



Concept | Prepare recipe

Watch the video

The [Prepare recipe](#) is a visual recipe in Dataiku that allows you to create data cleansing, normalization, and enrichment scripts in a visual and interactive way.

Adding transformation steps to the script

To prepare your data, you must add steps to the recipe script.

Using the processor library

An essential advantage of the Prepare recipe is its library of around 100 data processors. Most [processors](#) are designed to handle one specific task, such as filtering rows, rounding numbers, extracting regular expressions, concatenating or splitting columns, and much more.

The screenshot shows the Dataiku Prepare Recipe interface. The main window displays a data table with columns: order_date, pages_visited, order_id, customer_id, tshirt_category, tshirt_price, and tshirt_quantity. The table contains three rows of data. A 'Processors library' window is open in the foreground, listing various data processing tasks such as 'Filter data', 'Data cleansing', 'Strings', 'Math / Numbers', 'Split / Extract', 'Web logs', 'Dates', 'Geography', 'Enrich', 'Reshaping', 'Natural Language', 'Joins', 'Complex objects', 'Code', and 'Misc'. The 'Reshaping' category is selected, and a list of processors is shown, including 'Find and replace', 'Split column', 'Transform string', 'Formula', 'Extract with regular expression', 'Concatenate columns', 'Simplify text', 'Tokenize text', 'Extract ngrams', 'Extract numbers', and 'Negate boolean value'.

order_date	pages_visited	order_id	customer_id	tshirt_category	tshirt_price	tshirt_quantity
2016/09/04	9	HTS-0002	038040	White T-Shirt M	20.0	1
2014/11/14	11	HTS-0001	801797	White T-Shirt M	20.0	1
2017/03/26	10	HTS-0003	038040	White T-Shirt E	18.0	2

Processors empower you to perform a huge variety and combination of tasks. One processor, for example, is a [Formula language](#), similar to what you might find in a spreadsheet.

which you can use to create new columns from those already present, drawing on a range of built-in functions.

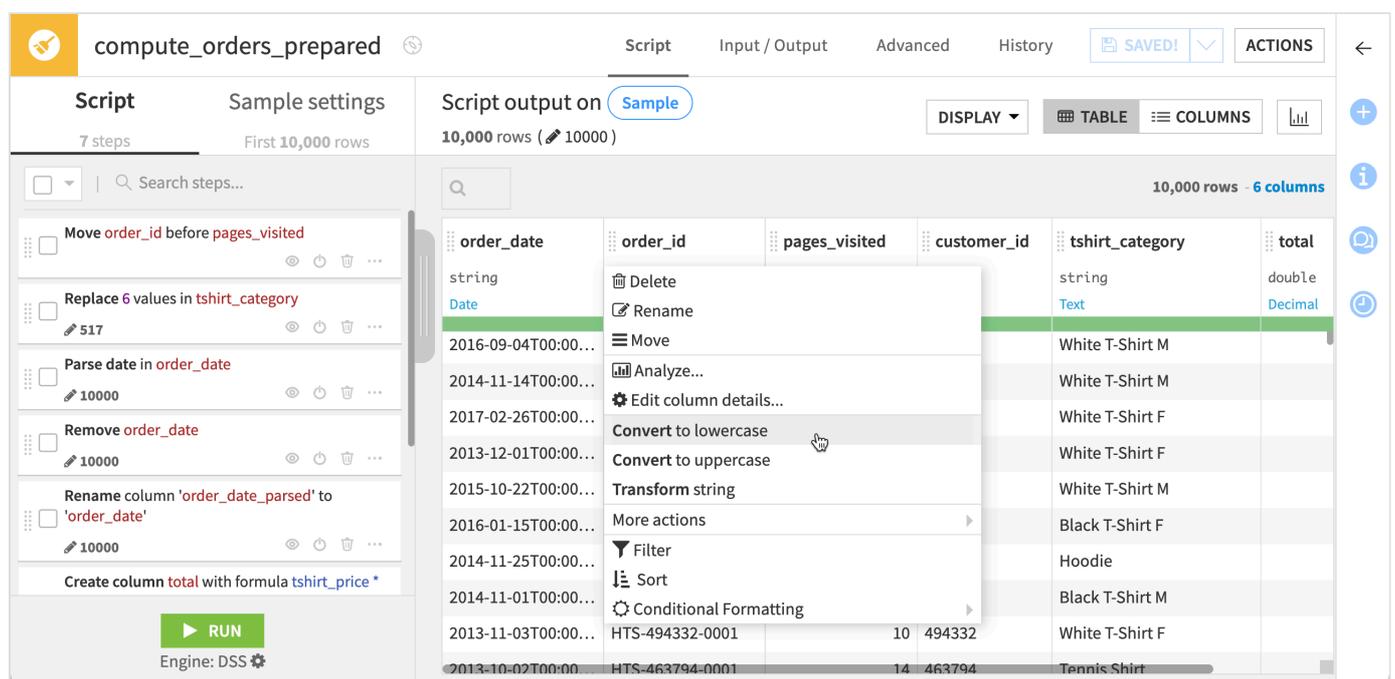
Another processor even lets you create a [Python function](#) for each row.

In addition to directly adding steps from the processor library, you can add steps to the script in a number of other ways.

Using the column context menu

In the column context menu, Dataiku will suggest steps to add based on the column's meaning.

For example, Dataiku will suggest to parse date columns, or remove rows with invalid values according to the column meaning. For a text column, it will suggest string transformations, such as converting to lowercase.



The screenshot shows the Dataiku interface for a script named 'compute_orders_prepared'. The script has 7 steps and is displaying the first 10,000 rows. The table view shows 10,000 rows and 6 columns: order_date (Date), order_id (string), pages_visited (double), customer_id (string), tshirt_category (Text), and total (Decimal). A context menu is open over the 'order_date' column, suggesting actions such as 'Delete', 'Rename', 'Move', 'Analyze...', 'Edit column details...', 'Convert to lowercase', 'Convert to uppercase', 'Transform string', 'More actions', 'Filter', 'Sort', and 'Conditional Formatting'. The 'Convert to lowercase' option is highlighted by the mouse cursor.

Using the Analyze window

Another method to add steps to the script is through the Analyze window.

Within a Prepare recipe, the Analyze window can guide data preparation, for example merging categorical values.

The screenshot displays the Dataiku interface for a script step named "compute_orders_prepared". A modal window titled '"tshirt_category" - (10 distinct)' is open, showing a "VALUES CLUSTERING" view. The modal includes a "SUMMARY" section with the following data:

Category	Count	%
Valid	7,597	100.0 %
Unique	0	0.0 %
Invalid	0	0.0 %
Empty	0	0.0 %
0 UNIQUES	0.0 %	
0 INVALIDS	0.0 %	

The modal also features a "Merging 2 values" section with a "Replace with" field containing "white t-shirt m" and a "MERGE" button. Below this is a table of values for clustering:

Value	Count	%	Cum. %
hoodie	1752	23.1	23.1
white t-shirt m	1582	20.8	43.9
black t-shirt m	1179	15.5	59.4
white t-shirt f	1014	13.3	72.8
black t-shirt f	897	11.8	84.6
tennis shirt	656	8.6	93.2
wh tshirt m	163	2.1	95.3
bl tshirt m	144	1.9	97.2

The background interface shows a data table with columns for date, order ID, and t-shirt category. The table contains the following data:

Date	Order ID	T-shirt Category
2015/03/26	2015-03-26T00:00:00.000Z	hoodie
2016/04/24	2016-04-24T00:00:00.000Z	black t-shirt f
2015/01/17	2015-01-17T00:00:00.000Z	white t-shirt m

Manually moving the columns

You can also directly drag columns to adjust their order, or switch from the Table view to the Columns view to apply certain steps to more than one column at a time.

Previewing and applying the script

When adding new steps to the script, you'll notice how the step output is immediately visible. This is possible because the step is being applied to the same **sample** of the dataset found in the Explore tab. The quick feedback allows you to work incrementally, quickly modifying your transformation steps.

compute_orders_prepared

Script Input / Output Advanced History

SAVE ACTIONS

Script 8 steps

Sample settings First 10,000 rows

Disable preview

Step preview on sample Perform to_lower on tshirt_category

View modified rows View all rows 10000

DISPLAY

TABLE COLUMNS

10,000 rows 6 columns

order_date	order_id	pages_visited	customer_id	tshirt_category	total
string Date	string Text	bigint Integer	string Text	string Text	double Decimal
2016-09-04T00:00...	HTS-038040-0002	9	038040	white t-shirt m	
2014-11-14T00:00...	HTS-801797-0001	11	801797	white t-shirt m	
2017-02-26T00:00...	HTS-vft1eu-0003	10	vft1eu	white t-shirt f	
2013-12-01T00:00...	HTS-914324-0001	10	914324	white t-shirt f	
2015-10-22T00:00...	HTS-88ua9r-0001	12	88ua9r	white t-shirt m	
2016-01-15T00:00...	HTS-061311-0003	9	061311	black t-shirt f	
2014-11-25T00:00...	HTS-479441-0001	6	479441	hoodie	
2014-11-01T00:00...	HTS-352809-0001	10	352809	black t-shirt m	
2013-11-03T00:00...	HTS-494332-0001	10	494332	white t-shirt f	
2013-10-02T00:00...	HTS-463794-0001	14	463794	tennis shirt	
2013-07-23T00:00...	HTS-132885-0001	17	132885	white t-shirt m	

Remove columns tshirt_quantity, tshirt_price 10000

Perform to_lower on tshirt_category 10000

Column single | multiple | pattern | all

tshirt_category

Mode

Convert to lowercase

For more advanced string transformations, use the Formula processor

+ ADD A NEW STEP

RUN

Engine: DSS

Notice that steps in the script constitute a list of instructions. These instructions are not immediately applied to the dataset itself.

For example, adding a **Delete Column** step removes that column from the step preview, but it does not actually delete the column in the dataset, as it would in a spreadsheet.

Only when you choose to actually run the recipe will Dataiku execute the instructions on the full input dataset, and thereby produce a new output dataset. The original input dataset always remains.

Managing the script

If a script starts to grow in complexity, a number of features can help you manage them. You can:

- Disable steps.
- Organize individual steps into groups of steps.
- Add colors and comments in order to send reminders to yourself and colleagues.
- [Copy and paste steps](#) within the same recipe or to another recipe, even if that recipe is in another project or another Dataiku instance.

The screenshot shows the Dataiku interface for a recipe named 'compute_orders_prepared'. The 'Script' tab is active, displaying a list of steps on the left and a 'Script output' table on the right. The table contains 7597 rows and 9 columns. A context menu is open over the table, showing options like 'Copy this step', 'Paste after this step', 'Comment', 'Color', and 'Duplicate step'. The 'ADD A GROUP' button in the bottom left is highlighted with a red box.

order_date	order_date_parsed	order_id	pages_visited	customer_id	tshirt_category	tshirt_price	tshirt_quantity
2016/09/04	2016-09-04T00:00:00.000Z	HTS-0002	9	038040	white t-shirt m	20.0	
2014/11/14	2014-11-14T00:00:00.000Z	HTS-0001	11	801797	white t-shirt m	20.0	
2013/12/01	2013-12-01T00:00:00.000Z	HTS-0001	10	914324	wh tshirt f	18.0	
2016/01/15	2016-01-15T00:00:00.000Z	HTS-0003	9	061311	black t-shirt f	17.5	
2014/11/25	2014-11-25T00:00:00.000Z	HTS-0001	6	479441	hoodie	23.0	
2014/11/01	2014-11-01T00:00:00.000Z	HTS-0001	10	352809	black t-shirt m	19.0	
2013/11/03	2013-11-03T00:00:00.000Z	HTS-0001	10	494332	wh tshirt f	18.0	
2013/11/03	2013-11-03T00:00:00.000Z	HTS-0001	14	463794	tennis shirt	24.0	
2013/11/03	2013-11-03T00:00:00.000Z	HTS-0001	17	132885	white t-shirt m	20.0	
2013/11/03	2013-11-03T00:00:00.000Z	HTS-0001	17	519113	bl tshirt f	17.5	
2015/10/09	2015-10-09T00:00:00.000Z	HTS-0002	8	252675	white t-shirt m	20.0	
2016/08/15	2016-08-15T00:00:00.000Z	HTS-0004	6	049348	black t-shirt f	17.5	
2015/10/09	2015-10-09T00:00:00.000Z	HTS-0003	10	989355	hoodie	23.0	
2015/03/26	2015-03-26T00:00:00.000Z	HTS-0002	11	679143	hoodie	23.0	
2016/04/24	2016-04-24T00:00:00.000Z	HTS-0001	12	301362	black t-shirt f	17.5	
2015/01/17	2015-01-17T00:00:00.000Z	HTS-0002	5	198773	white t-shirt m	20.0	

What's next?

In this article, you learned how to use the Prepare recipe for data cleansing, normalization, and enrichment.

Note

Instead of building recipes directly in the Flow, when your workflow is in production, to avoid disturbing it, you can use a [visual analysis in the Lab](#) for experimental work.

Continue getting to know the basics of Dataiku by learning about [date handling](#).

Tip

You can find this content (and more) by registering for the Dataiku Academy course, [Visual Recipes](#). When ready, challenge yourself to earn a [certification](#)!