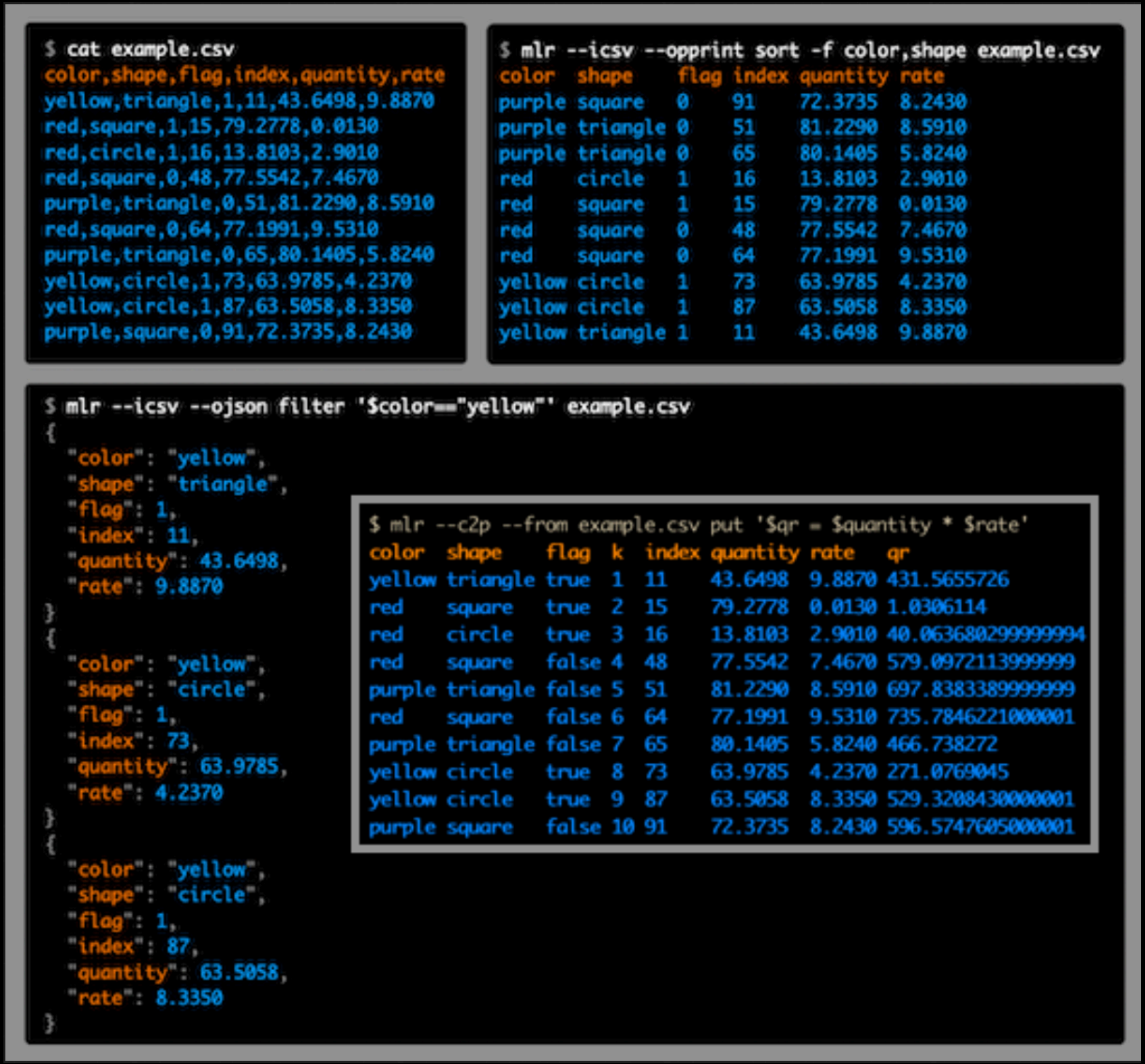
Miller

a swiss-army chainsaw for CSV and more

csv,conf,v7



```
$ cat example.csv
color,shape,flag,index,quantity,rate
yellow,triangle,1,11,43.6498,9.8870
red,square,1,15,79.2778,0.0130
red,circle,1,16,13.8103,2.9010
red,square,0,48,77.5542,7.4670
purple,triangle,0,51,81.2290,8.5910
red,square,0,64,77.1991,9.5310
purple,triangle,0,65,80.1405,5.8240
yellow,circle,1,73,63.9785,4.2370
yellow,circle,1,87,63.5058,8.3350
purple,square,0,91,72.3735,8.2430
```

```
$ mlr --icsv --opprint sort -f color,shape example.csv
color  shape    flag index quantity rate
purple square   0    91    72.3735  8.2430
purple triangle 0    51    81.2290  8.5910
purple triangle 0    65    80.1405  5.8240
red    circle   1    16    13.8103  2.9010
red    square   1    15    79.2778  0.0130
red    square   0    48    77.5542  7.4670
red    square   0    64    77.1991  9.5310
yellow circle   1    73    63.9785  4.2370
yellow circle   1    87    63.5058  8.3350
yellow triangle 1    11    43.6498  9.8870
```

```
$ mlr --icsv --ojson filter '$color=="yellow"' example.csv
{
  "color": "yellow",
  "shape": "triangle",
  "flag": 1,
  "index": 11,
  "quantity": 43.6498,
  "rate": 9.8870
}
{
  "color": "yellow",
  "shape": "circle",
  "flag": 1,
  "index": 73,
  "quantity": 63.9785,
  "rate": 4.2370
}
{
  "color": "yellow",
  "shape": "circle",
  "flag": 1,
  "index": 87,
  "quantity": 63.5058,
  "rate": 8.3350
}
```

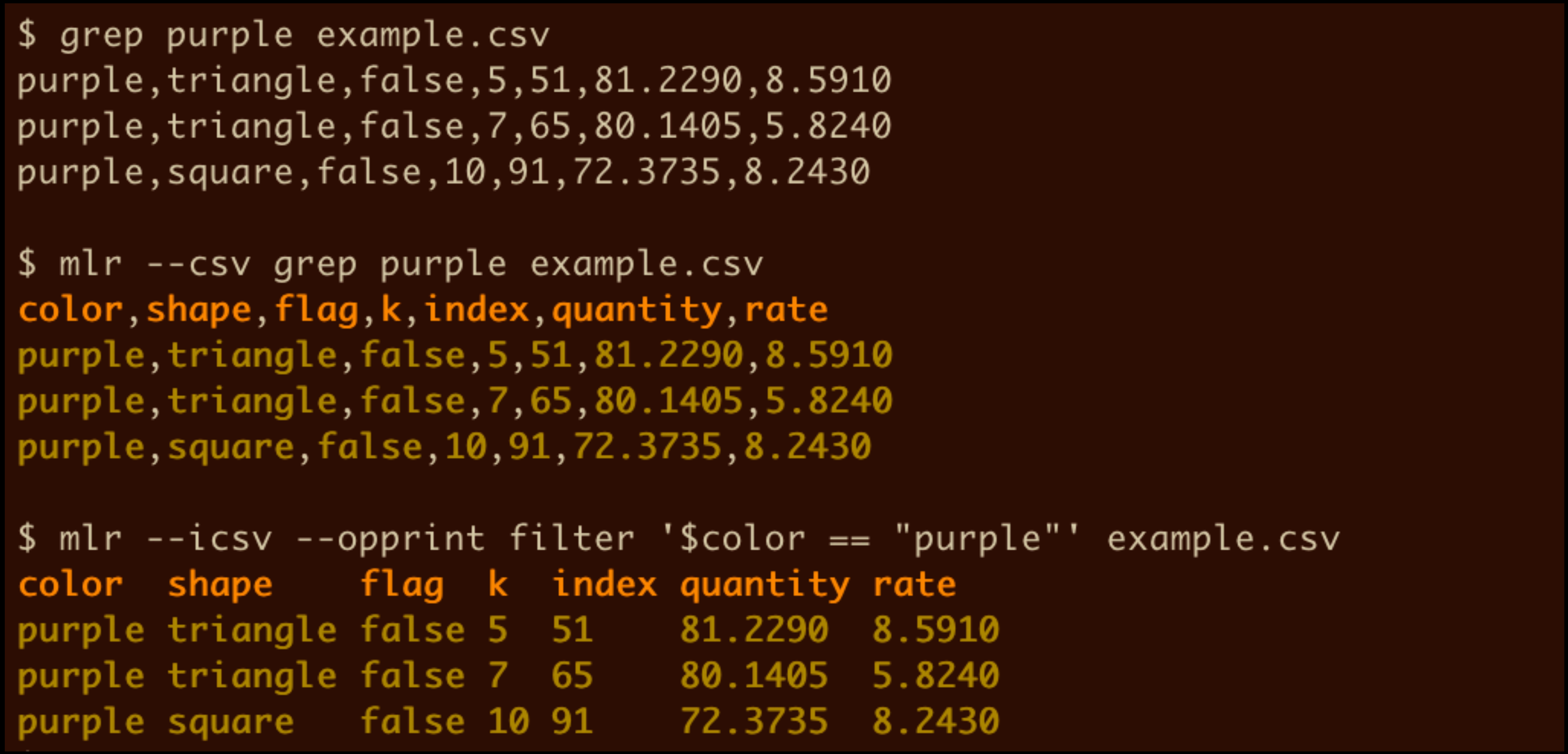
```
$ mlr --c2p --from example.csv put '$qr = $quantity * $rate'
color  shape    flag  k  index quantity rate   qr
yellow triangle true  1  11    43.6498  9.8870 431.5655726
red    square   true  2  15    79.2778  0.0130 1.0306114
red    circle   true  3  16    13.8103  2.9010 40.063680299999994
red    square   false 4  48    77.5542  7.4670 579.0972113999999
purple triangle false 5  51    81.2290  8.5910 697.8383389999999
red    square   false 6  64    77.1991  9.5310 735.7846221000001
purple triangle false 7  65    80.1405  5.8240 466.738272
yellow circle   true  8  73    63.9785  4.2370 271.0769045
yellow circle   true  9  87    63.5058  8.3350 529.3208430000001
purple square   false 10 91    72.3735  8.2430 596.5747605000001
```

John Kerl
Day job: TileDB
Project: https://tiledb.com/data-types/single-cell -- joint work with CZI
We're hiring!

Why? And, why Miller?

When you grep your `.csv`, I cry -- because you deserve better

- `grep`, `cut`, `sort` etc are line-aware and they're good for lines and integer column indices: `cut -d, -f 2,3; sort -k 7`
- Nacimiento, ~2015: I wanted a record-aware tool



```
$ grep purple example.csv
purple,triangle,false,5,51,81.2290,8.5910
purple,triangle,false,7,65,80.1405,5.8240
purple,square,false,10,91,72.3735,8.2430

$ mlr --csv grep purple example.csv
color,shape,flag,k,index,quantity,rate
purple,triangle,false,5,51,81.2290,8.5910
purple,triangle,false,7,65,80.1405,5.8240
purple,square,false,10,91,72.3735,8.2430

$ mlr --icsv --opprint filter '$color == "purple"' example.csv
color  shape    flag  k  index quantity rate
purple triangle false 5  51    81.2290  8.5910
purple triangle false 7  65    80.1405  5.8240
purple square   false 10 91    72.3735  8.2430
```

Many tools are out there

- **xsv** : custom indices, fast
- **zsv** : *really* fast
- **csvtk** : closer to **dplyr**
- **q** : supports SQL
- **jq** : *amazing* tool for JSON
- **nu** : a fully interactive data-aware shell, multiple file formats
- … and more including (of course!) **pandas**, **datasette**, **frictionless**, …

Miller (`mlr`) is ...

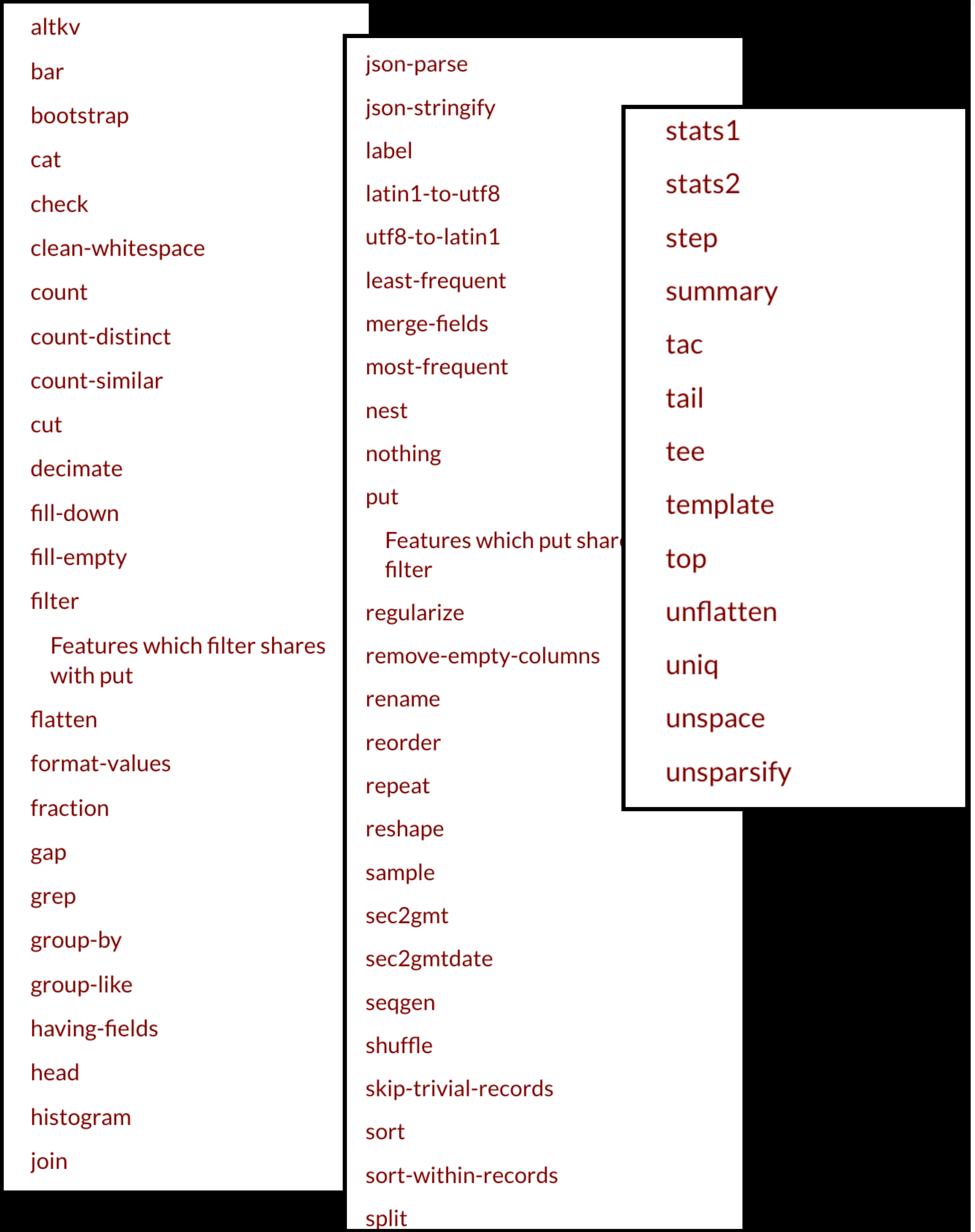
- Multiple file formats: CSV, TSV, JSON, JSON Lines, PPRINT, XTAB, DKVP, integer-indexed (like the Unix toolkit)
- Mix of:
  - **sort**/**cut**/etc equivalents
  - and an **awk**-like programming language
- **Unix-toolkit**/**Unix-pipe** family tree with **record awareness**
- Streaming/out-of-core/bigger-than-RAM when it can
- Open source, free, single binary without runtime dependencies

Installation

- MacOS, Windows, Linux, BSDs, ...
  - Older distros have older Miller versions
  - Latests: https://miller.readthedocs.io/en/latest/installing-miller/
- `brew install miller`
- `choco install miller`
- `yum install miller`
- `apt-get install miller`
- `conda install -c conda-forge miller`

Verbs; functions

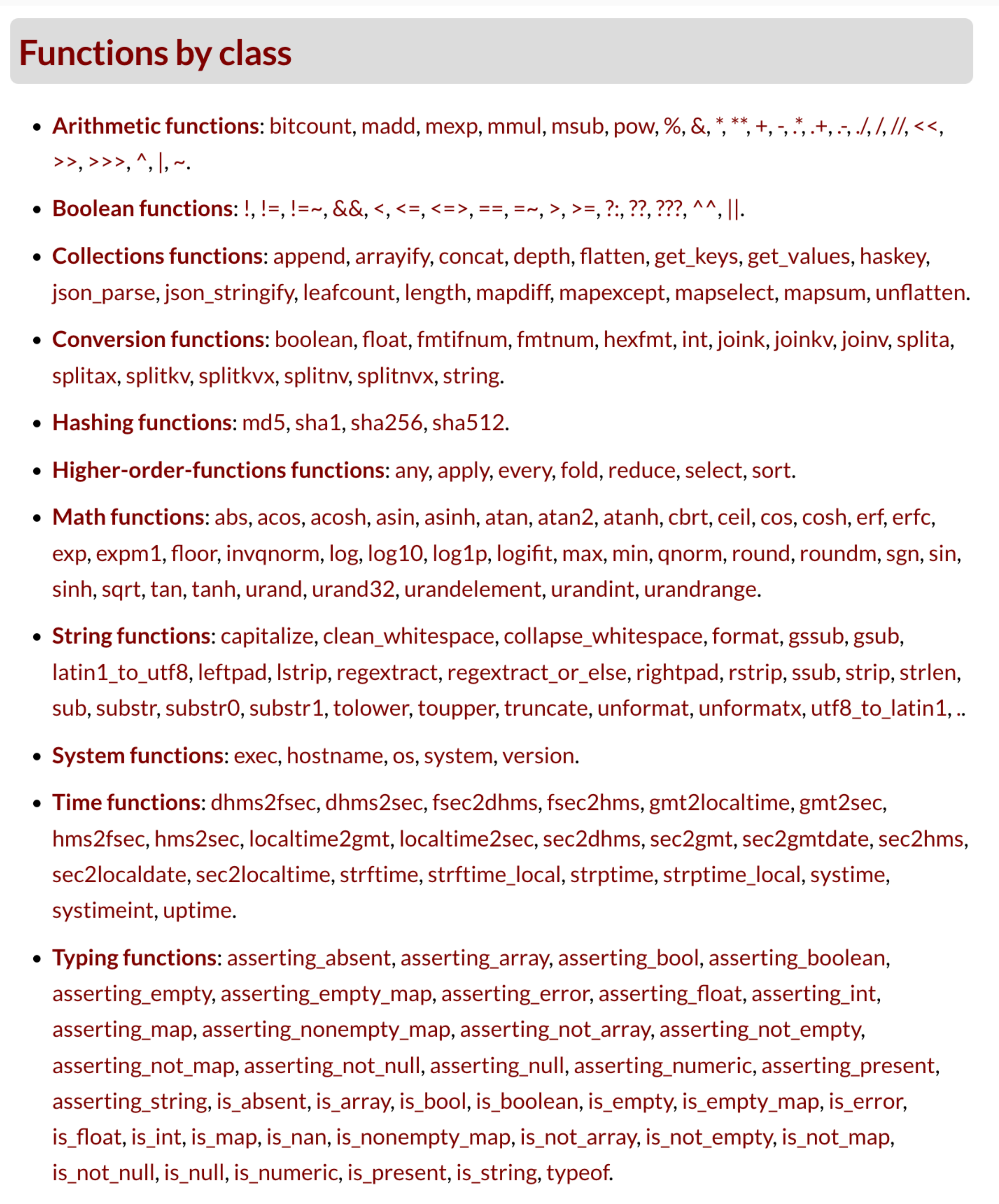
Like `cut`, `sort`, `sed`, `grep`; `awk`-like DSL



- altkv
- bar
- bootstrap
- cat
- check
- clean-whitespace
- count
- count-distinct
- count-similar
- cut
- decimate
- fill-down
- fill-empty
- filter
  - Features which filter shares with put
- flatten
- format-values
- fraction
- gap
- grep
- group-by
- group-like
- having-fields
- head
- histogram
- join

- json-parse
- json-stringify
- label
- latin1-to-utf8
- utf8-to-latin1
- least-frequent
- merge-fields
- most-frequent
- nest
- nothing
- put
  - Features which put shar filter
- regularize
- remove-empty-columns
- rename
- reorder
- repeat
- reshape
- sample
- sec2gmt
- sec2gmtdate
- seqgen
- shuffle
- skip-trivial-records
- sort
- sort-within-records
- split

- stats1
- stats2
- step
- summary
- tac
- tail
- tee
- template
- top
- unflatten
- uniq
- unspace
- unsparsify



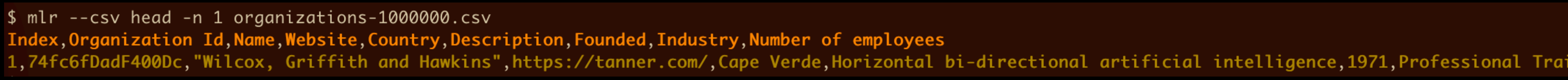
Functions by class

- **Arithmetic functions**: bitcount, madd, mexp, mmul, msub, pow, %, &, *, **, +, -, .*, .+, .-, ./, /, //, <<, >>, >>>, ^, |, ~.
- **Boolean functions**: !, !=, !=~, &&, <, <=, <=>, ==, =~, >, >=, ?:, ??, ???, ^^, ||.
- **Collections functions**: append, arrayify, concat, depth, flatten, get_keys, get_values, haskey, json_parse, json_stringify, leafcount, length, mapdiff, mapexcept, mapselect, mapsum, unflatten.
- **Conversion functions**: boolean, float, fmtifnum, fmtnum, hexfmt, int, joink, joinkv, joinv, splita, splitax, splitkv, splitkvx, splitnv, splitnvx, string.
- **Hashing functions**: md5, sha1, sha256, sha512.
- **Higher-order-functions functions**: any, apply, every, fold, reduce, select, sort.
- **Math functions**: abs, acos, acosh, asin, asinh, atan, atan2, atanh, cbrt, ceil, cos, cosh, erf, erfc, exp, expm1, floor, invqnorm, log, log10, log1p, logifit, max, min, qnorm, round, roundm, sgn, sin, sinh, sqrt, tan, tanh, urand, urand32, urandelement, urandint, urandrange.
- **String functions**: capitalize, clean_whitespace, collapse_whitespace, format, gssub, gsub, latin1_to_utf8, leftpad, lstrip, regextract, regextract_or_else, rightpad, rstrip, ssub, strip, strlen, sub, substr, substr0, substr1, tolower, toupper, truncate, unformat, unformatx, utf8_to_latin1, ..
- **System functions**: exec, hostname, os, system, version.
- **Time functions**: dhms2fsec, dhms2sec, fsec2dhms, fsec2hms, gmt2localtime, gmt2sec, hms2fsec, hms2sec, localtime2gmt, localtime2sec, sec2dhms, sec2gmt, sec2gmtdate, sec2hms, sec2localdate, sec2localtime, strftime, strftime_local, strptime, strptime_local, systime, systimeint, uptime.
- **Typing functions**: asserting_absent, asserting_array, asserting_bool, asserting_boolean, asserting_empty, asserting_empty_map, asserting_error, asserting_float, asserting_int, asserting_map, asserting_nonempty_map, asserting_not_array, asserting_not_empty, asserting_not_map, asserting_not_null, asserting_null, asserting_numeric, asserting_present, asserting_string, is_absent, is_array, is_bool, is_boolean, is_empty, is_empty_map, is_error, is_float, is_int, is_map, is_nan, is_nonempty_map, is_not_array, is_not_empty, is_not_map, is_not_null, is_null, is_numeric, is_present, is_string, typeof.

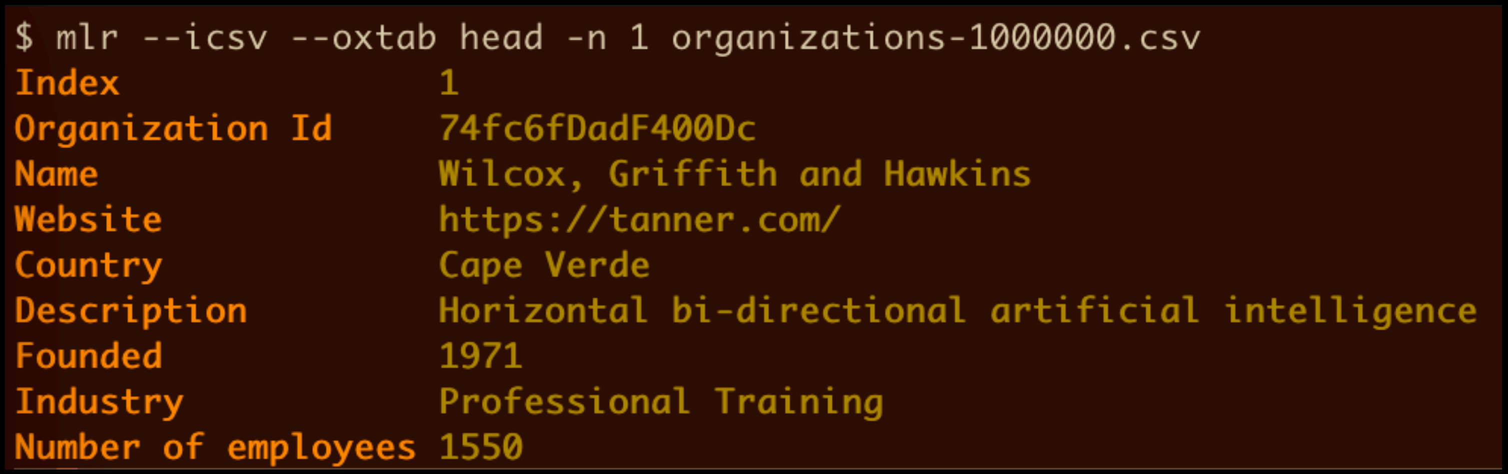
Exploring data

Data source: https://github.com/datablist/sample-csv-files

- What fields are in the data? (Use XTAB output for wide data)

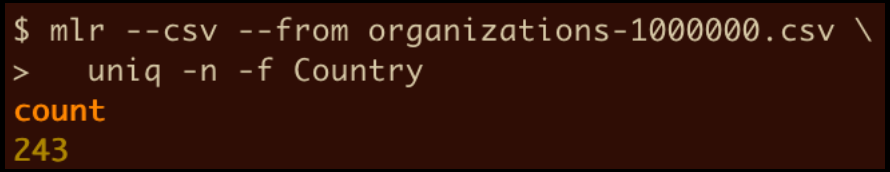


```
$ mlr --csv head -n 1 organizations-1000000.csv
Index,Organization Id,Name,Website,Country,Description,Founded,Industry,Number of employees
1,74fc6fDadF400Dc,"Wilcox, Griffith and Hawkins",https://tanner.com/,Cape Verde,Horizontal bi-directional artificial intelligence,1971,Professional Trai
```



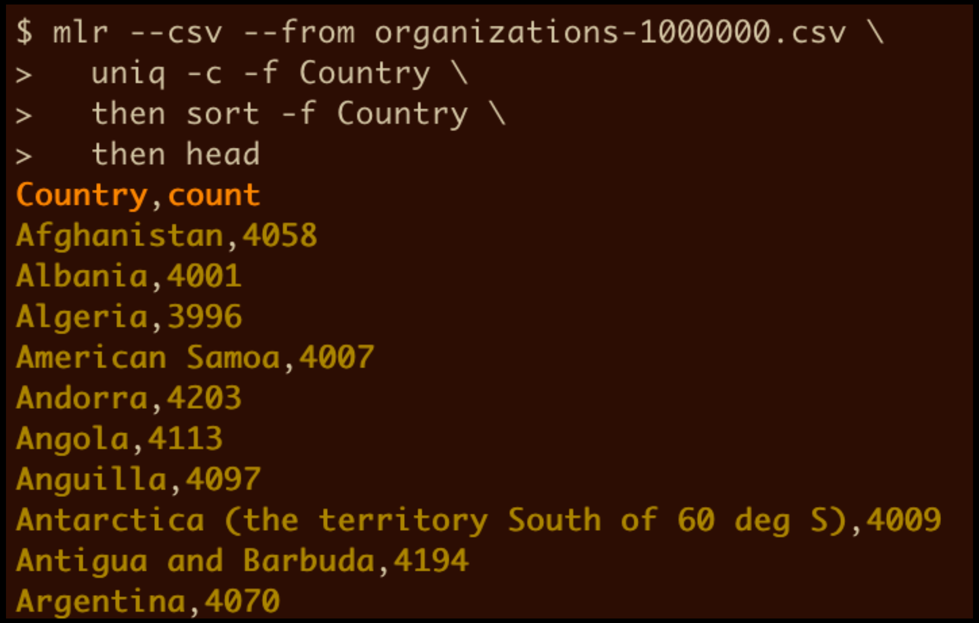
```
$ mlr --icsv --oxtab head -n 1 organizations-1000000.csv
Index               1
Organization Id     74fc6fDadF400Dc
Name                Wilcox, Griffith and Hawkins
Website             https://tanner.com/
Country             Cape Verde
Description         Horizontal bi-directional artificial intelligence
Founded             1971
Industry            Professional Training
Number of employees 1550
```

- How many countries?



```
$ mlr --csv --from organizations-1000000.csv \
>   uniq -n -f Country
count
243
```

- How many organizations in Argentina?

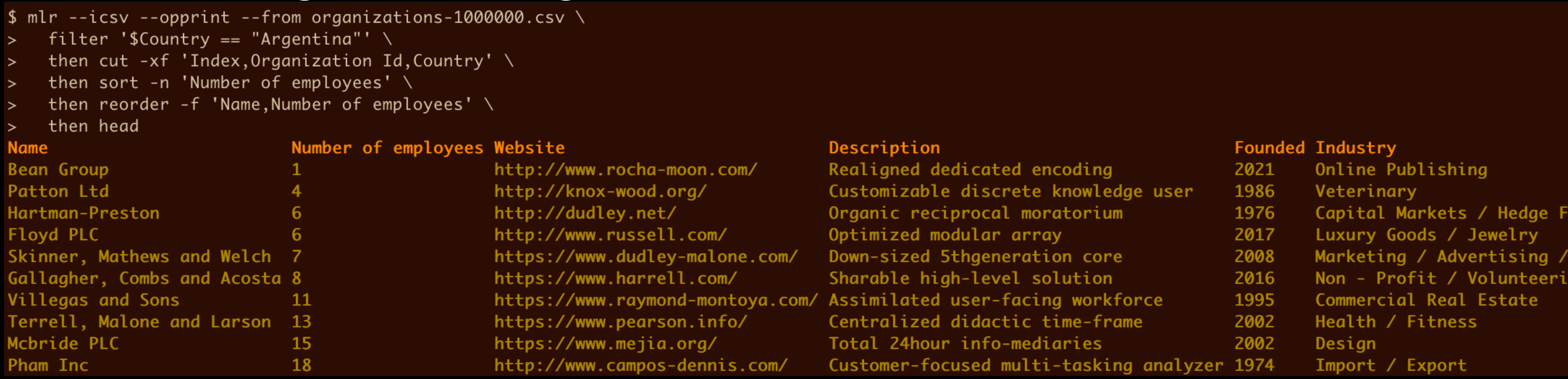


```
$ mlr --csv --from organizations-1000000.csv \
>   uniq -c -f Country \
>   then sort -f Country \
>   then head
Country,count
Afghanistan,4058
Albania,4001
Algeria,3996
American Samoa,4007
Andorra,4203
Angola,4113
Anguilla,4097
Antarctica (the territory South of 60 deg S),4009
Antigua and Barbuda,4194
Argentina,4070
```

- *Miller processes record streams*
- `--csv` *or* `--icsv --ojson`*: same format, or format conversion*
- `then` *is the pipe between verbs (transformations):* `mlr -l`
- *The* `filter` *and* `put` *verbs support a domain-specific language (DSL)*
- *You can set a* `~/.mlrrc` *file with defaults like* `--csv`

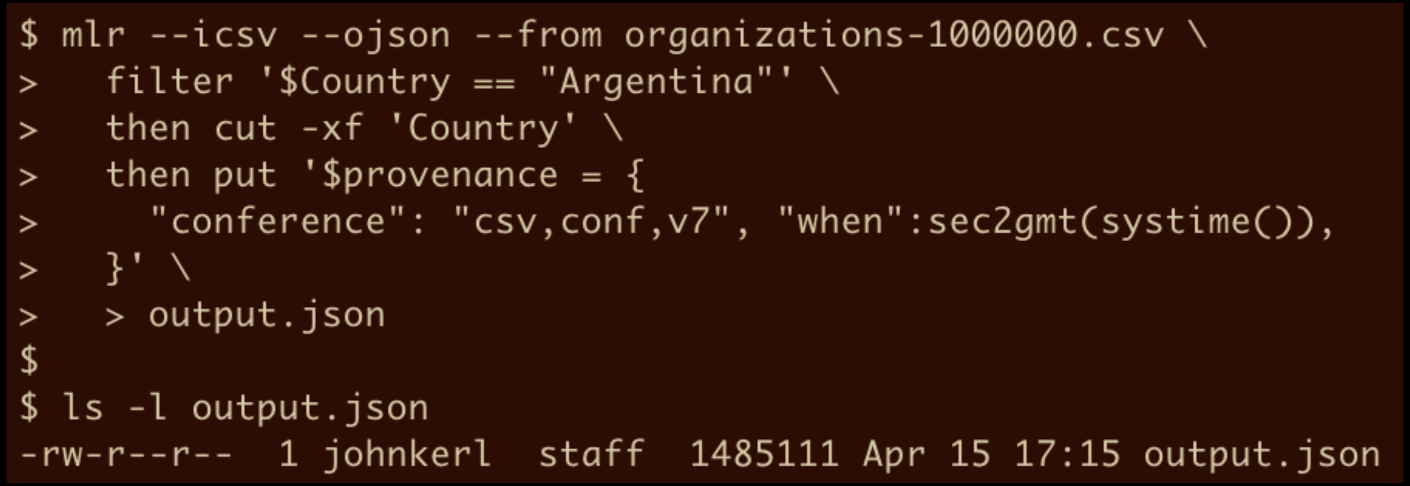
Exploring data

- Smallest Argentinian organizations?

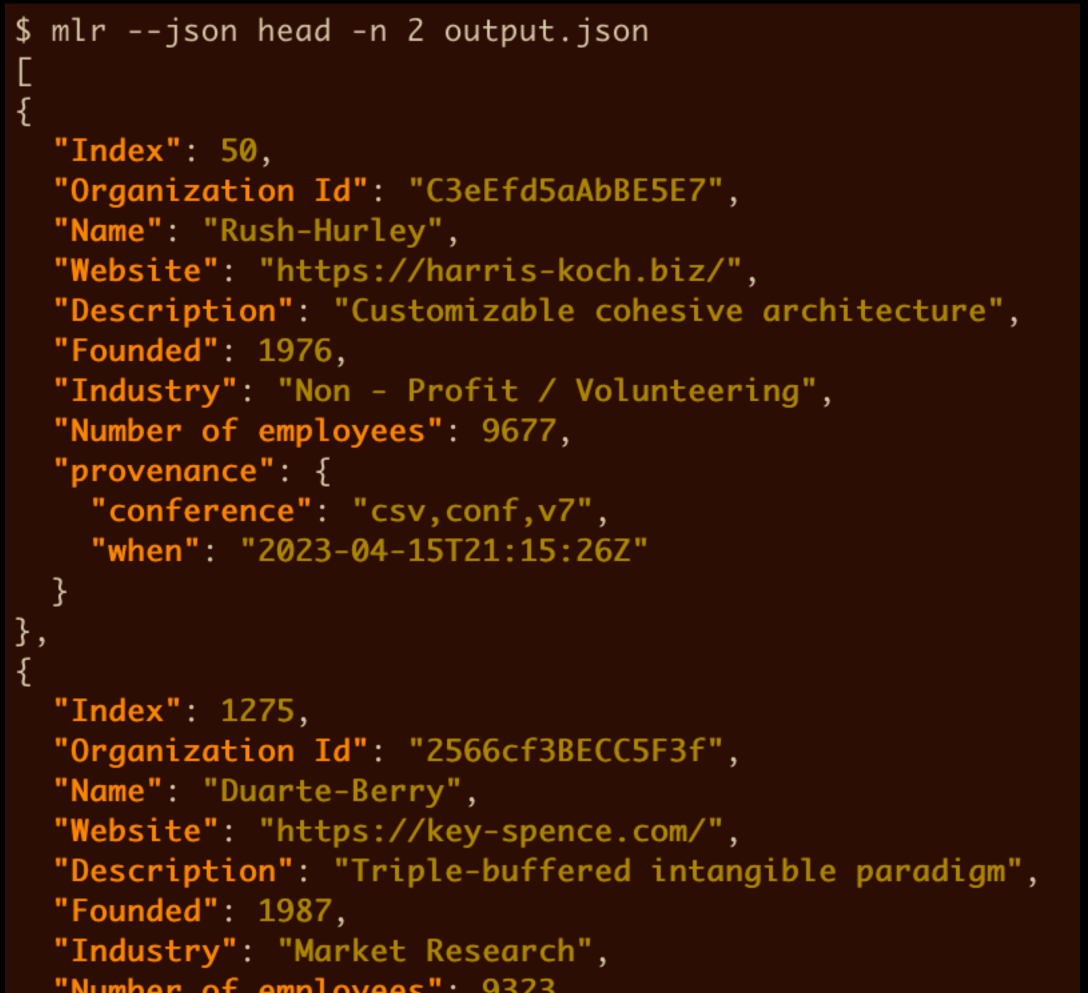


```
$ mlr --icsv --opprint --from organizations-1000000.csv \
>   filter '$Country == "Argentina"' \
>   then cut -xf 'Index,Organization Id,Country' \
>   then sort -n 'Number of employees' \
>   then reorder -f 'Name,Number of employees' \
>   then head
Name                        Number of employees Website                          Description                             Founded Industry
Bean Group                  1                   http://www.rocha-moon.com/       Realigned dedicated encoding            2021    Online Publishing
Patton Ltd                  4                   http://knox-wood.org/            Customizable discrete knowledge user    1986    Veterinary
Hartman-Preston             6                   http://dudley.net/               Organic reciprocal moratorium           1976    Capital Markets / Hedge F
Floyd PLC                   6                   http://www.russell.com/          Optimized modular array                 2017    Luxury Goods / Jewelry
Skinner, Mathews and Welch  7                   https://www.dudley-malone.com/   Down-sized 5thgeneration core           2008    Marketing / Advertising /
Gallagher, Combs and Acosta 8                   https://www.harrell.com/         Sharable high-level solution            2016    Non - Profit / Volunteeri
Villegas and Sons           11                  https://www.raymond-montoya.com/ Assimilated user-facing workforce       1995    Commercial Real Estate
Terrell, Malone and Larson  13                  https://www.pearson.info/        Centralized didactic time-frame         2002    Health / Fitness
Mcbride PLC                 15                  https://www.mejia.org/           Total 24hour info-mediaries             2002    Design
Pham Inc                    18                  http://www.campos-dennis.com/    Customer-focused multi-tasking analyzer 1974    Import / Export
```

- New fields, and JSON output



```
$ mlr --icsv --ojson --from organizations-1000000.csv \
>   filter '$Country == "Argentina"' \
>   then cut -xf 'Country' \
>   then put '$provenance = {
>     "conference": "csv,conf,v7", "when":sec2gmt(systime()),
>   }' \
>   > output.json
$
$ ls -l output.json
-rw-r--r--  1 johnkerl  staff  1485111 Apr 15 17:15 output.json
```

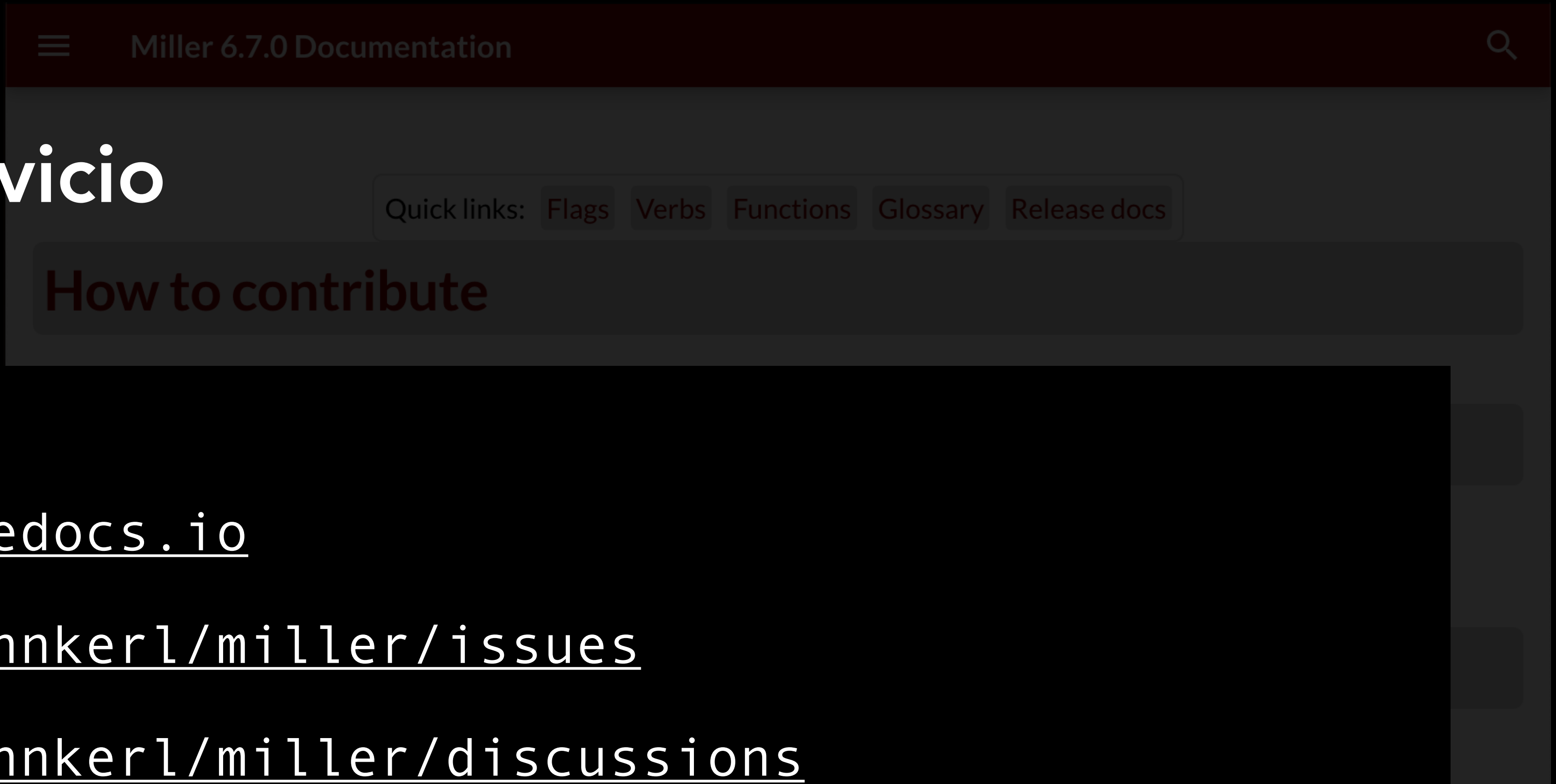


```
$ mlr --json head -n 2 output.json
[
{
  "Index": 50,
  "Organization Id": "C3eEfd5aAbBE5E7",
  "Name": "Rush-Hurley",
  "Website": "https://harris-koch.biz/",
  "Description": "Customizable cohesive architecture",
  "Founded": 1976,
  "Industry": "Non - Profit / Volunteering",
  "Number of employees": 9677,
  "provenance": {
    "conference": "csv,conf,v7",
    "when": "2023-04-15T21:15:26Z"
  }
},
{
  "Index": 1275,
  "Organization Id": "2566cf3BECC5F3f",
  "Name": "Duarte-Berry",
  "Website": "https://key-spence.com/",
  "Description": "Triple-buffered intangible paradigm",
  "Founded": 1987,
  "Industry": "Market Research",
  "Number of employees": 9323
```

Community

Ayuda / apoyo / servicio

- `man mlr` and `mlr help`
- https://miller.readthedocs.io
- https://github.com/johnkerl/miller/issues
- https://github.com/johnkerl/miller/discussions
- https://github.com/johnkerl/miller/discussions/1268 -- this talk
- Se necesita ayuda: lots of open feature requests (Go language
  - Miller is open source and development time is the most precious commodity


Miller 6.7.0 Documentation
Quick links: Flags Verbs Functions Glossary Release docs

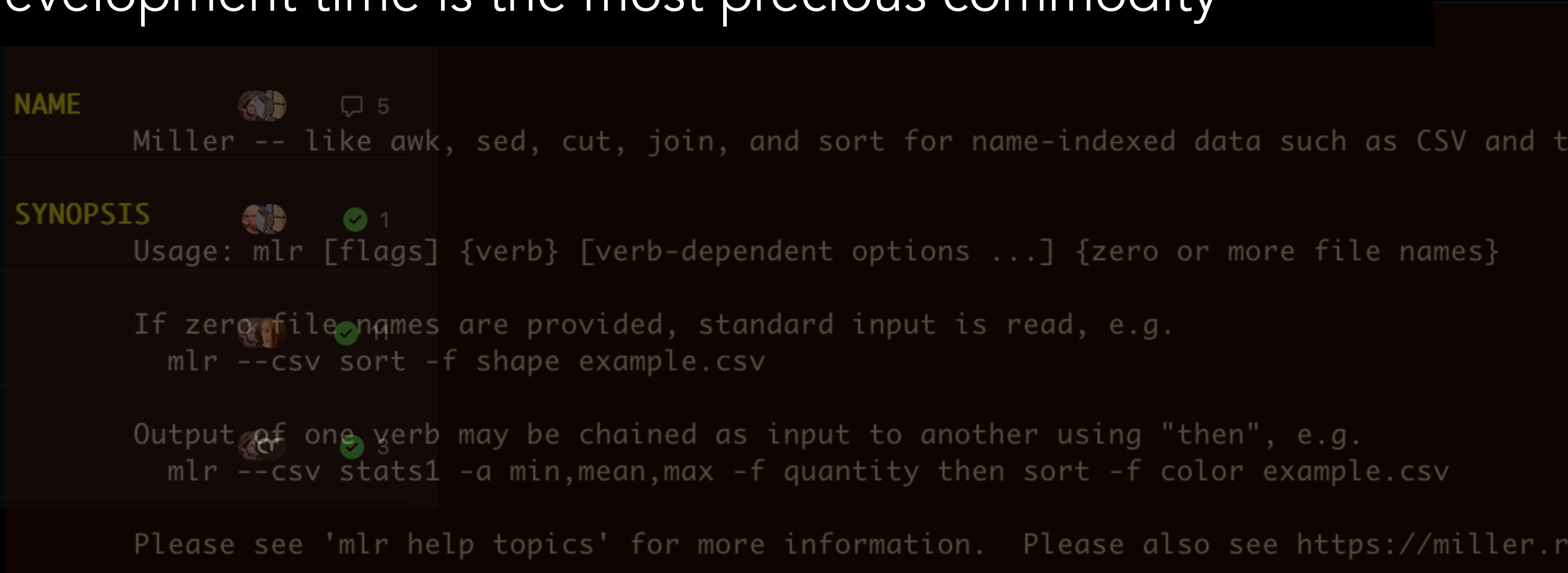
How to contribute

How to use environment variables for field name in DSL?
YokojimaSkewers started on Feb 25 in General

Can Miller remove records?
YokojimaSkewers asked on Feb 28 in Q&A · Answered

How can I right pad an integer with zeros?
derekmahar asked on Dec 19, 2022 in Q&A · Answered

strptime support for non zero-padded values?
archetyped asked on Feb 22 in Q&A · Answered



```
NAME
       Miller -- like awk, sed, cut, join, and sort for name-indexed data such as CSV and t

SYNOPSIS
       Usage: mlr [flags] {verb} [verb-dependent options ...] {zero or more file names}

       If zero file names are provided, standard input is read, e.g.
         mlr --csv sort -f shape example.csv

       Output of one verb may be chained as input to another using "then", e.g.
         mlr --csv stats1 -a min,mean,max -f quantity then sort -f color example.csv

       Please see 'mlr help topics' for more information.  Please also see https://miller.r
```

Questions?

Bonus slide: Data cleaning

Limpieza de datos

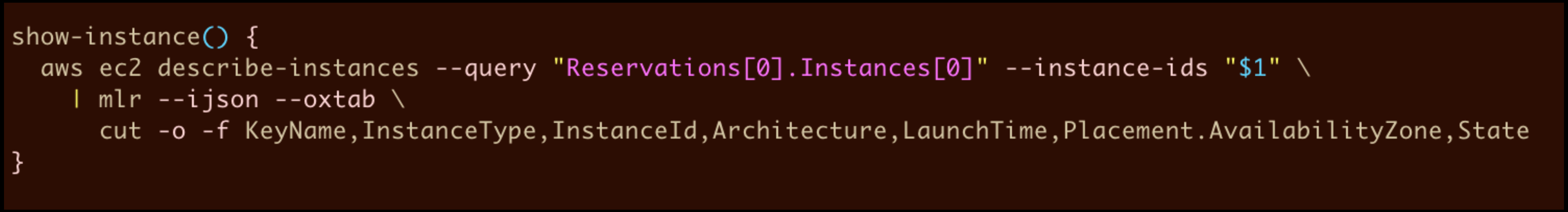
- (+) Miller has many verbs and functions for data cleaning
  - See also the data-cleaning examples page
- (-) It's a bit fussy about RFC 4180
  - Pro-tip: on parse failure, also try `--csvlite`
  - In the C implementation (Miller <= 5) I had a hand-written CSV parser
  - In the Go implementation I use the Go library's CSV parser -- better for me to revert to manual

Bonus slide: JSON processing

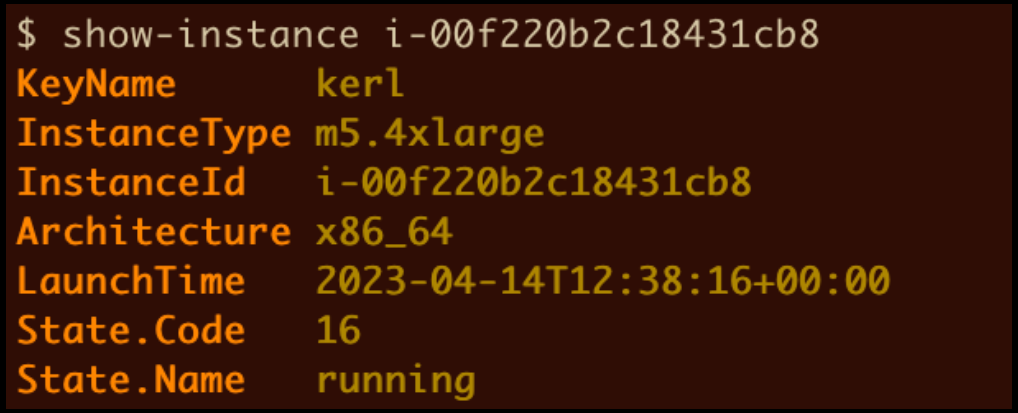
Ahorrando dinero con Miller

Here's one of a few shell aliases I use to manage my instances

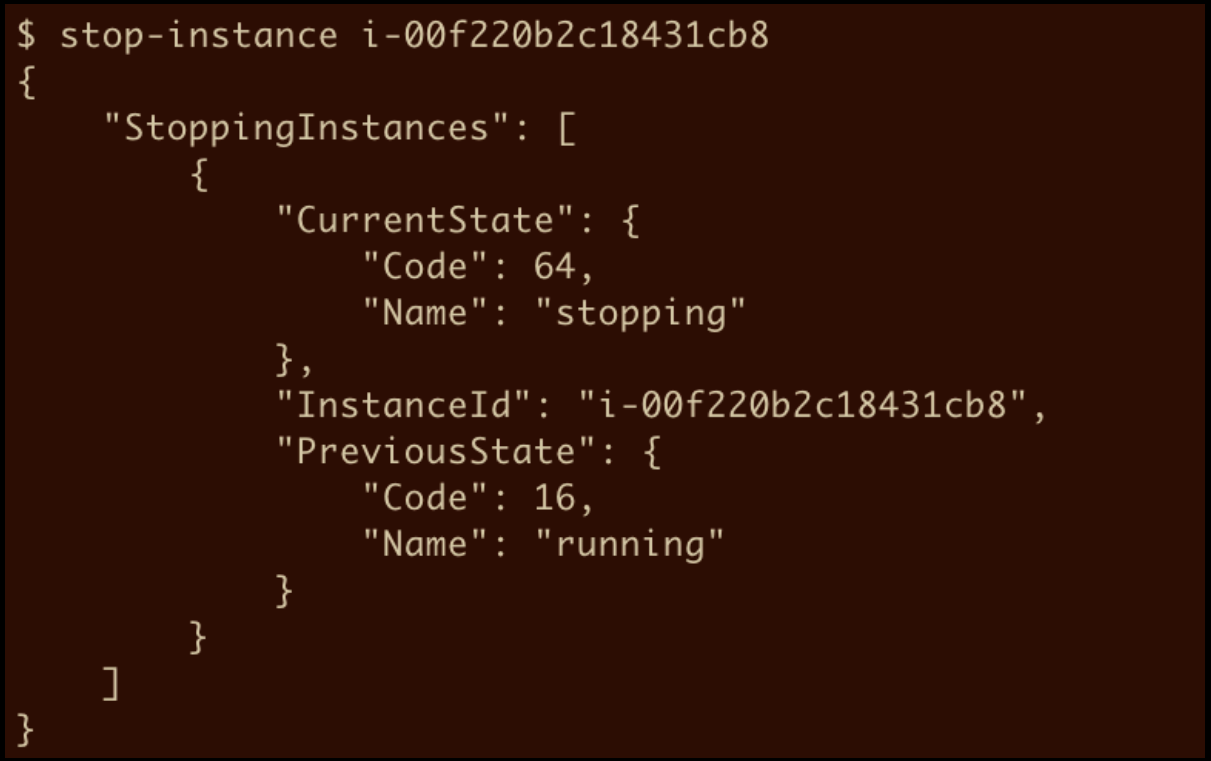
Together with a starter and a stopper alias -- my boss likes this!



```
show-instance() {
  aws ec2 describe-instances --query "Reservations[0].Instances[0]" --instance-ids "$1" \
    | mlr --ijson --oxtab \
      cut -o -f KeyName,InstanceType,InstanceId,Architecture,LaunchTime,Placement.AvailabilityZone,State
}
```



```
$ show-instance i-00f220b2c18431cb8
KeyName      kerl
InstanceType m5.4xlarge
InstanceId   i-00f220b2c18431cb8
Architecture x86_64
LaunchTime   2023-04-14T12:38:16+00:00
State.Code   16
State.Name   running
```



```
$ stop-instance i-00f220b2c18431cb8
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-00f220b2c18431cb8",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```