Miller Name-indexed data tool

CONCEPTS

Miller handles records with named fields, not lines: verbs see $name, so column order rarely matters.

Input format --icsv, --ijson
↓ records
Verb chain cat, sort, put ...
↓ then
Next verb each sees the previous output
↓ records
Output format --opprint, --ojson

Main flags come first, then verbs joined by then, then file names: mlr {flags} verb [opts] then verb [opts] {files}.

Put DSL expressions in single quotes so the shell leaves $name alone; fields are $name, ${name with space}.

FORMATS AND I/O

`mlr --icsv --opprint <verb> <file>`
Read CSV, pretty-print

`mlr --c2j <verb>`
CSV in, JSON out

`mlr --c2p <verb>`
CSV in, aligned table out

`mlr --c2t / --t2c <verb>`
CSV to TSV / TSV to CSV

`mlr --j2c <verb>`
JSON in, CSV out

`mlr --io json <verb>`
JSON in and out

`mlr --implicit-csv-header <verb>`
Fields named 1, 2, 3 (no header)

`mlr --headerless-csv-output <verb>`
Omit the CSV header line

INSPECT AND SLICE

`mlr <fmt> cat`
Pass records through (convert format)

`mlr <fmt> cat -n -g <field>`
Number records, per group

`mlr <fmt> head -n <n>`
First n records

INSPECT AND SLICE (cont.)

`mlr <fmt> head -n 1 -g <field>`
First record of each group

`mlr <fmt> tail -n <n>`
Last n records

`mlr <fmt> count -g <field>`
Records per group

`mlr <fmt> summary`
Per-column statistics overview

FIELDS

`mlr <fmt> cut -f a,b`
Keep only these fields

`mlr <fmt> cut -o -f b,a`
Keep fields in the order given

`mlr <fmt> cut -x -f a,b`
Drop these fields

`mlr <fmt> reorder -f a,b`
Move fields to the front

`mlr <fmt> fill-empty -v <value>`
Fill empty values

FILTER AND COMPUTE (DSL)

`mlr <fmt> filter '$x > 5 && $y == "a"'`
Keep records where true

`mlr <fmt> filter '$name =~ "^ab"'`
Regex match on a field

`mlr <fmt> put '$new = $a . "-" . $b'`
Create or set a field (dot joins)

`mlr <fmt> put 'unset $x'`
Remove a field

`mlr <fmt> put 'NR == 1 {$flag = "first"}'`
Pattern-action block

`mlr <fmt> put 'if ($x > 0) {$s = "pos"} else {$s = "neg"}'`
Conditional

`mlr <fmt> put '@sum += $x; end {emit @sum}'`
Accumulate and emit at end

`mlr <fmt> put '$r = fmtnum($x, "%.2f")'`
Format a number

AGGREGATE

`mlr <fmt> stats1 -a mean,sum,count -f <f>`
Summary stats of fields

`mlr <fmt> stats1 -a p50,min,max -f <f> -g <g>`
Percentiles and extremes by group

`mlr <fmt> count-distinct -f <field>`
Count each distinct value

`mlr <fmt> top -n 3 -f <f> -g <g> -a`
Top n records per group

SORT AND DEDUPE

`mlr <fmt> sort -f <field>`
Lexical ascending

`mlr <fmt> sort -nr <field>`
Numeric descending

`mlr <fmt> sort -nf a -r b`
Multi-key sort

`mlr <fmt> uniq -g a,b`
Distinct values of fields

JOIN AND COMBINE

`mlr <fmt> join -j <id> -f <left> <right>`
Inner join on a field

NEST AND RESHAPE

`mlr <fmt> nest --evar ';' -f <field>`
Split a cell into records

`mlr <fmt> reshape -i a,b -o key,value`
Wide to long

`mlr <fmt> reshape -s key,value`
Long to wide

TIME

`mlr <fmt> sec2gmt <field>`
Epoch seconds to ISO timestamp

HELP AND MISC

`mlr help verb <name>`
Usage for one verb

`mlr <verb> --help`
Usage for one verb