

# Trifacta Wrangler Tutorial

<b><u>Summary</u></b>	<a href="#">In this codelab you will get a well-rounded explanation of Trifacta's transformation capabilities and walkthrough of the end to end workflow - from connecting to the data to joining datasets and creating recipes to generating output</a>
<b><u>Dataset URL</u></b>	<a href="https://drive.google.com/drive/folders/11ZS06eEaAzs3wlpjE3YRbLJhdzO-wlc2?usp=sharing">https://drive.google.com/drive/folders/11ZS06eEaAzs3wlpjE3YRbLJhdzO-wlc2?usp=sharing</a>
<b><u>Category</u></b>	<a href="#">Trifacta</a>
<b><u>Author</u></b>	<a href="#">Nikhil Kohli</a>

---

## What is Trifacta Wrangler?

Trifacta Wrangler enables you to explore, combine, and transform diverse datasets for downstream analysis.

---

## Demo

This tutorial walks through editing a sample dataset using the free tool Trifacta Wrangler, which is available for download [here](#). The sample dataset used here is a CSV of Ecommerce 2019 Sales data and synthetically generated user data.

Once you have signed up for Trifacta Wrangler, select a new flow—this will create a new flow in which you can organize your dataset.

The screenshot displays the Trifacta Wrangler web interface. At the top, the 'TRIFACTA Wrangler' logo is visible. Below it, a table lists recent flows with columns for status and completion time. A 'View more' link is present next to the table. On the right side, there are buttons for 'Import Data' and 'Create Flow', followed by a 'Resources' section with links to 'What's New', 'Tutorials', 'Help Articles', 'Events & Webinars', 'Community', and 'Wrangle Exchange'. A 'Create Flow' dialog box is open in the center, prompting for a 'Flow Name' and 'Flow Description'. The background interface shows a 'Welcome back, Nikhil Kohli!' message, a 'Recently Updated' section with a '2019 Sales Insights' dataset, and a 'Recent Jobs' section with two jobs listed.

TRIFACTA Wrangler

Import Data

Create Flow

Resources

What's New

Tutorials

Help Articles

Events & Webinars

Community

Wrangle Exchange

View more

Today at 12:14 AM

Last Sunday at 10:46 PM

View more

Completed • Finished Last Sunday at 10:48 PM

Welcome back, Nikhil Kohli!

Recently Updated

2019 Sales Insights

Recent Jobs

2019\_Ecomm\_Sales\_...  
Job ID: 125284

2019\_Ecomm\_Sales\_...  
Job ID: 125230

Create Flow

Resources

Tutorials

Help articles

Events & webinars

Community

Wrangle Exchange

Create Flow

Flow Name

2019 Ecomm Sales Insights Demo

Flow Description

2019 Ecomm Sales Insights Demo

Cancel

Create

Then select add datasets to flow, and then import a dataset.



Add Datasets into this Flow to start wrangling.

[Add Datasets](#)

From the import dataset window, you can either drag and drop both the csv provided or choose the file from your computer. Once you have done that, make sure your dataset is selected, then click Import and add to flow.

Import Data and Add to Flow

Upload

+ New PRO

Upload from your computer

Drag & drop a file here or

Choose a file

Maximum upload file size: 100MB

NAME	SIZE
+ Ecomm_Sales_Data.csv	22MB

1 New Dataset

Clear All

Ecomm\_Sales\_Data.csv

Sales Data

#	column2	RBC	event_time
0			2019-10-11 16:57:06
1			2019-10-27 14:23:53
2			2019-11-25 12:57:47
3			2019-11-12 19:50:55
4			2019-11-13 04:59:13

Edit settings

Import & Add to Flow

Cancel

We can connect Trifacta to many more data sources like databases and cloud storages but in the pro version.

Then import the second csv too, it should look like this once you are done.

2019 Ecomm Sales Insights Demo  
2019 Ecomm Sales Insights Demo

Dataset  
Ecomm\_Sales\_Data.csv

Dataset  
user\_details.csv

Details

Ecomm\_Sales\_Data.csv  
Sales Data

Add View dataset details

Data Preview

#	column2	ABC	event_time	ABC en
0			2019-10-11 16:57:06 UTC	view
1			2019-10-27 14:23:53 UTC	view
2			2019-11-25 12:57:47 UTC	view
3			2019-11-12 19:50:55 UTC	view
4			2019-11-13 04:59:13 UTC	view

## Recipes

Trifacta Wrangler works by generating *scripts* that apply *transformations* to your data and then compiling multiple scripts into a *recipe*. Multiple datasets and recipes are organized in a *flow*.

Trifacta Wrangler will automatically generate an initial recipe for your dataset that will convert it from its original format to something Wrangler can transform. Because it is a CSV, this recipe will include steps such as converting newline characters and commas into new rows and columns. However, it will generate similar scripts from JSON files as well.

Details

Ecomm\_Sales\_Data.csv

Sales Data

Add

View dataset details

...

Recipe

Join

Union

view

Column2	ABC	event_time	ABC ev
		2019-10-11 16:57:06 UTC	view
1		2019-10-27 14:23:53 UTC	view
2		2019-11-25 12:57:47 UTC	view
3		2019-11-12 19:50:55 UTC	view
4		2019-11-12 19:50:55 UTC	view

Click on Add > Recipe > Edit recipe to bring up the transform builder and a preview of the dataset. Here is a quick rundown of the editor's important features.

Details

Ecomm\_Sales\_Data

Edit Recipe

Add

...

Recipe

Data

Steps Preview

### Data Quality indications

- For each column, Wrangler displays the percentage of the data that is *valid* (the same format as the selected or inferred data type), *invalid* (a different format), and *empty*.
- This is visible for each column directly below the column name. If the bar is completely green, the data is 100% valid values; invalid values are red and empty values are gray.

2019 ECOMM SALES INSIGHTS DEMO > Ecomm\_Sales\_Data Initial Sample Run Job

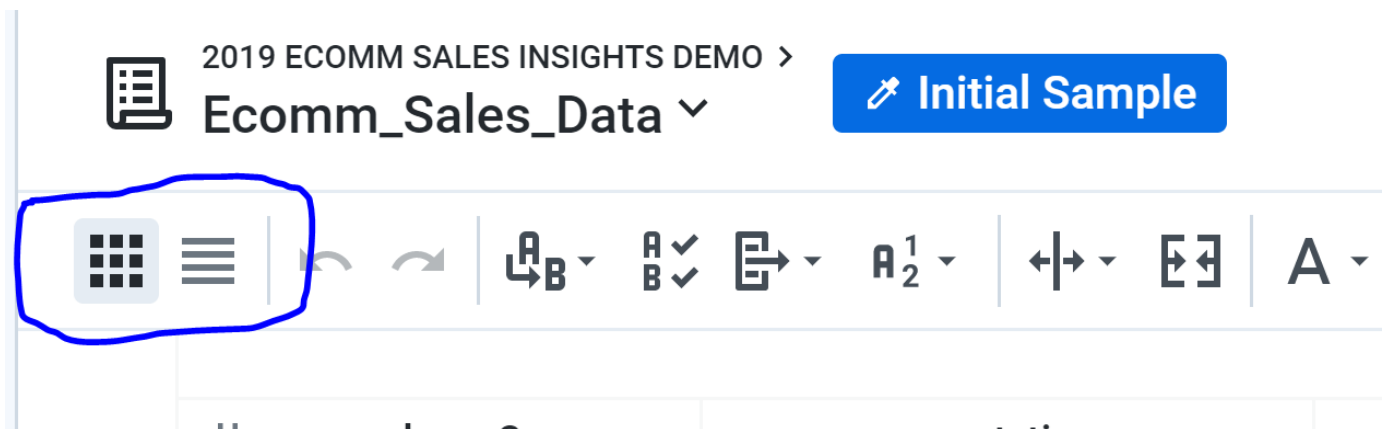
#	column2	ABC	event_time	ABC	event_type	#	product_id	#	category_id	category_code
0 - 108.1k		106,144 Categories		3 Categories		1 - 26.3k		0 - 305		71 Categories
0		2019-10-11 16:57:06 UTC		view		5016		183		appliances.kitchen.oven
1		2019-10-27 14:23:53 UTC		view		4240		43		electronics.video.tv
2		2019-11-25 12:57:47 UTC		view		7721		208		appliances.environment.vacuum
3		2019-11-12 19:50:55 UTC		view		597		71		electronics.smartphone
4		2019-11-13 04:59:13 UTC		view		13709		70		electronics.telephone
5		2019-11-20 08:00:18 UTC		view		871		71		electronics.smartphone

## Data Types

- Wrangler will also guess at the data type of each of your column and display an icon of the data type next to the column name.
- Knowing the data type helps Wrangler suggest which transformations might be useful/applicable.
- There are many different supported types, which can be viewed, along with more on data types [here](#).

## Grid vs. Column view

- With large datasets and/or datasets with many columns, you may want to see only a few of them when writing transformations.
- Selecting *columns* will allow you to select which columns are visible in the grid view.









- It will also give you the data quality indicators for each column.

## Transformation Builder

- This is the main feature of the tool. It allows you to choose from broad types of transformations and then customize them to edit your own data.
- Below, I'll walk through a quick tutorial of how to wrangle the sales data set you've just imported. I'll include both how to make and customize the transformations and the wrangle script that accompanies them.

First things first: matching the data types.

#	product_id	#	category_id	category_code	ABC	brand	##	price	#	user_id	#
											
1 - 26.3k		0 - 305		71 Categories		50 Categories	1 - 2,574		99 - 4.06M		15 - 15.
	5016	183	appliances.kitchen.oven	artel		36.01		1095536			
	4240	43	electronics.video.tv	lg		2445.08		1153084			
	7721	208	appliances.environment.vacuum	philips		257.38		364298			
	597	71	electronics.smartphone	huawei		163.2		3536496			

For the most part, Wrangler is good at guessing your data types, but it thinks 'category code' as a url.

- In the transform builder, choose the settype transform, *category code* as the column, and enter *Integer* as the new type, then select add to the recipe.

```
settype col: category_code type: 'String'
```

Run Job

Recipe

X

...

The recipe is empty

Add New Step

Search Transformations

ri

sett

Change column type

Change the data type of a column

Search documentation for "sett"



🔍 ☰ ✎
Run Job

🔍 ☰ ✎
⌵ ⌵ ⌵

Preview		
product_id	ABC	category_code
86 Categories		
183	appliances.kitchen.oven	
43	electronics.video.tv	
208	appliances.environment.vacuum	
71	electronics.smartphone	
70	electronics.telephone	
71	electronics.smartphone	
43	electronics.video.tv	
177	appliances.kitchen.washer	
182	appliances.kitchen.refrigerator	
71	electronics.smartphone	
71	electronics.smartphone	
71	electronics.smartphone	
71	electronics.smartphone	
77	construction.tools.saw	
71	electronics.smartphone	

<
Change column type
✕

**Columns** required

Multiple

🌐 category\_code
✕ ✕

**New type** required

String

Cancel
Add

Next, Lets convert the text format to proper case for 3 columns -  
Search text format and add the column names

📅 2019 ECOMM SALES INSIGHTS DEMO >
Initial Sample
🔍 ☰ ✎ Run Job

🔍 ☰ ✎
⌵ ⌵ ⌵

Initial Sample
Rows 108099

Collected on Today at 2:35 PM
Collected by Nikhil Kohli

product_id	category_code	category_code
86 Categories		
5016	appliances.kitchen.oven	Appliances.kitchen.oven
4240	electronics.video.tv	Electronics.video.tv
7721	appliances.environment.vacuum	Appliances.environment.vacuum
597	electronics.smartphone	Electronics.smartphone
13709	electronics.telephone	Electronics.telephone
871	electronics.smartphone	Electronics.smartphone
4152	electronics.video.tv	Electronics.video.tv
7462	appliances.kitchen.washer	Appliances.kitchen.washer
5485	appliances.kitchen.refrigerators	Appliances.kitchen.refrigerators
533	electronics.smartphone	Electronics.smartphone
720	electronics.smartphone	Electronics.smartphone
51	electronics.smartphone	Electronics.smartphone
707	electronics.smartphone	Electronics.smartphone
19294	construction.tools.saw	Construction.tools.saw
717	electronics.smartphone	Electronics.smartphone

<
Text format
✕

**Columns** required

Multiple

ABC event\_type
✕ ✕

ABC category\_code
✕ ✕

ABC brand
✕ ✕

**Format** required

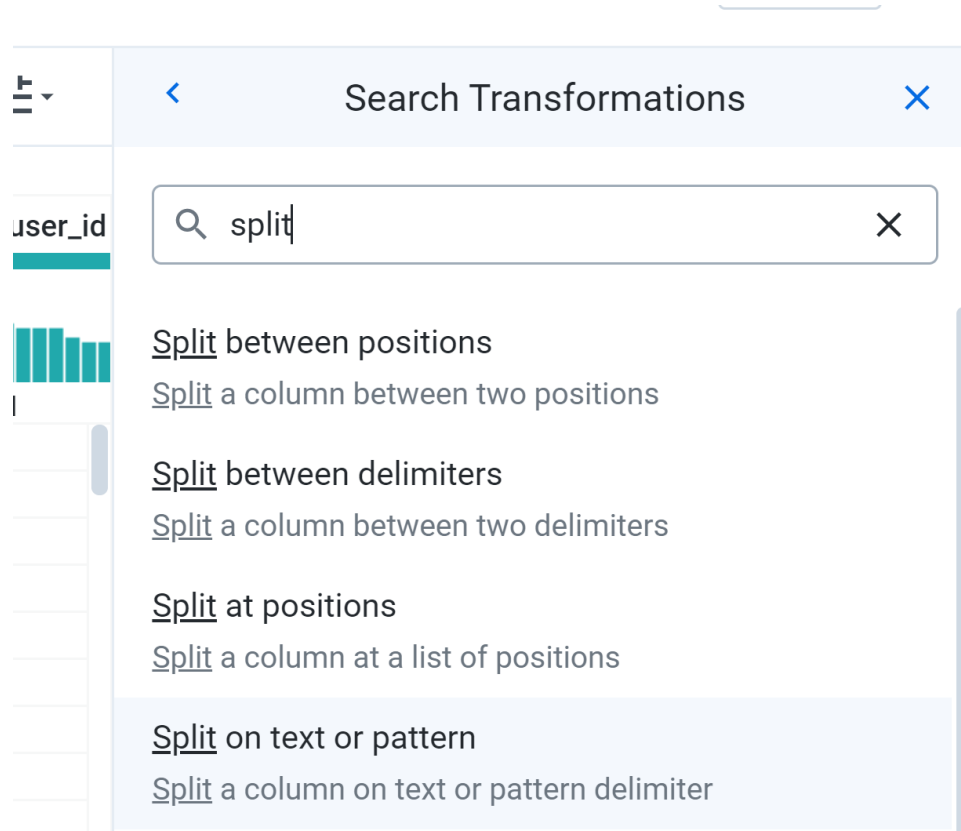
Convert to Proper Case

Convert text in column to ProperCase

Cancel
Add

## Splitting and feature Extraction

Now, we will split the date time into date and time columns so we can use the date for extracting further columns -



Search - Split in the transformations and select - 'split between delimiters'



ABC

event\_time2

event\_time1

ABC

55,951 Categories

00:00 - 23:00

1 Category

16:57:06 UTC	16:57:06	UTC
14:23:53 UTC	14:23:53	UTC
12:57:47 UTC	12:57:47	UTC
19:50:55 UTC	19:50:55	UTC
04:59:13 UTC	04:59:13	UTC
08:00:18 UTC	08:00:18	UTC
15:05:15 UTC	15:05:15	UTC
08:10:32 UTC	08:10:32	UTC
16:04:19 UTC	16:04:19	UTC
21:24:25 UTC	21:24:25	UTC
16:33:30 UTC	16:33:30	UTC
03:30:54 UTC	03:30:54	UTC
08:11:44 UTC	08:11:44	UTC
14:36:18 UTC	14:36:18	UTC
18:08:36 UTC	18:08:36	UTC

Split by delimiter

Column

required

ABC event\_time2

Option

required

By delimiter

Delimiter

required

'|'

Advanced options

Cancel

Add

Rename the column to event\_timezone

Rename columns

Option

required

Manual rename

Specify the new name for each column

Columns (1)

Add

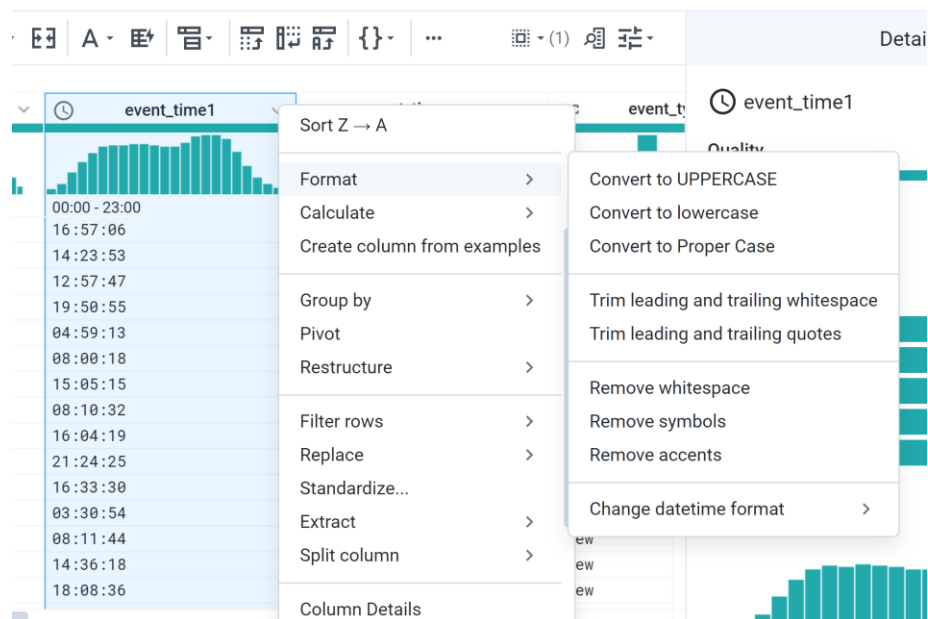
ABC event\_time3

event\_timezone

Cancel

Add

Extract hour from event\_time



Extract > Datetime > Hours

Extract Day of week column from event date

Extract > Datetime > Day of Week

New formula

Formula type required

Single row formula

Create a new column from a single row formula

Formula required

WEEKDAY(event\_date)

New column name

DayofWeek

Cancel Add

Now lets use if conditions to update the day of week

```
IF(DayofWeek == 1, 'Monday', IF(DayofWeek == 2, 'Tuesday', IF(DayofWeek == 3, 'Wednesday',  
IF(DayofWeek == 4, 'Thursday', IF(DayofWeek == 5, 'Friday', IF(DayofWeek == 6, 'Saturday',  
'Sunday'))))))))
```

Now lets join this dataset with the other file - user\_data

- Search Join Datasets

Join - Add New Step

Choose dataset or recipe to join with Ecomm\_Sales\_Data

Search...

Recipes in current flow Datasets in current flow All datasets

Name	Last Updated	Source	Data
✓ user_details.csv	Today at 2:33 PM	Upload	# column2 # user_id State # User_
Ecomm_Sales_Data.csv	Today at 2:31 PM	Upload	0 1095536 PA 3
			1 1153084 CT 4
			2 364298 HI 3

Cancel Accept

We can select different types of joins as well. Here we will select Left join to get all the data from our main table

Join - Keys & Conditions

Search row values...

Join Key

#	user_id	#	user_id
99 - 4.06M	1095536	99 - 4.06M	1095536
	1153084		1153084
	364298		364298
	3536496		3536496
	3542250		3542250
	1246770		1246770
	468520		468520
	2388173		2388173
	2382829		2382829
	2162691		2162691
	3145632		3145632
	1589151		1589151
	1119887		1119887

Join type: Left

Join keys: # user\_id

= (Equal to)

Suggested 67% match

Results summary

Based on current samples

Rows in Current	Rows in Joined-in	Rows in Output
108098	204630	108098

Back Next

108,098 Rows in 204,630 Rows in 108,098 Rows in Output

Show only: ☒ Included Rows ☐ Excluded Rows

Select the columns you want to add from other table

Columns

Search columns

All (18) Current (14) Joined-In (4)

<input type="checkbox"/>	Column	Source
<input checked="" type="checkbox"/>	brand	Current (14)
<input checked="" type="checkbox"/>	price	Current (14)
<input checked="" type="checkbox"/>	user_session	Current (14)
<input type="checkbox"/>	column2	Joined-In (4)
<input checked="" type="checkbox"/>	State	Joined-In (4)
<input checked="" type="checkbox"/>	User_Score	Joined-In (4)

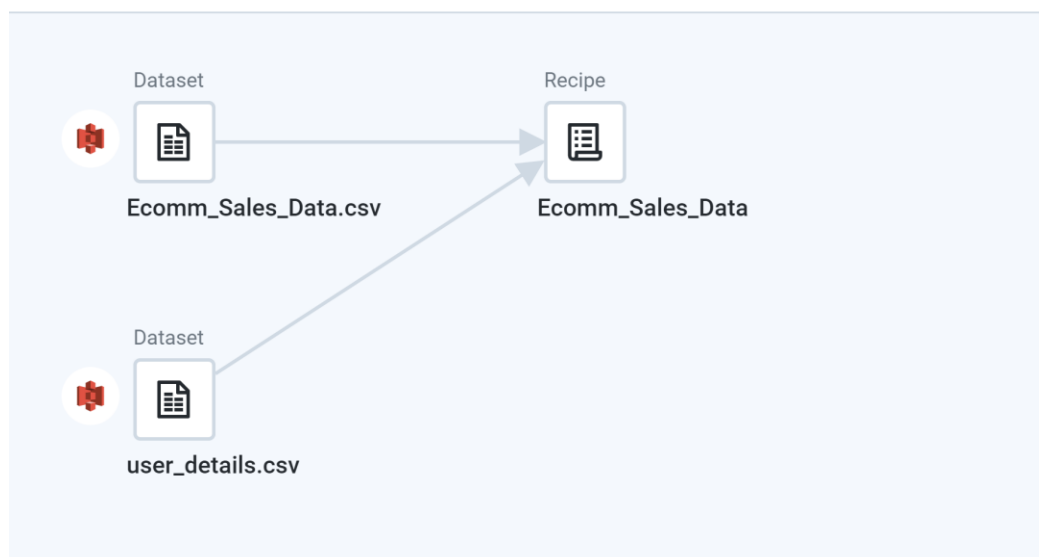
Advanced options

Back Review

If you check the flow now it will look something like this

## 2019 Ecomm Sales Insights Demo

2019 Ecomm Sales Insights Demo



## Download the transformed data -

After making any additional modifications to the dataset (find the full range of transformations [here](#)), you can generate results for the entire dataset by clicking Generate Results.

- You can choose the format to download your data, an optional method of compression, and view a summary of the dataset.

The screenshot shows the '2019 Ecomm Sales Insights Demo' interface. At the top, there's a search bar, an info icon, a zoom level of 100%, and buttons for 'Add Datasets' and a menu. The main workspace displays a data pipeline: two 'Dataset' nodes ('Ecomm\_Sales\_Data.csv' and 'user\_details.csv') feed into a 'Recipe' node ('Ecomm\_Sales\_Data'), which then feeds into an 'Output' node ('Ecomm\_Sales\_Data'). On the right, a 'Details' panel for 'Ecomm\_Sales\_Data' is open, showing a 'Run Job' button, tabs for 'Destinations' and 'Jobs', and a 'Manual Destinations' section with a table.

Manual Destinations	
Create-CSV	
Profiling	yes

You can generate json or csv file as well for your output

The screenshot shows the 'Publishing settings' panel. It includes an 'Options' section with a checked 'Profile results' checkbox and a note: 'When enabled, this will generate a profile of your results'. Below is the 'Publishing Actions' section with two buttons: 'Create CSV' and 'Create JSON'. At the bottom is a table with columns 'Actions' and 'Settings'.

Actions	Settings
Create-CSV	no compression, single file, with headers, with quotes, with delimiter: ,
Create-JSON	no compression, single file

Click on run job





Details

Ecomm\_Sales\_Data

[Run Job](#) [...](#)

[Destinations](#) [Jobs \(1\)](#)

[Job 225069](#) • In progress  
Started Today at 6:11 PM

Click on that job id

2019 Ecomm Sales Insights Demo > Ecomm\_Sales\_Data  
Job 225069  
Started Today at 6:11 PM

Cancel job

[Overview](#) [Output Destinations](#) [Profile](#) [Dependencies](#)

### In progress stages

Transforming with profile...  
Started Today at 6:11 PM • In progress for 3 min  
Environment Spark  
[View steps and dependencies](#)

Publishing...  
Started Today at 6:11 PM • In progress for 3 min  
Activity  

Ecomm_Sales_Data.json	In progress
Ecomm_Sales_Data.csv	In progress

  
[View all](#)

### Job summary

Job ID 225069  
Job status In progress  
Flow [2019 Ecomm Sales Insights D...](#)  
Output [Ecomm\\_Sales\\_Data](#)

### Execution summary

Job type Manual  
User Nikhil Kohli  
Start time September 18th 2020, 6:11 pm  
Last update September 18th 2020, 6:11 pm  
Duration 3 minutes

2019 Ecomm Sales Insights Demo > Ecomm\_Sales\_Data  
Job 225069  
Finished Today at 6:24 PM

Download results

...

[Overview](#) [Output Destinations](#) [Profile](#) [Dependencies](#)

### Completed stages

Transform with profile  
Completed Today at 6:24 PM, started Today at 6:11 PM • Ran for 13 min  
Environment Spark  
  
99.9% valid values 0.1% mismatching values 0% missing values  
[View steps and dependencies](#) [View profile](#)

Publish  
Completed Today at 6:24 PM, started Today at 6:24 PM • Ran for <1 sec  
Activity  

Ecomm_Sales_Data.json	Completed
Ecomm_Sales_Data.csv	Completed

  
[View all](#)

### Job summary

Job ID 225069  
Job status Completed  
Flow [2019 Ecomm Sales Insights D...](#)  
Output [Ecomm\\_Sales\\_Data](#)

### Execution summary

Job type Manual  
User Nikhil Kohli  
Start time September 18th 2020, 6:11 pm  
Finish time September 18th 2020, 6:24 pm  
Last update September 18th 2020, 6:24 pm  
Duration 13 minutes

Click on Download results and it will download the json and csv for output

## Other Transformations that you can apply

1. One hot encoding:

Source

Preview

#	IsPenalty	#	0	#	1	#	IsPenaltyAccepte_New
0-1	<div></div>	0-1	<div></div>	0-1	<div></div>	0-1	<div></div>
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null
	0		1		0		null

One-hot encode values to columns

Column

required

# IsPenalty

Max number of columns to create

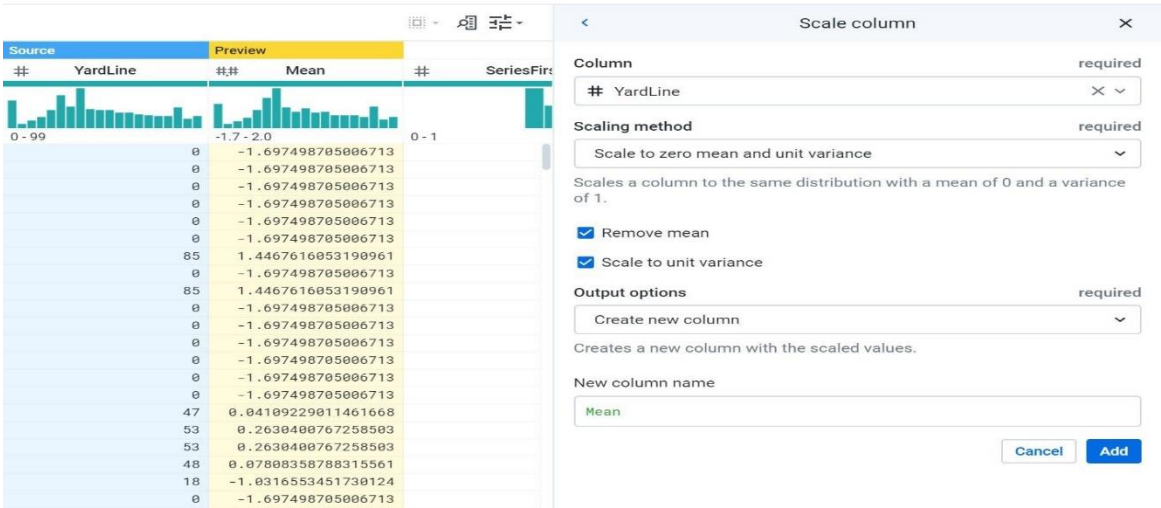
required

2

Cancel

Add

2. Scale Column:



3. Window:

<

Window

×

Formulas

required

Formulas

We support functions typically found in most desktop spreadsheet packages. Functions can be used to create formulas that manipulate data in your columns. [Learn more](#)

Examples

`IF(age_col < 18, "Minor", "Adult")`

`(unit_price_col - unit_cost_col) * number_units_col`

Browse

[Functions](#) [Columns](#)

#### 4. Filtering Rows:

##### Filter rows

##### Remove duplicate rows

Remove duplicates if values in every column are the same

##### Filter contains

Filter rows which contain a value

##### Filter custom formula

Filter rows using a custom formula

##### Filter ends with

Filter rows which end with a value

##### Filter exact

Filter rows which exactly equal a value

##### Filter not equals

Filter rows which do not equal a value

##### Filter from top

Filter rows from the top of a dataset

##### Filter greater than

Filter rows which are greater than a value

##### Filter at interval

Filter rows at regular intervals

##### Filter less than

Filter rows which are less than a value

##### Filter missing

Filter rows with missing values

##### Filter mismatched

Filter rows with mismatched values

##### Filter in

Filter rows which are in a list of values

##### Filter range

Filter rows within a range of values

##### Filter starts with

Filter rows that start with a value

#### 5. Functions:

**MODE**  
(Function) Returns the most frequent value for each group. If multiple values ...

**NULL**  
(Function) Returns a null value

**NUMFORMAT**  
(Function) Converts the value into a custom number format

**PAD**  
(Function) Pads a value to a specified length

**PARSEDATE**  
(Function) Parses a string into a datetime object

**PI**  
(Function) Returns the value of PI to fifteen decimal places

**PROPER**  
(Function) Converts a string to Propercase by capitilizing the first letter of eac...

**RADIANS**  
(Function) Converts degrees into radians

**RAND**  
(Function) Returns a random number between 0 and 1

**RANDBETWEEN**  
(Function) Returns a random integer between two specified integers, inclusive

**RANGE**  
(Function) Returns an array of integers sequenced between two values by a st...

**REMOVESYMBOLS**  
(Function) Removes all characters that are not alphanumeric or whitespace fr...

**REMOVEWHITESPACE**  
(Function) Removes all whitespace characters from a string

**REPEAT**  
(Function) Repeats a string a specified number of times

**RIGHT**  
(Function) Returns a sub-string from the end of a string

**RIGHTFIND**  
(Function) Returns the position at which a string or a pattern is last found with...

**ROUND**  
(Function) Rounds the value to the specified decimal place

**SIN**  
(Function) Returns the sine of an angle provided in radians

**SINH**  
(Function) Returns the hyperbolic sine of the value provided

---

## References

<https://docs.trifacta.com/display/SS/Documentation>

