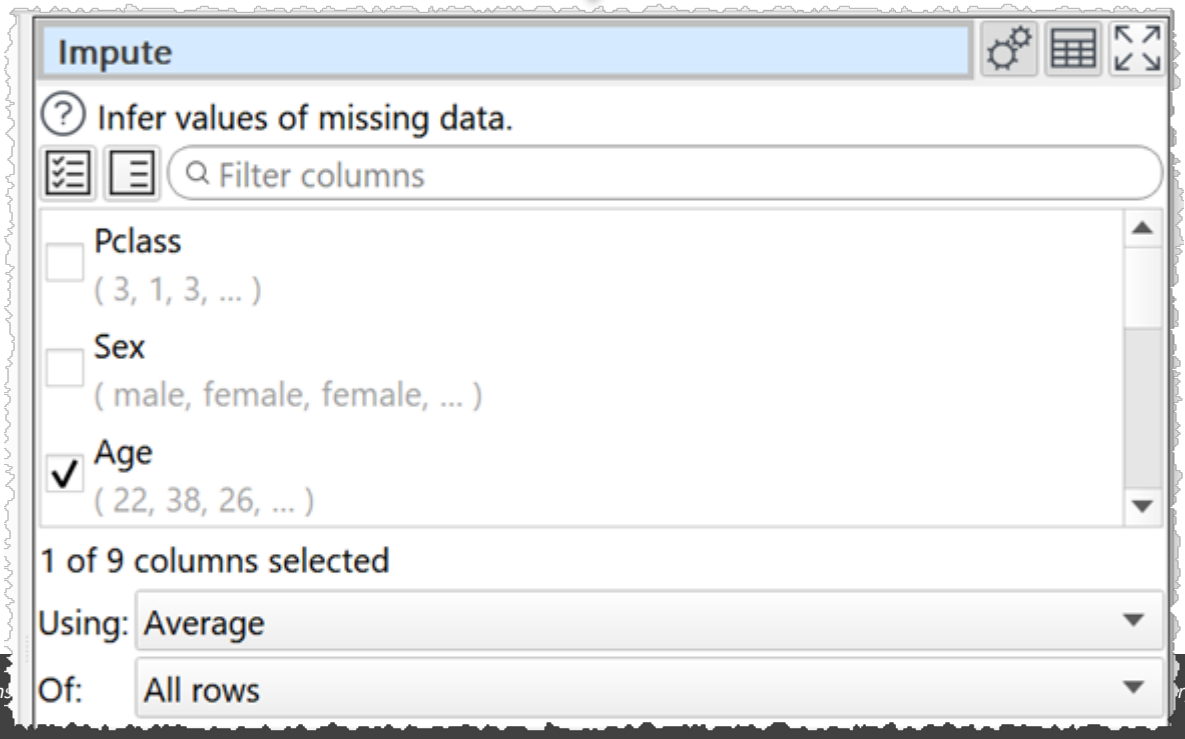
## 3.3 Add missing data values

You can add missing values based on the average (mean), median or mode of non-empty values in the same column using the **Impute** transform.

For example you can add missing ages for Titanic passengers based on the average age supplied for the other passengers:


*© 2023 Oryx Digital Ltd*

| | Pclass | Sex | Age | SibSp | Parch |
|---|---|---|---|---|---|
| 21 | 2 | male | 35 | 0 | 0 |
| 22 | 2 | male | 34 | 0 | 0 |
| 23 | 3 | female | 15 | 0 | 0 |
| 24 | 1 | male | 28 | 0 | 0 |
| 25 | 3 | female | 8 | 3 | 1 |
| 26 | 3 | female | 38 | 1 | 5 |
| 27 | 3 | male | | 0 | 0 |
| 28 | 1 | male | 19 | 3 | 2 |
| 29 | 3 | female | | 0 | 0 |
| 30 | 3 | male | | 0 | 0 |
| 31 | 1 | male | 40 | 0 | 0 |



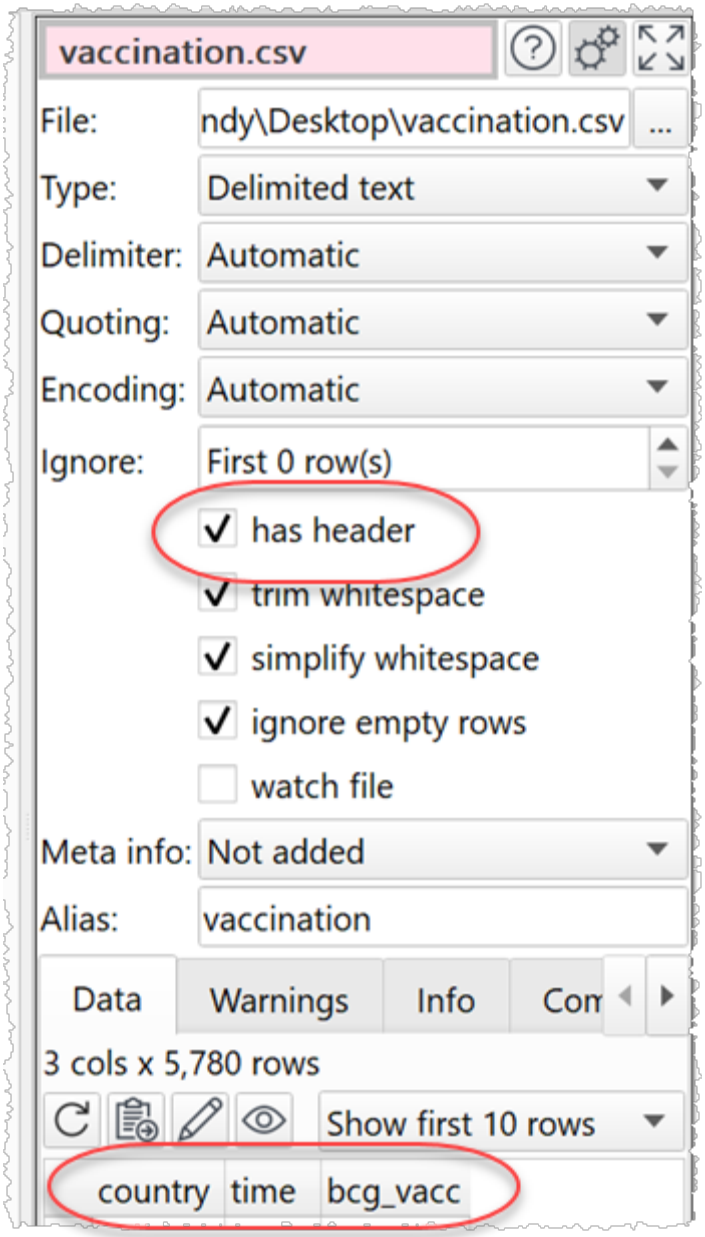
See the Impute transform documentation for more details.

See also:

- Video: How to impute missing data
- Replace empty values
- Profile a dataset

## 3.4 Add or remove a header

To add or remove a header just check or uncheck the **has header** checkbox for the appropriate input item.





© 2023 Oryx Digital Ltd

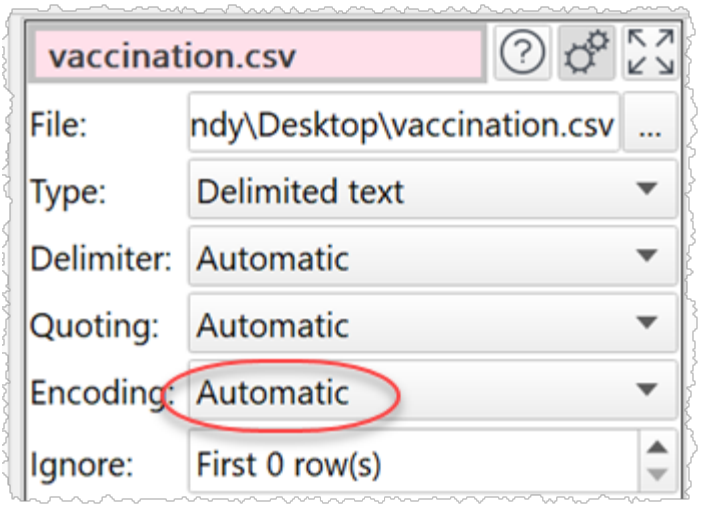
You can also make the first row of your dataset into a header using the Header transform.

## 3.5 Change a connection

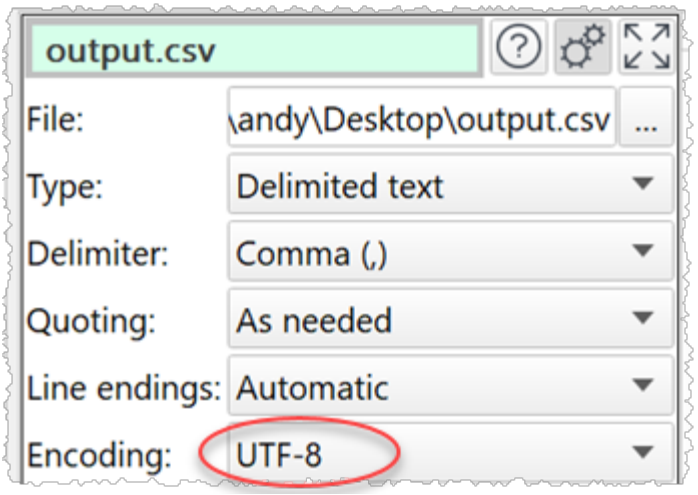
To change a connection see connections.

## 3.6 Change encoding

When Easy Data Transform inputs a text file (e.g. a CSV file) it will make a guess at the encoding. You can explicitly set the encoding by selecting an input item and changing **Encoding** from **Automatic** to one of the other encodings in the **Right** pane.




Similarly you can also set the encoding of a text file output by selecting the output item and changing **Encoding** in the **Right** pane.

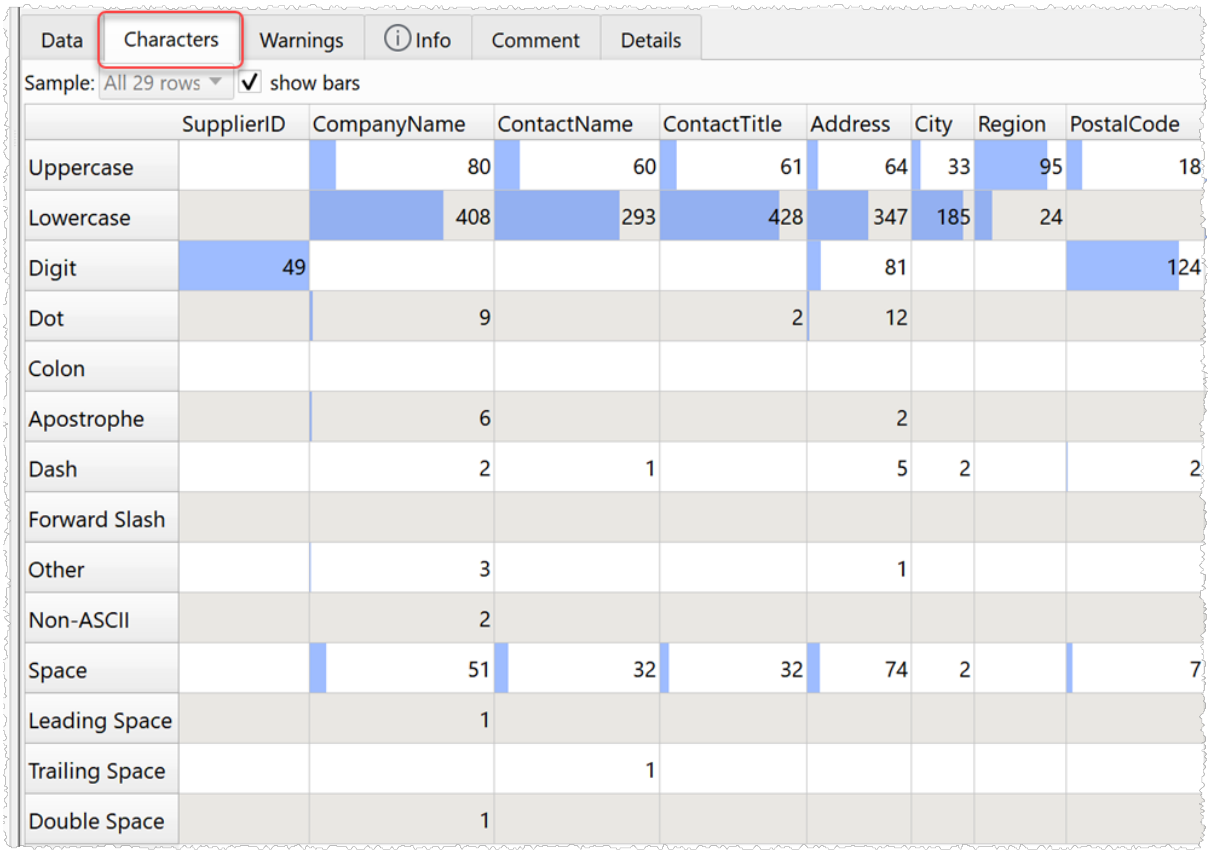




© 2023 Oryx Digital Ltd

## 3.7 Clean a dataset

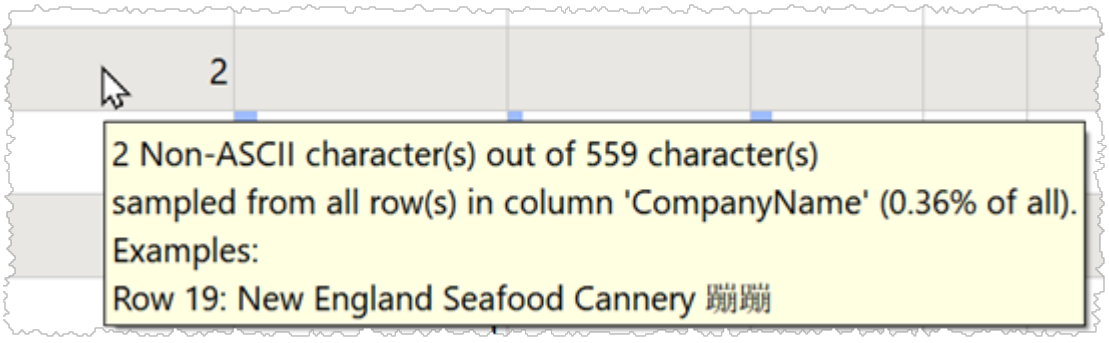
Data is often 'dirty' and needs to be cleaned up before further processing. You can quickly find unwanted characters by looking in the **Characters** tab of the **Right** pane to see what types of characters occur in which columns.




| | SupplierID | CompanyName | ContactName | ContactTitle | Address | City | Region | PostalCode |
|---|---|---|---|---|---|---|---|---|
| Uppercase | | 80 | 60 | 61 | 64 | 33 | 95 | 18 |
| Lowercase | | 408 | 293 | 428 | 347 | 185 | 24 | |
| Digit | 49 | | | | 81 | | | 124 |
| Dot | | 9 | | 2 | 12 | | | |
| Colon | | | | | | | | |
| Apostrophe | | 6 | | | 2 | | | |
| Dash | | 2 | 1 | | 5 | 2 | | 2 |
| Forward Slash | | | | | | | | |
| Other | | 3 | | | 1 | | | |
| Non-ASCII | | 2 | | | | | | |
| Space | | 51 | 32 | 32 | 74 | 2 | | 7 |
| Leading Space | | 1 | | | | | | |
| Trailing Space | | | 1 | | | | | |
| Double Space | | 1 | | | | | | |

Note that some of these categories are non-exclusive. For example: a space might be counted as a space, whitespace and a leading space, and a symbol might also be a non-ASCII character.

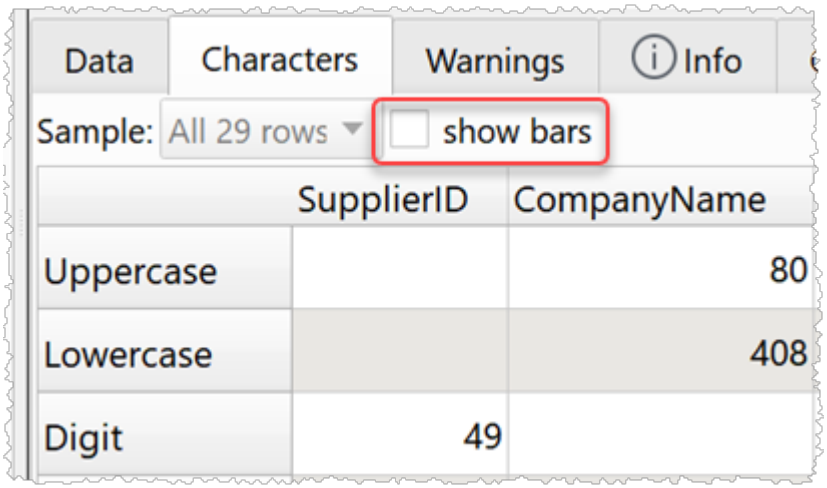
Hover over a cell for more details.



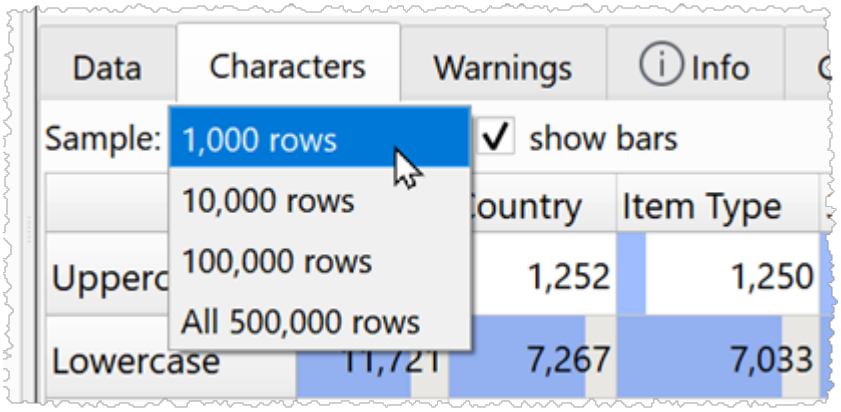


© 2023 Oryx Digital Ltd

You can turn the colored bars off.

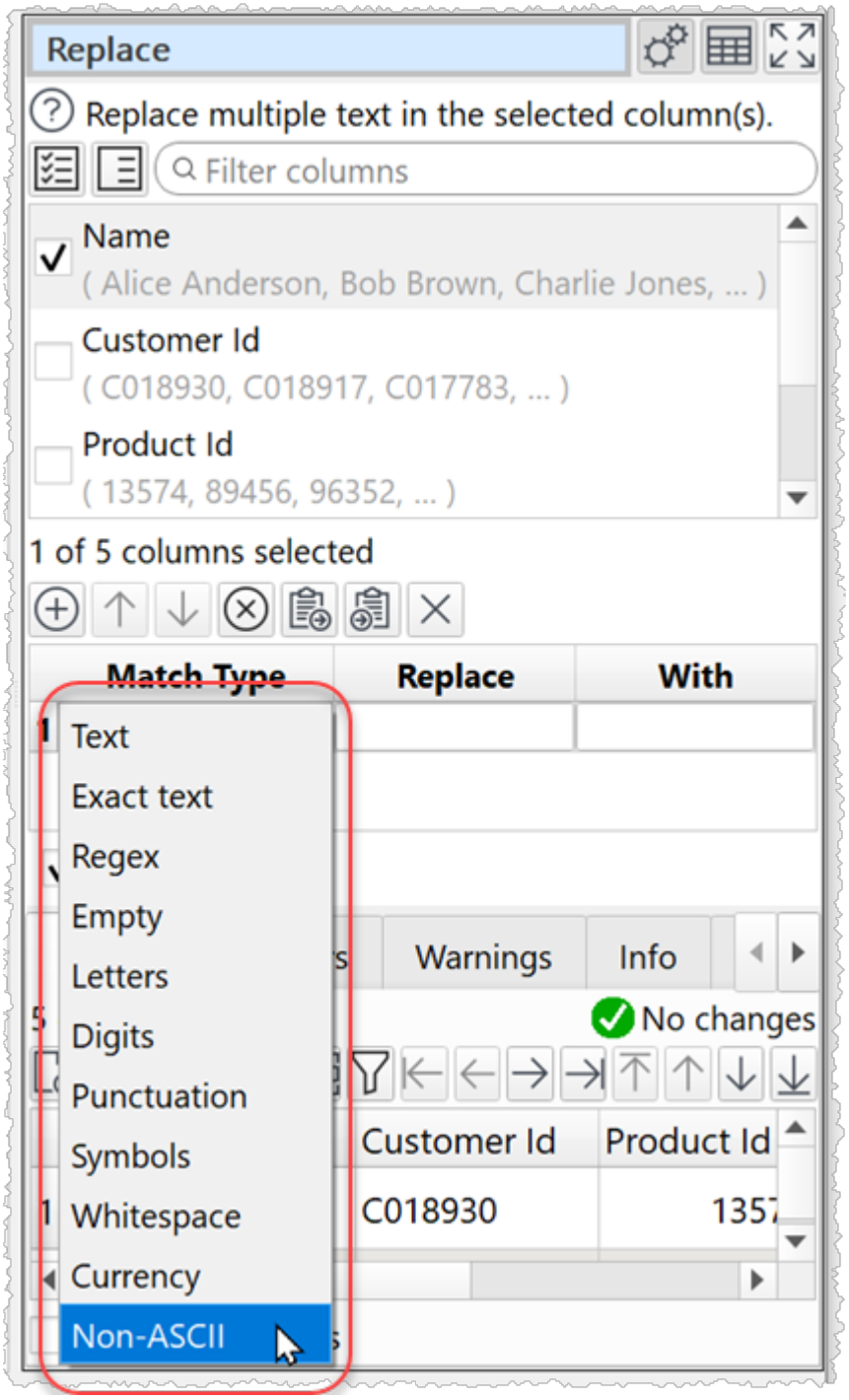



And you can restrict it to sampling only a subset of rows for improved speed in large datasets.




You can use the **Characters** tab in conjunction with the Replace and Whitespace transforms to quickly clean your data. **Replace** can easily remove non-ASCII, symbols etc by replacing them with nothing:


*© 2023 Oryx Digital Ltd*




See also:

- Video: How to clean data
- Add missing data values
- Profile a dataset

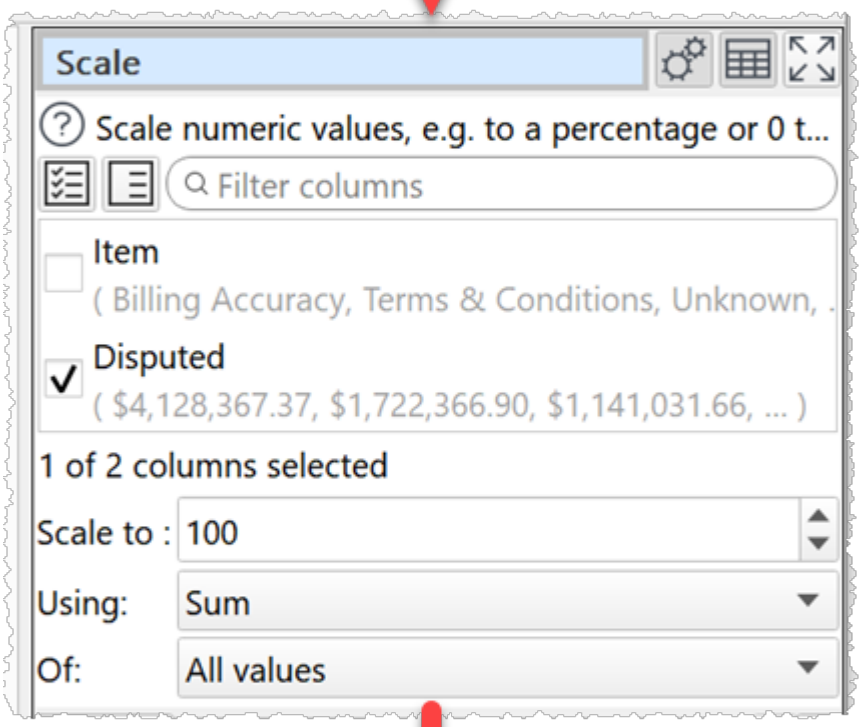

© 2023 Oryx Digital Ltd

## 3.8 Convert to percentages

You can convert to percentages using the Scale transform.


*© 2023 Oryx Digital Ltd*

| | Item | Disputed |
|---|---|---|
| 1 | Billing Accuracy | $4,128,367.37 |
| 2 | Terms & Conditions | $1,722,366.90 |
| 3 | Unknown | $1,141,031.66 |
| 4 | Delivery & Installation | $1,091,424.82 |
| 5 | Service Delivery | $238,914.81 |
| 6 | Technical Issues | $53,816.33 |

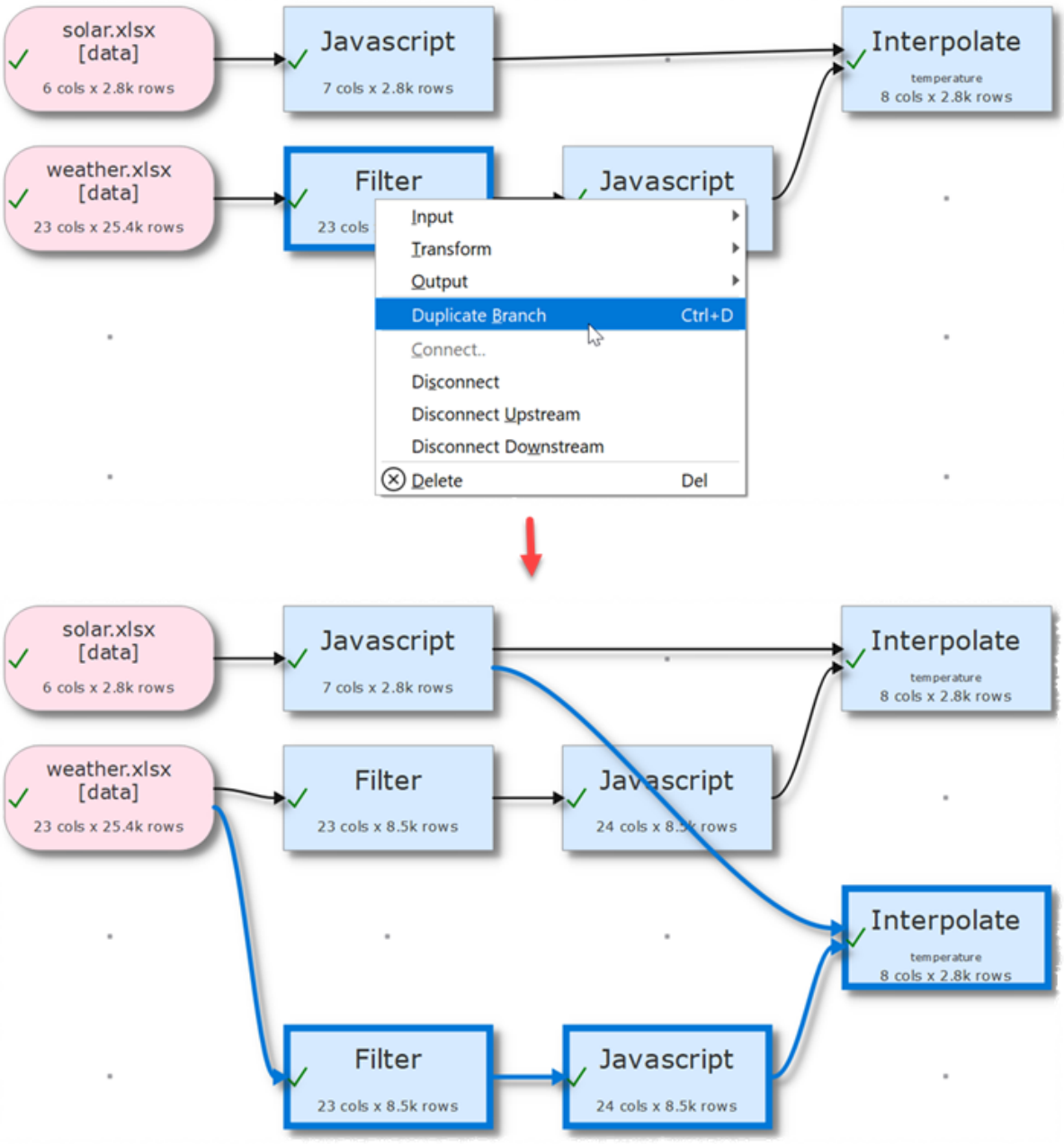



| | Item | Disputed |
|---|---|---|
| 1 | Billing Accuracy | 49.2885132433 |
| 2 | Terms & Conditions | 20.5633113897 |
| 3 | Unknown | 13.6227590823 |
| 4 | Delivery & Installation | 13.0305037981 |


© 2023 Oryx Digital Ltd

## 3.9 Duplicate a series of transforms

To duplicate multiple items in the center pane (e.g. a sequence of transforms) right click on the first item and select **Duplicate Branch**.




## 3.10 Dedupe a dataset

There are 2 transforms for removing duplicate rows from datasets:

- Dedupe removes duplicate rows, keeping only the first row in each group that it considers to be duplicates of each other


*© 2023 Oryx Digital Ltd*

- Unique creates a single aggregate row from each group that it considers to be duplicates of each other

Dedupe is simpler. Unique has more flexibility.

## Using Dedupe

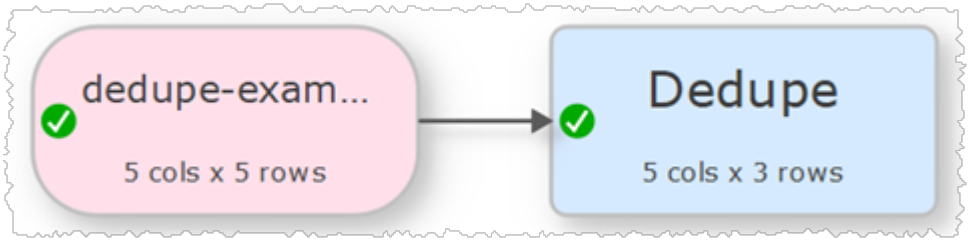
To keep only the first row with each **Customer Id** from this dataset:

| | Name | Customer Id | Product Id | Cost | Date |
|---|---|---|---|---|---|
| 1 | Alice Anderson | C018930 | 13574 | 29.95 | 01/10/2020 |
| 2 | Bob Brown | C018917 | 89456 | 10.55 | 01/10/2020 |
| 3 | Charlie Jones | C017783 | 96352 | 19.95 | 02/10/2020 |
| 4 | Robert Brown | C018917 | 98526 | 10.00 | 02/10/2020 |
| 5 | Charles Jones | C017783 | 28746 | 25.00 | 03/10/2020 |

To get this dataset:

| | Name | Customer Id | Product Id | Cost | Date |
|---|---|---|---|---|---|
| 1 | Alice Anderson | C018930 | 13574 | 29.95 | 01/10/2020 |
| 2 | Bob Brown | C018917 | 89456 | 10.55 | 01/10/2020 |
| 3 | Charlie Jones | C017783 | 96352 | 19.95 | 02/10/2020 |

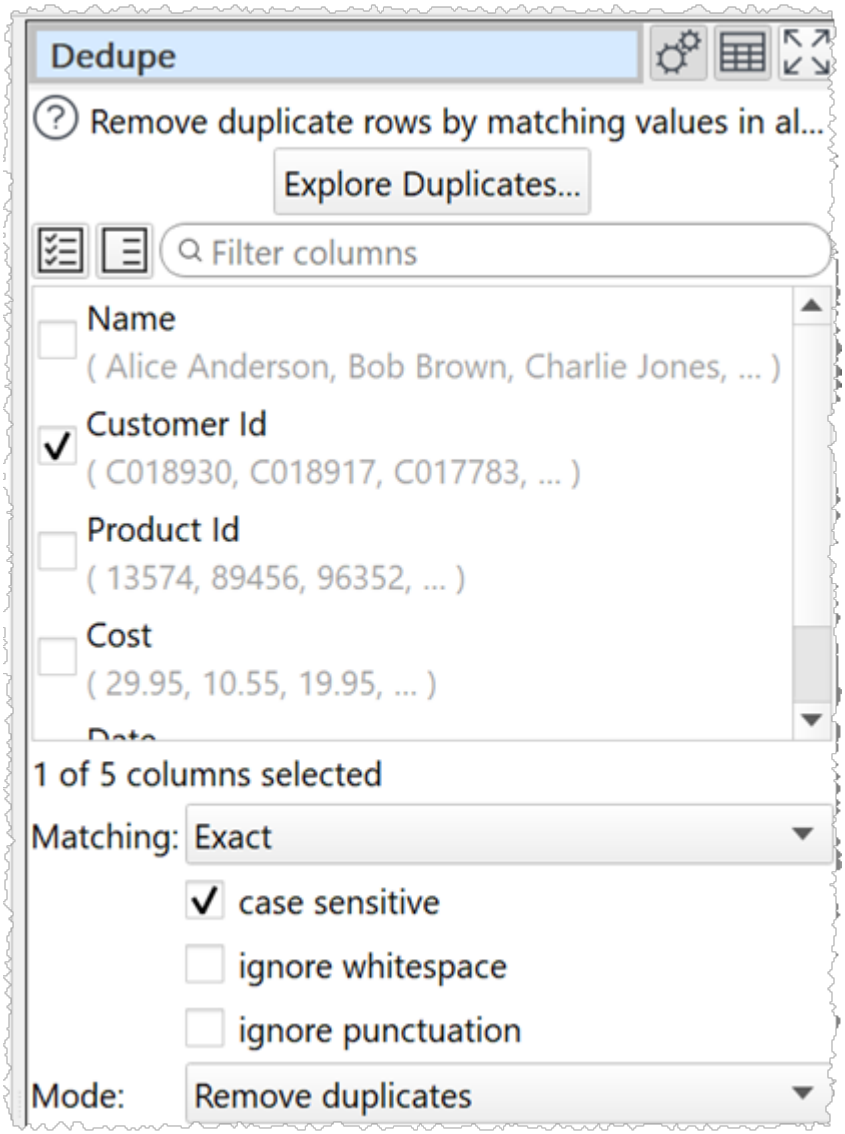
Drag the dataset file onto the **Center** pane of Easy Data Transform. Then click the **Dedupe** transform in the **Left** pane.





© 2023 Oryx Digital Ltd

Then check the **Customer Id** column in the **Right** pane.



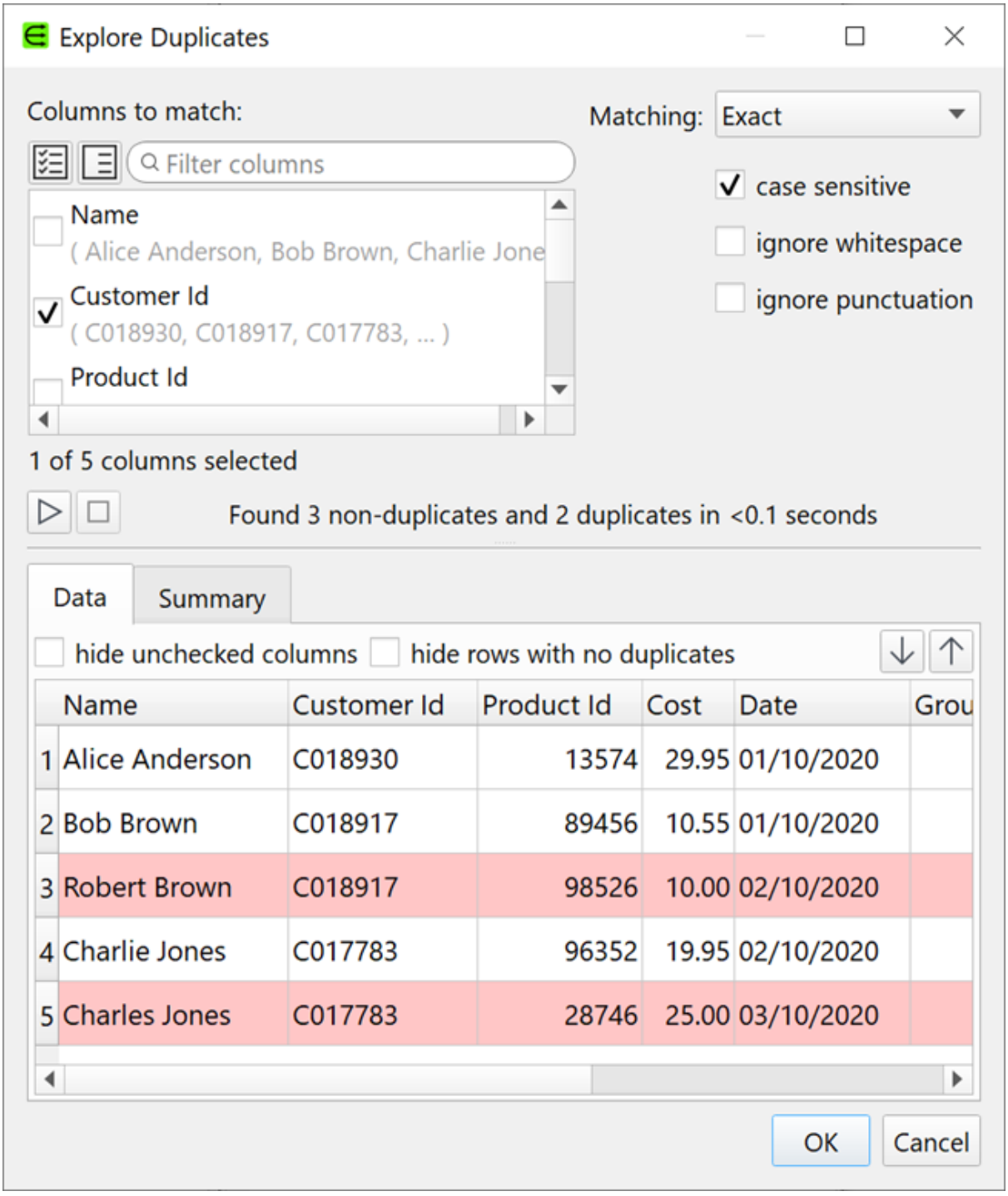

Only the first row with each **Customer Id** is kept. Use Sort if you want to change the order before the **Dedupe** transform.

If you only want to remove rows with the same **Customer Id** and **Product Id**, check both the **Customer Id** and **Product Id** columns.

If you want to use fuzzy matching to also remove rows that are similar, but not identical, set **Matching** to **Fuzzy**.


*© 2023 Oryx Digital Ltd*

To see what rows will be removed and experiment with different options, click the **Explore Duplicates...** button.




Note that **Dedupe** takes account of whitespace. So you might need a Whitespace transform before the **Dedupe** transform.


*© 2023 Oryx Digital Ltd*

See also the Dedupe documentation.

## Using Unique

To aggregate all rows with the same **Customer Id**:

| | Name | Customer Id | Product Id | Cost | Date |
|---|---|---|---|---|---|
| 1 | Alice Anderson | C018930 | 13574 | 29.95 | 01/10/2020 |
| 2 | Bob Brown | C018917 | 89456 | 10.55 | 01/10/2020 |
| 3 | Charlie Jones | C017783 | 96352 | 19.95 | 02/10/2020 |
| 4 | Robert Brown | C018917 | 98526 | 10.00 | 02/10/2020 |
| 5 | Charles Jones | C017783 | 28746 | 25.00 | 03/10/2020 |

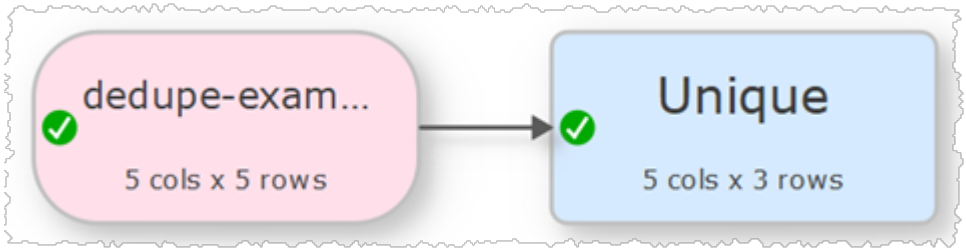
To get this dataset with:

- The first **Name** for that **Customer Id**
- A comma separated list of **Product Ids** for that **Customer Id**
- The total **Cost** for that **Customer Id**
- The last **Date** for that **Customer Id**

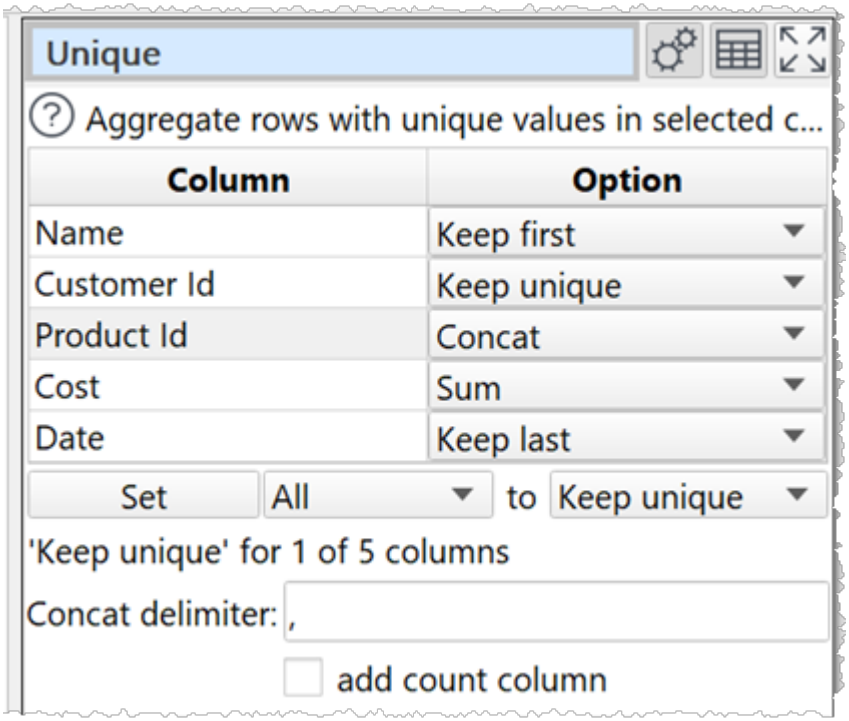
| | Name | Customer Id | Product Id | Cost | Date |
|---|---|---|---|---|---|
| 1 | Alice Anderson | C018930 | 13574 | 29.95 | 01/10/2020 |
| 2 | Bob Brown | C018917 | 89456,98526 | 20.55 | 02/10/2020 |
| 3 | Charlie Jones | C017783 | 96352,28746 | 44.95 | 03/10/2020 |

Drag the dataset file onto the **Center** pane of Easy Data Transform. Then click the **Unique** transform in the **Left** pane (if you can't see **Unique**, then check **show advanced** is checked in the **Left** pane).


*© 2023 Oryx Digital Ltd*




Then set the **Unique** options as shown below:




One aggregated row is created for each **Customer Id**. Use Sort if you want to change the order before the **Unique** transform.

If you only want to create a new aggregate row for rows with the same **Customer Id** and **Product Id**, set both the **Customer Id** and **Product Id** columns to **Keep unique**.

Note that **Unique** takes account of whitespace. So you might need a Whitespace transform before the **Unique** transform.

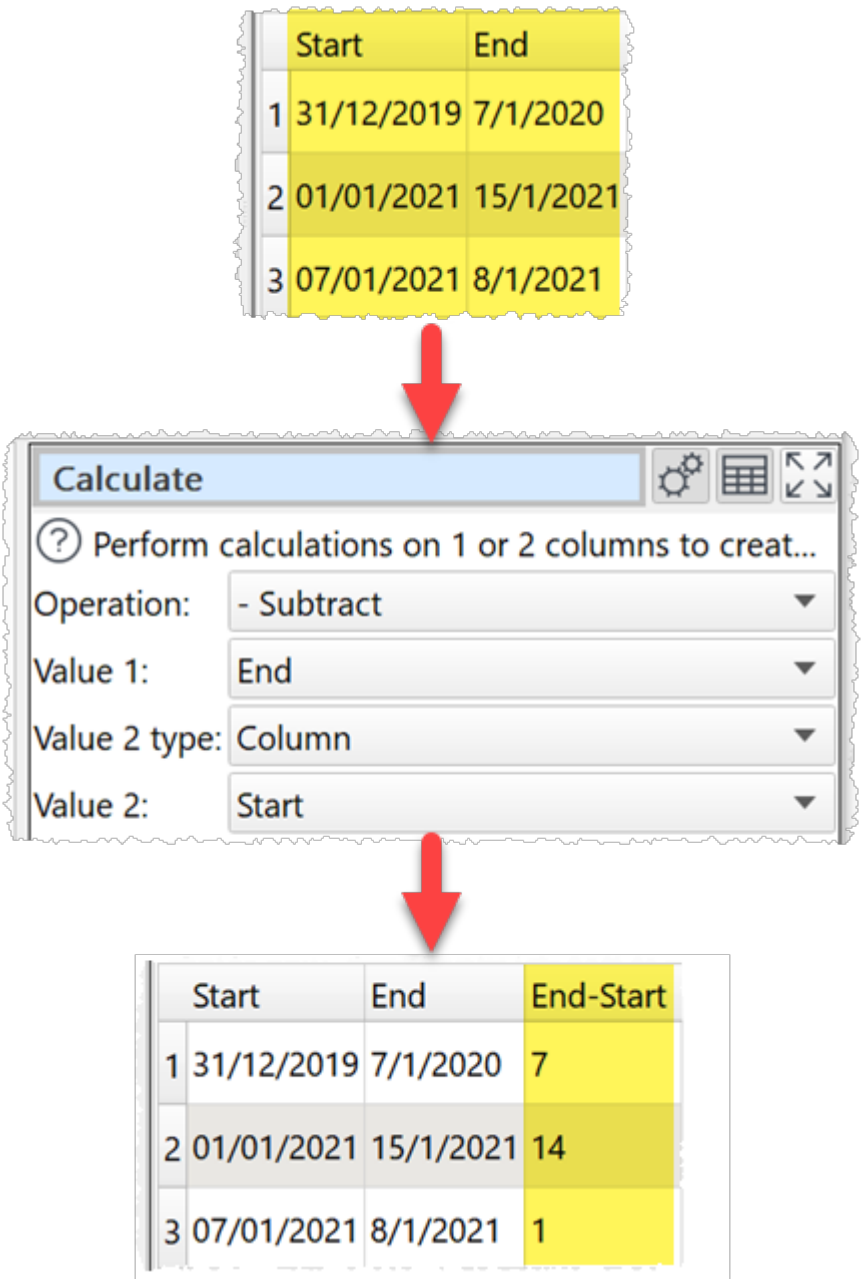
See also the Unique documentation.

See also:
- Video: How to remove Excel duplicates
- Video: How to merge rows in Excel


*© 2023 Oryx Digital Ltd*

## 3.11 Find the difference between dates/datetimes

You can calculate the number off days difference between two dates using the Calculate transform. For example:




You can convert a datetime into a date using a transform such as Extract.

Alternatively, you can calculate the difference between two dates or datetimes using Date objects in the Javascript transform.


© 2023 Oryx Digital Ltd

There are 4 ways to create a Javascript Date object:

| Date format | Description |
|---|---|
| `new Date(year, month, day, hours, minutes, seconds, milliseconds)` | Specified date and time specified as numeric parameters (January is month 0!). |
| `new Date(text date)` | Date and time specified as text. |
| `new Date(milliseconds)` | Milliseconds after 1st January 1970. |
| `new Date()` | Current date and time. |

Notes:
- A text date should be in `yyyy-mm-dd` format.
- A Date object always includes a time. If no time is set, then the time is assumed to be midnight GMT.
- One and two digit years will be interpreted from 1900.

## Examples

To calculate the number of milliseconds between a date in the 'date' column and 31st Dec 2000:

```
return new Date( $(date) ) - new Date( "2000-12-31" );
```

Or:

```
return new Date( $(date) ) - new Date( 2000, 11, 31 );
```

To calculate the difference between datetimes in the 'start' and 'end' columns in hours:

```
return ( new Date( $(end) ) - new Date( $(start) ) ) / ( 60 * 60 *
1000 );
```

To calculate how many days ago 'date' occurred (rounded down):

```
return Math.floor( ( new Date() - new Date( $(date) ) ) / ( 24 * 60 *
60 * 1000 ) );
```

For more information see the Javascript documentation.


© 2023 Oryx Digital Ltd

## 3.12 Handle column name/order changes in inputs

If you have a .transform file that you want to run multiple input files through (perhaps with a different input file each month, or as a batch process) you need to be aware of differences in column name and column order in the input files.

To change the file being used by an input, select the input item and change the file location in the **Right** pane (e,g, by clicking the '...' browse file button), rather the disconnecting the input and connecting a new one. Otherwise column-related parameters downstream will be reset.

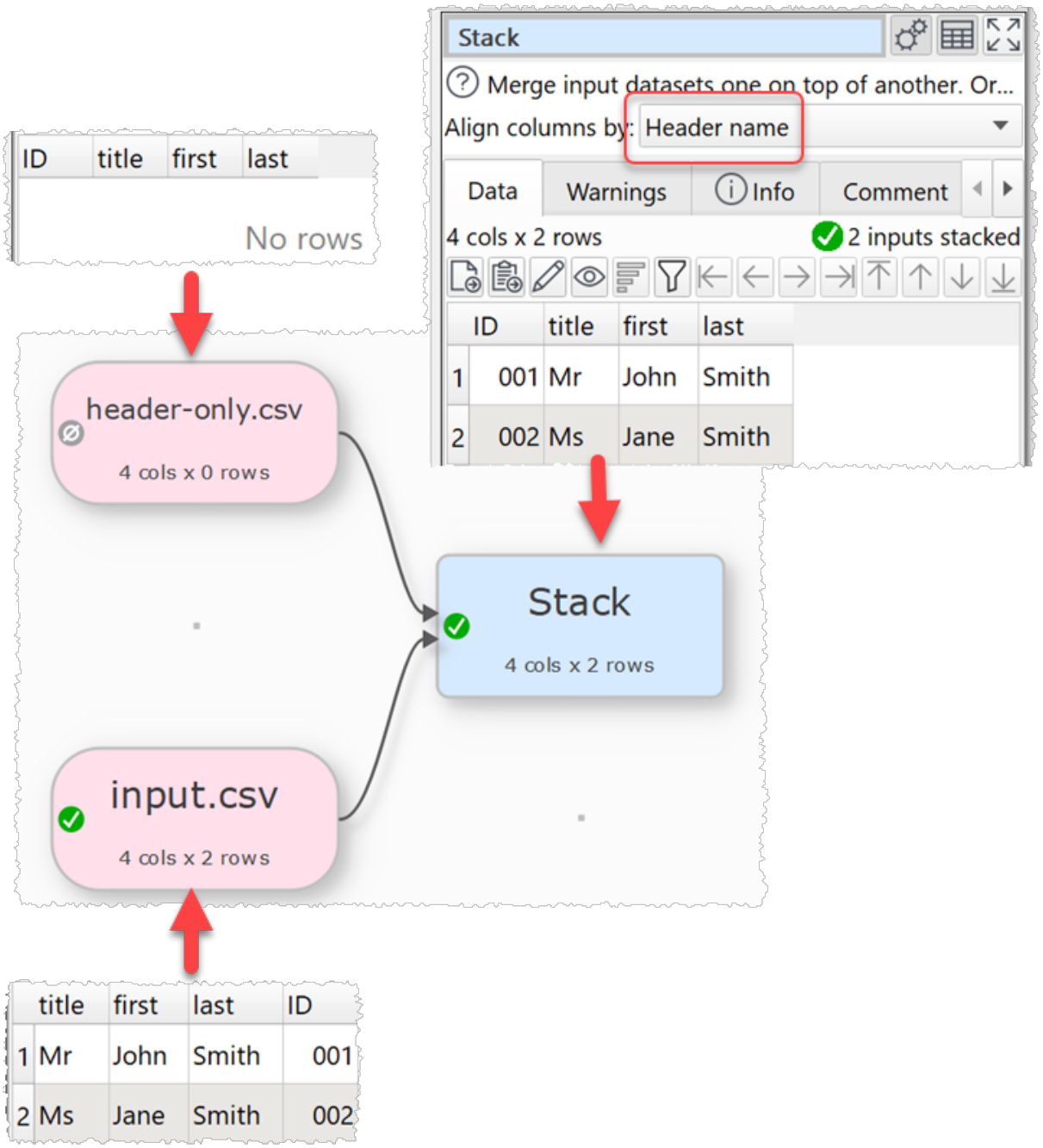
### Same columns in the same order, but with different names

Easy Data Transform references columns by their position (e.g. 3rd column from the left) not their column name. So differences in column names (e.g. first column is called "id" in input 1 and "UniqueID" in input 2) are not generally an issue. But you need to be careful if you are using the Stack transform with **Align columns by** set to **Header name**, as this will reorder columns by name. If you want to always output the same column names, regardless of the input column names, you should use a Rename Cols transform to set the names.

### Same columns with the same names, but in a different order

If columns are in different orders in different input files (e.g. the "ID" column in the first column in input 1 and the last column in input 2) you need to sort the input columns into a standard order before applying other transforms. You can so this using the Stack transform with **Align columns by** set to **Header name**. Stack your input under a dataset with just the columns in the correct order.


*© 2023 Oryx Digital Ltd*




## Same columns with different names, in a different order

Easy Data Transform can't handle this automatically. But you can create a new .transform and use Reorder Cols and/or Rename Cols transforms to output to a new file with the correct column names/ordering. You can then input this to the original .transform.

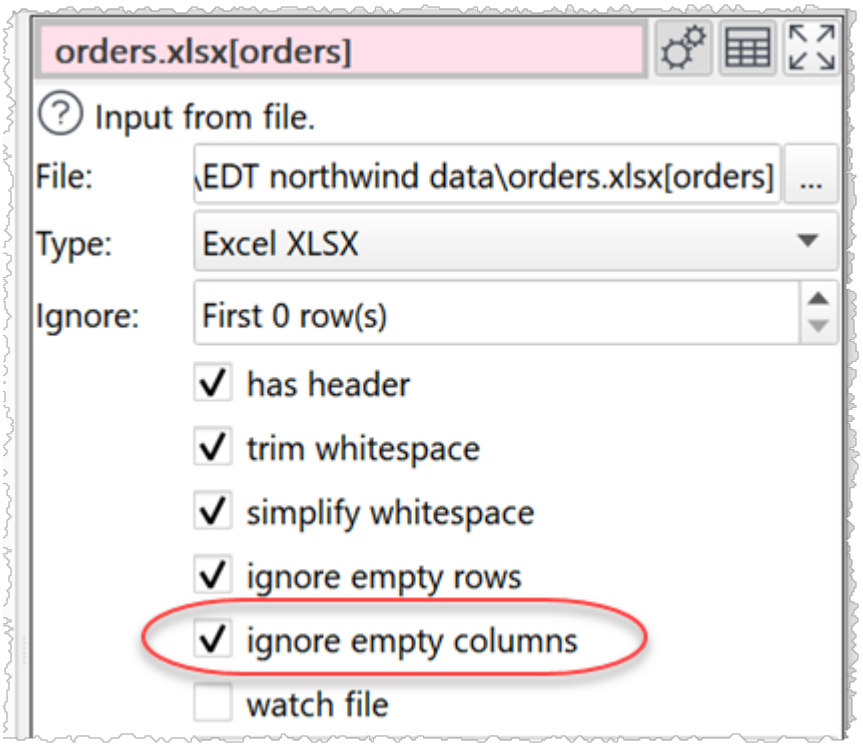

*© 2023 Oryx Digital Ltd*

## 3.13 Handle datasets with many columns

Easy Data Transform can help you to easily remove or rename columns in datasets with large numbers of columns.

### Remove multiple empty columns

If your dataset has a large number of empty columns you can remove them by checking **ignore empty columns** when you input it.



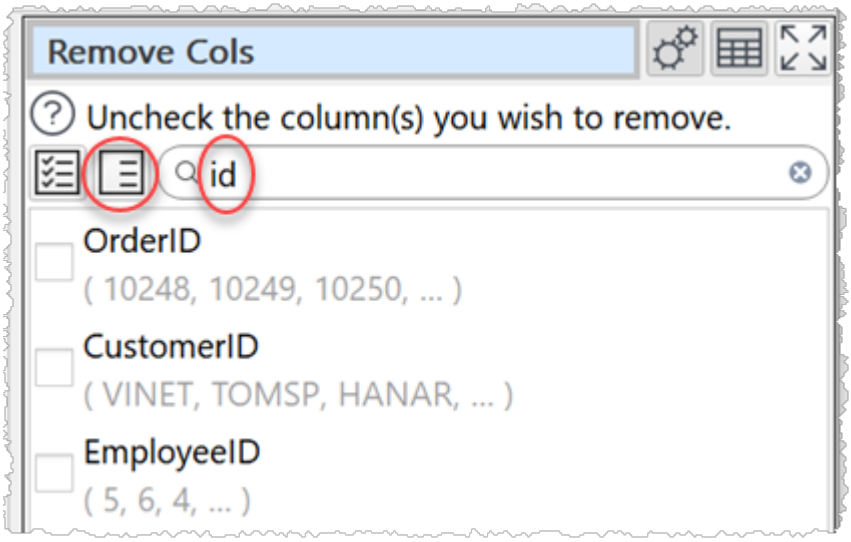

### Remove multiple columns

To quickly remove a large number of columns using the Remove Cols transform, you can do it by filtering column names. For example, to remove the columns with name that contain "id":

- Add a **Remove Cols** transform.
- Type `id` into the filter field. All the columns whose names contain "id" should now be visible (not case sensitive).
- Click the **Unselect all** button.
- Click the **Clear** button.

All the columns whose names contain "id" should now be removed by the transform. You can repeat this process as often as needed.

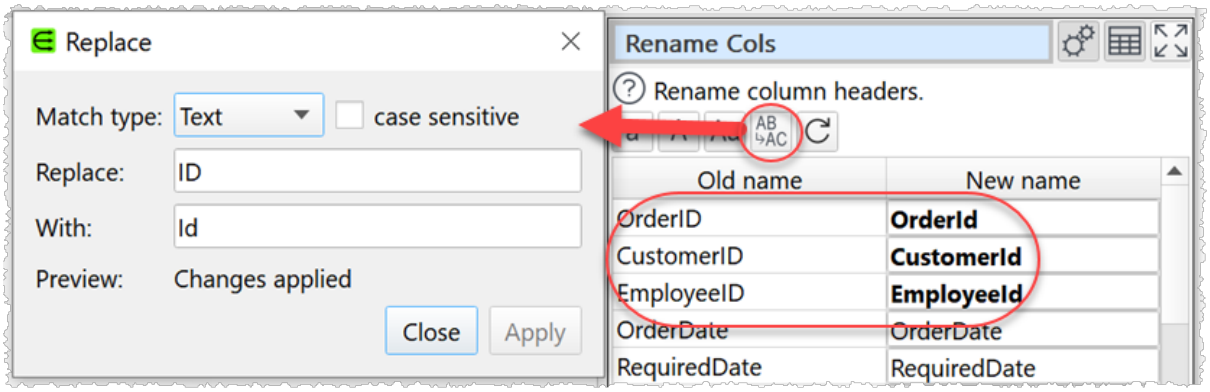

*© 2023 Oryx Digital Ltd*




### Rename multiple columns

To quickly rename a large number of columns using the Rename Cols transform, you can do it using **Replace text**:

- Add a **Rename Cols** transform.
- Click the **Replace text** button.
- Enter the text you want to **Replace** and the text you want to replace it **With**. You can match as **Text**, **Exact text** or **Regex**, with **case sensitive** matching optional.
- Click **Apply** to make the change.
- Click **Close**.




See also:

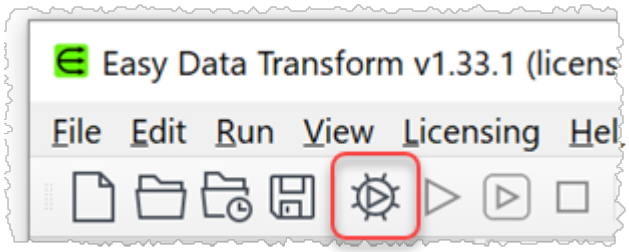
- Handle large datasets

## 3.14 Handle large datasets

Large datasets (e.g. with millions of rows) can slow down processing. If slow processing is a problem you can:


© 2023 Oryx Digital Ltd

- Add a sample transform straight after the input and set **Rows** to pass through only the first 100 or so rows. Once you have completed all your transforms you can then **Disable sampling** to pass through all rows.
- Set **Maximum memory usage** to a higher value in the Preferences window, to ensure you don't run out of memory.
- Set **Optimize processing for** to **Minimum memory use** in the Preferences window, to reduce memory usage.
- Uncheck **Run>Auto Run** to give yourself full control over processing.




or

- Check **Run>Auto Run** and:
  - Set **Right pane processing delay** in the Preferences window to a longer time (say 5 seconds) to ensure that changes aren't processed until you have finished making the changes.
  - Set **Write mode** to **Disabled** in output files, until you are ready to write them.

See also:

- Handle datasets with many columns

## 3.15 Input a fixed width format file

To input data from a fixed width file see fixed width format.

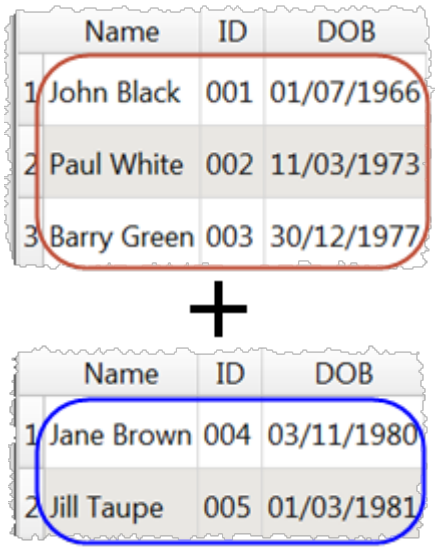
## 3.16 Merge datasets

Easy Data Transform has two main options for merging two datasets. Stack and Join.

### Stack datasets

If you want to merge the two datasets so they are one on top of another, use the Stack transform. For example, to Stack these two datasets:


© 2023 Oryx Digital Ltd

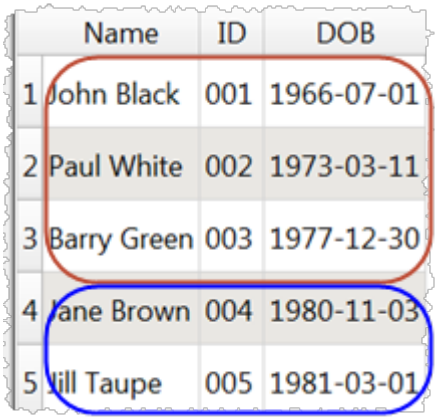


| | Name | ID | DOB |
|---|---|---|---|
| 1 | John Black | 001 | 01/07/1966 |
| 2 | Paul White | 002 | 11/03/1973 |
| 3 | Barry Green | 003 | 30/12/1977 |

+

| | Name | ID | DOB |
|---|---|---|---|
| 1 | Jane Brown | 004 | 03/11/1980 |
| 2 | Jill Taupe | 005 | 01/03/1981 |

To get this dataset:

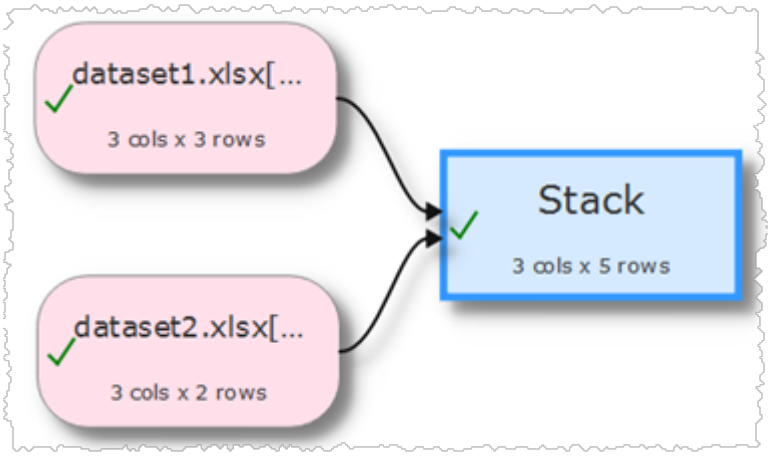


| | Name | ID | DOB |
|---|---|---|---|
| 1 | John Black | 001 | 1966-07-01 |
| 2 | Paul White | 002 | 1973-03-11 |
| 3 | Barry Green | 003 | 1977-12-30 |
| 4 | Jane Brown | 004 | 1980-11-03 |
| 5 | Jill Taupe | 005 | 1981-03-01 |

Drag the two dataset files onto the **Center** pane of Easy Data Transform.


*© 2023 Oryx Digital Ltd*

Select the two datasets using `Ctrl`+click then click the **Stack** transform in the **Left** pane.




The datasets are now stacked in the vertical order that the datasets are shown on the screen. The top dataset is shown first. You can swap the vertical positions of the datasets to change the order in which they are stacked.

If you want to stack column n of the first dataset above column n of the second dataset, set **Align columns by** to **Column number**.

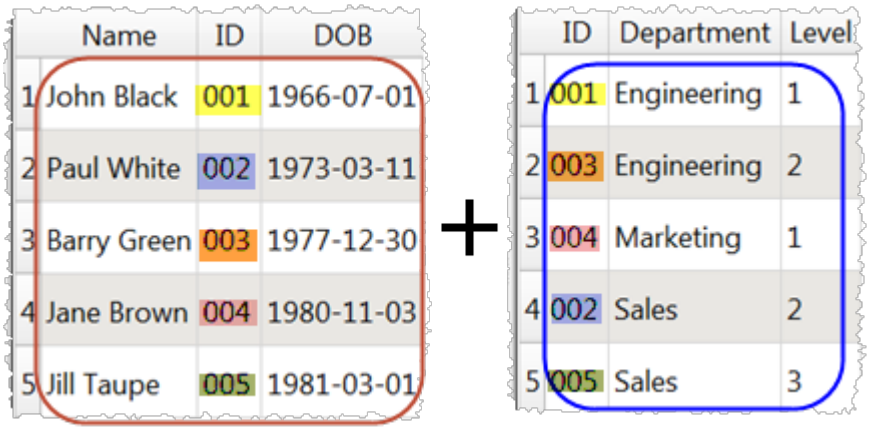
If you want to stack columns by common header names (even if they aren't in the same order), set **Align columns by** to **Header name**.

If you want to stack a large number of files you can do it by using batch processing to write to an output item with **Write Mode**=Append.


*© 2023 Oryx Digital Ltd*

## Join datasets

If you want to merge the two datasets side-by-side using a common ('key') column, use the Join transform. For example, to Join these two datasets:

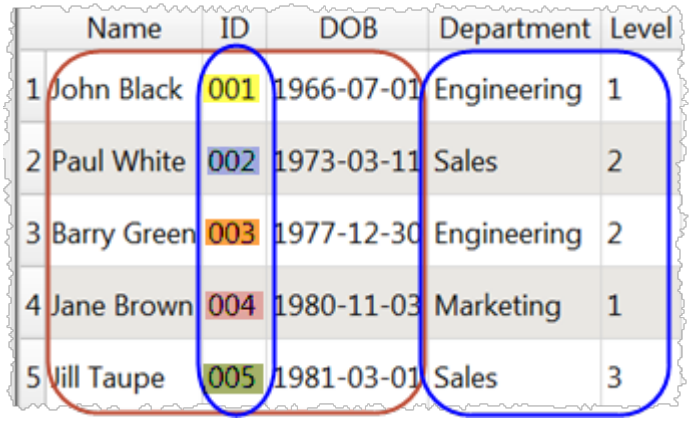


| | Name | ID | DOB |
|---|---|---|---|
| 1 | John Black | 001 | 1966-07-01 |
| 2 | Paul White | 002 | 1973-03-11 |
| 3 | Barry Green | 003 | 1977-12-30 |
| 4 | Jane Brown | 004 | 1980-11-03 |
| 5 | Jill Taupe | 005 | 1981-03-01 |

+

| | ID | Department | Level |
|---|---|---|---|
| 1 | 001 | Engineering | 1 |
| 2 | 003 | Engineering | 2 |
| 3 | 004 | Marketing | 1 |
| 4 | 002 | Sales | 2 |
| 5 | 005 | Sales | 3 |

By common ID value to get this dataset:

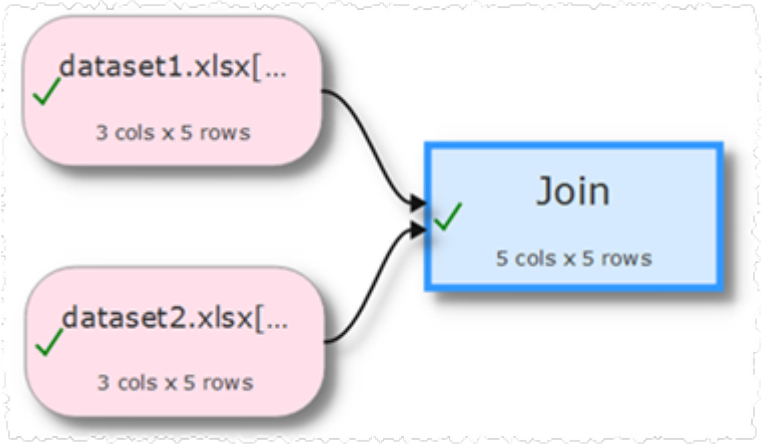


| | Name | ID | DOB | Department | Level |
|---|---|---|---|---|---|
| 1 | John Black | 001 | 1966-07-01 | Engineering | 1 |
| 2 | Paul White | 002 | 1973-03-11 | Sales | 2 |
| 3 | Barry Green | 003 | 1977-12-30 | Engineering | 2 |
| 4 | Jane Brown | 004 | 1980-11-03 | Marketing | 1 |
| 5 | Jill Taupe | 005 | 1981-03-01 | Sales | 3 |

Drag the two dataset files onto the **Center** pane of Easy Data Transform.


*© 2023 Oryx Digital Ltd*

Select the two datasets using `Ctrl`+click then click the **Join** transform in the **Left** pane.




Set both **Top key column** and **Bottom key column** to the common ('key') column.

The datasets are now joined side-by-side using the common column. The top dataset is shown on the left. You can swap the vertical positions of the datasets to change the order in which they are joined.

If you just want to join row N of one dataset to row N of another dataset, you can use the Row Num transform to create a common column in each dataset.

Set **include top non-matching rows** and **include bottom non-matching rows** depending on what you want to do with top and bottom dataset rows for which there are no matches.


*© 2023 Oryx Digital Ltd*

Note that matching columns takes account of whitespace. So you might need to do Whitespace transform before the join.

If you are merging numerical datasets you can also use an Interpolate transform.

See also:
- Video: How to join Excel files

## 3.17 Move a .transform file

To move a .transform file to a different location on the same computer use **File>Save As...** or **Windows Explorer**. You either leave the Input files at the original location or move them to the same location relative to the .transform file (e.g. if they were in the same folder as the .transform file before, move them to the same folder as new .transform file).

To move a .transform file to a different computer, move the Input files to the same location relative to the .transform file (e.g. if they were in the same folder as the .transform file before, move them to the same folder as new .transform file).

See also .transform files.

## 3.18 Open multiple main windows

To open an additional Easy Data Transform **Main** window, select **File>New Window**.

## 3.19 Output nested JSON or XML

You can use the dot ('.') character in the column header to show nesting. For example:

| | name | carb | cholesterol | fiber | minerals.ca | minerals.fe | protein | sodium | vitamins.a | vitamins.c |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Avocado Dip | 2 | 5 | 0 | 0 | 0 | 1 | 210 | 0 | 0 |

Is output to JSON as:

```
[
  {
    "name": "Avocado Dip",
    "carb": "2",
    "cholesterol": "5",
    "fiber": "0",
    "minerals": {
      "ca": "0",
      "fe": "0"
    },
    "protein": "1",
    "sodium": "210",
    "vitamins": {
      "a": "0",
```


© 2023 Oryx Digital Ltd

```
      "c": "0"
    }
  }
]
```

And to XML as:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <record>
    <name>Avocado Dip</name>
    <carb>2</carb>
    <cholesterol>5</cholesterol>
    <fiber>0</fiber>
    <protein>1</protein>
    <sodium>210</sodium>
    <minerals>
      <ca>0</ca>
      <fe>0</fe>
    </minerals>
    <vitamins>
      <a>0</a>
      <c>0</c>
    </vitamins>
  </record>
</root>
```

For more details see:

- JSON format
- XML format

See also:

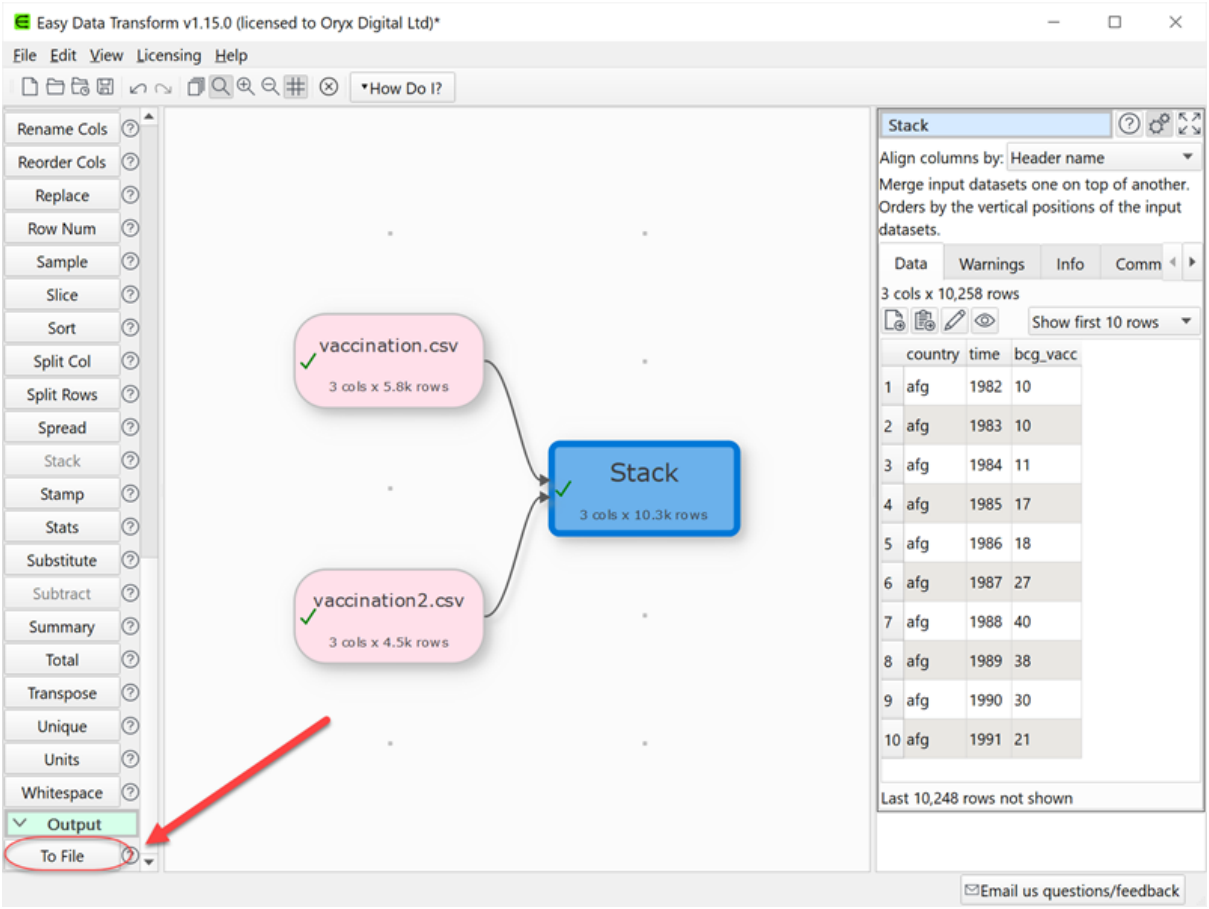
- Video: How to convert CSV to XML
- Video: How to convert CSV to JSON

## 3.20 Output to Excel

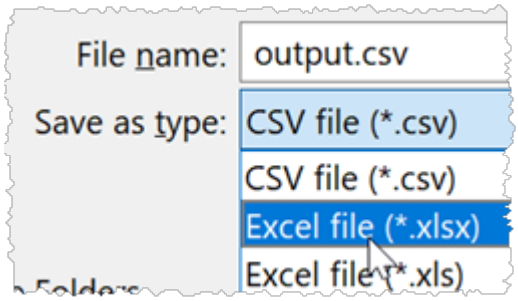
To output results from a transform to an Excel .xlsx/.xls file:

- Select the transform item in the **Center** pane.
- Click **To File** at the bottom of the **Left** pane.


© 2023 Oryx Digital Ltd




- Select ***.xlsx** or ***.xls** from the file type drop-down list that appears.




Note that Excel .xlsx files are typically limited to 1,048,576 rows and 16,384 columns.

See also:

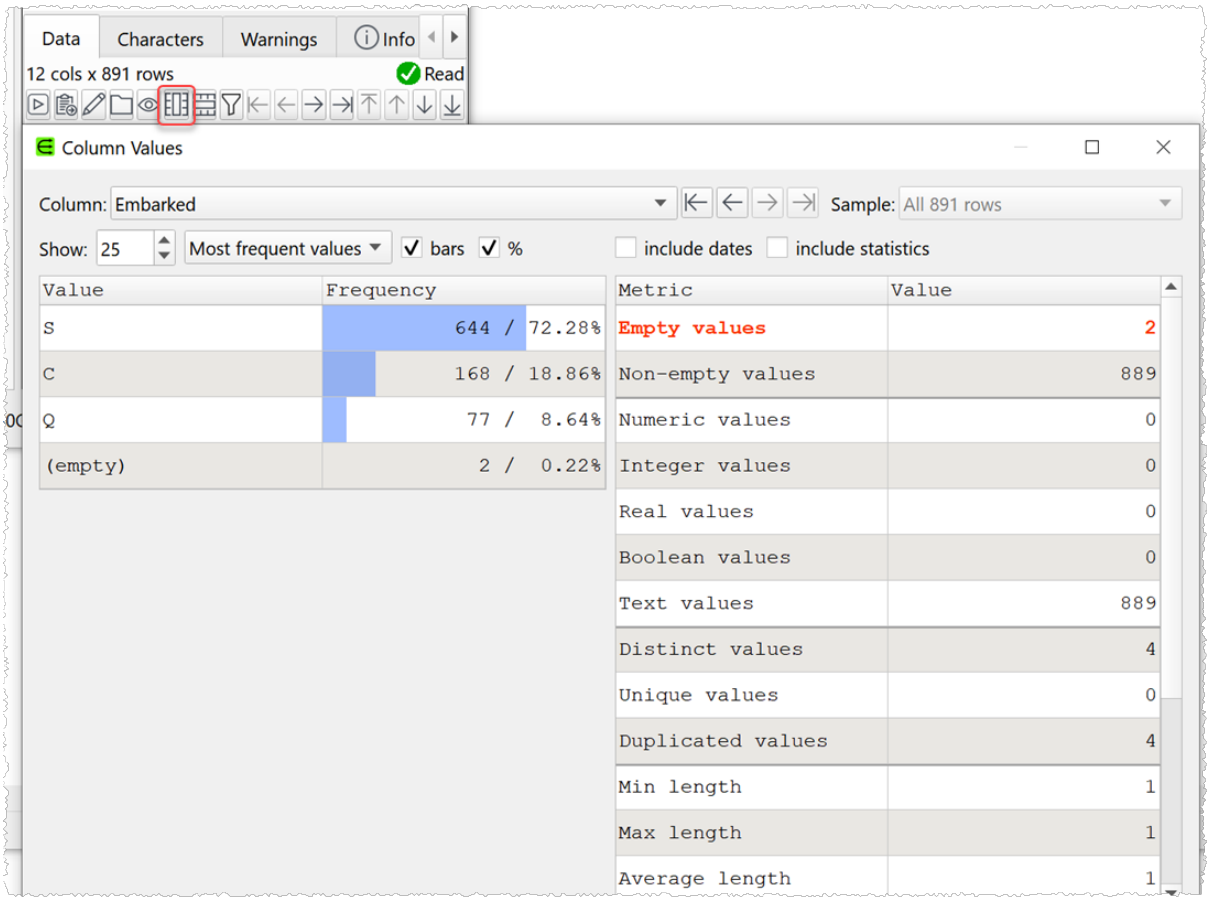
- Write to multiple sheets of an Excel file
- Video: How to convert fixed column width to CSV or Excel
- Video: How to convert JSON to Excel
- Video: How to convert XML to Excel


*© 2023 Oryx Digital Ltd*

## 3.21 Profile a dataset

In the **Right** pane you can double click on a column header or click the corresponding button to show a profile of the data in each column. For example the different data types (text, numeric etc) and the frequencies of different values. Potential discrepancies (such as missing values) are highlighted in red.

You can use profiling in conjunction with Filter, Remove Cols and Replace transforms to remove unwanted rows, columns or values; and with the Impute transform to add missing values.




For large datasets you might want to set **Sample** to less than the full dataset for speed. You can also leave **include dates** and **include statistics** unchecked for speed (**include dates** checks against every date format listed in **Preferences**).

See also:

- Video: How to profile data
- Summary
- Add missing data values
- Clean a dataset


*© 2023 Oryx Digital Ltd*

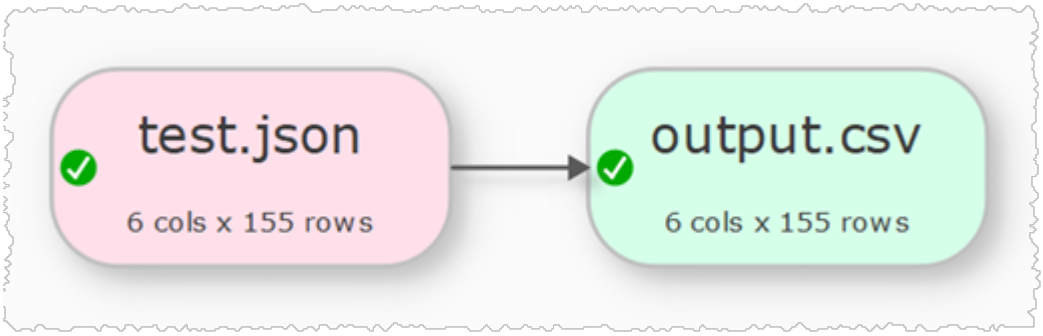
## 3.22 Perform the same transforms on many files

You can perform the same set of transforms on multiple inputs in one operation using batch processing or command line arguments.

### Example 1

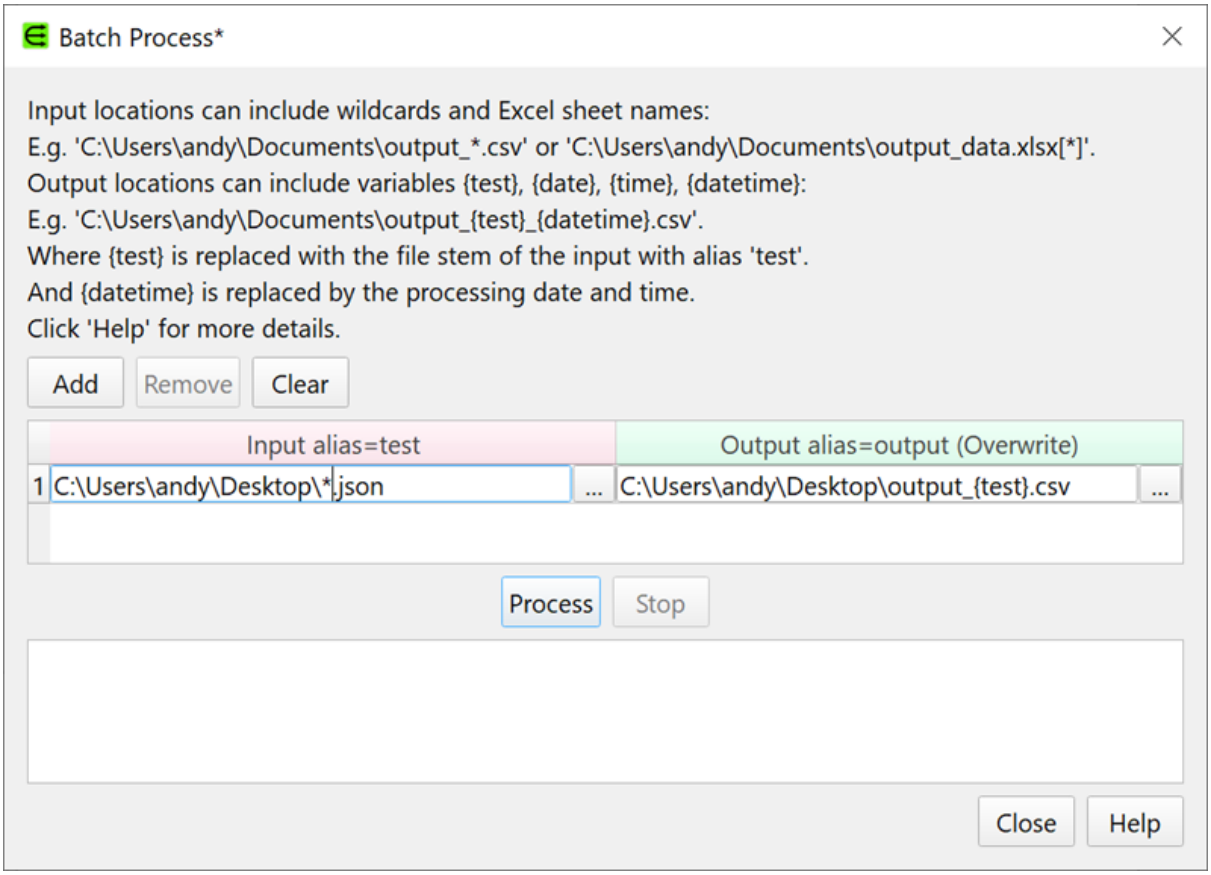
To convert a folder full of .json files to .csv files:

1. Select **File>New** to create a new `.transform` file.
2. Drag one of the .json files onto the **Center pane**. Ensure the options (encoding etc) are correct in the **Right** pane.
3. Click on the **To File** button at the bottom of the **Left** pane and set the location of a .json file to create. Ensure the options (encoding etc) are correct in the **Right** pane.




4. Select **File>Batch Process**.
5. In the **Batch Process** window change the `.json` file name to `*.json`, so that all the `.json` files in that folder will be processed.


© 2023 Oryx Digital Ltd




6. Press the **Process** button. A `.csv` file will now be created for each `.json` file in the folder. In the example above the `{test}` file name variable will be replaced by the file name of the input file with alias `test` (if you just output to `output.csv` then you would be continually overwriting the same file).
7. Select **File>Save** to save your `.transform` for future use.

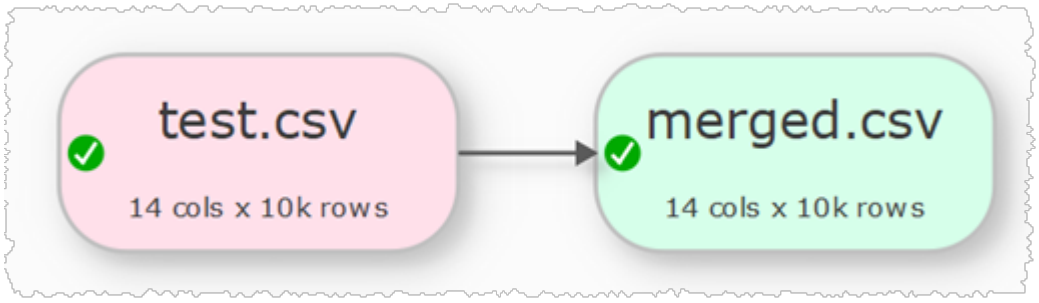
If you want to process input files from another folder then click **Add** to add a new row and change the input folder.

## Example 2

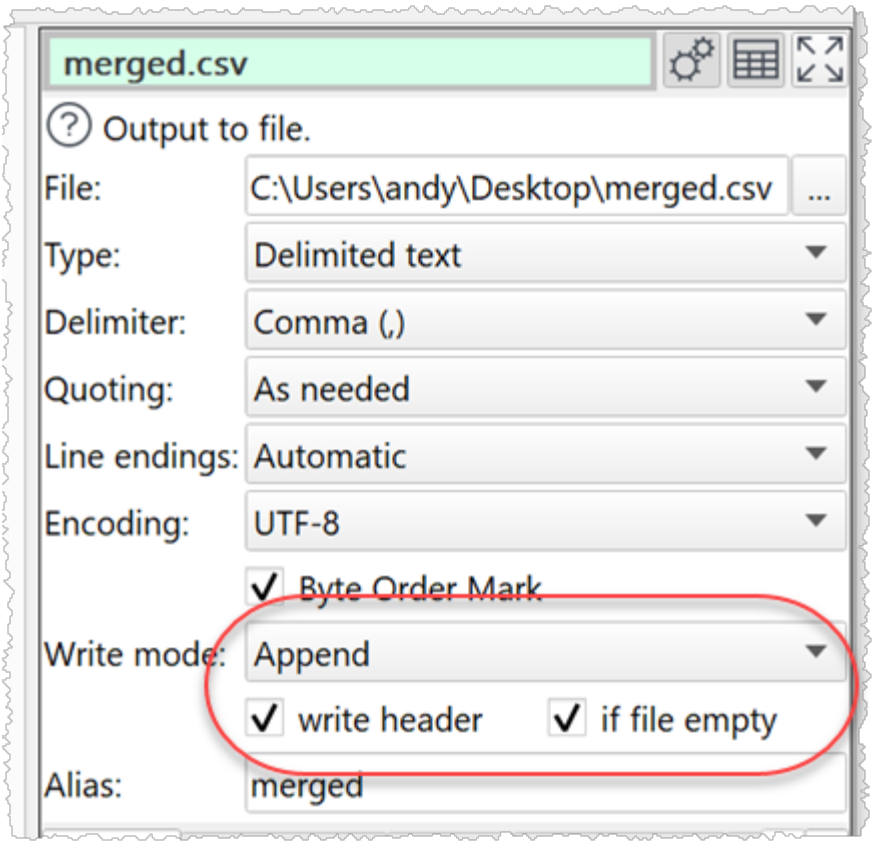
To merge multiple .csv files into a single .csv file:

8. Select **File>New** to create a new `.transform` file.
9. Drag one of the .csv files onto the **Center pane**. Ensure the options (encoding etc) are correct in the **Right** pane.
10. Click on the **To File** button at the bottom of the **Left** pane and set the location of a `merged.csv` file to create, in a different folder to the input .csv files. Ensure the options (encoding etc) are correct.


*© 2023 Oryx Digital Ltd*



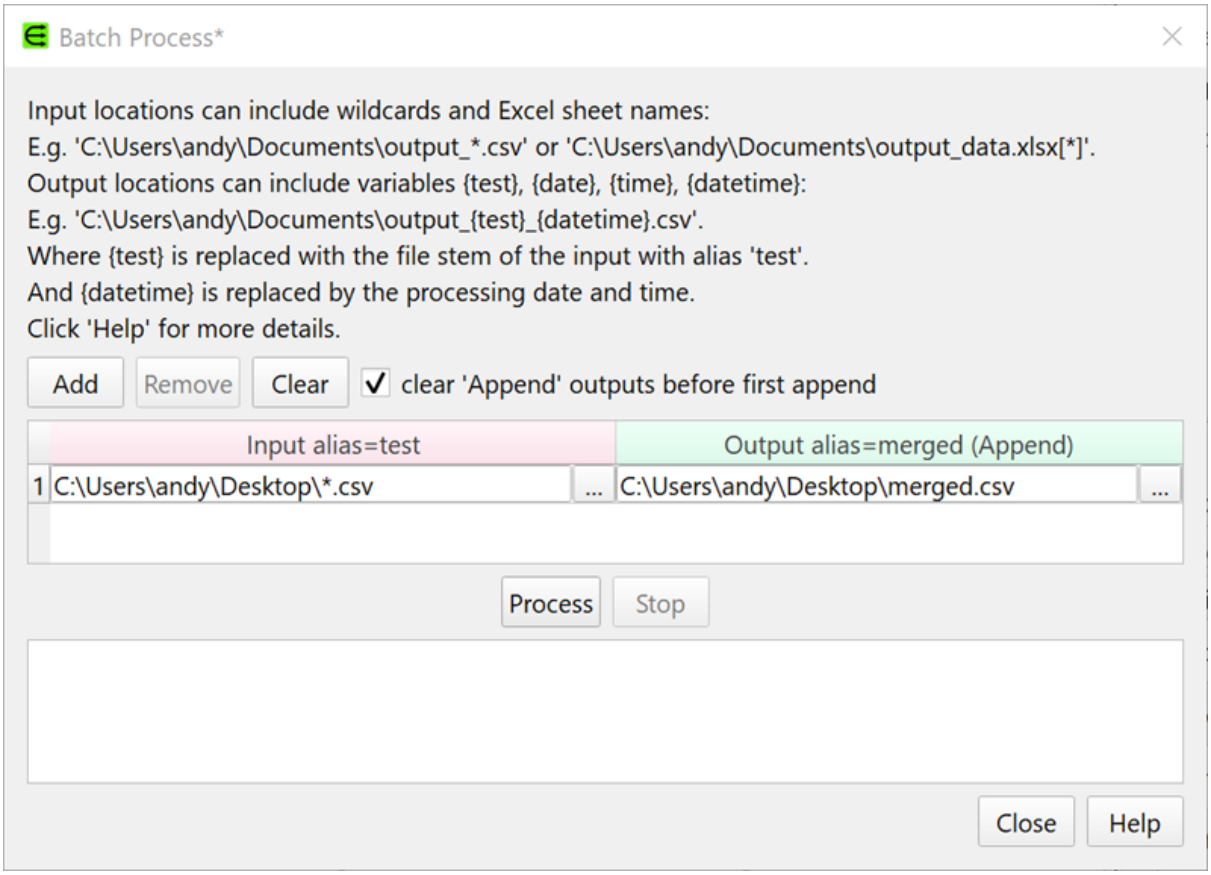

4. In the **Right** pane set **Write Mode** to **Append** and check **write header** and **if empty**. This ensures that only one header will be written, rather than one for every input file.




5. Select **File>Batch Process**.
6. In the **Batch Process** window:

- change the input .csv file name to `*.csv`, so that all the `.csv` files in that folder will be processed.
- Check **clear 'Append' outputs before first append** so that any existing data in the output file is cleared before we start appending to it.

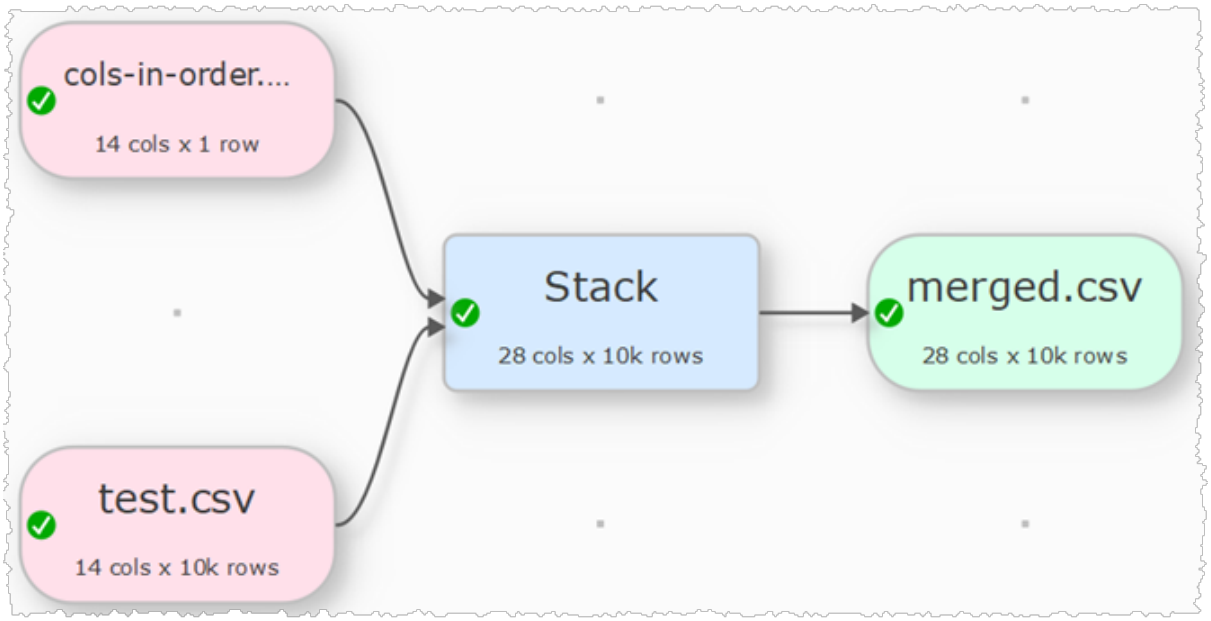

*© 2023 Oryx Digital Ltd*

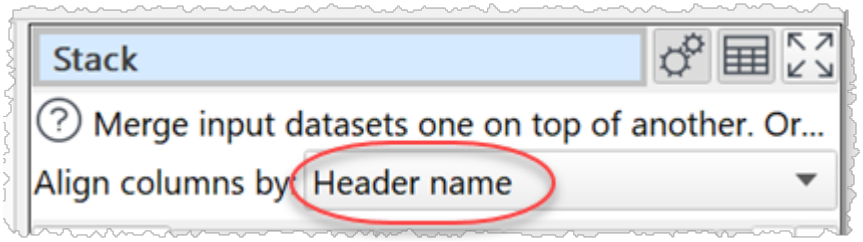



7. Press the **Process** button. A single `merged.csv` file will now be created that contains a concatenation of all the other `.csv` files.
8. Select **File>Save** to save your `.transform` for future use.

If the headers are different orders in different .csv files, then you can Stack by header name in your `.transform` to get a consistent column order before outputting.


*© 2023 Oryx Digital Ltd*



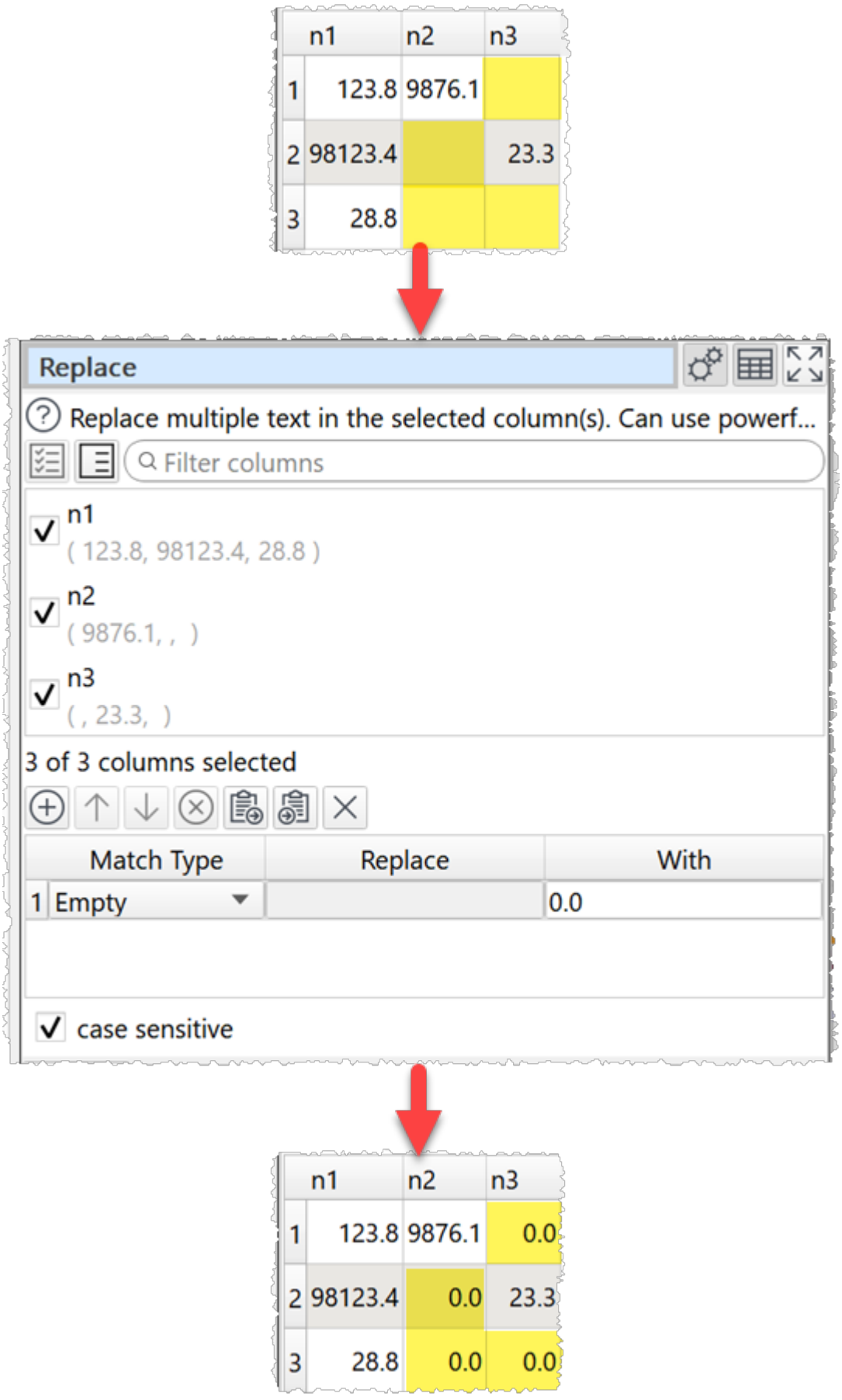

See also:

- Video: Batch processing

## 3.23 Replace empty values

You can replace empty values using the Replace transform.


© 2023 Oryx Digital Ltd





© 2023 Oryx Digital Ltd

See also:

- Add missing data values

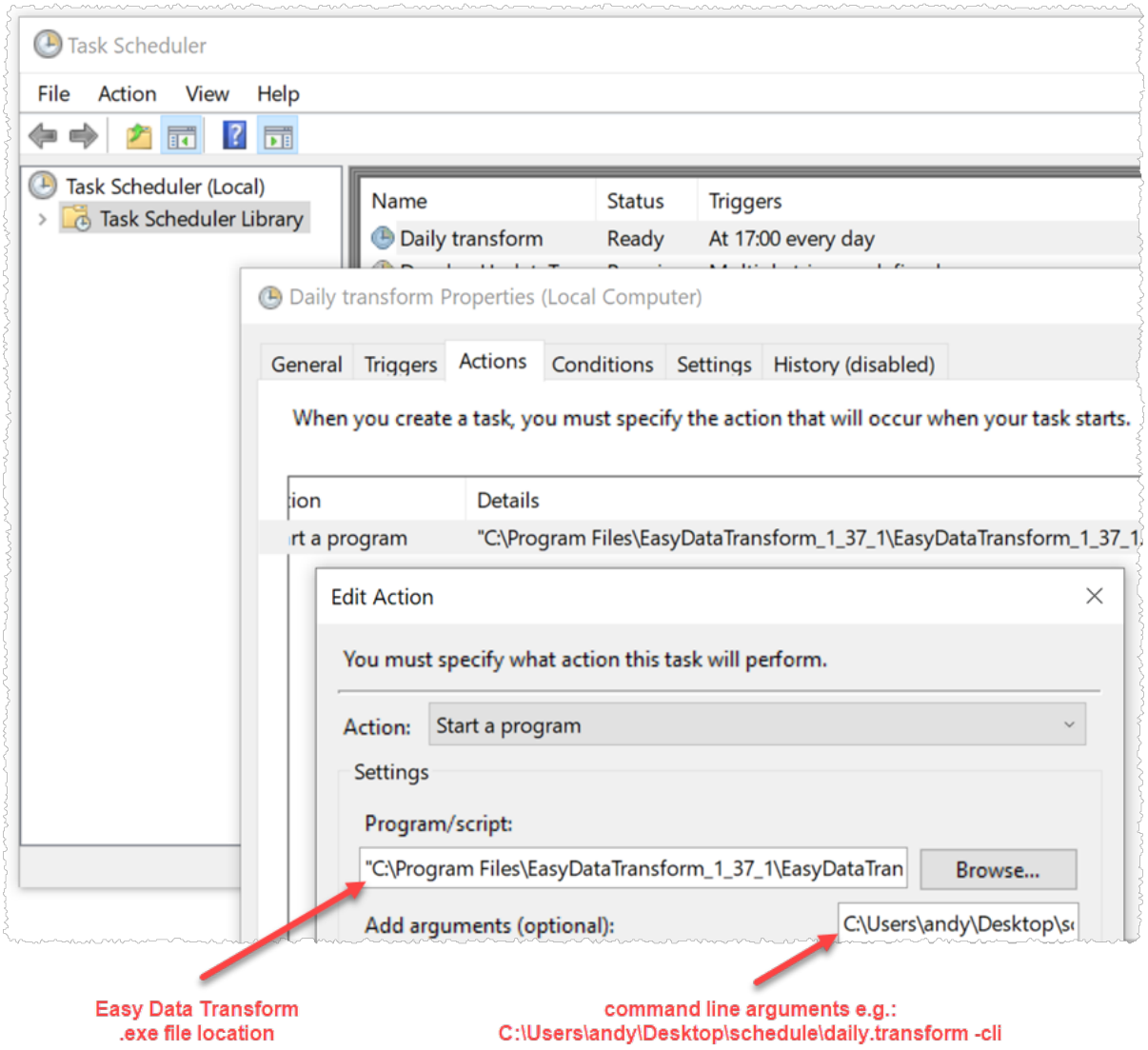
## 3.24 Run jobs on a schedule

Easy Data Transform doesn't currently have it's own scheduler. However you can run jobs on a schedule (e.g. at 5pm every day) by using pretty much any scheduler to call the Easy Data Transform command line interface:

1. Create the *.transform* file that details the transformations you want to carry out.
2. Call Easy Data Transform command line interface (or a script that calls the command line interface) from your scheduler.

You can use Task Scheduler, which is part of Windows. Or you can use any number of third party scheduler applications .


*© 2023 Oryx Digital Ltd*




You can change the input and output file names through the command line interface. Add `-cli` to close Easy Data Transform after the job is finished.

Some schedulers will also allow you to call Easy Data Transform on an event, e.g. when a file is added to or updated in a particular folder.

## 3.25 Scrape web data

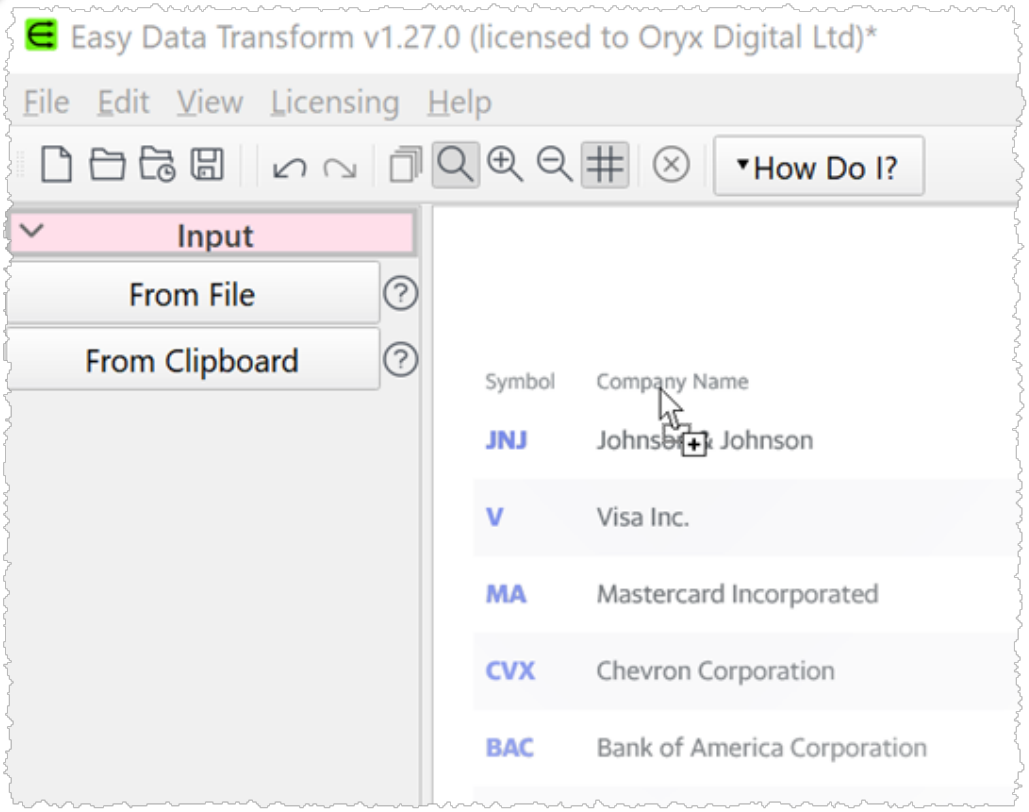
You can scrape data from a web browser by selecting it and dragging into the **Center** pane of Easy Data Transform (or copying and clicking **From Clipboard**).


*© 2023 Oryx Digital Ltd*

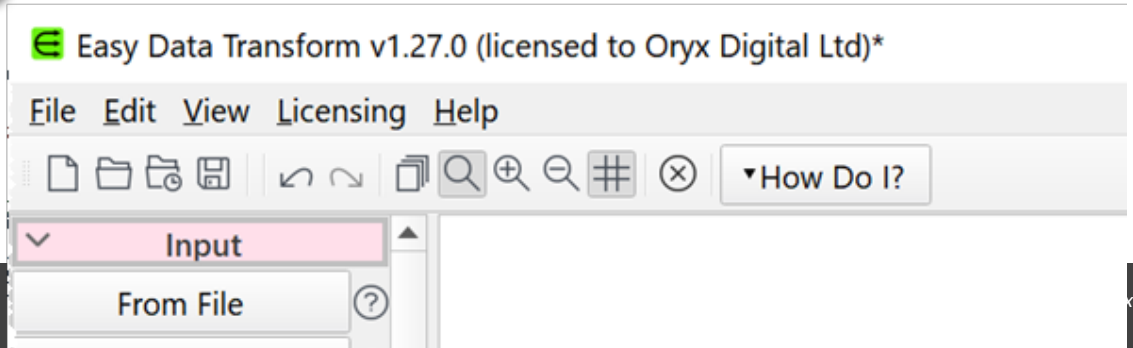
## 1 Select data in web a browser

| Symbol | Company Name | Last Price | Change | % Change |
|---|---|---|---|---|
| JNJ | Johnson & Johnson | 181.54 | -1.82 | -0.99% |
| V | Visa Inc. | 208.17 | -8.28 | -3.83% |
| MA | Mastercard Incorporated | 351.18 | -13.26 | -3.64% |
| CVX | Chevron Corporation | 160.95 | -3.63 | -2.21% |

## 2 Drag onto Easy Data Transform




## 3 Drop



See also:

- Video: How to scrape web data

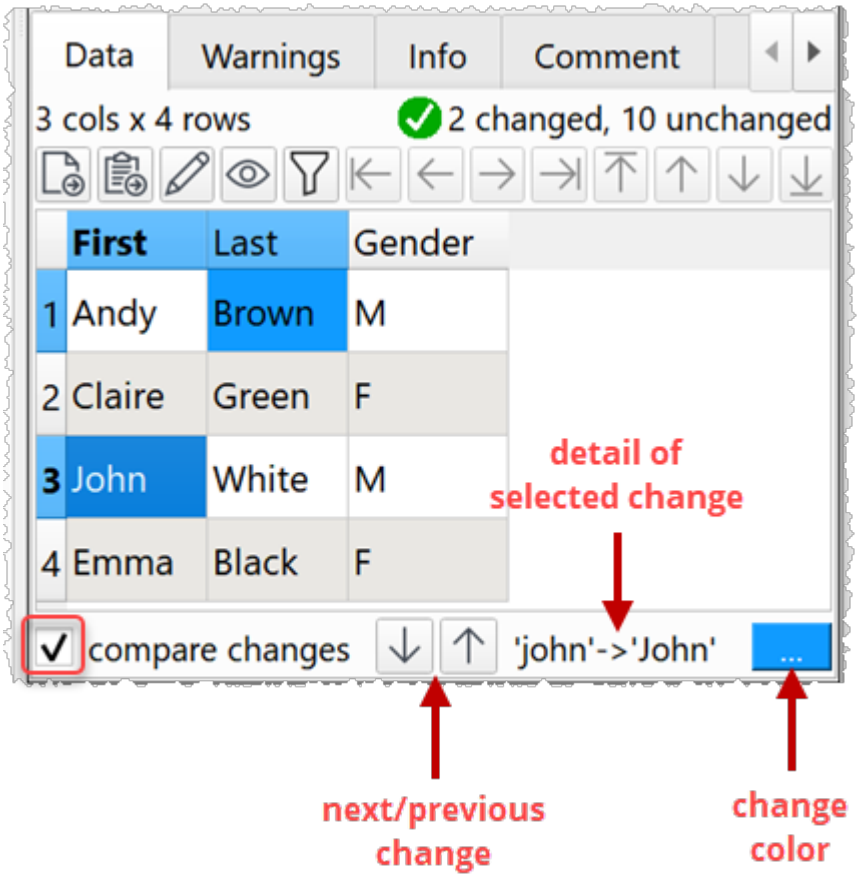
## 3.26 See changes from a transform

The following transforms allow you to highlight changes made by the transform:

- Case
- Chop
- DateTime Format
- Decode
- Extract
- Fill
- Hash
- Impute
- Insert
- Number Format
- Offset
- Pad
- Replace
- Scale
- Unfill
- Units
- Whitespace

Check the **compare changes** checkbox below the data table to highlight changes made by the transform.


*© 2023 Oryx Digital Ltd*

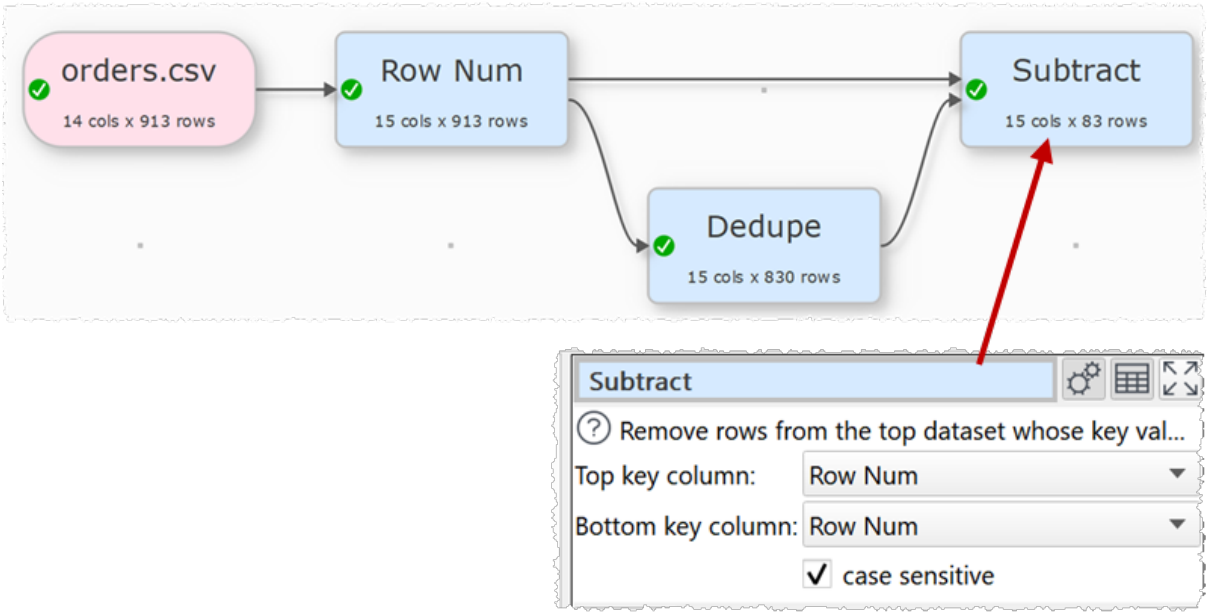



Click on or hover over a cell to see details of each change.

Click on the up and down arrows to move between changes. Changes hidden by dataset filtering are skipped.

If you want to see rows that were removed by a transform such as Dedupe, Unique or Filter, you can do that using Subtract. The dataset needs to have a column of unique key values to use in the **Subtract**. If it doesn't have one, then add one first using Row Num.

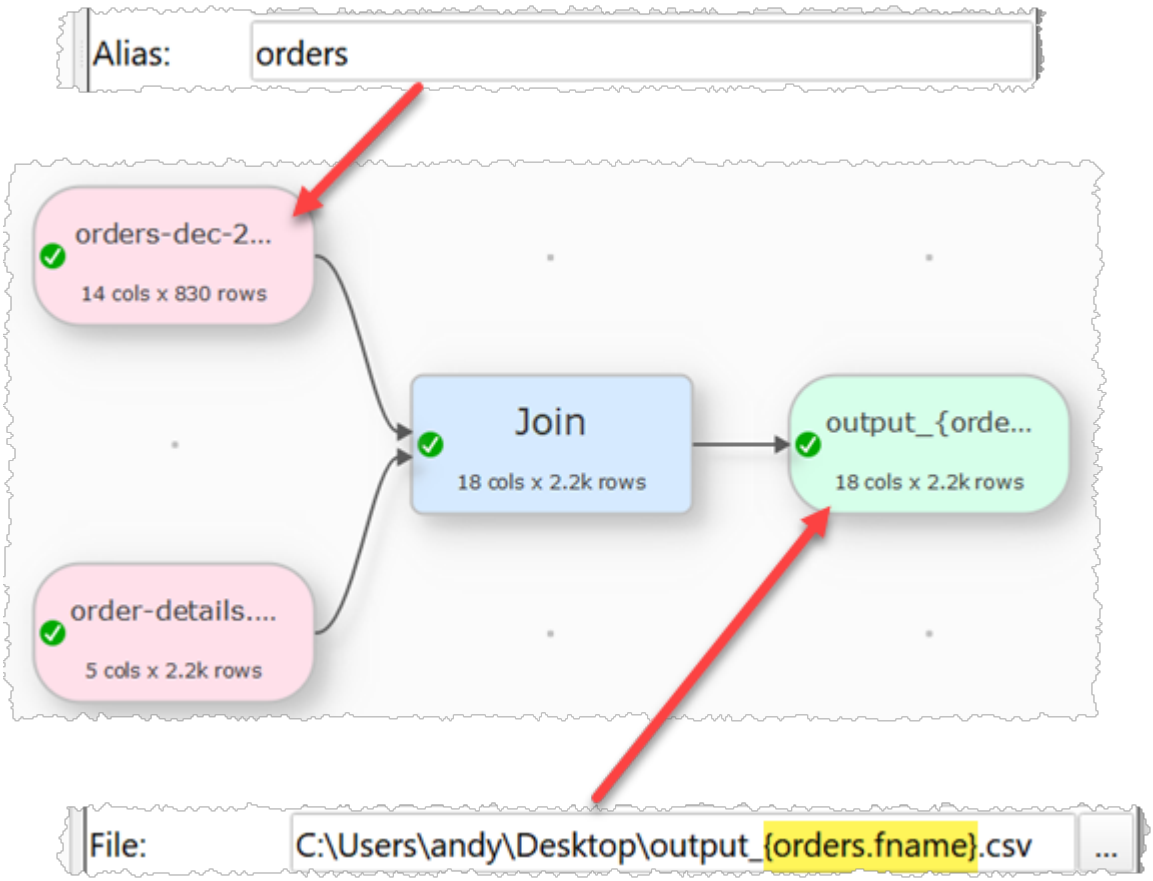

*© 2023 Oryx Digital Ltd*




## 3.27 Set output file name from input file

To set an output file name based on the name of one or more input files, see File name variables.

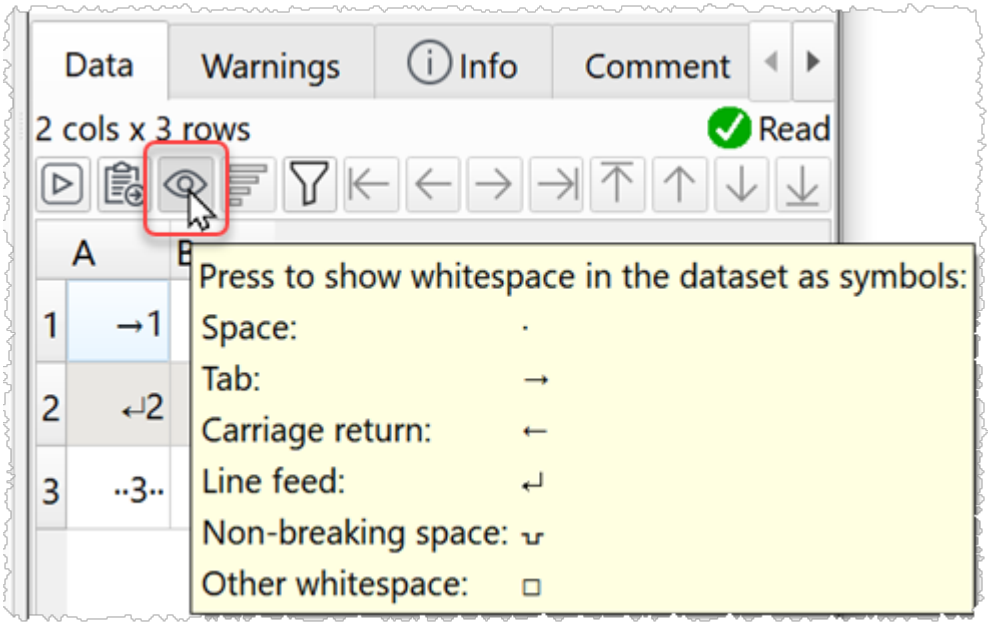

© 2023 Oryx Digital Ltd




## 3.28 Show whitespace

Click the eye icon in the **Right** pane to show whitespace in the data.


*© 2023 Oryx Digital Ltd*



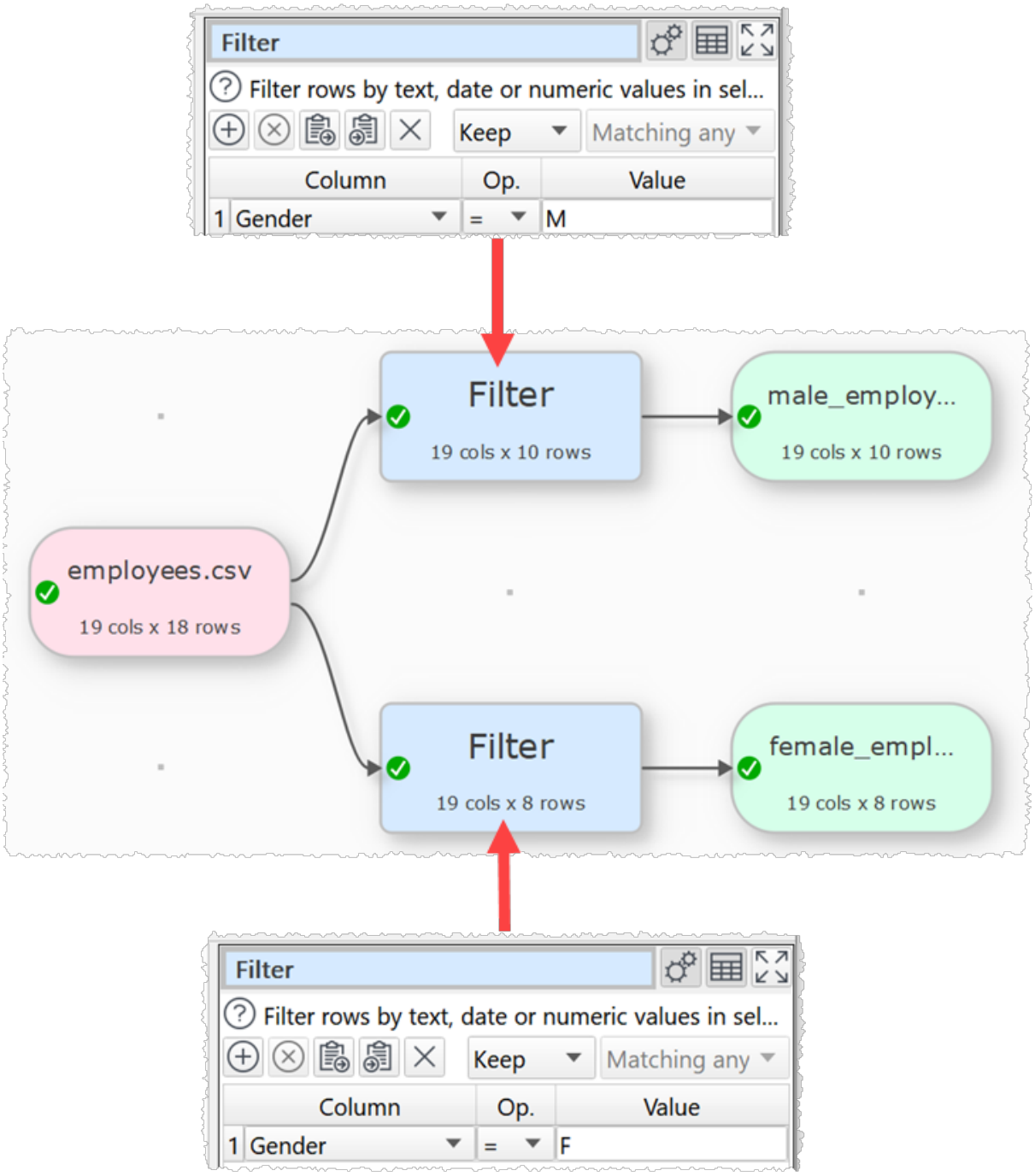

## 3.29 Split a dataset into multiple files

Easy Data Transform supports splitting datasets into multiple files in 2 different ways.

### Example 1: Simple splitting

To split a dataset according to whether the `Gender` column contains `M` or `F`, use a pair of Filter transforms.


*© 2023 Oryx Digital Ltd*




## Example 2: Split according to row values

To split a dataset according to `Region` column values:

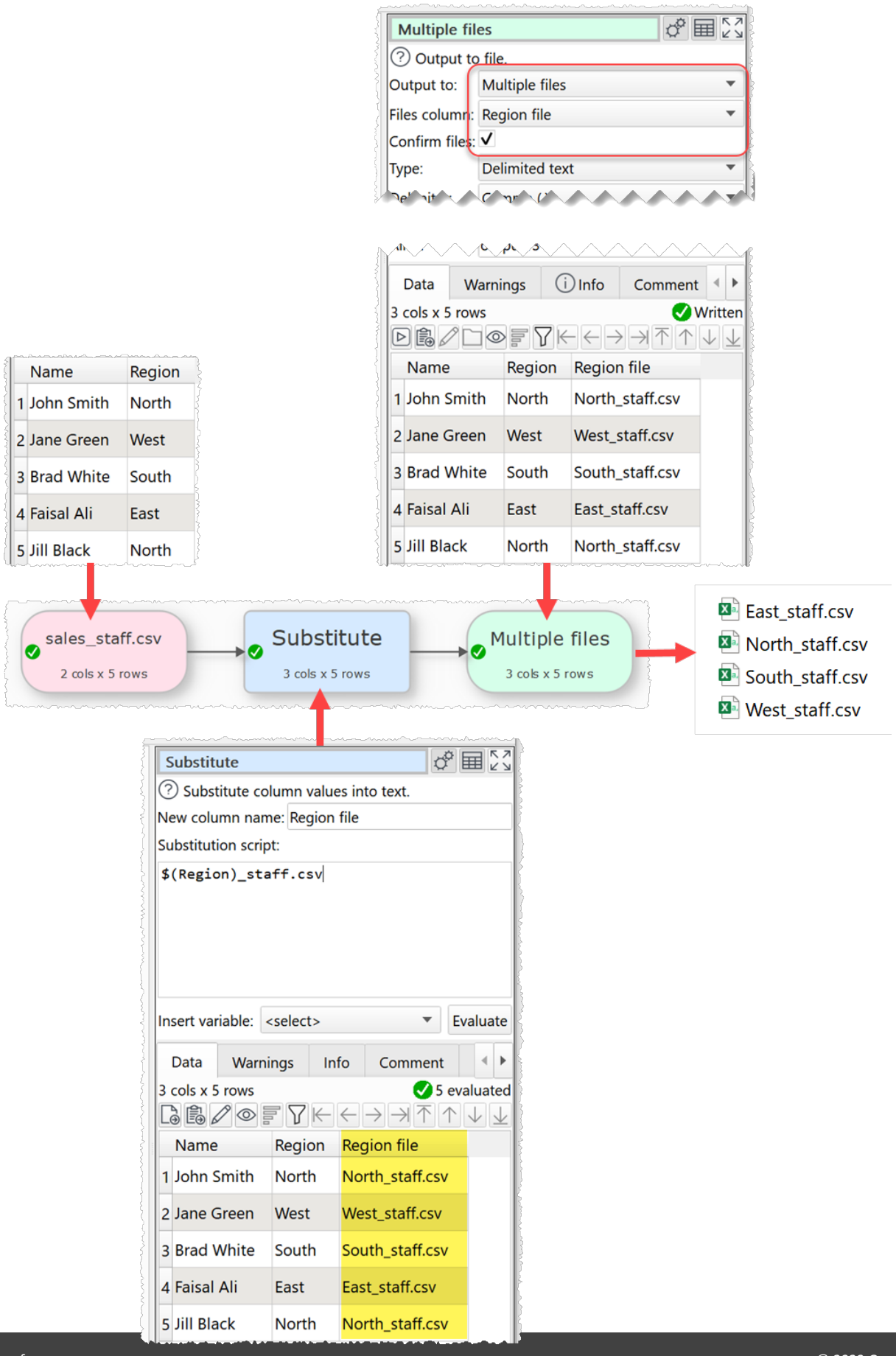
- Use transforms (such as Substitute) to create a column with the file location you want to output to, based on the `Region` column. The location can be an absolute path (e.g. `c:\users\andy\output.csv`) or a path relative to the .transform file location (e.g.


*© 2023 Oryx Digital Ltd*

`results\output.csv`). Folders output to must already exist. Empty or invalid values are ignored. This column is not output (use the **Copy Cols** transform if you want it to be).

- Create an output with **Output to** set to **Multiple files** and the **Files column** selected.
- Set **Confirm files** checked to be prompted before writing (recommended).


*© 2023 Oryx Digital Ltd*

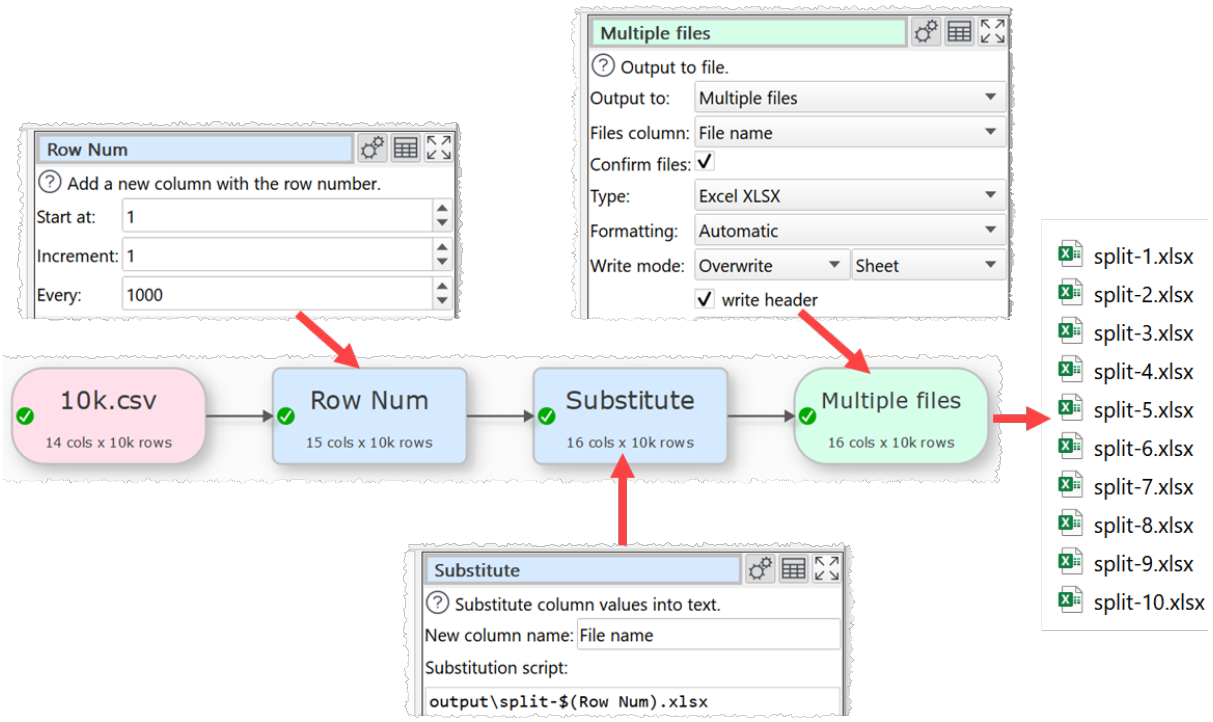




© 2023 Oryx Digital Ltd

### Example 3: Split by size

To split a dataset into files of 1000 rows each:



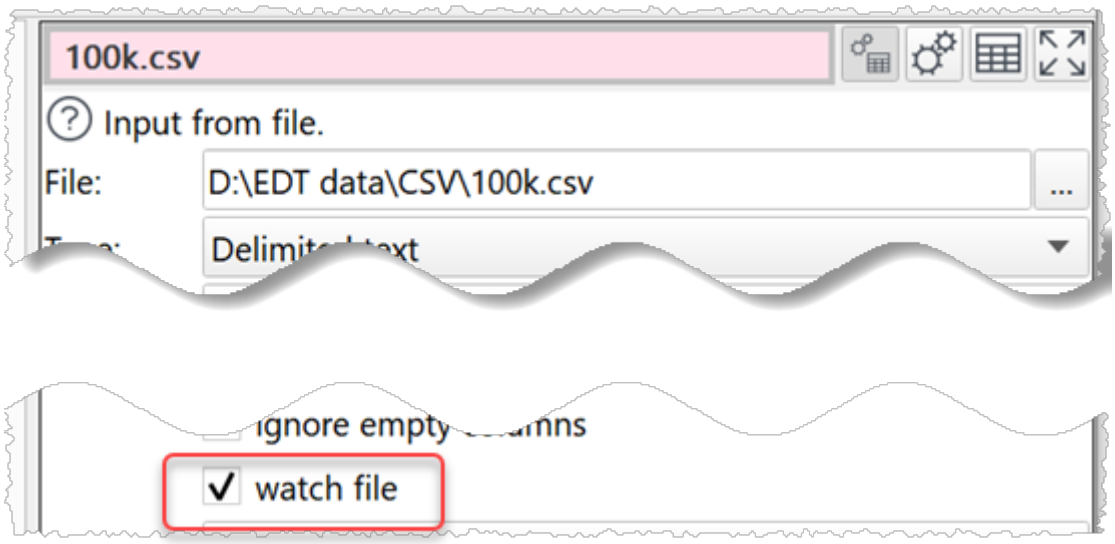

See also:

- Video: How to split CSV into multiple files

## 3.30 Trigger an update when an input changes

Check **watch file** in the **Right** pane for an input file if you want to trigger processing (**Run>Auto Run** checked) or set it to 'Needs update' (**Run>Auto Run** unchecked) whenever the input file changes.


*© 2023 Oryx Digital Ltd*




Note:

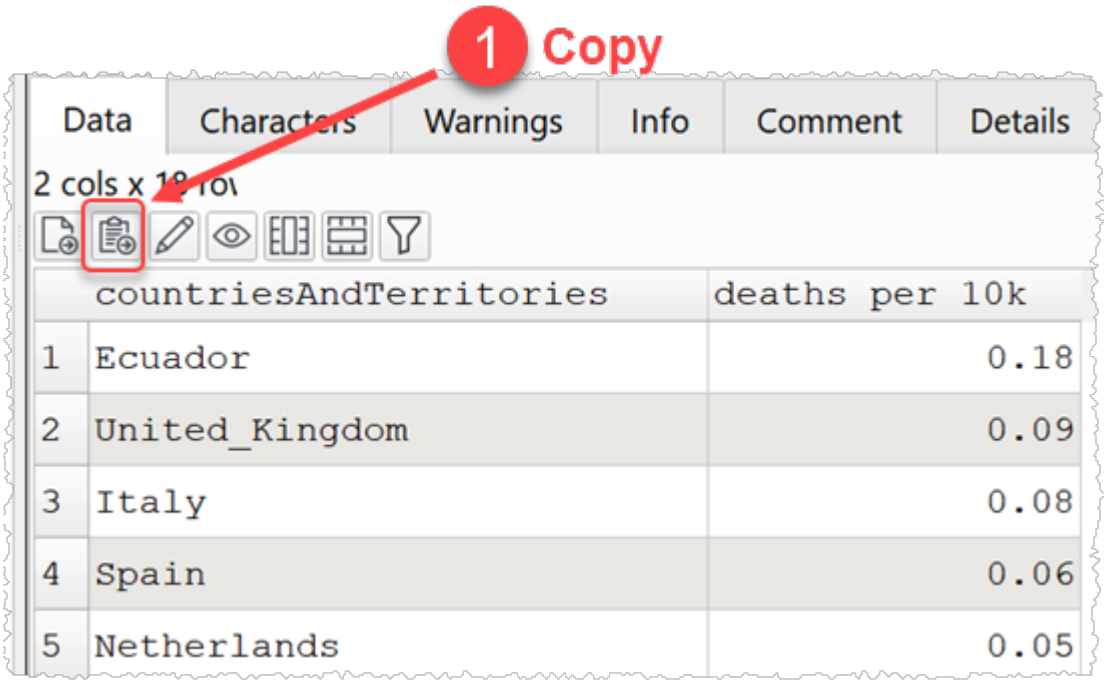
- If you are also outputting to this file in the same *.transform* you are inputting it, it may cause the *.transform* to run forever.
- The update may not be triggered if the file is deleted and replaced with a new file with the same name.

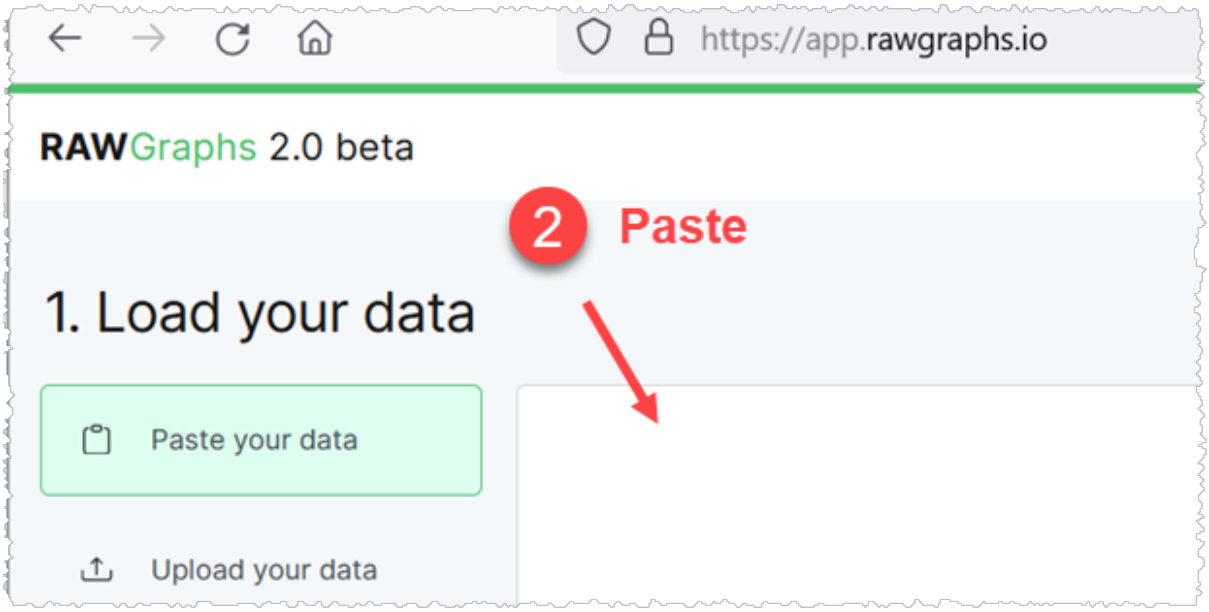
## 3.31 Visualize data

Currently Easy Data Transform does not support charting beyond data profiling. However it can easily integrate with other tools that do. For example. by outputting a CSV for import into the visualization tool, or copying and pasting via system clipboard.

For example, you can copy and paste data direct from Easy Data Transform to rawgraphs.io:

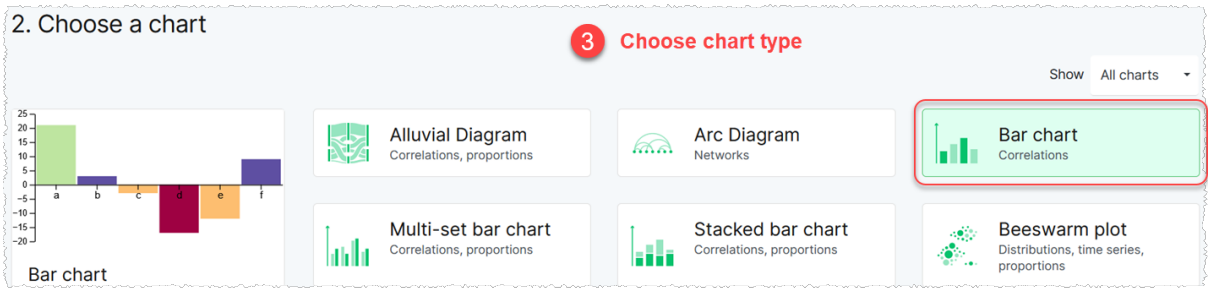

*© 2023 Oryx Digital Ltd*

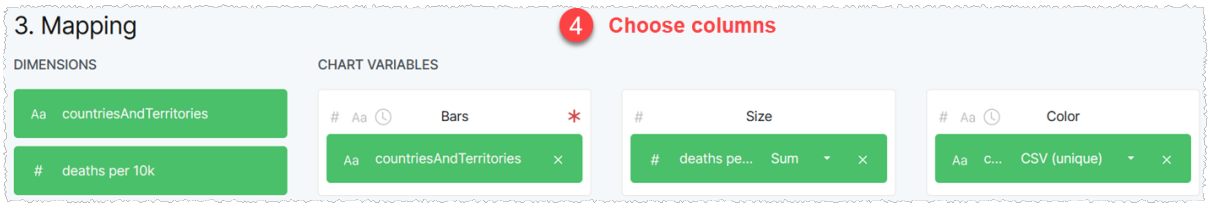


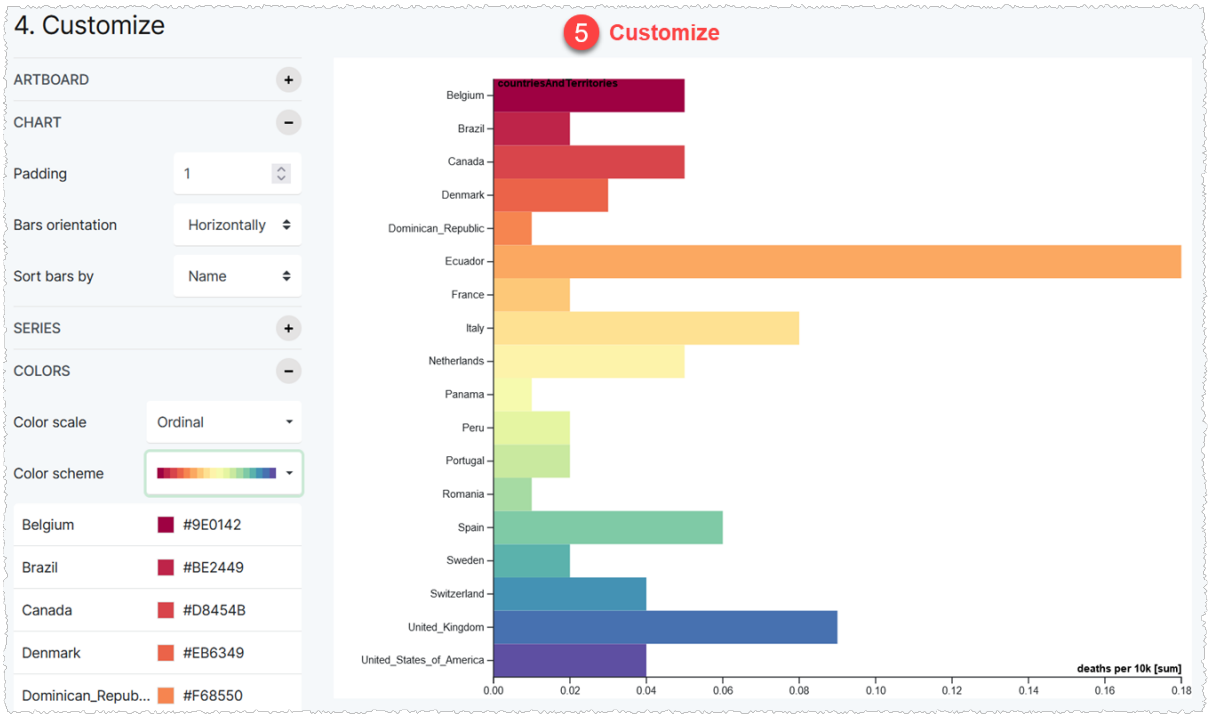





© 2023 Oryx Digital Ltd



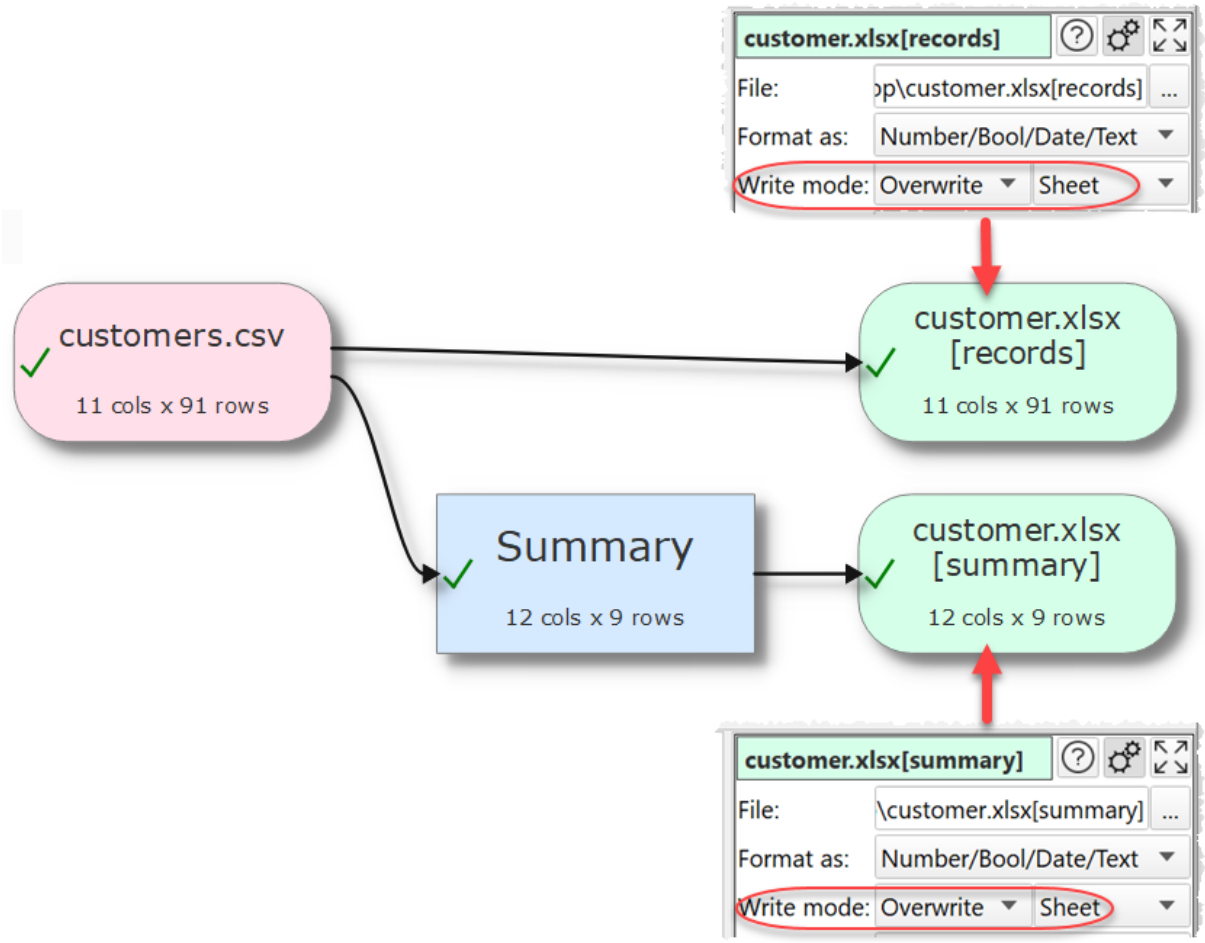

See also:

- Video: How to visualize data with rawgraphs.io


*© 2023 Oryx Digital Ltd*

## 3.32 Write to multiple sheets of an Excel file

To write to multiple sheets (tabs) of the same Excel file you need to set the **Write mode** of each output item to **Overwrite/Sheet** (to clear the sheet first) or **Append** (to add to existing sheet data).




If you set the **Write mode** to **Overwrite/File** for an item then the write will remove existing sheets.


*© 2023 Oryx Digital Ltd*

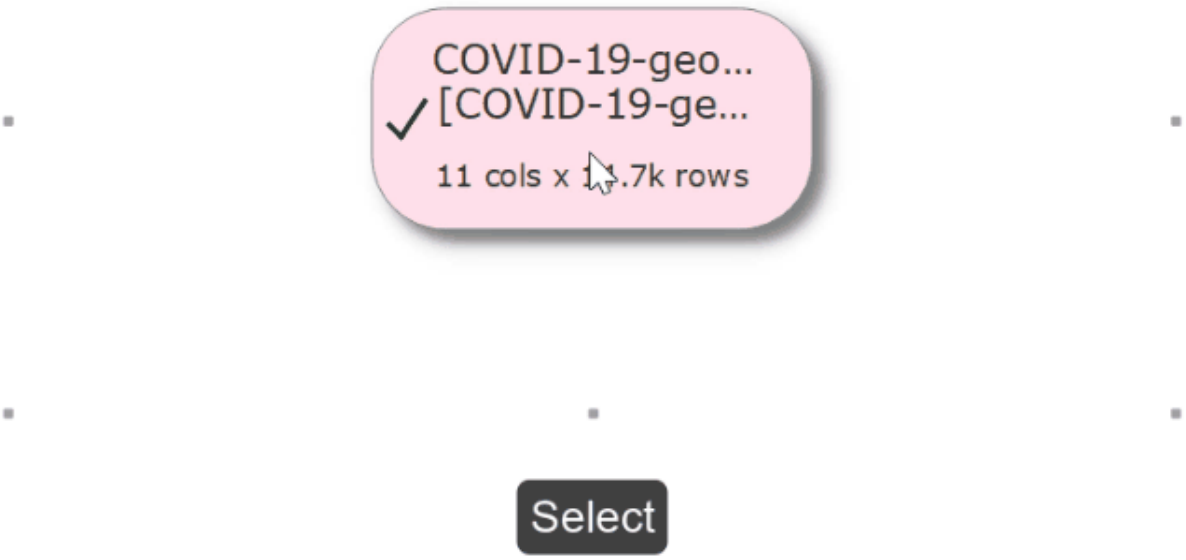
# Expert tips

# 4 Expert tips

## 4.1 Expert tips

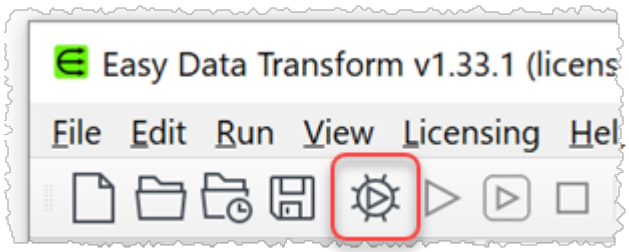
Here are some tips to help you be more productive with Easy Data Transform.

1. To add a transform click an existing Center pane item and type the first few letters of the transform name.



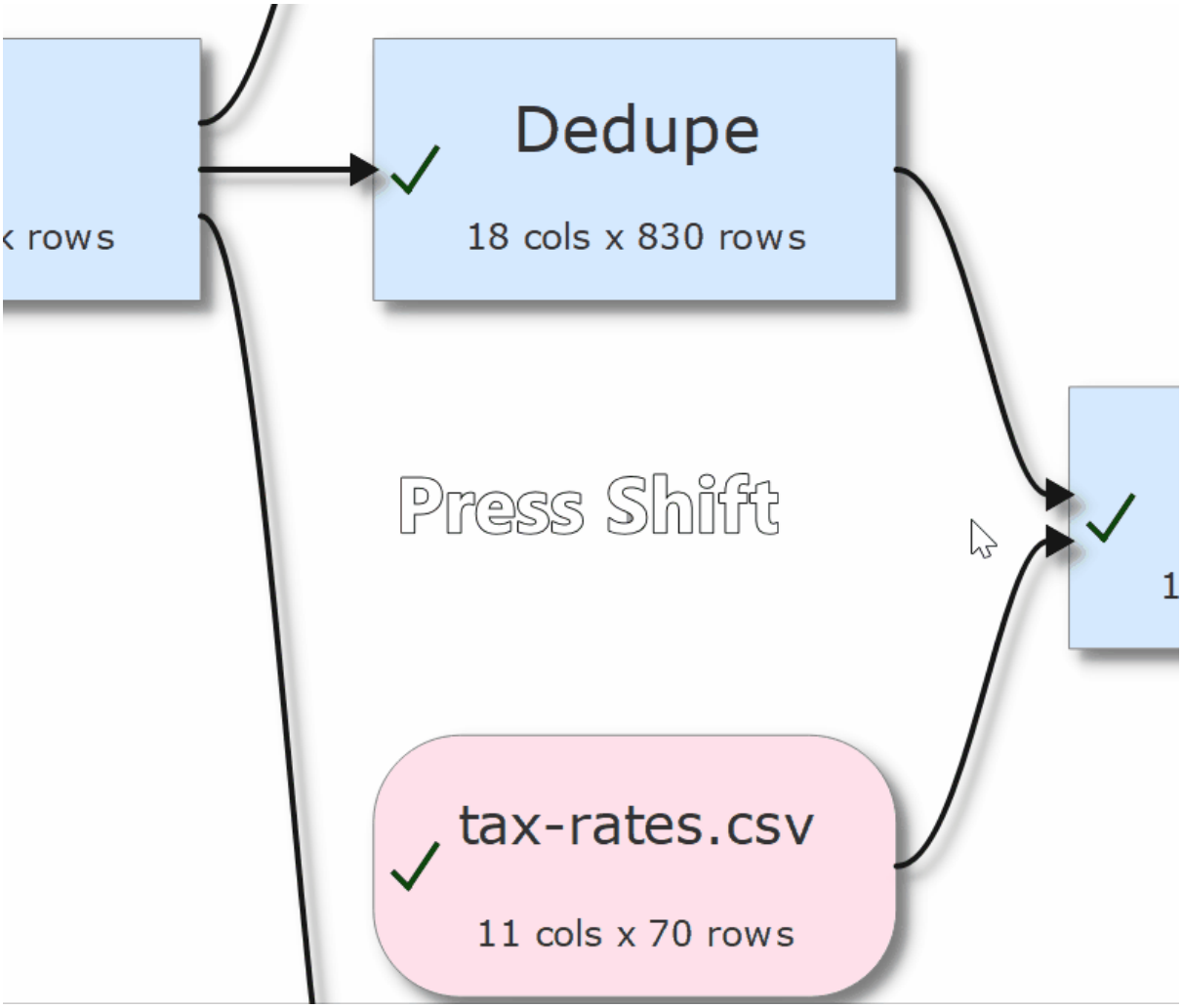

2. Check the **show advanced** checkbox in the **Left** pane to see all available transforms.

3. Uncheck **Run>Auto Run** to take full control of when items are processed.



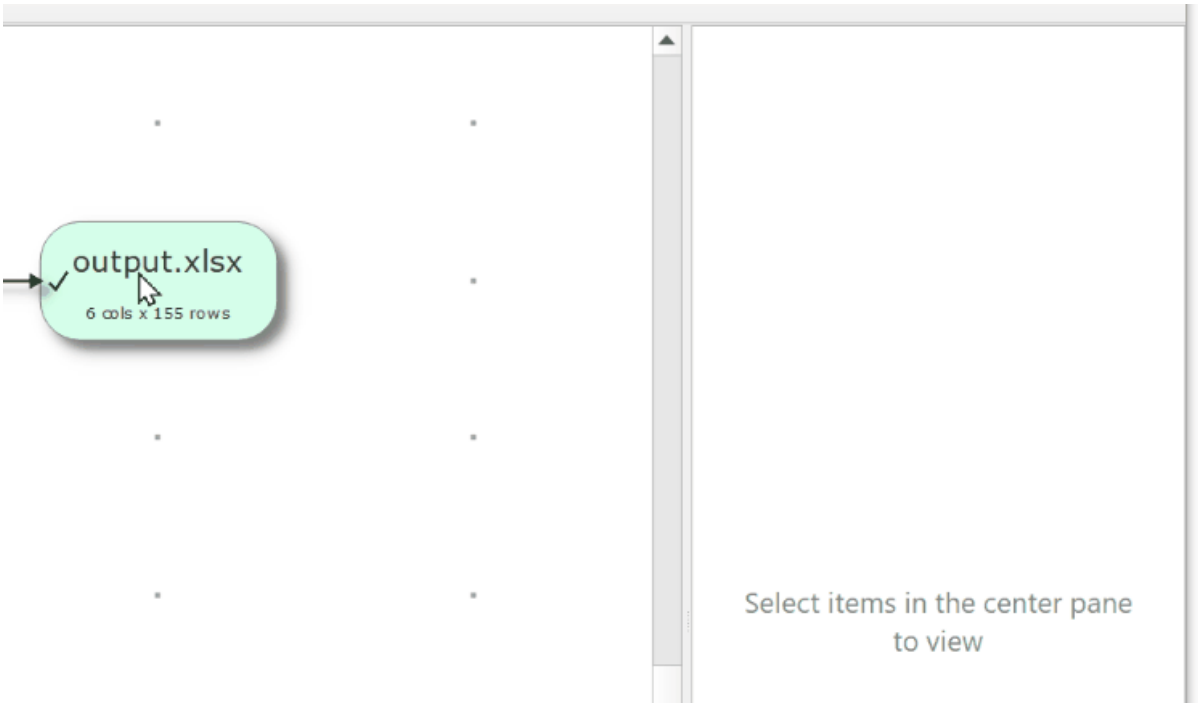

4. Hold down the `Shift` key and drag on the **Center** pane to scroll left/right/up/down.


*© 2023 Oryx Digital Ltd*

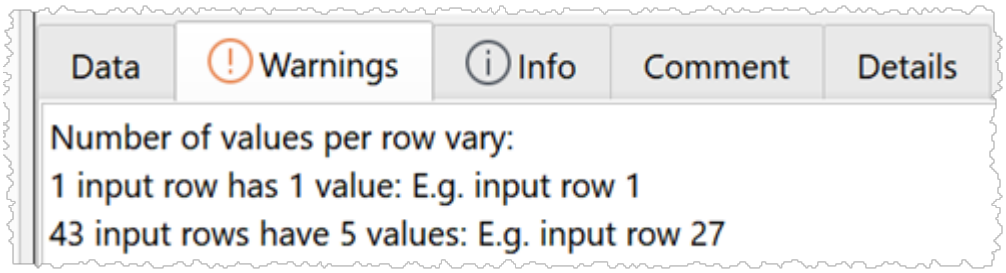



5. Use Note items to remind yourself how a complex `.transform` works.


© 2023 Oryx Digital Ltd




6. Look in the **Warnings** and **Info** tabs in the **Right** pane to troubleshoot potential issues.



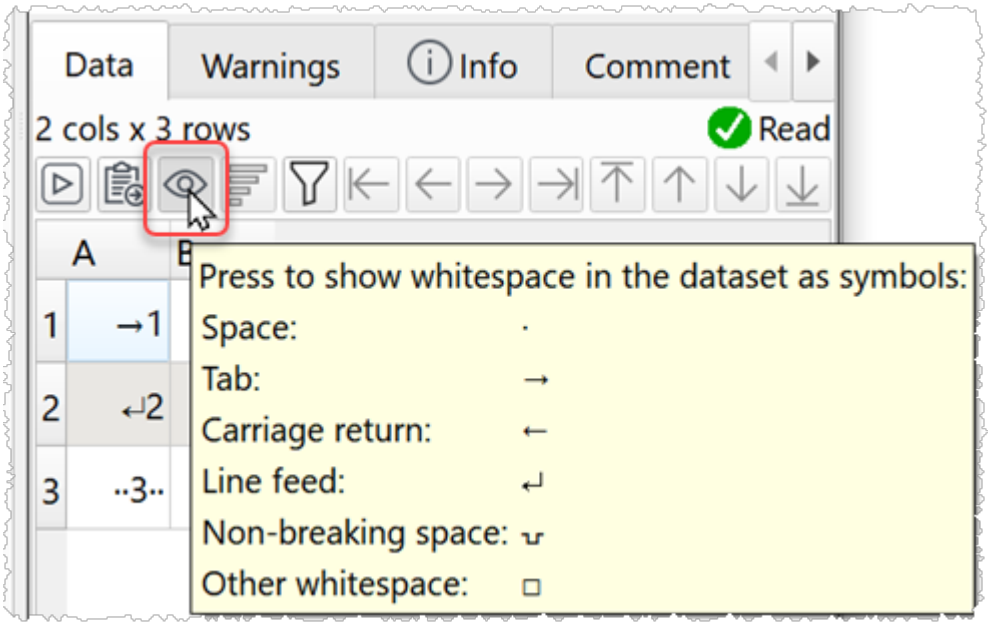

7. To select multiple items in the **Center** pane, left click with the mouse and drag a box over the items you want to select.

8. Try to avoid disconnecting transforms, as this can reset column related options downstream. Add a transform to an existing connection instead. You can also delete a transform without deleting it's connections.

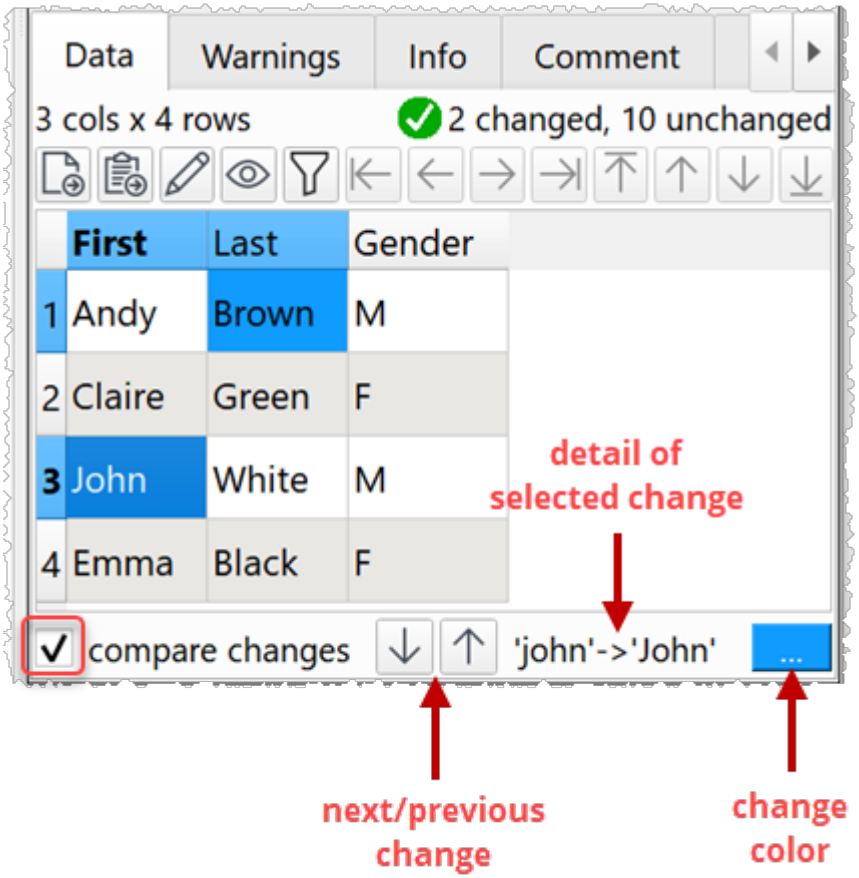
9. Add a Sample transform straight after the input of a large dataset and set **Rows** to pass through only the first 100 or so rows. Once you have completed all your transforms you can then **Disable sampling** to pass through all rows.

10. Click the eye icon in the **Right** pane to show whitespace in the data.


*© 2023 Oryx Digital Ltd*



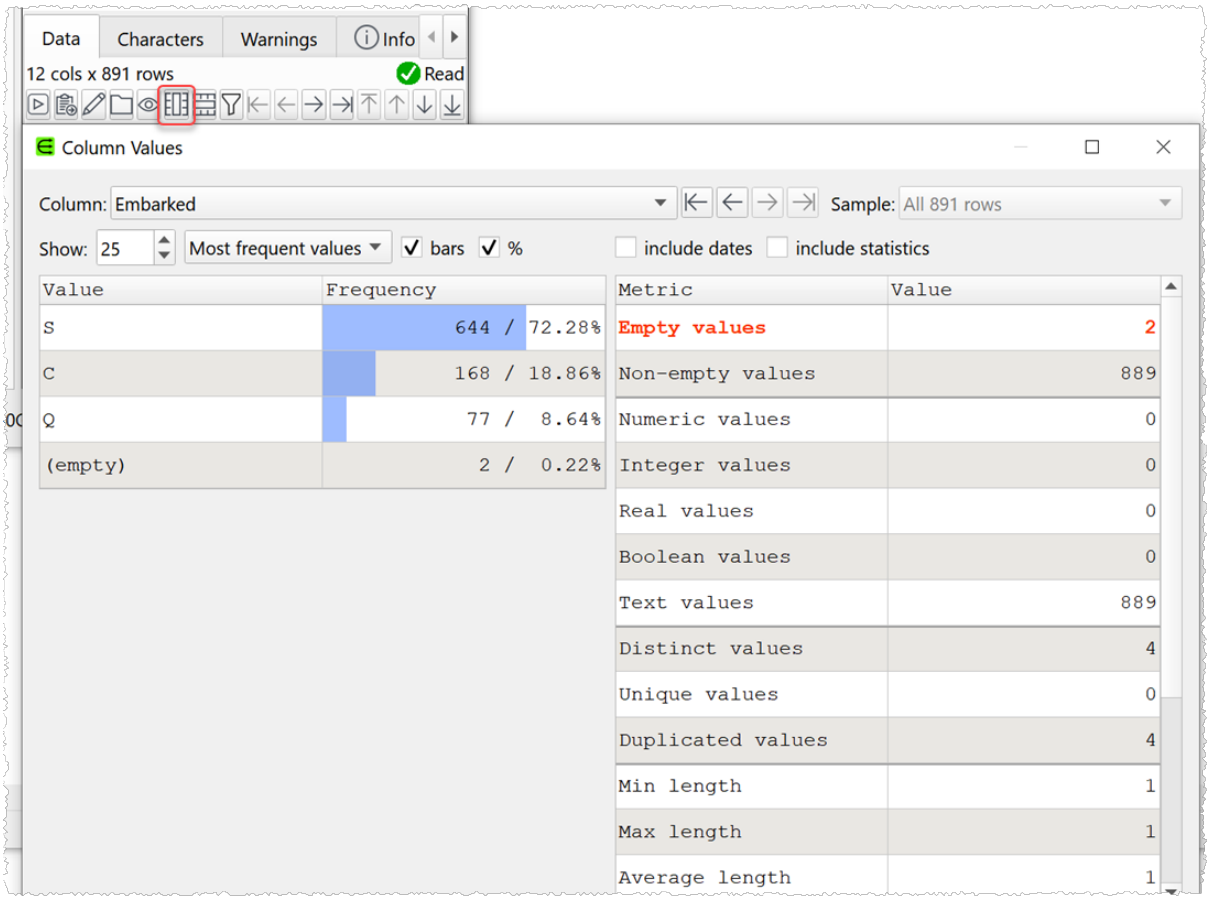

11. Check **compare changes** in the **Right** pane to [see changes from a transform] (selected transforms only).





© 2023 Oryx Digital Ltd

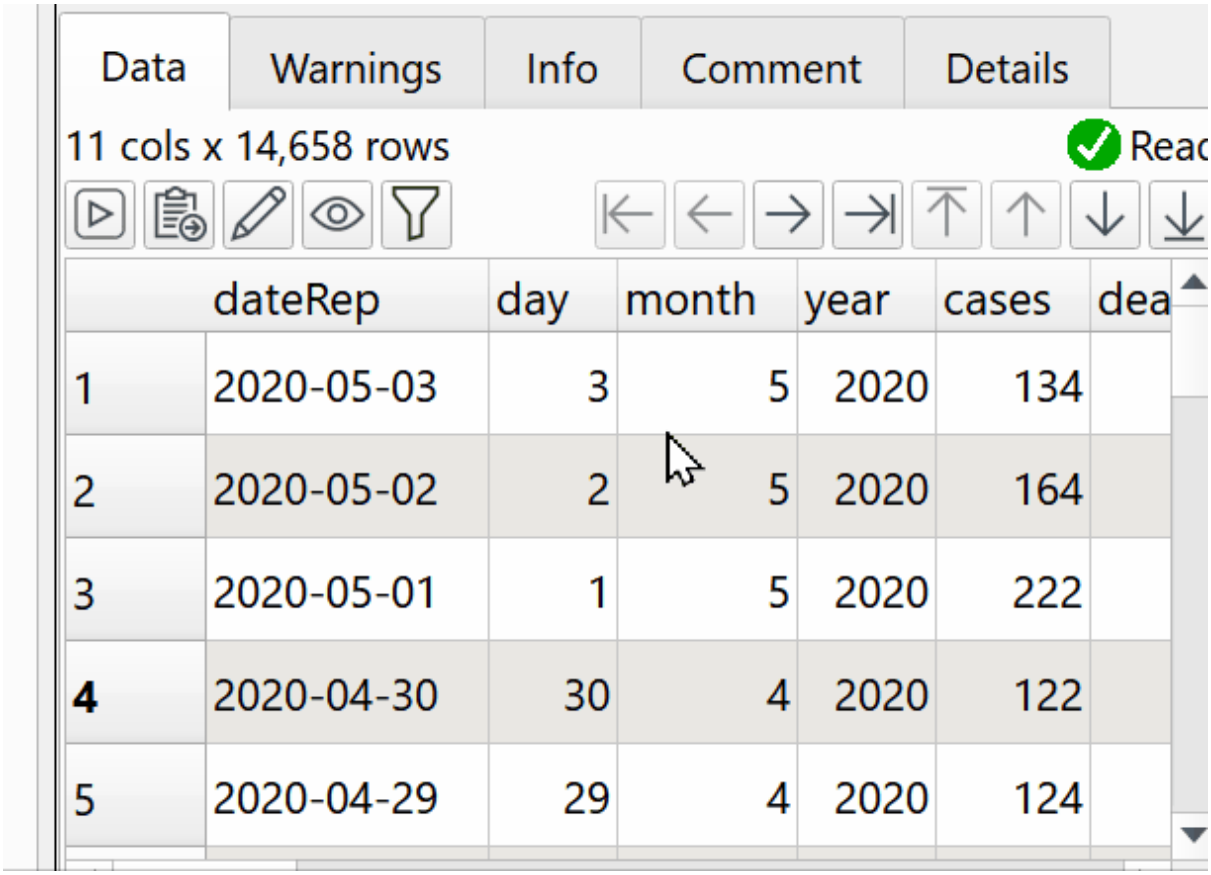
12. Select a column and click the bars icon in the **Right** pane to show the frequency of values. Potential issues are shown in red.




13. Enable drilldown to double-click on a cell and see how the information was derived.

14. Filter data in the **Right** pane.


*© 2023 Oryx Digital Ltd*




| | dateRep | day | month | year | cases | dea |
|---|---|---|---|---|---|---|
| 1 | 2020-05-03 | 3 | 5 | 2020 | 134 | |
| 2 | 2020-05-02 | 2 | 5 | 2020 | 164 | |
| 3 | 2020-05-01 | 1 | 5 | 2020 | 222 | |
| **4** | 2020-04-30 | 30 | 4 | 2020 | 122 | |
| 5 | 2020-04-29 | 29 | 4 | 2020 | 124 | |

15. If you need several similar branches of transforms you can create one branch and duplicate it.

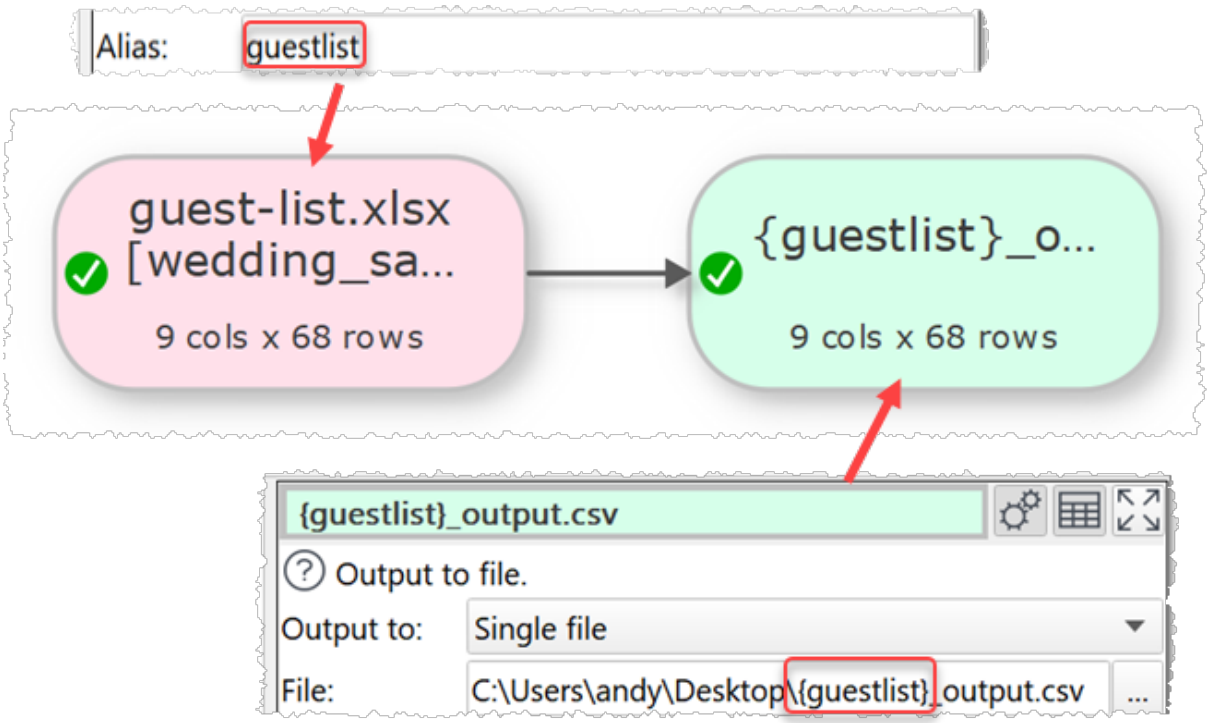
16. If you have 2 or more monitors connected to your computer try **View>Two Screen Mode** to show the **Right** pane on your second monitor.

17. Use batch processing with wildcards (e.g. *.csv) to process a folder full of input files in one operation.

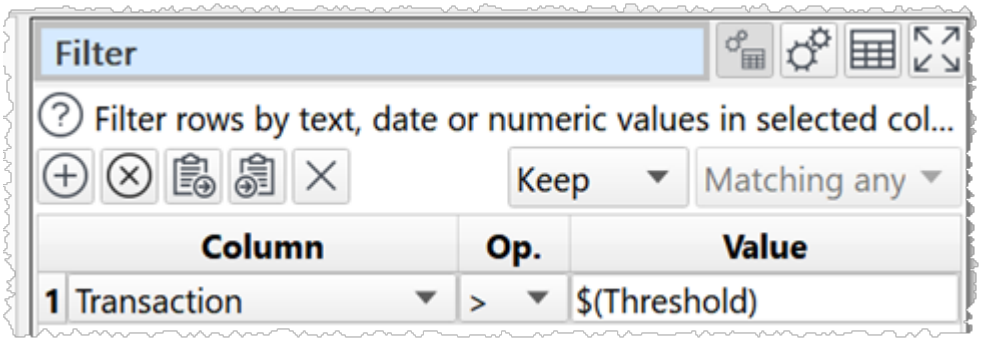
18. Use command line arguments to run `.transform` files from a batch or script file.

19. Use file name variables to set the name of an output file based on the name of an input file, date created etc.


*© 2023 Oryx Digital Ltd*

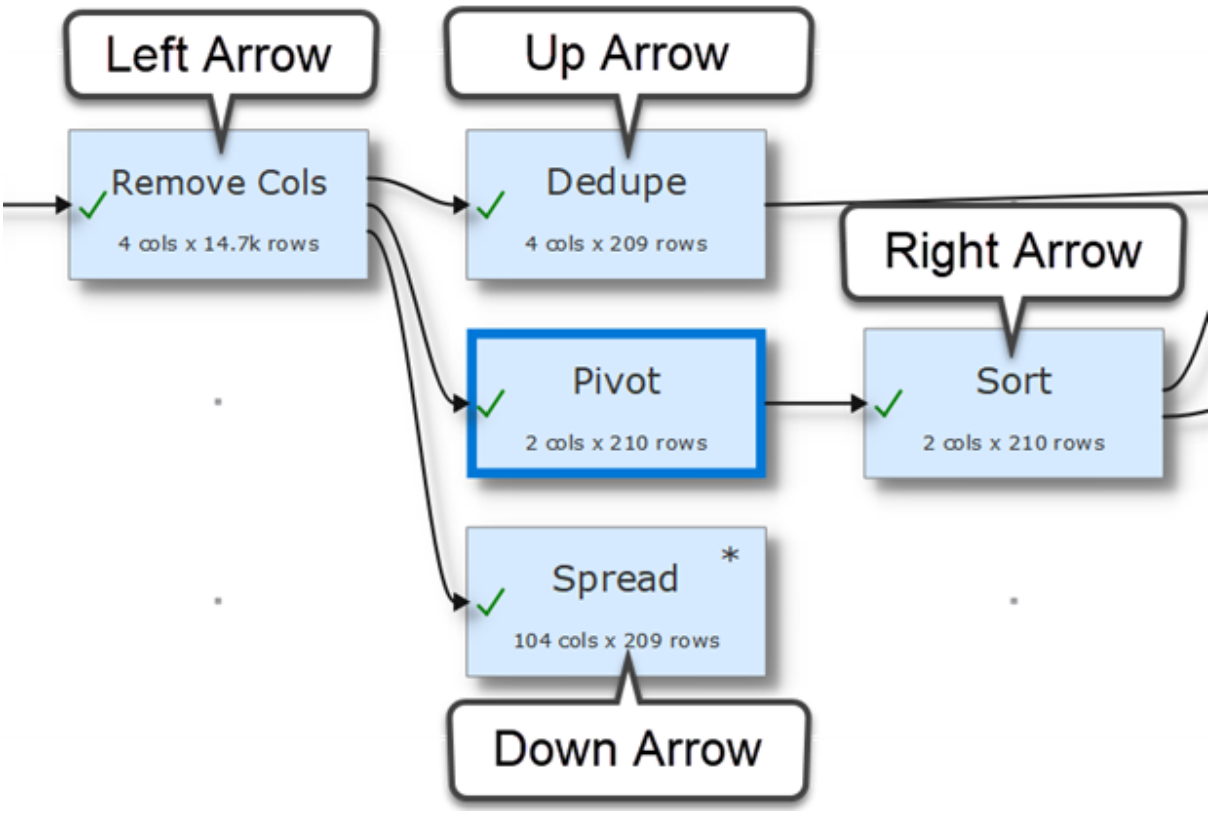



20. Use column variables when you want to use values from columns in transforms such as If and Filter.




21. Learn keyboard shortcuts. For example, you can quickly change selection in the **Center** pane using arrow keys with the `Ctrl` key.


*© 2023 Oryx Digital Ltd*




22. Change focus between the **Center** and **Right** panes using `Alt+Left` and `Alt+Right`.

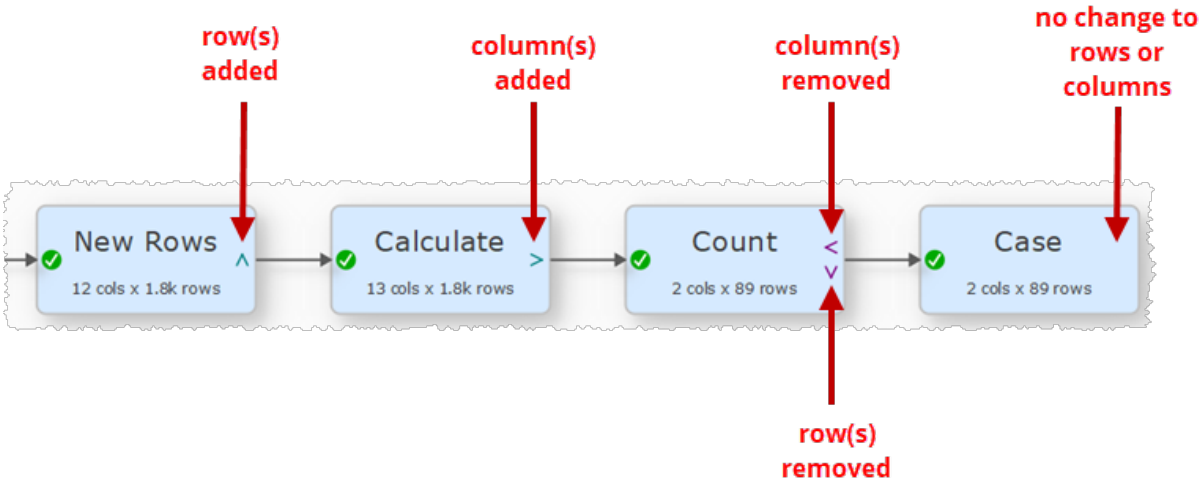
23. Check the **gray Left pane buttons when not available** checkbox in the **General** tab of the **Preferences** window for **Left** pane buttons to be disabled until the appropriate items in the **Center** pane are selected.

24. Double-click on an item in the **Center** pane to show it fullscreen.

25. Check **watch file** in the **Right** pane for an input file if you want to trigger processing (**Run>Auto Run** checked) or set it to 'Needs update' (**Run>Auto Run** unchecked) whenever it changes.

26. Check **View>Show Row/Col Changes** to show changes in the number of rows and columns in the **Center** pane.

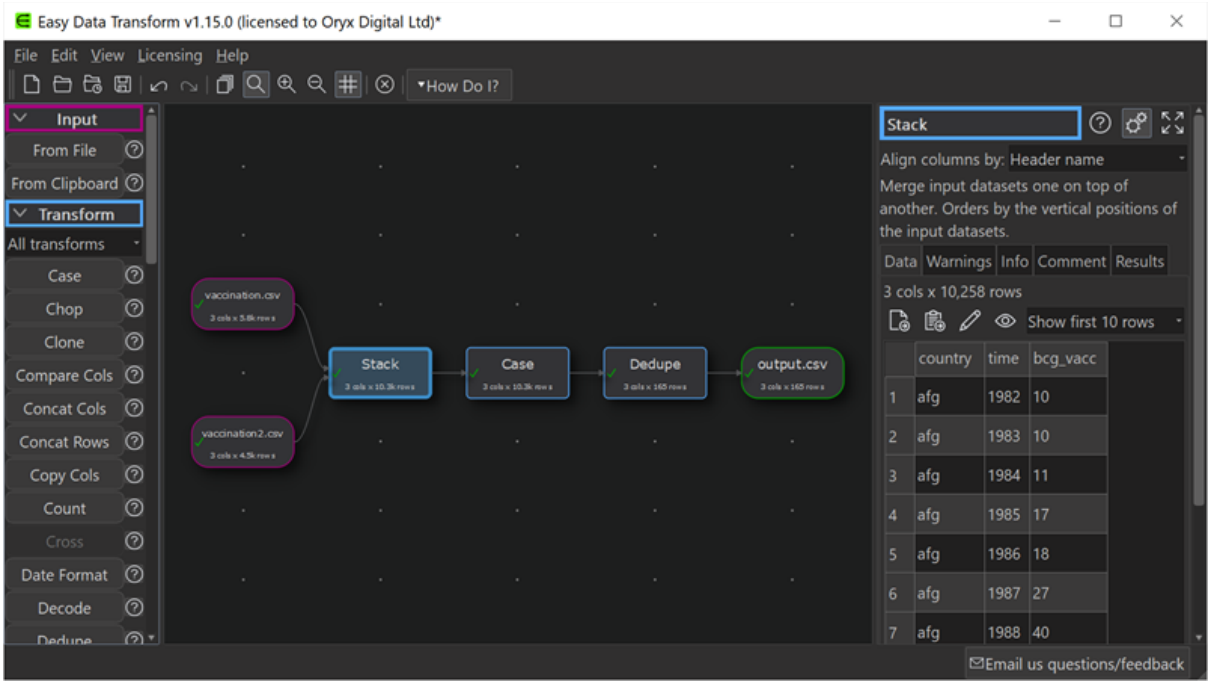

*© 2023 Oryx Digital Ltd*




27. Change the **Center** pane color scheme in the **Colors** tab of the **Preferences** window.

28. Experiment with changing **Optimize processing for** from **Minimum memory use** to **Maximum speed** to see if that improves performance. It may use up all your memory.

29. Select **View>Toggle UI Theme** to swap between light and dark user interface themes.




30. Select **File>New Window** to open a new **Main** window.


*© 2023 Oryx Digital Ltd*

# Support

# 5 Support

## 5.1 Forum

There is a discussion forum online at forum.easydatatransform.com. You can use this to ask questions, share what you are doing and keep up to date with news.

## 5.2 Contact support

If you have any questions or suggestions, please post a question on forum.easydatatransform.com or email us at support@easydatatransform.com.

## 5.3 Report a bug

Please report any bugs you find to support@easydatatransform.com Please include:

- your operating system (e.g. Windows 11)
- the version of Easy Data Transform (from **Help>About**)
- a step-by-step description of how we can reproduce the problem
- your .transform file and input data files (where appropriate)
- a screen capture or video can often be helpful

The step-by-step description is particularly important - if we can't reproduce your problem, then we probably won't be able to fix it.

We treat all data sent to us as confidential, unless you tell us otherwise. If your data is particularly sensitive, it might be enough to send us just the first few rows with sensitive values changed. Keep the same column structure though.

## 5.4 Request an enhancement

We are always very interested to hear your suggestions on how the software can be improved. Please post a feature suggestion on our forum.easydatatransform.com or email us at support@easydatatransform.com.


*© 2023 Oryx Digital Ltd*

## - . -

.transform file 270

## - A -

abs 29
add 29
aggregate 213
and 29, 92
Automatic and manual processing. 246

## - B -

base (numeric) 131
base64 decode/encode 68
batch processing 264
binary 131
boolean 255

## - C -

calculate 29
case 37
Center pane 23
chop 39
classify 41
cleaning data 282
clone 41
cluster 41
column variables 259
command line arguments 266
comments 248
compare cols 45
concat cols 48
concat rows 50
connections 251
copy 287
copy cols 56
count 58
cross 60
crosstab 189
CSV format 227

## - D -

dark mode 274
date format 61
dates 253
day of week/month/year 29
decode 68
dedupe 70
divide 29
drilldown 262
duplicate 287

## - E -

Excel format 230
expert tips 331
extensions 25
extract 73

## - F -

file extensions 25
file formats 227
file splitter 321
filename variables 268
fill 79
filter 81
Fixed width format 231
forum 341
fuzzy matching 261

## - G -

gather 85
group by 85

## - H -

hashing 87
header 90, 250
hexadecimal 131
HTML 236
HTML escape/unescape 68


*© 2023 Oryx Digital Ltd*

## - I -

if 92
impute 96
input 26
insert 101
interpolate 103
intersect 106
Introduction 8

## - J -

Javascript 107
join 111
JSON format 233

## - L -

Left pane 23
locale 256
Log pane 23
logarithm 29
long pivot 85
lookup 115

## - M -

Main window 22
Markdown format 238
meta information 257
month 29
moving 118
moving average 118
multiply 29

## - N -

new col 122
new rows 124
ngram 127
num base 134
num format 134
numbers 255

## - O -

octal 131
offset 138
or 29, 92
outliers 139
output 221

## - P -

pad 144
paste 287
percent 285
pivot 146
pivot longer 85
pivot wider 189
Plain text format 238
power 29
preferences 23
Preferences window 23
processing 246
profiling data 307

## - Q -

Quick start guide 8

## - R -

rank 177
regular expressions 260
remove cols 149
rename cols 151
reorder cols 152
replace 154
Right pane 23
rolling average 118
row num 163
run processing 246
running average 118

## - S -

sample 165
scale 167


*© 2023 Oryx Digital Ltd*

scheduling 266, 314
scrape web data 315
scripting 107
slice 175
sort 177
split col 182
split file 321
split rows 186
spread 189
stack 191
stamp 193
standard deviation 197
stats 197
substitute 199
subtract 29, 201
summary 202
system requirements 8

## - T -

text 253
time format 61
total 206
transpose 209
TSV format 239

## - U -

unfill 210
unique 213
units 215
Unix timestamp 29
unpivot 85
URL encode/decode 68

## - V -

vCard format 241
vcf format 241
visualization 326

## - W -

web scraping 315
whitespace 218, 320
wide pivot 189

## - X -

XML escape/unescape 68
XML format 241
xor 29

## - Y -

YAML format 246
year 29


*© 2023 Oryx Digital Ltd*