Overview of Sampling

Contents:

To prevent overwhelming the client or significantly impacting performance, Trifacta® Wrangler generates one or more samples of the data for display and manipulation in the client application. Since Trifacta Wrangler supports a variety of clients and use cases, you can change the size of samples, the scope of the sample, and the method by which the sample is created. This section provides background information on how the product manages dataset sampling.

How Sampling Works

Initial Sample

When a dataset is first created, a background job begins to generate a sample using the first set of rows of the dataset. This **initial sample** is usually very quick to generate, so that you can get to work right away on your transformations.

- The default sample is the initial sample.
- By default, each sample is 10 MB in size or the entire dataset if it's smaller.
- If your source of data is a directory containing multiple files, the initial sample for the combined dataset is generated from the first set of rows in the first filename listed in the directory.
    - If the matching file is a multi-sheet Excel file, the sample is taken from the first sheet in the file.
    -

Generating samples

Additional samples can be generated from the context panel on the right side of the Transformer page. Sample jobs are independent job executions. When a sample job succeeds or fails, a notification is displayed for you.

As you develop your recipe, you might need to take new samples of the data. For example, you might need to focus on the mismatched or invalid values that appear in a single column. Through the Transformer page, you can specify the type of sample that you wish to create and initiate the job to create the sample. This sampling job occurs in the background.For more information on creating samples, see *Samples Panel*.

Sampling methods

Sampling mechanics

When a non-initial sample is executed for a single dataset-recipe combination, the following steps occur:

1. All of the steps of the recipe are executed on the dataset on the backend cluster, up to the currently selected recipe step.
2. The generated sample is executed on the current state of the dataset.

> **NOTE:** When a sample is executed from the Samples panel, it is launched based on the steps leading up to current location in the recipe steps. For example, if your recipe includes joining in other datasets, those steps are executed, and the sample is generated with dependencies on these other datasets. As a result, if you change your recipe steps that occur before the step where the sample was generated, you can invalidate your sample. More information is available below.

When your flow contains multiple datasets and flows, all of the preceding steps leading up to the currently selected step of the recipe are executed, which can mean:

- The number of datasets that must be accessed increases.
- The number of recipe steps that must be executed on the backend increases.
- The time to process the sampling job increases.

Implications:

- When the sample is displayed in the Transformer page, all steps after the one from which it was executed are computed in the web browser. So, if you have a lengthy series of steps or complex operations after the step where you generated a sample, you can cause performance issues of the Transformer page, including the occasional browser crash. Try generating a new sample later in your flow for better performance.
- If you have added an expensive transformation step, such as a complex union or join, you can improve performance of the Transformer page by generating and using a new sample after the transformation step.

> **NOTE:** When a flow is shared, its samples are shared with other users. However, if those users do not have access to the underlying files that back a sample, they do not have access to the sample and must create their own.

Important notes on sampling

- When sampling from compressed data, the data is uncompressed and then expanded. As a result, the sample size reflects the uncompressed data.
- Changes to preceding steps that alter the number of rows or columns in your dataset can invalidate the current sample, which means that the sample is no longer a valid representation of the state of the dataset in the recipe. In this case, Trifacta Wrangler automatically switches you back to the most recently collected sample that is currently valid. Details are below.

Choosing Samples

After you have collected multiple samples of multiple types on your dataset, you can choose the proper sample to use for your current task, based on:

1. **How well each sample represents the underlying dataset.** Does the current sample reflect the likely statistics and outliers of the entire dataset at scale?
2. **How well each sample supports your next recipe step.** If you're developing steps for managing bad data or outliers, for example, you may need to choose a different sample.

> **Tip:** You can begin work on an outdated yet still valid sample while you generate a new one based on the current recipe.

Limitations

- Some advanced sampling options are available only with execution across a scan of the full dataset.
- Undo/redo do not change the sample state, even if the sample becomes invalid.

- When a new sample is generated, any Sort transformations that have been applied previously must be re-applied. Depending on the type of output, sort order may not be preserved.

Sample Invalidation

With each step that is added or modified to your recipe, Trifacta Wrangler checks to see if the current sample is valid. Samples are valid based on the state of your flow and recipe at the step when the sample was collected. If you add steps before the step where it was created, the currently active sample can be invalidated. For example, if you change the source of data, then the sample in the Transformer page no longer applies, and a new sample must be displayed.

> **Tip:** After you have completed a step that significantly changes the number of rows, columns, or both in your dataset, you may need to generate a new sample, factoring in any costs associated with running the job. Performance costs may be displayed in the Transformer page.

> **NOTE:** If you modify a SQL statement for an imported dataset, any samples based on the old SQL statement are invalidated.

- The Transformer page reverts to displaying the most recently collected sample that is currently valid.
- You can generate a new sample of the same type through the Samples panel. If no sample is valid, you must generate a new sample before you can open the dataset.
- A sample that is invalidated is listed under the Unavailable tab. It cannot be selected for use. If subsequent steps make it valid again, it re-appears in the Available tab.

Sample Types

Trifacta Wrangler currently supports the following sampling methods.

Random samples

Random selection of a subset of rows in the dataset. These samples are comparatively fast to generate.

Filter-based samples

Find specific values in one or more columns. For the matching set of values, a random sample is generated.

You must define your filter in the Filter textbox.

Anomaly-based samples

Find mismatched or missing data or both in one or more columns.

You specify one or more columns and whether the anomaly is:

1. mismatched
2. missing
3. either of the above

Optionally, you can define an additional filter on other columns.

Stratified samples

Find all unique values within a column and create a sample that contains the unique values, up to the sample size limit. The distribution of the column values in the sample reflects the distribution of the column values in the dataset. Sampled values are sorted by frequency, relative to the specified column.

Optionally, you can apply a filter to this one.

> **Tip:** Collecting samples containing all unique values can be useful if you are performing mapping transformations, such as values to columns. If your mapping contains too many unique values among your key-value pairs, you can try to delete all columns except the one containing key-value pairs in a step, collect the sample, add the mapping step, and then delete the step where all other columns are removed.

Cluster-based samples

Cluster sampling collects contiguous rows in the dataset that correspond to a random selection from the unique values in a column. All rows corresponding to the selected unique values appear in the sample, up to the maximum sample size. This sampling is useful for time-series analysis and advanced aggregations.

Optionally, you can apply an advanced filter to the column.