Playbook

The Dataiku Playbook

What will you find in this Playbook?

Our objective at Dataiku is to create a tool that helps data teams — analysts, scientists, and managers — to collaborate on data projects. One of the key success factors for these teams is to allow analysts to work on big data as easily as they do on smaller data with Excelas well as to help them find new use cases specific to the data available and the tools at hand. Hence, the Dataiku Playbook. The goal of this document is to be a reference to business and data analysts while using Dataiku Data Science Studio (DSS). It's divided into two parts:

1. From Excel to Dataiku: a mapping of usual Excel functions
2. Select use cases for analysts

Part 1: From Excel to Dataiku

Before you start

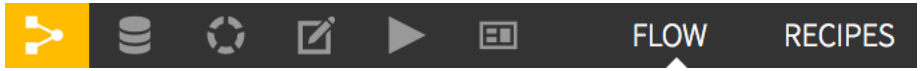
- Navigating Projects
- From the Lab to the Flow
- Recap on Dataiku Recipes
- Sampled vs Complete Data

Navigating Projects

Dataiku is organized by projects, which you can see from your home page. Once you click on a project, you'll access a project homepage with information about it, including a timeline. From here, navigate the project from the toolbar:

Datasets

In this section you will find all your datasets. They can be stored in a database to which Dataiku connects or they can be files.



The Flow and Recipes

Working in Dataiku, you'll build Recipes with data manipulation steps applied to an input dataset, thus creating an output dataset. Build and see these relations and dependencies between datasets in the Flow.
This is the main space in Dataiku.

The Lab: Visual Analysis

The Visual Analysis Lab is where you build your data preparation steps, enrich your data and create new features, explore with visualizations, and experiment with machine learning models. All this is done in a code-free visual interface to accelerate your work.

The Lab: Code Notebooks

This is a place to explore data with code. Notebooks integrated with Dataiku include SQL, Hive, Impala, Python, and R. Notebooks make it possible to save and share code snippets with teammates or to create visualizations.

Jobs and Scenarios

Here you'll find logs of every Job you have run.
This is also where you can easily set up complex automation Scenarios to make your work reproducible, and Monitor automated tasks.

Dashboards & Insights

Share insights about your data with Dashboards. You can publish graphs, notebooks, or even datasets to share with your team. You can also code advanced data visualisations and web apps.

From the Lab to the Flow

With Dataiku, projects have two phases: 1. Experimenting and exploring, and 2. Building stable and reproducible data manipulations. This is represented in Dataiku by two spaces: the Lab and the Flow.

LAB

The Lab

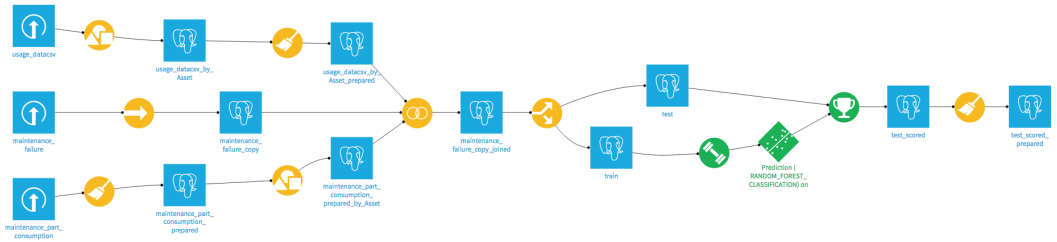
A sandbox where you can explore your data, experiment, and begin your analysis. There are two different Lab environments: code-based notebooks and visual analysis. Both environments allow you to prepare data iteratively and build charts or models on that prepared data.

The Flow

Once you've done analysis in the Lab that you're happy with, you have to deploy it on your whole dataset and run it to build a new dataset based on your preparation steps.

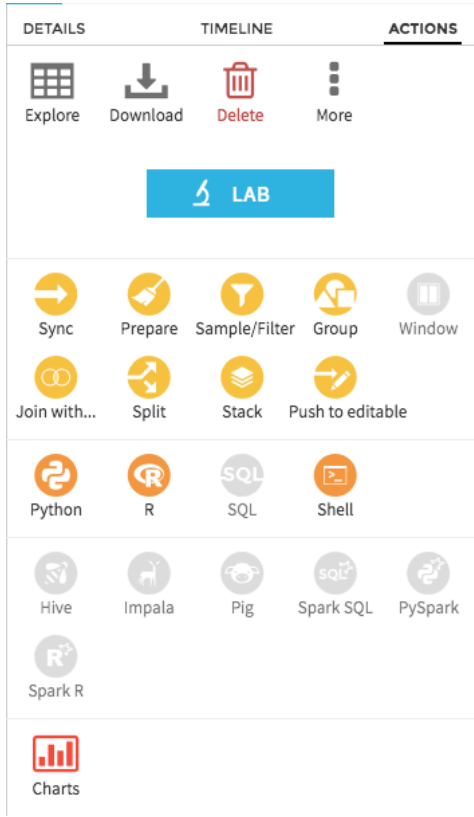
It then becomes a Prepare Recipe and is represented in the Flow. Once it's in the Flow, it's no longer an experimental sandbox; it becomes a stable data manipulation with dependencies that are visually represented (an input and an output).

In your Flow you'll see all datasets and Recipes (whether code or visual) as well as models and managed folders:



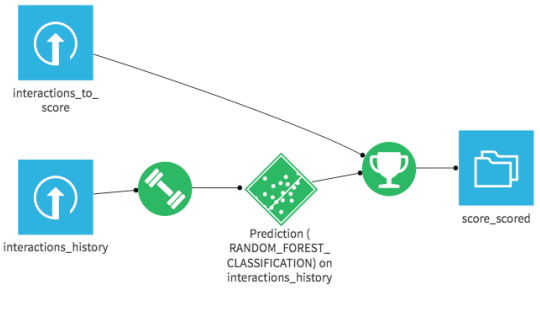
Recap on Dataiku Recipes

Lost in your Flow? A quick review of Recipes and datasets in Dataiku.



Visual Recipes are represented by yellow icons:

- ***Sync Recipe****: To move datasets between different storage databases*
- ***Prepare Recipe****: Use visual processors to clean your data fast*
- ***Sample/filter Recipe****: Extract parts of your dataset based on your criteria - see Filter*
- ***Group Recipe****: Aggregate data based on a chosen key - see Aggregations*
- ***Window Recipe****: Apply sliding (e.g. in time) operations - see Sliding calculations, order, and ranking*
- ***Join with… Recipe****: Enrich a dataset with the information from another - see Lookups and IndexMatch*
- ***Split Recipe****: Split datasets in multiple smaller ones - see Splitting datasets*
- ***Stack Recipe****: Stack multiple datasets into a bigger one -see Merging datasets*
- ***Push to editable Recipe****: Create small datasets where you can change the value of a cell (used to create small referential tables)*



There are also **code Recipes** available to manipulate your data.

Finally, you can **build models** in the Lab. When deployed, they appear as green icons in the flow.

Sampled vs Complete Data

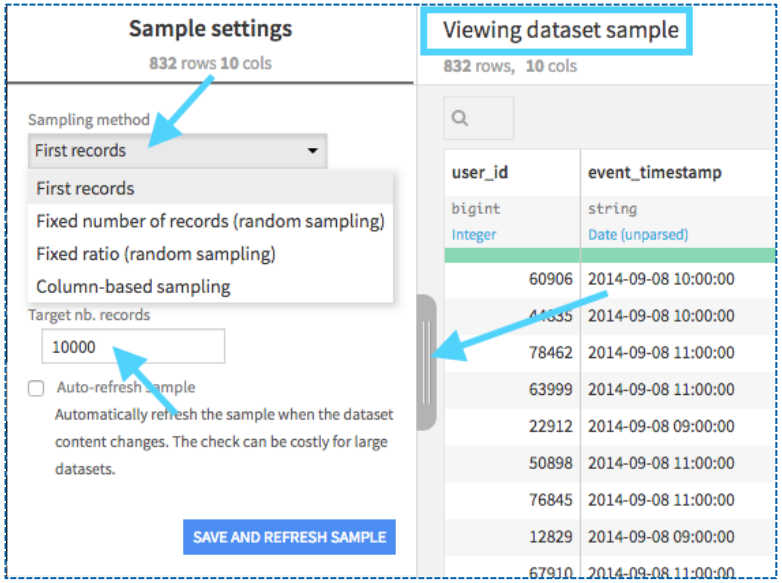
Real-time visual feedback of your operations in Dataiku happens on a sample of your datasets and not the complete dataset.

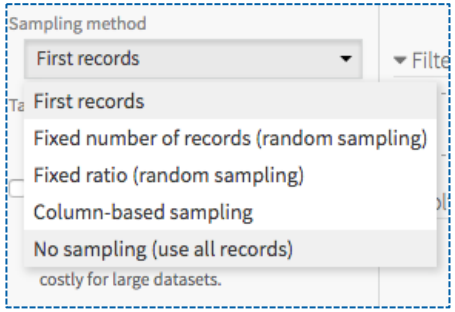
Why is that?
Because it would take too much computing power to get immediate visual feedback on calculations for a dataset that's very big.

When you explore a dataset, you will see only a sample. By default, the first 10,000 records of your dataset are selected for the sample.

You can change the sample size or sampling method by clicking on the grey tab to the left of your dataset and then selecting a different sampling method from the dropdown or adjusting the size of your ***Target nb. Records***.

Note that when you run a recipe, it will run on the entire dataset – only the live visual feedback uses a sample.





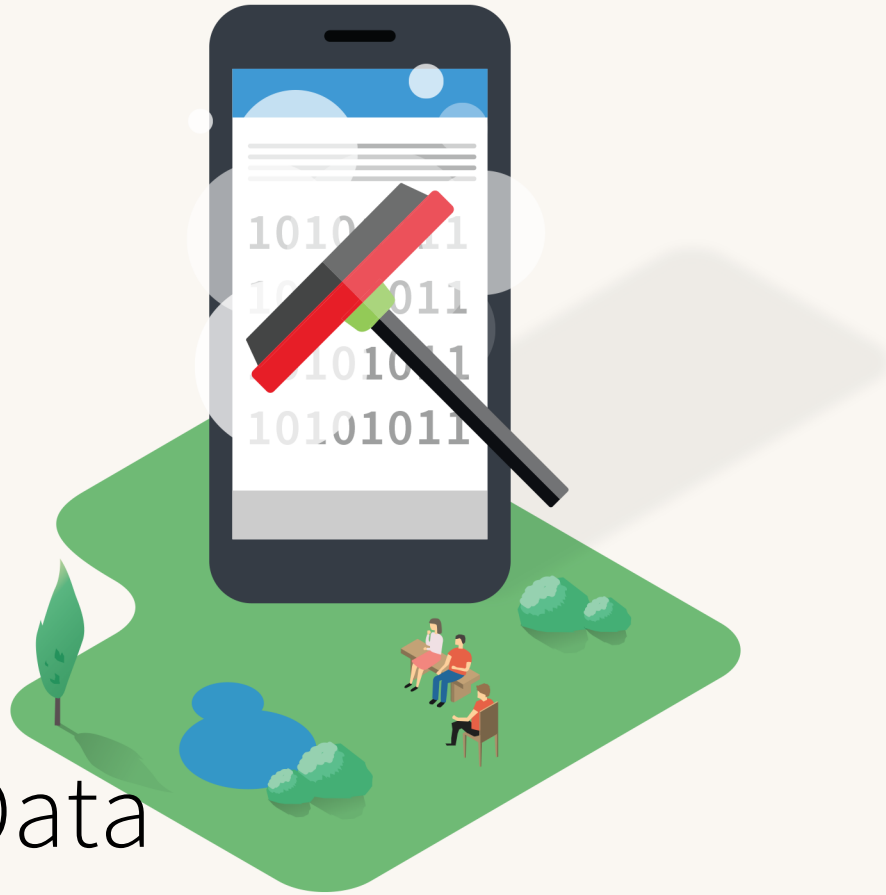
The Lab

When you perform visual analysis in the Lab, it's done on a sample as well; you can change the sampling method like you would for the dataset view.

Charts

Charts created with visual analyses in the Lab are computed on samples (in memory). Charts created in the Charts tab of a dataset can be computed using specific engines (in database).

Part 1: From Excel to Dataiku DSS



A. Cleaning Data

Renaming Columns

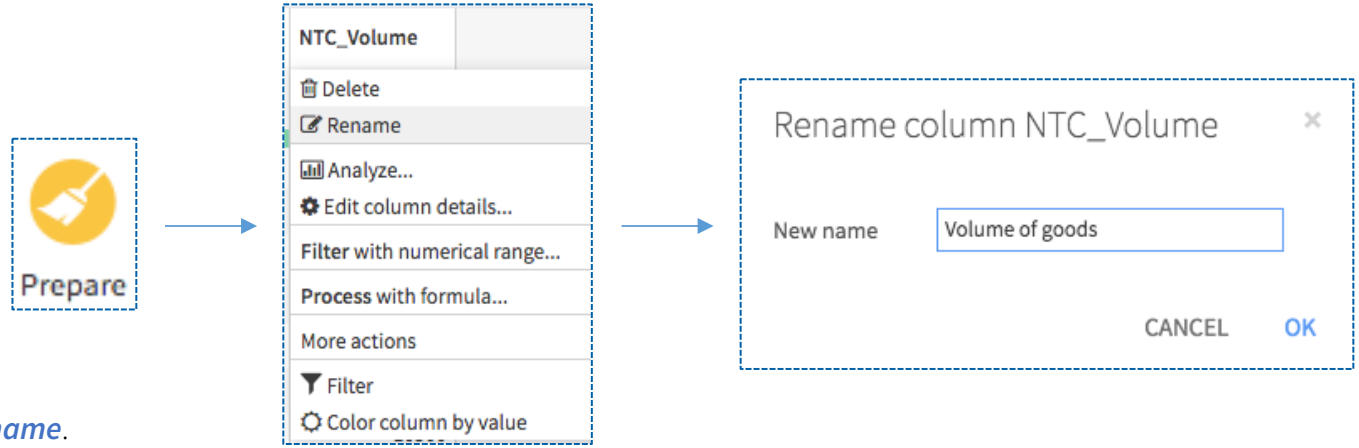
It's often helpful to rename newly created columns; for example, "since_birth_parsed_days" could be renamed "age." And it's as simple to do with Dataiku as it is in Excel.

Excel

- Click in the top cell of the relevant column.
- Type in the new name.

Dataiku

- In a ***Prepare Recipe***.
- Click on the column header, and select ***Rename***.
- Type the new name in the pop-up, and click ***OK***.



Note
Most processors allow you to name your output column, overriding the default name.

Since accents and, in general, non-Unicode characters in column names can sometimes lead to errors, it's always good to get rid of them.

Assessing Data Quality

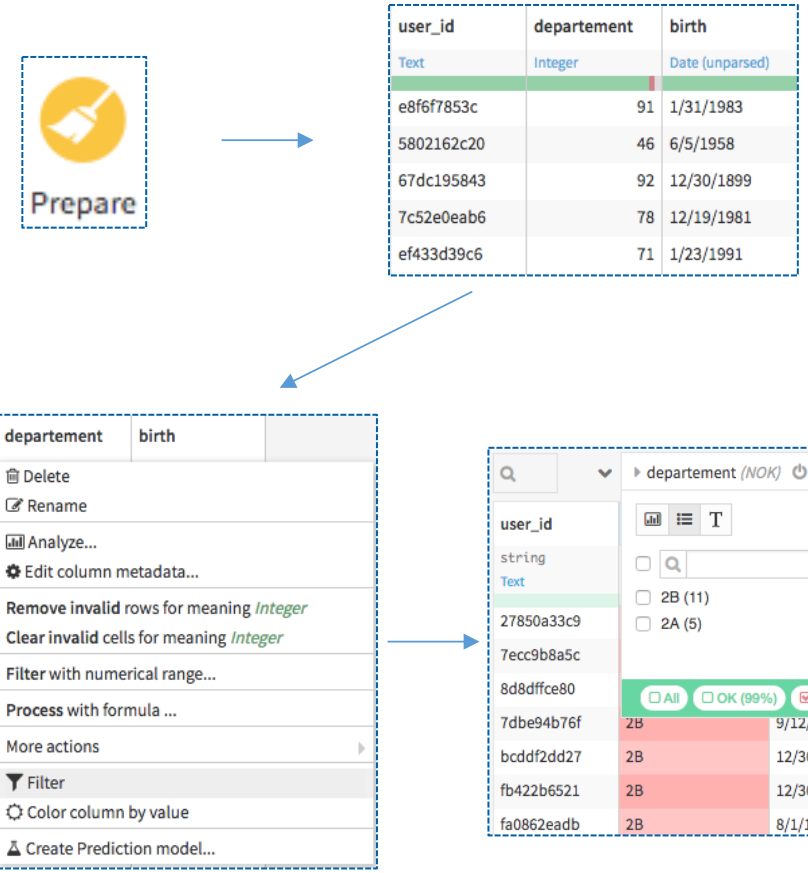
It's good practice to make sure data is complete and in the right format before beginning analysis. Dataiku has some features that make this process straightforward.

Excel

- Use the ***Filter & Sort*** functions to identify blank data and different data types (e.g., text vs. numerical, data ranges, etc.) .

Dataiku

- In a ***Prepare Recipe***.
- At the top of each column is a bar representing **OK data as green**, **not OK data as red**, and **missing data as gray**.
- Click on the column header, then click on ***Filter***, and you will be able to examine the data classified as not OK.



Find and Replace

Find and replace is a useful function when you want to remove the dots and the underscores from a text column or, for example, replace "é" with "e" (no accent).

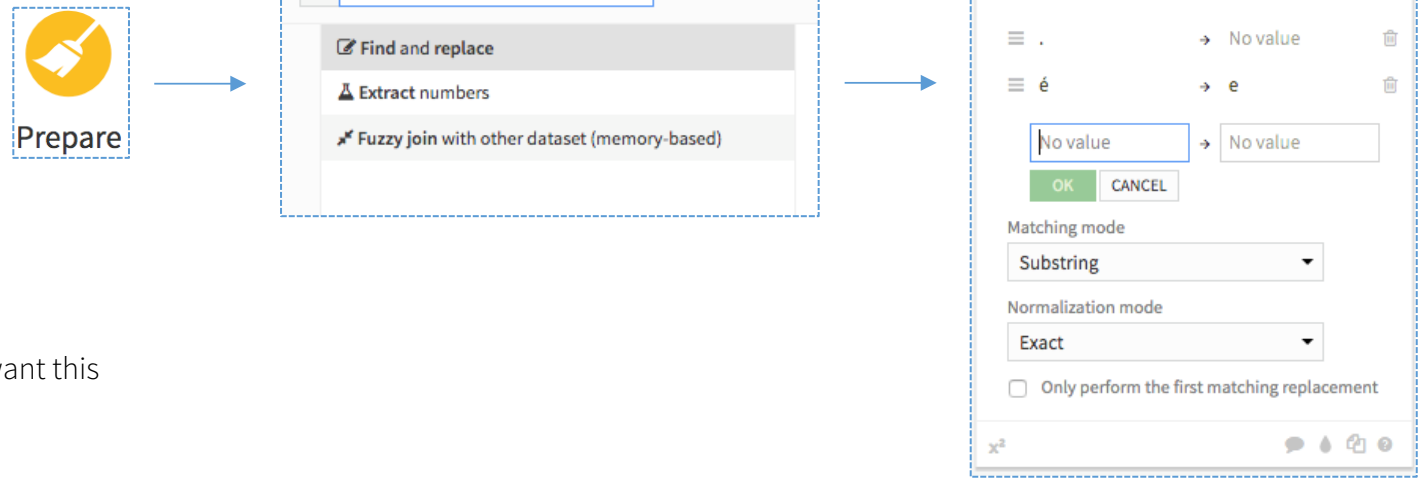
Excel

Perform three ***Find & Replace*** Operations:

- Replace dots with nothing
- Replace underscores with nothing
- Replace "é" with "e"

Dataiku

- In a ***Prepare Recipe***
- Use the find and replace processor
- Choose the columns on which you want this replacement to be performed



Note
A Dataiku find and replace preparation Recipe will be reapplied automatically if you replace your input data with new data

Merge and Clean Cell Values (1/2)

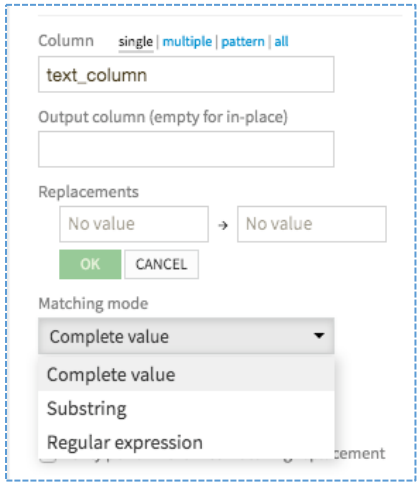
When working on a file with different data sources, for instance, you'll often find the same brand is written in various ways and want to unify these names to merge the data.

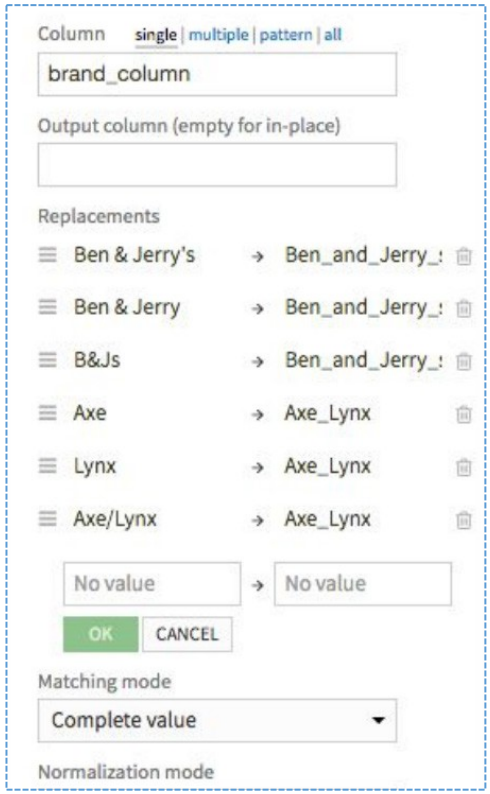
Excel

This is done in Excel using ***Find & Replace***. It can also be done manually or with complex IF conditioning

Dataiku

- In a ***Prepare Recipe***
- Use the find and replace processor
- Set the ***Matching mode*** to ***Complete value***



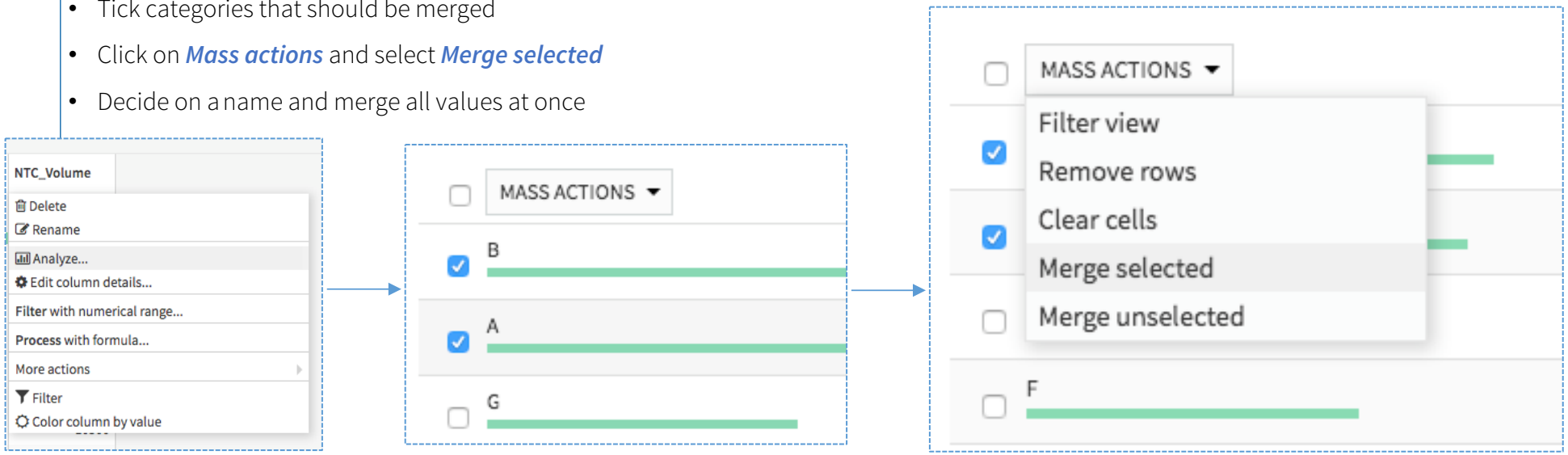


Merge and Clean Cell Values (2/2)

Alternate method:

Dataiku

- In a ***Prepare Recipe***
- Click on the column header and choose the ***Analyze*** option
- You will see the repartition of values in that column
- Tick categories that should be merged
- Click on ***Mass actions*** and select ***Merge selected***
- Decide on a name and merge all values at once



Note

If you find that you're doing this process often and the unification process is always the same, you can create a custom meaning to share across the whole Dataiku instance (see next page)

Create Custom Meaning (1/2)

Custom meanings are used to define which values are valid in a column, and they can also be used to create a mapping from the valid values to consistent ones.

✓ **For example…**

A dataset with country codes might consider the following values valid:

US, USA., UK, DE, AUS…

and any other strings not listed here as invalid. For example, AU is too ambiguous to be valid.

One could remap the codes to actual country names to end up with:

United States, United Kingdom, Germany, Australia…

Note
Custom meanings will apply to your entire Dataiku instance, i.e., all of your projects

Continue

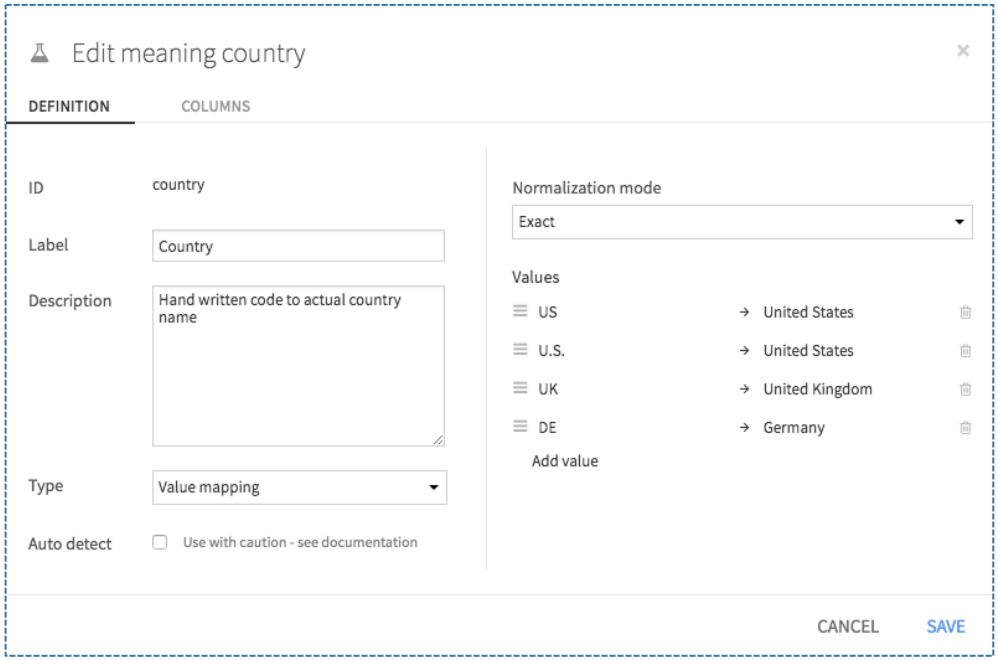
Create Custom Meaning (2/2)

Excel

- Use ***Find & replace***
- A better way to do this is to create a table with the valid values and their mappings and use ***Match*** to check if the values are valid and ***Index*** to retrieve the mapped values according to the match

Dataiku

- Create a new custom meaning: ***DSS Home Page*** > ***Catalog*** > ***Meaning***
- Set the meaning on a column header : click on header meaning and choose ***User defined***
- Map values: Click on the column header, and in the menu choose ***Translate***



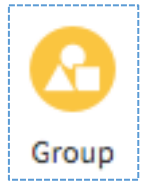
Note
You can also perform the mapping using a referential dataset and a join, but you won't have the data quality bar functionality

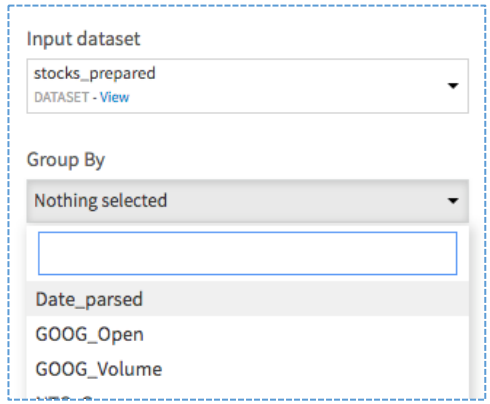
Remove Duplicates

Remove all lines in a dataset that are identical with remove duplicate functionality. You can also choose to remove rows that only have certain common values.

Excel

This is done in Excel using ***Remove duplicates***

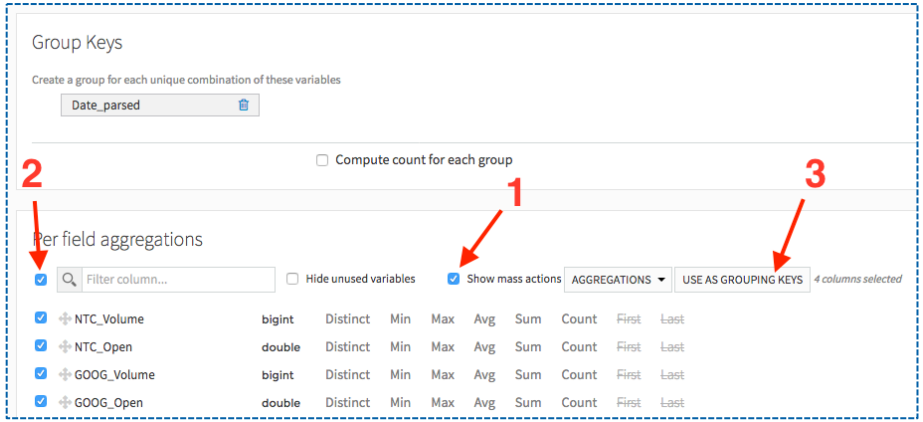




Dataiku

Dataiku uses ***Group Recipes*** to group together all lines that have the same values:

- Create a ***Group Recipes***
- Select the first column for the ***Group By*** field
- Add all other columns to ***Group Keys*** (you can use a mass action for this)
- To remove rows where some columns are the same, choose these columns only



Note
Group recipes are powerful and generally used to calculate aggregates (see Aggregate section)

B. Manipulating Data

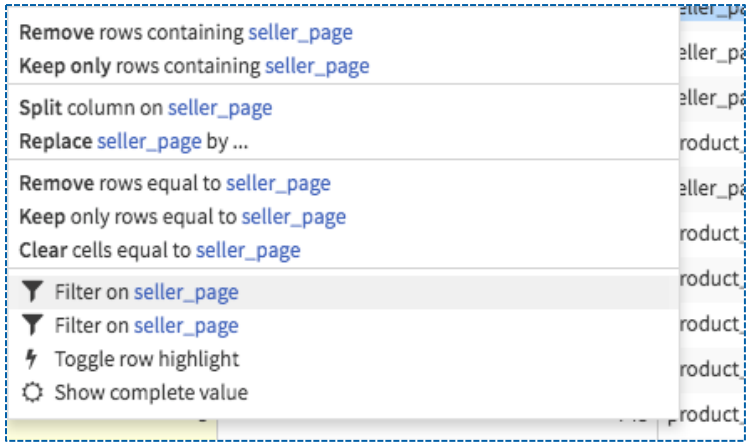
Filters (1/2)

Filtering data is downsizing a dataset by hiding rows based on rules. Filtering is useful during data exploration to display parts of the data without changing it or to create a new downsized dataset.

Excel

Filtering in Excel requires you to:

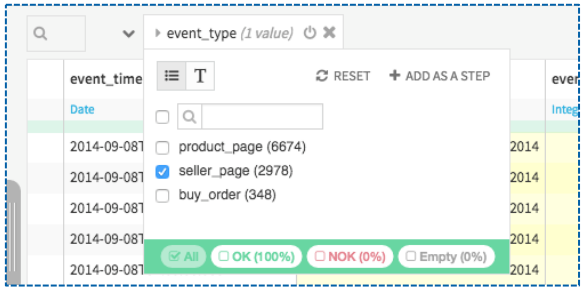
- Select the zone you want to filter
- Click on ***Sort & Filter***
- Then click on the column header you want to filter on



Dataiku

You can easily add filters in Dataiku:

- By adding a ***Filter processor*** on value, numerical range, or invalid data
- By right clicking on the value of a column you want to filter on and click on ***Filter on***- this will automatically detect whether it should filter on range or value



Note: There could be another way!

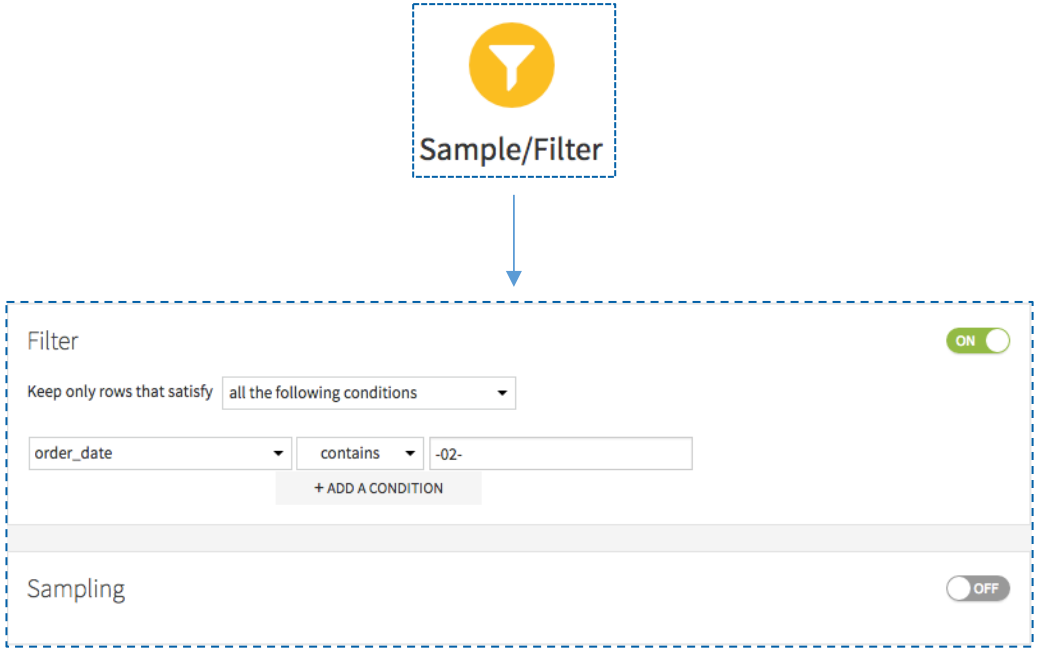
- If you're using a filter to see data only from a certain user or brand, for example, you can also use the search bar on top of each dataset
- If you're filtering to delete data, you can do this directly by right-clicking to Delete or Keep rows by value

Filters (2/2)

Filter recipes are used to downsize your data by creating a new, smaller dataset with only the data you need.

Dataiku

- Create a ***Sample/Filter Recipe***
- Turn filtering ***On*** and sampling ***Off***
- Choose the column you want to filter on, and enter the condition(s) you want to filter on
- In our example, let's say we want to see only rows from February of any year
- Click on ***Run***, and your filtered dataset is ready



Tip

Knowing how dates are rendered in this dataset, we know we that the condition "-02-" will only return records from February due to the hyphens on either side of the two-digit month

Note

There are filter steps integrated in *group Recipes* that will drop rows that do not fit the filter rules as a pre-processing step

Split Datasets

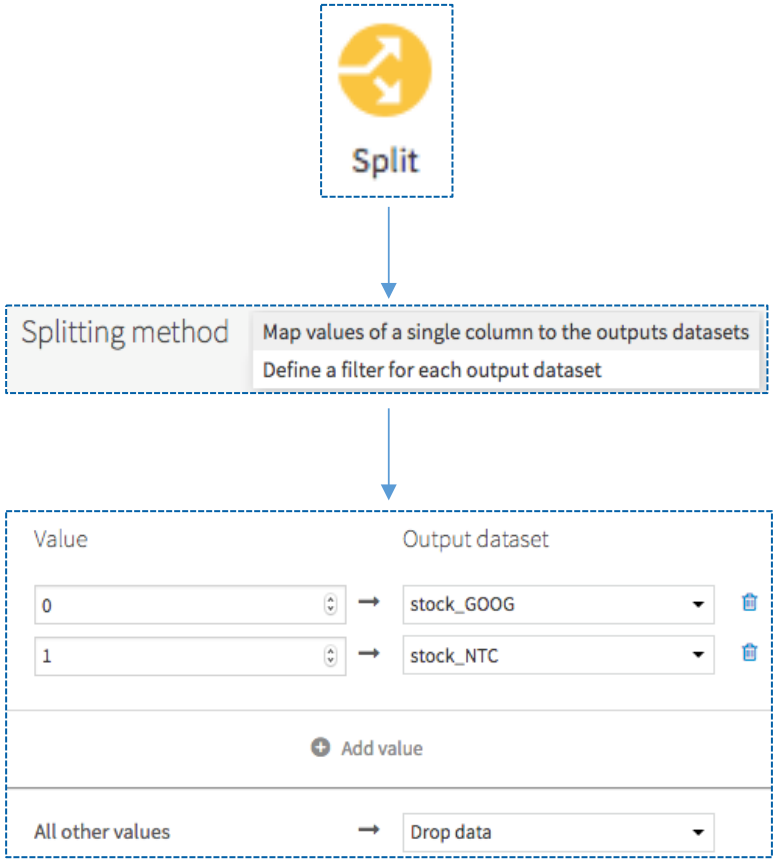
Splitting datasets creates multiple datasets by dispatching the rows according to the same rules.

Excel

- Select the data you need
- Copy it to a new file

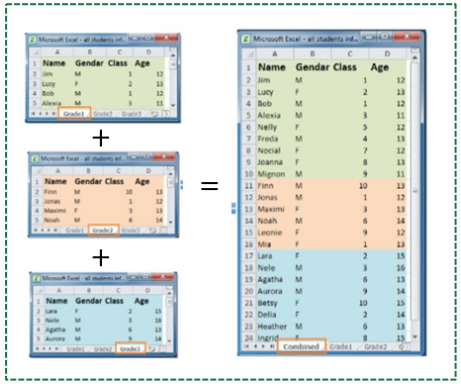
Dataiku

- Create a ***Split Recipe***
- Add as many outputs as you need
- Map values of a column to each output dataset
- Or define filters with conditions corresponding to your splitting criteria



Merge Datasets

Use the merge function to append multiple datasets into a single, larger dataset.

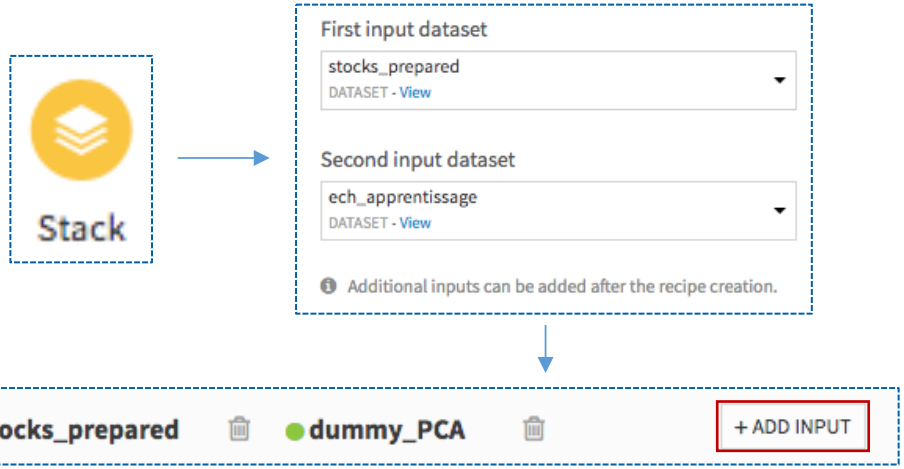


Excel

- Open two files
- Copy the content of the second one and paste at the end of the first one

Dataiku

- Create a ***Stack Recipe***
- Select the first and second datasets you want to stack
- Add all other files you want to merge by clicking on ***Add Input*** within the recipe



Note
You can easily add new files and automate the next steps in your process - when working on several datasets with the same format, stack all of them, perform cleanup and enriching steps on a single dataset, and split it back

Aggregate

By grouping your data based on the values contained in a column, you can compute aggregations (e.g., count the number of records, compute an average value, etc.).

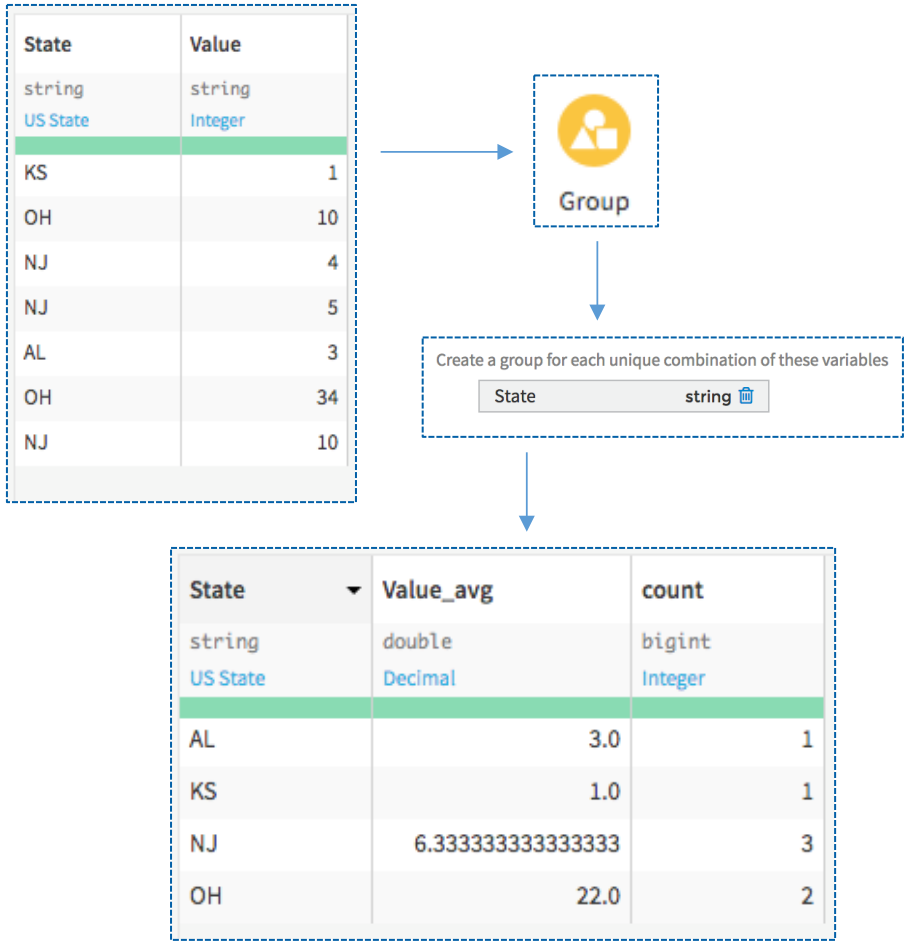
When working with big data, you often need to make it smaller before making sense of it. For instance, you'll start with a large amount of sales per country, or Tweets, or answers to a survey, and you'll find a way to aggregate them to a level that you can work with.

In Dataiku, this is done with a **Group Recipe**. It allows you to go from logs (so lines with an event like a sale or a Tweet and a timestamp) to an aggregated dataset with sums per week or users that have Tweeted about a certain subject.

The way the **Group Recipe** works is that you choose a key (or something to group by). For instance, you'll chose to go from a list of Tweets to a list of users by grouping on the user id.

You can then choose to have columns that count the number of values (say, distinct Tweets) or have minimum, maximum, sum, and average for numbers or dates.

Note
See the ***Remove Duplicates*** and the ***Basic Statistics*** pages for practical applications of aggregations



C. Reshaping Datasets

Going from Wide to Long Format

This section is on the process of reorganizing how data is displayed in a tabular format. Reshaping is often necessary as a preliminary step for merging datasets of various shapes.

For instance, if you have a really wide document with one column per date, you can change this to one line per day, making it a long dataset.

In Dataiku, this can be performed in a **Prepare Recipe** using the **Fold multiple columns** processor. You just pick the columns to fold, and give the new columns a name.

There is also an **Unfold** processor that does the exact opposite and reshapes the data from long to wide format.

Most of the operations in the pages to follow are not possible in Excel. However, they are very powerful tools, and they are worth learning so that you can handle data that does not have the right shape for your analysis.

Processors documented here:

- ***Transpose***: transforms lines into columns (diagonal symmetry)
- ***Pivot***: Uses an index to create multiple columns with values
- ***Fold***: Creates lines from the values in different columns
- ***Unfold***: Creates columns from the values in different lines
- ***Split and fold***: Creates columns by splitting a value on a delimiter

Note
Reshaping looks similar to Excel's pivot table, but it does not compute statistics per column or row values; it only reshapes data (see the pivot table section for more information)

Transpose

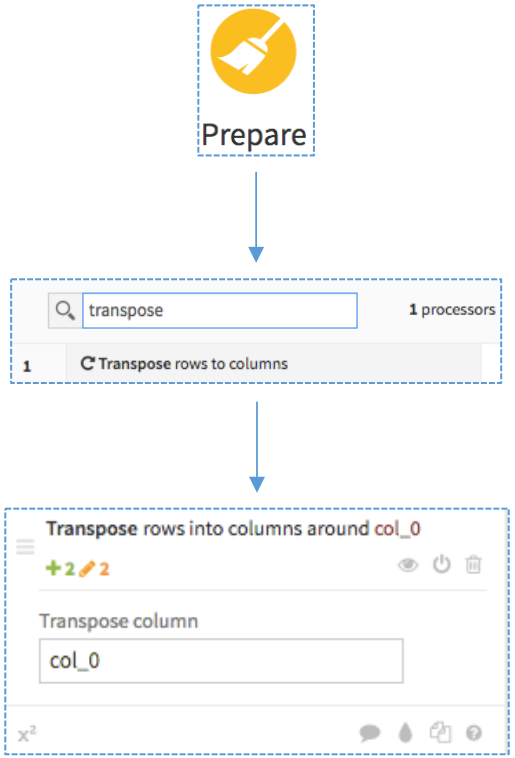
Transpose flips your data so that rows become columns and columns become rows.

Excel

- Select and cut/copy the data you would like to transpose, including row and column headers
- Choose ***Edit*** > ***Paste Special*** > ***Transpose*** and paste your data

Dataiku

- In a ***Prepare Recipe***
- Use the ***Transpose*** processor
- Choose the column whose values you want to be the column headers of the new dataset



Fold

Folding takes values from multiple columns and transforms them to one line per column (the opposite of a Pivot):

This operation is done using the ***Fold multiple columns*** processor.

For example, with the following dataset representing monthly scores:

person	age	01/2014	02/2014	03/2014
John	24	3	4	3
Sidney	31		6	9
Bill	33	1		4

Applying the the ***Fold multiple columns*** processor with:

- Three columns in the columns list: 01/2014, 02/2014, 03/2014
- Month as the "Column for fold name"
- Score as the "Column for fold value"

Will generate the following result:

person	age	month	score
John	24	01/2014	3
John	24	02/2014	4
John	24	03/2014	6
Sidney	31	01/2014	
Sidney	31	02/2014	6
Sidney	31	03/2014	9
Bill	33	01/2014	1
Bill	33	02/2014	
Bill	33	03/2014	4

- The column headers of the folded columns appear in the "Column for fold name"
- The entries of the folded columns appear in the "Column for fold value"
- The folded columns are removed
- All other columns are copied
- Empty values are preserved in the folded result

Unfold

Unfolding is used for categorical data and transforms cell values into binary columns. This process is also called Dummification:

This operation is done by using the **Unfold** processor, for example, with the following dataset:

id	type
0	A
1	A
2	C
3	B

Note
Due to the way the schema is handled in prepare Recipes, only the values that were found at least once in the sample will create columns in the output dataset.

Make sure your sampling is the right one!

Applying the **Unfold** processor on the "type" column will generate the following result:

id	type_A	type_C	type_B
0	1		
1	1		
2		1	
3			1

Each value of the unfolded column will create a new column.

This new column:

- Contains the value 1 if the original column contained this value
- Remains empty otherwise

Unfolding is often used to find some correlations to a particular value or for creating graphs.

Split and Fold

The split and fold operation creates new lines by splitting the values within a column on a delimiter:

This operation is done by using the *Split and Fold processor*.

For example, with the following dataset:

customer_id	events	browser
1	login,product,buy	Mozilla
2	login,product,logout	Chrome

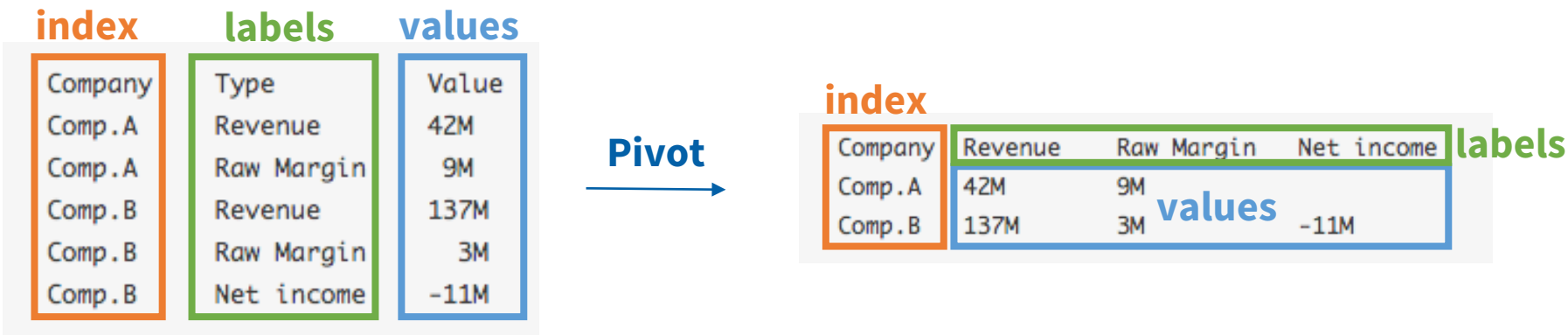
Applying the Split and Fold processor on the events column with , as the separator will generate the following result:

customer_id	events	browser
1	login	Mozilla
1	product	Mozilla
1	buy	Mozilla
2	login	Chrome
2	product	Chrome
2	logout	Chrome

All columns except the folded column are copied in each new line.

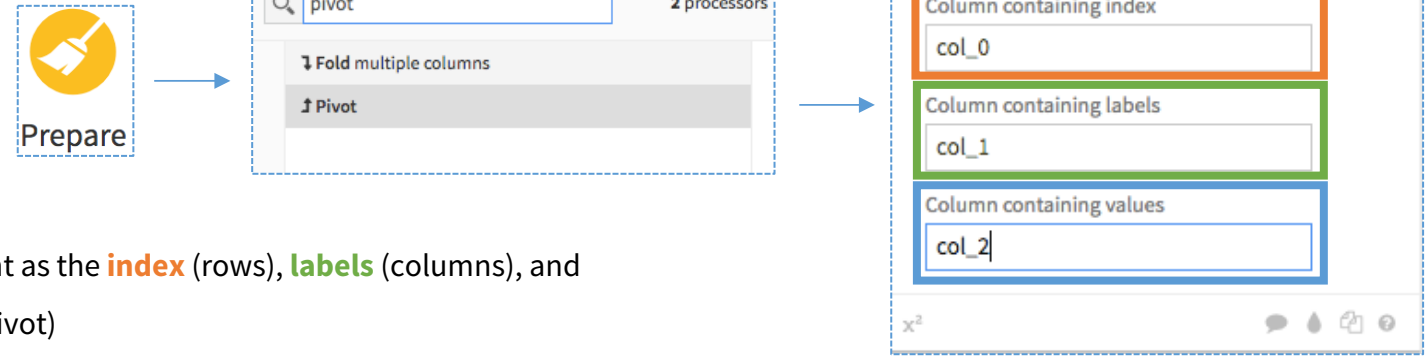
Pivot

The Pivot processor transforms multiple rows into columns, and the way it works is illustrated in an example below:



Dataiku

- In a **Prepare Recipe**
- Use the *Pivot* processor
- Choose which *columns* you want as the **index** (rows), **labels** (columns), and **values** (rest of the cells in the pivot)



D. Enriching Data

Parsing Dates (1/2)

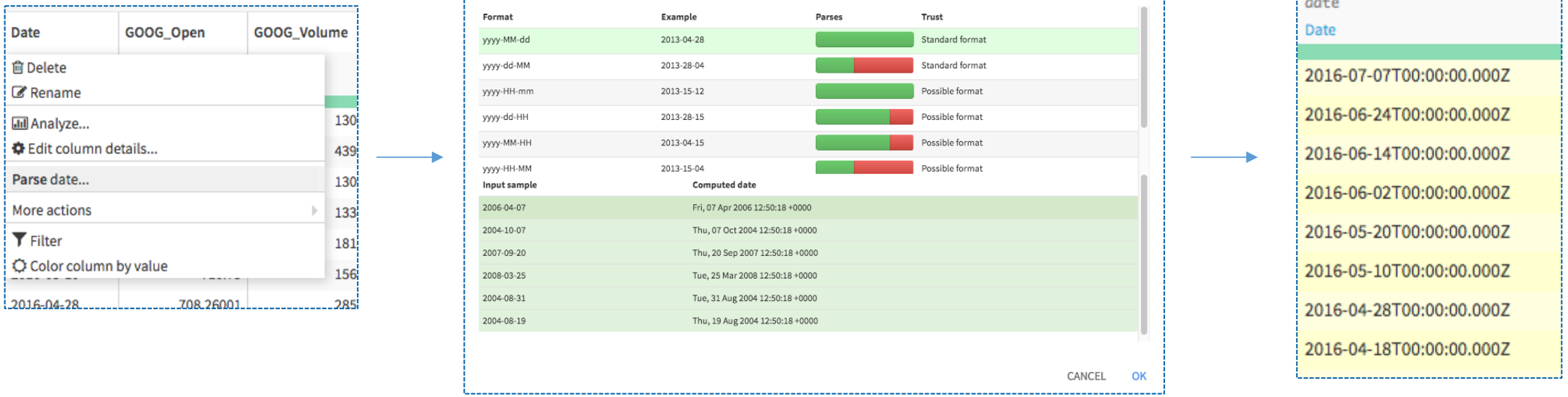
Parsing dates enables you to extract date-related information and create new columns based on it:

Excel

Excel recognises date formats in your columns

Dataiku

- *Click* on the *header* of the date column
- *Click* on *Parse date*, and Dataiku will automatically detect the most likely date format and change dates to *timestamps*



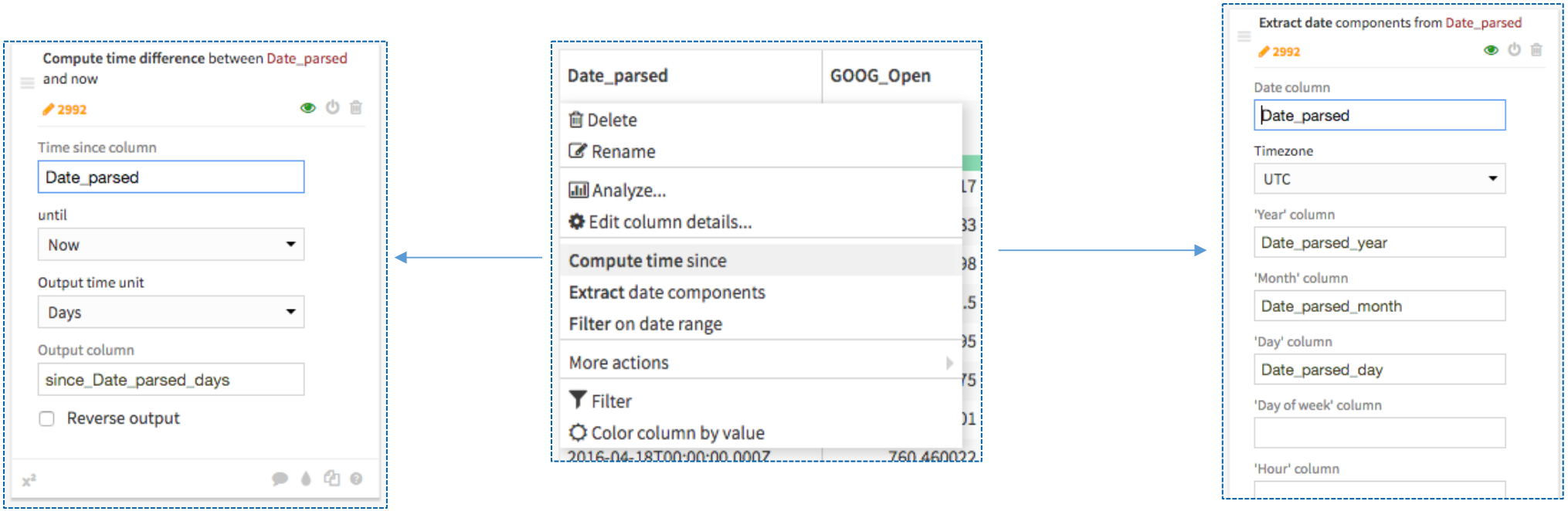
Parsing Dates (2/2)

Parsing dates enables you to extract date-related information and create new columns based on it:

Dataiku

Click on the parsed date column *header* to find advanced tools

- **Compute time since:** Like the DATEDIF function in Excel, except it outputs the time difference in days, weeks, years, or even seconds - you can decide what date the difference is calculated from.
- **Extract data component:** Create new columns by extracting the month, year, day of week, time, etc.

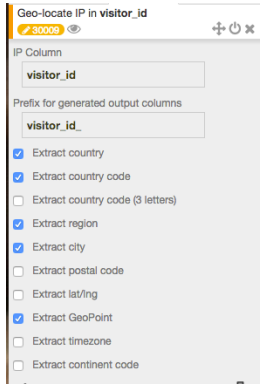


Geographic Data

Dataiku provides several processors to work with geographic information:

Resolve GeoIP

This processor uses the GeoIP database maxmind.com to resolve an IP address to the associated geographic coordinates:



It produces two kinds of information:

- Administrative data (country, region, city, ...)
- Geographic data (latitude, longitude)

The output GeoPoint can be used for Geographic charts

Reverse Geocoding

This processor provides Geographic coordinates –> Administrative data resolution to the city level:

- You need a column containing a Geo Point or a Geometry as input
- The processor outputs columns with city, region, country, etc.

Zipcode Geocoding

This processor provides Country + zip code –> Geographic coordinates resolution to the city level:

- You need a column containing the country (name or ISO code) and a column containing a zip code
- The processor outputs a GeoPoint column

Geocode

This processor performs the geocoding of an address using either the MapQuest or Bing API.

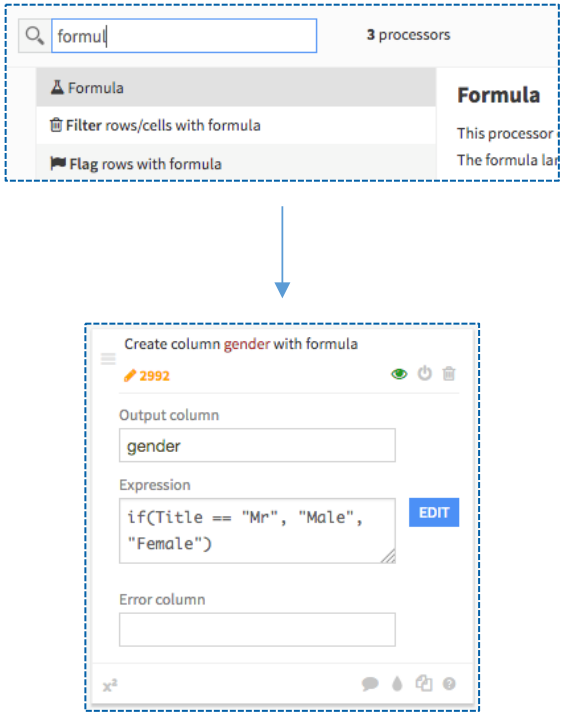
For details on which kind of addresses are handled, please see the doc of the associated API.

How To Use Formulas

Formulas in Dataiku are used to create new columns based on the values in other columns. You can find formulas as a step in the Lab or in the prepare Recipe.

Dataiku

- In a **Prepare Recipe** , create a *formula* step
- Click on *edit* to open the full editor
- A *reference document* is available on the right together with a preview of the result



Here are some examples of formulas you can use:

```
Nb_events + 3

max(x1, x2)

If (x1 > x2, “big”, “small”)
```

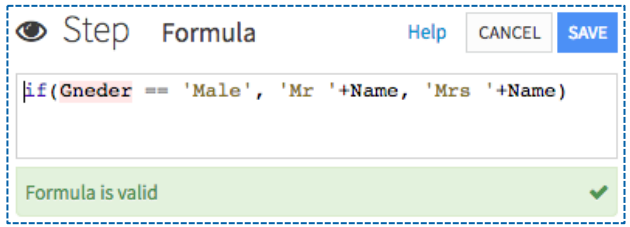
Note
Dataiku formulas work with columns, not with cells like Excel, which means that they cannot take a range of cells as an input

How To Debug Formulas

Formulas in Dataiku are a very powerful tool – the formula editor will help you to debug by displaying errors:

Dataiku

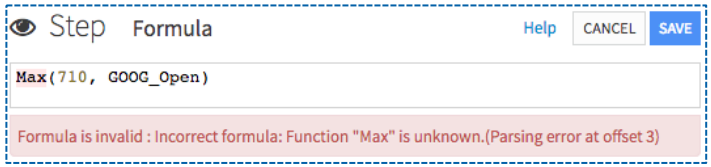
The formula editor will validate your formulas and highlight errors.



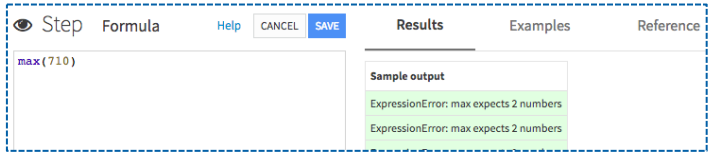
Green means that your formula is valid; however, if you reference non-existing columns, they will be highlighted in your formula.

Red means invalid; carefully read the error message to understand what is wrong.

In the example at right, the function should be max(), not Max()



Sometimes your formula is valid, but the functions return errors—you will easily see this in the Results tab that provides a preview of the output of your formula.



Lookups and IndexMatch

VLOOKUPS are used when you need to find things in a table or a range by row. They allow you to link all your tags and files together so you can be sure to have the same information in several places in a stable way, updated where it has to be.

The way to do this in Dataiku is by using a Join. You will use a visual Join recipe:

Join with

Principles of a JOIN

It combines two datasets (or joins them) by using a column (or several) that is common to both datasets (ON), just like the lookup value in a VLOOKUP.

- **Dataiku** You *join* two datasets *on* a common *key*
- **Excel** You *lookup in* a dataset the *key* you know from another dataset

✓ **Example**

VLOOKUP: search for a name, get a phone number from the phone book dataset

JOIN: search for a name, get all the information associated to that name from the phone book dataset

In the recipe

- Select your input datasets, and create or select an output
- Dataiku automatically finds columns in common
- Go to the *Join section* to add more datasets for the join operation (Add Input)
- Go to *Selected Columns* to decide which columns you want to keep by ticking the boxes and renaming columns that appear in red.

The *output* will be the lines for which the key you did the join on was in both datasets. The columns in the output will be the ones you selected from the first dataset as well as the ones you selected from the second dataset.

✓ **Example**

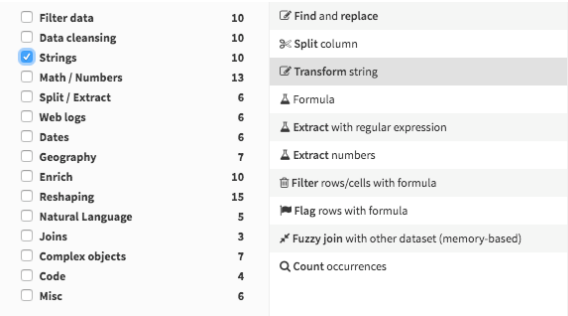
From your starting dataset containing names, create an enriched dataset with columns from the phone book, like phone number, address, city, etc.

Manipulating Text

There are several processors in Dataiku to manipulate text, and you'll find them in the processors library under Strings and Natural Language.

Dataiku

Transform string processor: convert to lower or upper case, capitalize, remove spaces, truncate (equivalent of the Left and Right Excel functions,) normalize (remove capitals and accents). You can also manipulate URLs, XML formats, and Unicode strings.



There are other *Strings* processors in the Visual Analysis and Prepare recipes.

- *Find and Replace* is covered in a specific section of this playbook
- *Split Columns* will create several columns from values with delimiters
- *Extract numbers*
- *Extract with regular expressions* to extract certain elements of a text, such at part of an email, using regexp
- *Count occurrences* of the complete cell value, or a substring, in each observation (useful for things like *Does this Tweet contain the word X?*)

You can find a few manipulators in the Natural Language section

- *Simplify Text* is used to Normalize, clear stop words, reduce words to their grammatical stem, and sort them alphabetically
- *TokenizeText* splits words into columns, lines, or puts them in a Json; you can also *Simplify* your text to clean it up (this is very useful to build word clouds or perform statistics on text)
- *Extract ngrams* is just like *Tokenize* but for a group of several words, so you can build statistics on a series of words used together

There is also a processor to *Concatenate*, just like you would in Excel.

E. Final Data Product

Basic Statistics (min, max, avg. . .)

After cleaning, enriching, and merging your data, you may want to see statistics!

Dataiku

To do this in a format that can be exported and automatized (and be sure it's performed on your whole dataset), you can use the *group Recipe*. We've mentioned how aggregations work, so you can refer to that section.

You can calculate Counts, Counts of different (i.e., distinct) values– for example, if a user Tweets twice, he'll be counted as two in count and one in Distinct.

On numerical values and dates, you can use it to compute minimum, maximum, sum, or average values.

Row Labels	Count of user_id
08/09/14 09:00	252
08/09/14 10:00	277
08/09/14 11:00	302
Grand Total	**831**

event_timestamp	user_id_distinct
string Text	bigint Integer
9/8/14 10:00	168
9/8/14 11:00	181
9/8/14 9:00	166

Charts

From bar charts to maps, Dataiku allows visual interaction with data.

Excel

To create a chart:

- Create the data you want to chart (often with *pivot table*)
- Select the *range* you want to plot
- Choose the type of charts you want

Dataiku

To create a chart:

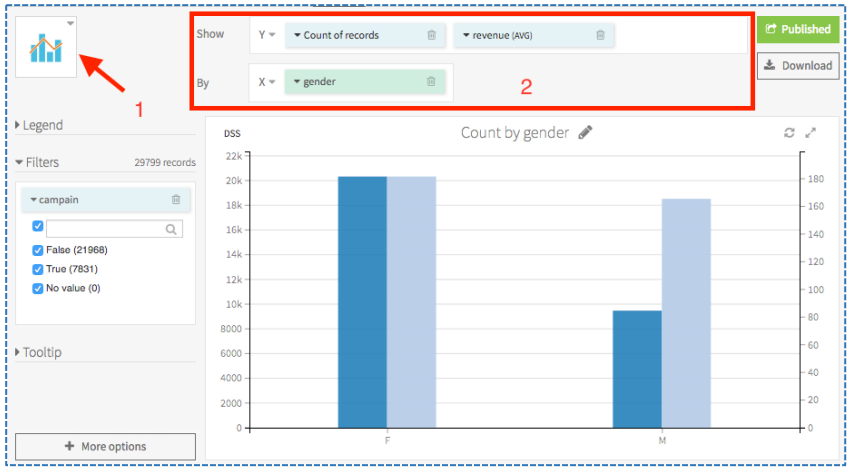
- Choose a type of *Chart*
- *Drag and drop* what you want to display
- In the *Sampling* tab, you can choose the *amount of data* to use to compute your chart

		lost_cause	persuadable	sleeping_dog	sure_thing	
control	Avg of outcome	0	0	1	1	0.50
	Count of records	1298	1252	1287	1226	5063
target	Avg of outcome	0	1	0	1	0.49
	Count of records	1256	1219	1241	1221	4937
	Avg of outcome	0	0.49	0.51	1	
	Count of records	2554	2471	2528	2447	

Options

Charts Options (1/2)

Charts have many options:



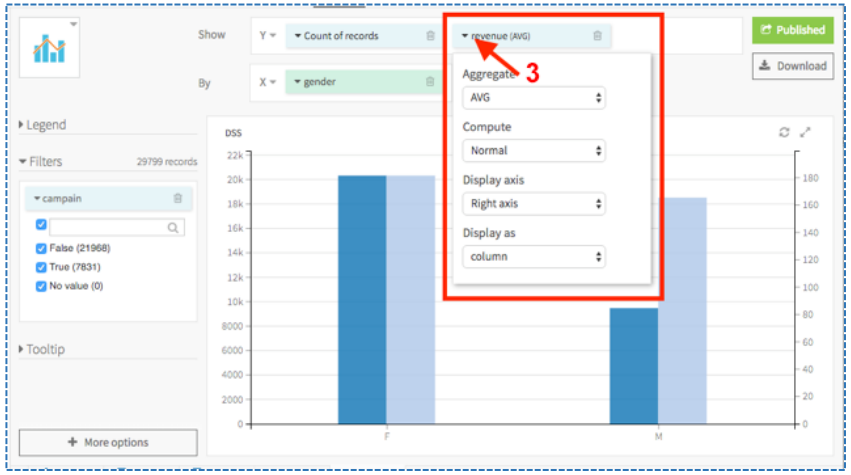
1. Set your chart type

Navigate in the *chart categories* to choose your chart.

2. Set the chart global settings (basic)

Drag & drop columns from the left panel in the chart settings zone.

Note that you can display the *count of records* in the blue areas by dragging it to any blue zone in the chart settings.



3. Set the chart global settings (advanced)

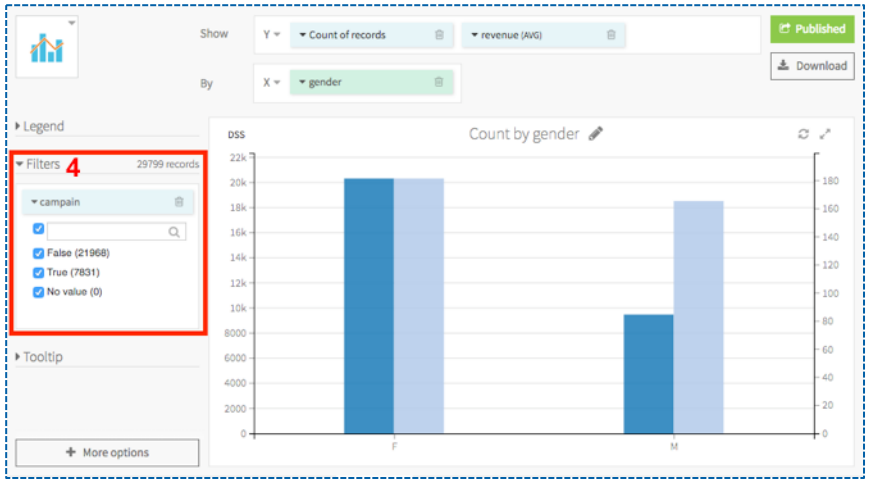
Click on the caret left of the column name in the settings area to set what you want to compute :

- How you want to *bin* your data?
- The kind of *statistics* you want to visualize (Average, Sum, etc)?

Continue

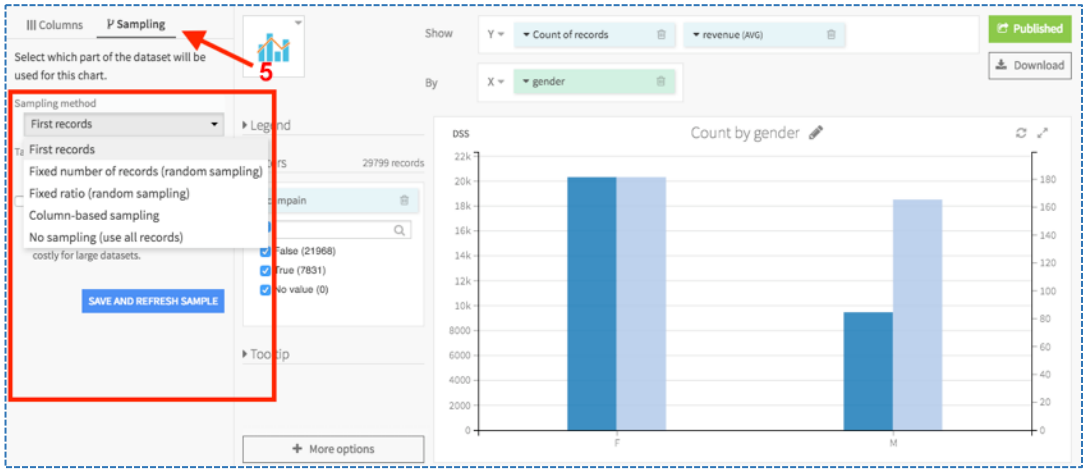
Charts Options (2/2)

Set the chart underlying data and the chart engine:



4. Filter the data

Drag & drop columns from the left panel in the filter settings zone.
Select the values to filter in/out.



5. Set the underlying chart data

Click on the *Sampling* tab to the left of the chart and select the data on which your chart will be based for its computation (SUM, AVG).

Note

Charts within visual analysis can use the whole dataset *only if it fits into memory*; you must create your charts on a *Dataset* in order to *use the whole data* when it does not fit into memory.

Chart computations will be a lot quicker if your datasets reside in a connection that allows efficient *Live Engine* computations (e.g., a SQL connection or HDFS with Impala).

The Pivot Table

The pivot table in Excel is used to compute statistics on your data. The exact same operation is possible in Charts:

Dataiku

- Browse charts and select the *Tables category*
- Click on *Pivot Table*
- *Drag and drop* the columns you want to see as rows, columns, and values
- You can apply filters by dragging columns to the *Filters* cell

Note
Click on the caret next to your Content values to choose between AVG, MIN, MAX and SUM.

If you use dates, you can select different grouping (e.g., month of year). The resulting table can be exported as an image or a .csv/.xlsx file.

Sliding Calculations with Window Recipes

When you're monitoring KPIs, it's often very important to track moving averages or sums to compare over similar timelines, for example. This can be done using Window Recipes:

Dataiku

Using an input dataset where *each line corresponds to a time unit* (you can perform a *group by* on hour-day-week before)

- Create a *Window Recipe*
- Set up the *Window Frame* you want to compute by defining how many lines below or after you want to calculate on.
- Define what columns you want to *order* your dataset by (this is going to define what you build your windows on)
- Select your *Aggregations*, choosing from **minimum, maximum, average, sum, count** (number of values), **first and last** for each of your columns.

The output of your recipe will be a dataset with the same format as your input along with extra columns with the computations you added. For each line, you'll have the computation for the window you defined.

Example: setting up a window on a daily dataset with a limit of seven before and zero after and computing the average will get you the average over the past seven days for each day.

In *Aggregations*, you can also use **Lag** and **Lead** to show the value X days (lines) before or after. **LagDiff (LeadDiff)** show the difference between the value X lines before (after) and the one on the current line. You can use several values by separating them with commas.

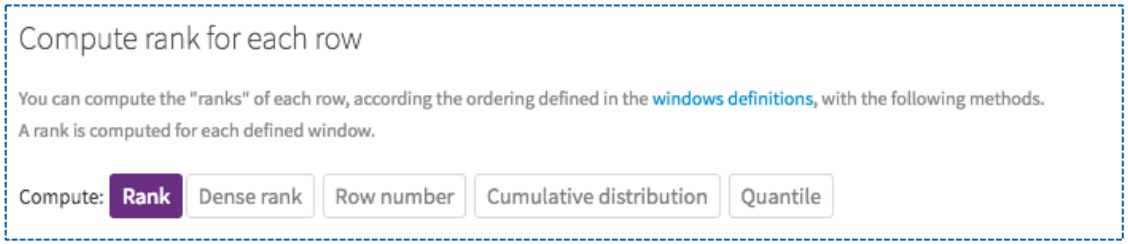
Note
This recipe only works on compatible engines, so if it appears all grey, that means you need to synch your dataset to a SQL, Hadoop, Spark, or Impala database.

Calculating Ranks or Quartiles

You can also use Window Recipes to calculate ranks, dense ranks, row numbers, and cumulative distribution on your datasets:

Dataiku

- Set up your *Window Recipe* by toggling Order columns and deciding which column you want to rank on.
- Go to the Aggregations section
- Select your ranking method.(hover over the methods to see a definition and choose between rank, dense rank, row numbers, and cumulative distribution)



In the *Windows definition* section, you can choose a partitioning column. Partitioning your datasets based on a column value will apply calculations independently for each value in that column. For instance, you could partition based on a product_category column (or on dates, etc.). This way, when you compute ranks, it will output ranks by product_category (rank count resets for each different category).

Hence partitioning can also be used to set up automatic calculations of **sums**, **min**, **max**, or **average** per product category (just like a COUNT_IF, AVERAGE_IF or SUM_IF in Excel).

Note
Have you figured out that you can use a Window Recipe to order a dataset?
You can even order on several columns. However, It only works if you set up at least one aggregation to compute.

data
iku

Part 2 : Use Cases

A. Merging Datasets with Different Time Units

A. Time Unit Homogenization

Time-based data inputs do not always have the same sampling frequency:

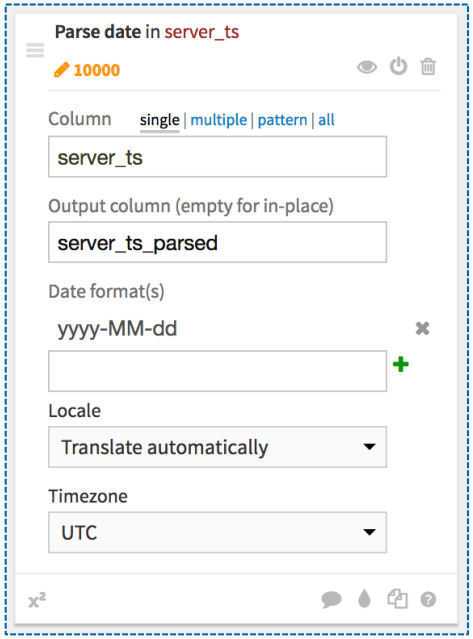
Goal	Issues
Unify data based on dates	The minimal common timescale has to be used

Steps

- Start by deciding what is the *longest acceptable timescale* for your study.
- Explore your datasets and determine the ***sampling frequency*** for each of them.
- Once you have decided what sampling frequency to use, you will have to ***downscale*** all the datasets that have a shorter time scale.
- In a prepare recipe, add a *Parse date* processor
- Replace the date format with the downscaled one
- **Date formats** use this syntax yyyy-MM-dd'T'HH:mm:ss.SSS

Examples:

- to downscale to the hour use yyyy-MM-dd'T'HH
- to downscale to the day use yyyy-MM-dd



Note

Sampling frequency: the smallest time resolution available in your data. Is it a machine log with timestamps down to the millisecond? Do you get a few values each day? Is it a daily report?

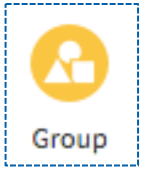
Downscaling: getting rid of the shorter time elements you don't need. Downscaling to the minute would have you drop seconds and milliseconds

A. Time Unit Homogenization

Time-based data inputs do not always have the same sampling frequency

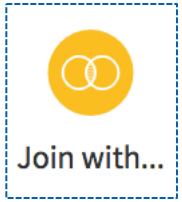
Steps

Your down-sampled dataset will have multiple rows for each time stamp. You can use a group recipe to get a single row for each time and choose how to aggregate your data (e.g., keeping the **minimum**, **maximum**, or **average** value)



Once all your datasets are prepared with the same time unit, you can join them:

- Create a *Join* Recipe
- Start by adding *two datasets* to join
- You can add *additional input* datasets from within the recipe by clicking on the *Add input* button



Note

See the ***Aggregate*** slide for more information on Grouping.
See the ***Lookup and IndexMatch*** slide to understand how Joining works

Thanks for reading!

You can find additional content on the Learn section of our website:

http://www.dataiku.com/learn/guide/

Or refer to the reference documentation:

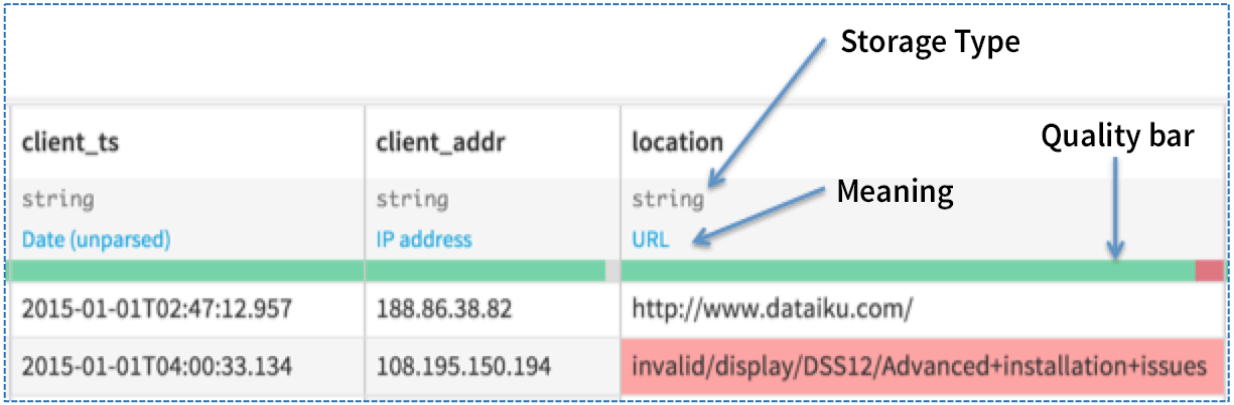
http://doc.dataiku.com/dss/latest/

Data Types (1/2)

In Dataiku DSS datasets have a schema. The schema of a dataset is the list of columns, with their names and types.

Dataiku DSS

- Change the *storage type* by clicking in column header on the type in black
- Change the *meaning* by clicking in the column header on the type in blue
- The *data quality* jauge refers to the validity of the values with respect to the meaning



Continue

Data Types (2/2)

In Dataiku DSS datasets have a schema. The schema of a dataset is the list of columns, with their names and types.

There are two kinds of « types » in DSS.

- The storage type is used to indicate how the dataset backend should store the column data. Storage type is technical : typical storage types are integer, float (16bits), float (32bits), boolean, array, …
- The meaning is a “rich” semantic type. Meanings are automatically detected from the contents of the columns. The meaning is more business oriented: typical meanings are integer, gender, URL, email address, Country code…

Note

By default, the meaning is set to auto: DSS automatically infers the most probable meaning by scanning a sample of the data. You can force the meaning type by setting it explicitly. A locker appears next to the meaning.

Storage types and meanings are related but with large amounts of flexibility that allow Data Science Studio to handle invalid data while retaining advanced meanings.

How do types affect data manipulation?

- Storage type of a column determines the kind of aggregations that can be computed in a group recipe, e.g. you can not compute an average of a non numerically stored column, it does not mean anything.
- Meaning affects cleansing: the validity of the values and the associated cleansing or the default processing of a column. For example, a column with email meaning can be split in login and domain.

Common errors

- When building a dataset, DSS might raise errors because some values are not compatible with the storage type of their columns. For example, it is not possible to write « England » in a column that has integer as storage type.
- In a group recipe. The numerical computation (AVG, SUM) are disabled. Go to the input dataset and check the storage type, correct it, reconstruct the inputs.

About Schemas

When you build certain recipes (like Group recipes), or modify recipes, you'll sometimes get a popup notifying you that the schema is different and has to be updated. What's this about?

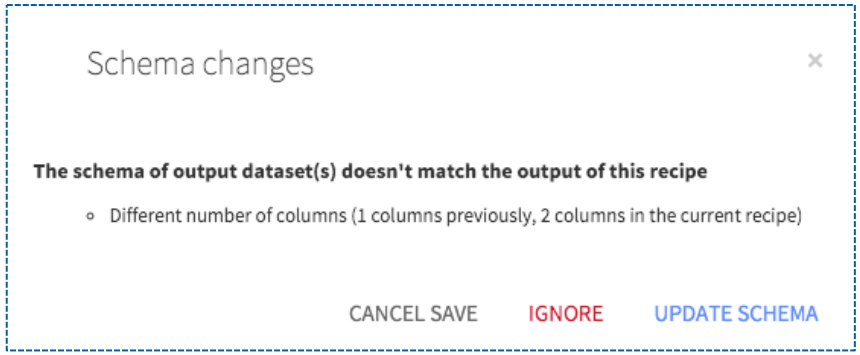
What are schemas?

A database schema represents the structure of a database. It defines how data is organized in datasets and how these datasets are associated (with key columns).
The names and number of columns in your datasets, as well as the type of data stored in each, are part of the schema.

Why does this come up?

Dataiku DSS infers the schema of your data automatically when an external dataset is created, from the sample data. When you build a recipe, it will automatically set the schema of your output dataset as identical to the input dataset.

DSS never automatically modifies the schema of a dataset without an explicit user action. That's why for some recipes, if you've altered columns names or meanings, or added or deleted columns, you'll get a warning. You can go ahead and click Update Schema.



Note
If you're working with code outside Dataiku DSS, you often have to explicitly create the schema for your output datasets, while most recipes in Dataiku DSS (including code) generate it automatically.