

DATA WRANGLING CHEATSHEET

COMBINE DATASETS

Union

Append rows from 2 or more datasets by name or position



Join...

inner

Append columns from datasets, and retain only rows that are found in both sets



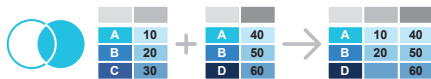
left

Append columns from datasets, and retain rows that match the left (1st) set



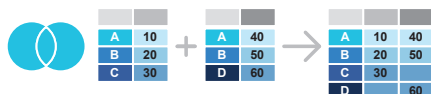
right

Append columns from datasets, and retain rows that match the right (2nd) set



outer

Append columns from datasets, and retain all rows from both sets



RESHAPE DATASETS

Pivot

Aggregate values in columns



Unpivot

Turn columns into rows



Group by...

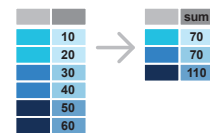
as a new column(s)

Aggregate a group of columns and show added values in a new column, a.k.a. partition by



as a new table

Aggregate a group of columns and show added values in a new table



COLUMN & ROW MANIPULATIONS

Merge Columns

Concatenate 2 or more columns as a new column



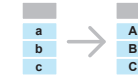
Split Columns

Split a column into multiple columns based on a condition



Set/Replace Transformation

Apply a formula to the current value in a cell



Create a New Column from a Formula

Apply a transformation to 1 or more columns which results in a new column



Filter

Keep/Remove row based on a condition



Header

Convert row(s) into the column name



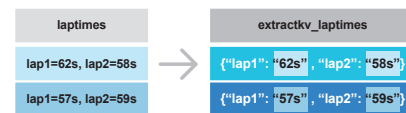
RESTRUCTURING DATASETS

Objects / JSON

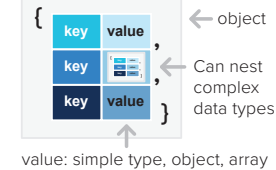
Nested data structure using key-value pairs

Convert Key Value into Object

Create an object from string value

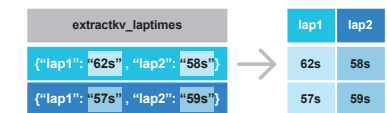


Data Structure



Unnest Elements into Columns

Create new columns for key values

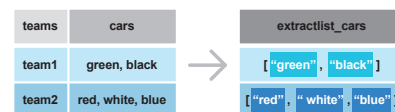


Array / List

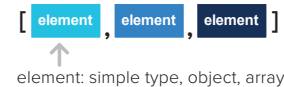
List containing multiple elements

Extract Matches to Array

Create an array / list from string value

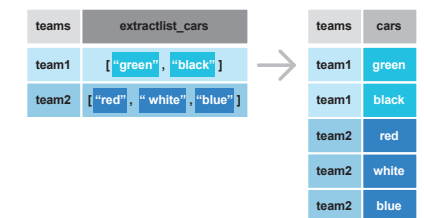


Data Structure

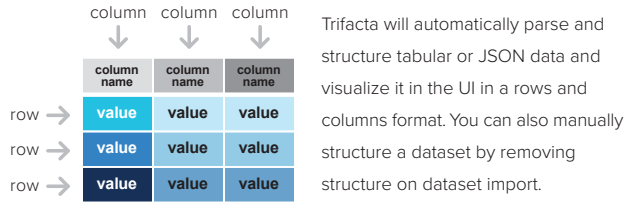


Flatten Elements in New Rows

Create new rows for each array element



COLUMNAR DATASETS

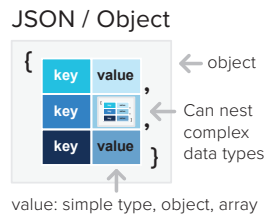


DATA TYPES

Simple Type

Item	Description
String	Any text
Integer	Positive or negative numeric values with no decimal point
Decimal	Floating points up to 15 digits in length
Boolean	True or false
Social Security Number	United States Social Security number
Phone Number	Common pattern that express a U.S. phone number
Email Address	Email address name@domain.ext
Credit Card	14-digit or 16-digit pattern for credit card
Gender	Expressing for male/female distinction
Zip Code	Five- and nine-digit U.S. zipcode
State Data	Full name or the two-letter abbreviation for U.S. states
IP Address	IPv4 address
URL	Generalized pattern of URLs
HTTP Code	HTTP Status Code
Datetime	Variety of Datetime formats

Complex Type



TOP TRIFACTA PATTERN SYNTAX

Pattern Types

Trifacta Pattern backticks <code>`...`</code>	Literal Strings: single quotes <code>'...'</code> or double quotes <code>"..."</code>	Regular Expression forward slash <code>/.../</code>	Column Name curly brace <code>{...}</code>
---	--	---	--

Character Patterns

Pattern	Description
%	Match any character, exactly once
/?	Match any character, zero or one times
%*	Match any character, zero or more times
%+	Match any character, one or more times
{3}	Match any character, exactly three times
{3,5}	Match any character, 3, 4, or 5 times
#	Digit character [0-9]
{any}	Match any character, exactly once
{alpha}	Alpha character [A-Za-z_]
{upper}	Uppercase alpha character [A-Z_]
{lower}	Lowercase alpha character [a-z_]
{digit}	Digit character [0-9]
{delim}	Single delimiter character e.g. ;, ,, , /, ~, \s
{delim-ws}	Single delimiter and all the whitespace around it
{alpha-numeric}	Match a single alphanumeric character
{alphanum-underscore}	Match a single alphanumeric char or underscore char
{at-username}	Match @username values
{hashtag}	Match #hashtag values
{hex}	Match hexadecimal number (e.g. 2FA3)

Position Patterns

Pattern	Description
{start}	Match the start of the line
{end}	Match the end of the line

Data Type Patterns

Pattern	Description
{phone}	Match a valid U.S. phone number
{email}	Match a valid email address
{url}	Match a valid URL
{ip-address}	Match a valid IP address
{hex-ip-address}	Match a valid hexadecimal IP address
{bool}	Match a valid Boolean value
{street}	Match a U.S.-formatted street address
{occupancy}	Match a valid U.S.-formatted occupancy address value
{city}	Match a city name within U.S.-formatted address value
{state}	Match a valid U.S. state value
{state-abbrev}	Match a valid two-letter U.S. state abbreviation value
{zip}	Match a valid five-digit zip code

Date Patterns

Pattern	Description
{time}	Match time value in HOUR:MINUTE:SECOND format
{period}	Match time period of the day: AM/PM
{month-abbrev}	Match short name of month (e.g. Jan)
{dayofweek-abbrev}	Match short name for day of the week (e.g. Sun)
{dayofweek}	Match long name for day of the week (e.g. Sunday)
{month}	Match full name of month (e.g. January)
{utcoffset}	Match a valid UTC offset value (e.g. -0500, +0400, Z)

Group Patterns

Pattern	Description
{[...]}	Character class matches characters in brackets
{![...]}	Negated class matches characters not in brackets
{(...)}	Grouping, including captures
{#, %, ?, *, +, {, }, (,), \, ', \n, \t}	Escaped characters or pattern modifiers Use a double backslash (\) for an escaped string literal
	logical OR