Miller Name-indexed data tool

START HERE

`mlr --icsv --opprint cat <file>`
Read CSV, print aligned

`mlr --icsv --opprint stats1 -a mean,sum -f <f> -g <g> <file>`
Group and aggregate

`mlr --c2j cat <file>`
CSV to JSON

`mlr --icsv --opprint cut -f a,b then sort -f a <file>`
Chain verbs

`mlr --icsv --opprint head -n 5 <file>`
First rows

`mlr --icsv --opprint sort -nr <field> <file>`
Sort numerically

`mlr --icsv --opprint filter '$x > 5' <file>`
Keep matching rows

`mlr --icsv --opprint put '$z = $x * $y' <file>`
Add a computed column

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

CONCEPTS

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

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 --c2m <verb>`
CSV to Markdown table

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

FORMATS AND I/O (cont.)

`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

`mlr <fmt> head -n 1 -g <field>`
First record of each group
Also deduplicates by a field while keeping all other columns.

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

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

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

`mlr <fmt> grep -i <regex>`
Match records as text

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> cut -r -f '^t_'`
Select fields by regex

`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

FILTER AND COMPUTE (DSL) (cont.)

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

`mlr <fmt> put '$z = $x > 0 ? "pos" : "neg"'`
Ternary

`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

SORT AND DEDUPE (cont.)

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

`mlr <fmt> sort -nf a -r b`
Multi-key sort
Records missing a sort field go to the end regardless of order.

`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

RECIPES

CSV to pretty JSON, selected columns

```
mlr --icsv --ojson cut -f
id,name,price then sort -nr
price data.csv
```

Top 3 products per category

```
mlr --icsv --opprint \
  top -n 3 -f revenue -g
category -a sales.csv
```

Group totals with a count column

```
mlr --icsv --opprint \
  stats1 -a count,sum -f amount
-g customer \
  then sort -nr amount_sum
orders.csv
```

Enrich with a lookup table

```
mlr --icsv --opprint \
  join -j id --lp left_ -f
users.csv \
  then cut -x -f internal_notes
orders.csv
```

Per-key files

```
# writes out_east.csv,
out_west.csv, ...
mlr --icsv --ocsv split -g
region --prefix out data.csv
```