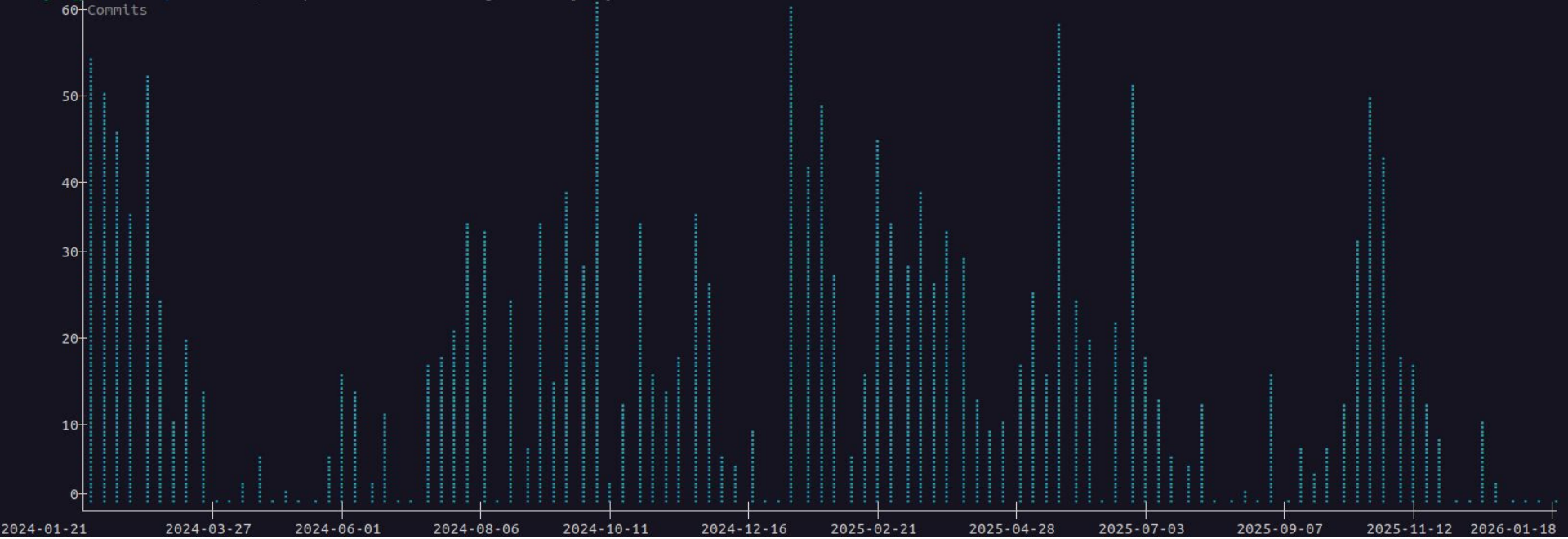


Data science from the command line

A look back at 2 years of using xan

Béatrice Mazoyer, Benjamin Ooghe-Tabanou, Guillaume Plique
médialab, Sciences Po Paris

01/02/2026

Who we are

- Our team of engineers develops research tools and methods for various applications in social sciences

- The médialab is an interdisciplinary research laboratory comprised of sociologists, engineers and designers

- **Guillaume Plique** (aka **Yomguithereal**) is the main developer of **xan**

- I (**Béatrice Mazoyer**) developed some functionalities and I'm a daily user

- **Benjamin Ooghe-Tabanou** is an occasional (but enthusiastic) user

What is xan?

- a command-line tool designed to manipulate CSV files

- originally, a fork of another tool called **xsv** (now unmaintained) developed in **Rust** by **Andrew Gallant**, aka **BurntSushi**

- now completely rewritten to fit our needs

Why we use xan

1. we love the CSV format
2. we love the terminal
3. xan is easy to use
4. xan is fast

Displaying 6 cols from 46 first rows of cartofriches_october_2023_enriched.csv

-	site_id	site_type	site_adresse	site_identif_date	site_url	unite_fonciere_surface
0	54425_23576	mixte	R GUY MOQUET	2022-10-10	NA	10555
1	66136_8117	inconnu	NA	2022-10-10	http://fiches-risques.b…	219
2	49099_4048	inconnu	NA	2022-10-10	http://fiches-risques.b…	2258
3	49007_10337	inconnu	NA	2022-10-10	https://basol.developpe…	132684
4	76640_14629	inconnu	NA	2022-10-10	http://fiches-risques.b…	2321
5	54173_9994	inconnu	NA	2022-10-10	http://fiches-risques.b…	877761
6	57550_23395	friche industrielle	NA	2023-08-08	NA	973834
7	63263_17591	inconnu	NA	2022-10-10	http://fiches-risques.b…	2204
8	71230_13472	inconnu	NA	2022-10-10	NA	142759
9	79081_13948	inconnu	NA	2022-10-10	NA	194295
10	32393_13587	inconnu	NA	2022-10-10	https://fiches-risques.…	262332
11	23096_17934	inconnu	NA	2022-10-10	http://fiches-risques.b…	92980
12	67365_3929	inconnu	NA	2022-10-10	http://fiches-risques.b…	633
13	59197_4498	inconnu	NA	2022-10-10	http://fiches-risques.b…	53719
14	57296_23585	friche logistique	NA	2022-10-10	NA	23181
15	19033_8770	inconnu	NA	2022-10-10	http://fiches-risques.b…	2737
16	59526_4276	inconnu	NA	2022-10-10	http://fiches-risques.b…	19780
17	30298_10878	inconnu	NA	2022-10-10	NA	122013
18	30176_11964	inconnu	NA	2022-10-10	https://fiches-risques.…	47893
19	36168_5487	inconnu	NA	2022-10-10	http://fiches-risques.b…	2263
20	01307_23181	friche commerciale	rue de l'Ancre	2021-02-01	NA	6702
21	67154_19790	inconnu	NA	2022-10-10	http://fiches-risques.b…	5185
22	45176_10149	inconnu	NA	2022-10-10	http://fiches-risques.b…	113830
23	41018_6190	inconnu	NA	2022-10-10	https://basol.developpe…	563
24	55296_24038	friche industrielle	25 RUE HENRI CHEVALIER,…	2022-10-10	NA	23665
25	63250_17406	inconnu	NA	2022-10-10	http://fiches-risques.b…	3993
26	57001_23477	friche ferroviaire	NA	2022-10-10	NA	11280
27	54542_24042	friche ferroviaire	R MARECHAL LYAUTEY	2022-10-10	NA	208121
28	57221_24090	autre	NA	2022-10-10	NA	29679
29	89387_19527	inconnu	NA	2022-10-10	http://fiches-risques.b…	1570
30	80677_15827	inconnu	NA	2022-10-10	http://fiches-risques.b…	6912
31	88011_23985	friche industrielle	15 RUE DE LA GARE, ARCH…	2022-10-10	NA	27163
32	48095_16715	inconnu	NA	2022-10-10	http://fiches-risques.b…	14862
33	42155_13047	inconnu	NA	2022-10-10	https://fiches-risques.…	97863
34	59569_10250	inconnu	NA	2022-10-10	http://fiches-risques.b…	292
35	33056_13201	inconnu	NA	2022-10-10	https://fiches-risques.…	64164
36	38344_17142	inconnu	NA	2022-10-10	https://basol.developpe…	72949
37	16230_11705	inconnu	NA	2022-10-10	http://fiches-risques.b…	3925
38	38423_17041	inconnu	NA	2022-10-10	https://basol.developpe…	173590
39	29103_20050	inconnu	NA	2022-10-10	http://fiches-risques.b…	58474
40	76216_22582	friche industrielle	NA	2017-10-24	NA	37363
41	54305_24080	friche hospitalière	LE SANATORIUM, LAY-SAIN…	2022-10-10	NA	87138
42	67482_5876	inconnu	NA	2022-10-10	http://fiches-risques.b…	561
43	67482_19963	inconnu	NA	2022-10-10	http://fiches-risques.b…	643
44	54425_24169	autre	24 RUE DE VERDUN, PIENN…	2022-10-10	NA	9728
45	56223_7199	inconnu	NA	2022-10-10	http://fiches-risques.b…	540
…	…	…	…	…	…	…

Why we love the CSV format

There are many reasons

- Guillaume already wrote a **love letter to the CSV format** that can be found here:

https://medialab.sciencespo.fr/en/news/a-love-letter-to-the-csv-format/

- I guess it is an unpopular opinion

Y **Hacker News** new | threads | past | comments | ask | show | jobs | submit

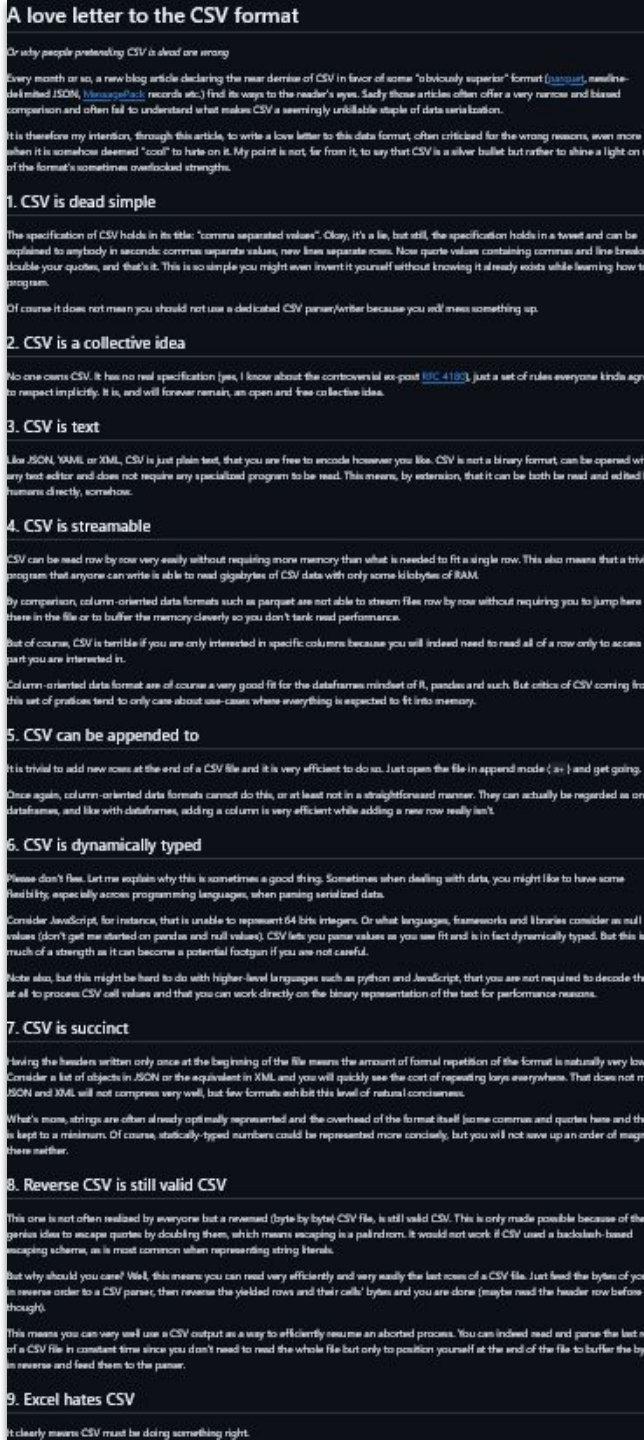
▲ A love letter to the CSV format (github.com/medialab)

708 points by Yomguithereal 10 months ago | hide | past | favorite | 689 comments

▲ jwr 10 months ago | next [–]

I so hate CSV.

see: https://news.ycombinator.com/item?id=43484382



A love letter to the CSV format

Or why people pretending CSV is dead are wrong

Every month or so, a new blog article declaring the near demise of CSV in favor of some "obviously superior" format (parquet, newline-delimited JSON, MessagePack records etc.) find its ways to the reader's eyes. Sadly those articles often offer a very narrow and biased comparison and often fail to understand what makes CSV a seemingly unkillable staple of data serialization.

It is therefore my intention, through this article, to write a love letter to this data format, often criticized for the wrong reasons, even more when it is somehow deemed "cool" to hate on it. My point is not, far from it, to say that CSV is a silver bullet but rather to shine a light on of the format's sometimes overlooked strengths.

1. CSV is dead simple

The specification of CSV holds in its title: "comma separated values". Okay, it's a lie, but still, the specification holds in a tweet and can be explained to anybody in seconds: commas separate values, new lines separate rows. Now quote values containing commas and line breaks double your quotes, and that's it. This is so simple you might even invent it yourself without knowing it already exists while learning how to program.

Of course it does not mean you should not use a dedicated CSV parser/writer because you will mess something up.

2. CSV is a collective idea

No one owns CSV. It has no real specification (yes, I know about the controversial ex-post RFC 4180), just a set of rules everyone kinda agr to respect implicitly. It is, and will forever remain, an open and free collective idea.

3. CSV is text

Like JSON, YAML or XML, CSV is just plain text, that you are free to encode however you like. CSV is not a binary format, can be opened wi any text editor and does not require any specialized program to be read. This means, by extension, that it can be both be read and edited humans directly, somehow.

4. CSV is streamable

CSV can be read row by row very easily without requiring more memory than what is needed to fit a single row. This also means that a triv program that anyone can write is able to read gigabytes of CSV data with only some kilobytes of RAM.

By comparison, column-oriented data formats such as parquet are not able to stream files row by row without requiring you to jump here there in the file or to buffer the memory cleverly so you don't tank read performance.

But of course, CSV is terrible if you are only interested in specific columns because you will indeed need to read all of a row only to access part you are interested in.

Column-oriented data format are of course a very good fit for the dataframes mindset of R, pandas and such. But critics of CSV coming fro this set of pratices tend to only care about use-cases where everything is expected to fit into memory.

5. CSV can be appended to

It is trivial to add new rows at the end of a CSV file and it is very efficient to do so. Just open the file in append mode (`a+`) and get going.

Once again, column-oriented data formats cannot do this, or at least not in a straightforward manner. They can actually be regarded as on dataframes, and like with dataframes, adding a column is very efficient while adding a new row really isn't.

6. CSV is dynamically typed

Please don't flee. Let me explain why this is sometimes a good thing. Sometimes when dealing with data, you might like to have some flexibility, especially across programming languages, when parsing serialized data.

Consider JavaScript, for instance, that is unable to represent 64 bits integers. Or what languages, frameworks and libraries consider as null values (don't get me started on pandas and null values). CSV lets you parse values as you see fit and is in fact dynamically typed. But this is much of a strength as it can become a potential footgun if you are not careful.

Note also, but this might be hard to do with higher-level languages such as python and JavaScript, that you are not required to decode th at all to process CSV cell values and that you can work directly on the binary representation of the text for performance reasons.

7. CSV is succinct

Having the headers written only once at the beginning of the file means the amount of formal repetition of the format is naturally very lo Consider a list of objects in JSON or the equivalent in XML and you will quickly see the cost of repeating keys everywhere. That does not m JSON and XML will not compress very well, but few formats exhibit this level of natural conciseness.

What's more, strings are often already optimally represented and the overhead of the format itself (some commas and quotes here and th is kept to a minimum. Of course, statically-typed numbers could be represented more concisely, but you will not save up an order of magn there neither.

8. Reverse CSV is still valid CSV

This one is not often realized by everyone but a reversed (byte-by-byte) CSV file, is still valid CSV. This is only made possible because of the genius idea to escape quotes by doubling them, which means escaping is a palindrom. It would not work if CSV used a backslash-based escaping scheme, as is most common when representing string literals.

But why should you care? Well, this means you can read very efficiently and very easily the last rows of a CSV file. Just feed the bytes of yo in reverse order to a CSV parser, then reverse the yielded rows and their cells' bytes and you are done (maybe read the header row before though).

This means you can very well use a CSV output as a way to efficiently resume an aborted process. You can indeed read and parse the last of a CSV file in constant time since you don't need to read the whole file but only to position yourself at the end of the file to buffer the by in reverse and feed them to the parser.

9. Excel hates CSV

It clearly means CSV must be doing something right.

Social scientists are used to tabular data

- We often think in terms of variables (columns) spread across a population of individuals (rows)

- We use CSV to interoperate a variety of tools dealing with tabular data (Excel, Stata, R, Pandas...)

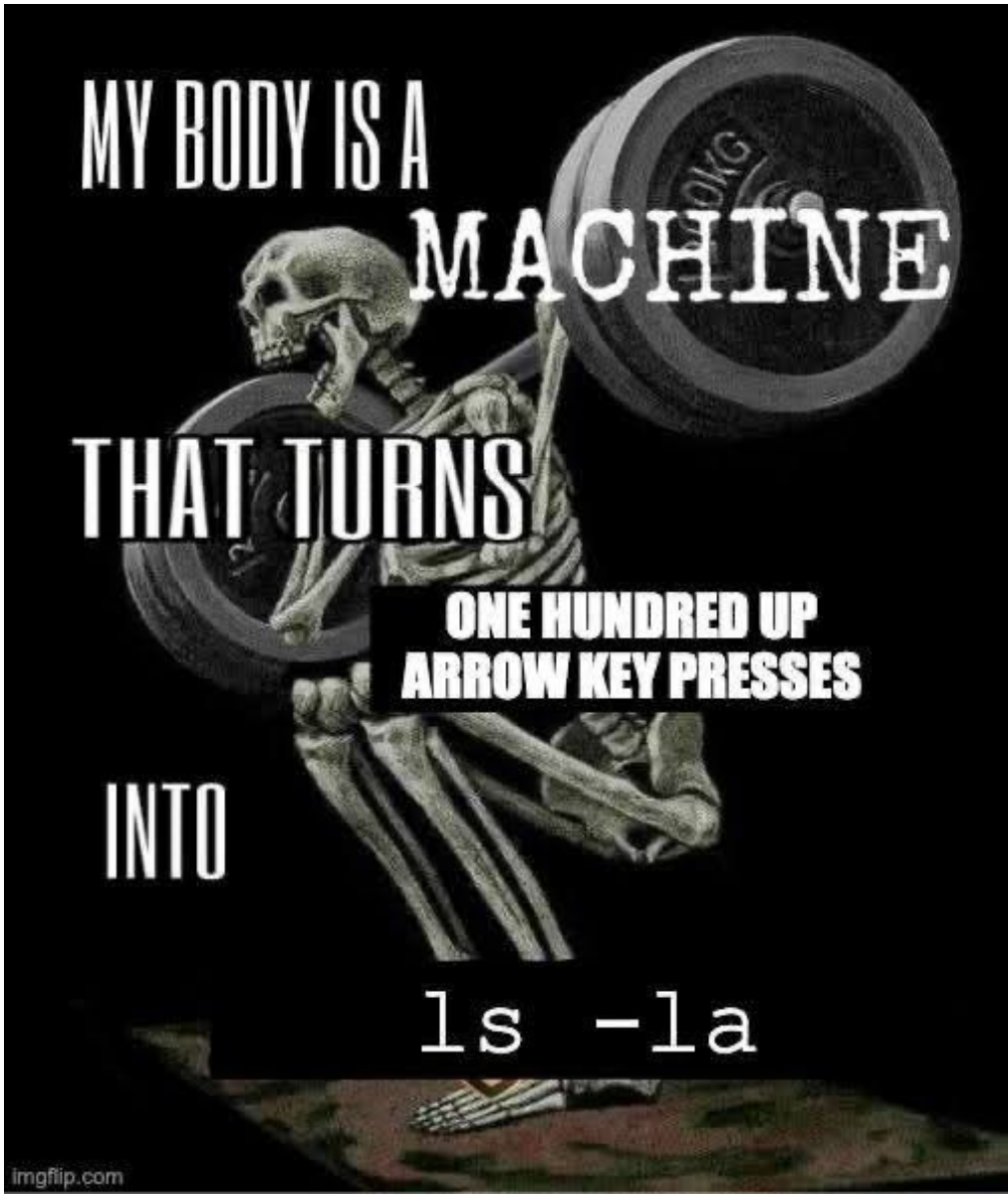
```
$ xan freq -s urba_zone_lib cartofriches_october_2023_enriched.csv -l 46 | xan hist

Histogram for urba_zone_lib (bars: 47, sum: 22,864, max: 11,890):

NA                                                  | 8,246  36.07%|
A - Zone agricole                                   |   247   1.08%|
N - Zone naturelle                                  |   215   0.94%|
UF - UF : Zone urbaine des faubourgs d'Angoulême    |   164   0.72%|
UA - UA                                             |   113   0.49%|
N - N                                               |   111   0.49%|
UA - UA : Noyaux historiques des communes de la premièr… |   107   0.47%|
UE - Zone d'activités diversifiées                  |   104   0.45%|
A - A                                               |    95   0.42%|
UM - UM : Zone urbaine à dominante résidentielle en mut… |    88   0.38%|
UB - UB : Zone urbaine des extensions urbaines des bour… |    86   0.38%|
UCA3.1 - Villes-centres d'agglomération - Tissu résiden… |    78   0.34%|
UB - UB                                             |    71   0.31%|
UCM2.1.1 - Coeur métropolitain - Tissu mixte dense  |    67   0.29%|
U - Zone urbaine                                    |    64   0.28%|
N - N : Zone naturelle                              |    54   0.24%|
UAB - Zone urbaine mixte à dominante habitat de central… |    54   0.24%|
UXM - Zone urbaine d'activités économiques mixtes   |    54   0.24%|
A - A : Zone agricole                               |    52   0.23%|
PSMV - Secteur sauvegardé                           |    50   0.22%|
UBA1 - Zone urbaine mixte à dominante habitat individuel |    48   0.21%|
N - Zone naturelle et forestière                    |    47   0.21%|
UEi1 - Cette zone regroupe les espaces qui accueillent … |    41   0.18%|
UEi2 - Cette zone regroupe les espaces qui accueillent … |    41   0.18%|
UG - Zones urbaines générales                       |    41   0.18%|
UCa - UCa : Secteur central général : Danton, Sainte Ma… |    40   0.17%|
A - zone agricole                                   |    39   0.17%|
UA - Zone urbaine dense du bourg et des villages    |    36   0.16%|
UAA - Zone urbaine mixte à dominante habitat de central… |    35   0.15%|
UCA3.1.2 - Villes-centres d'agglomération - Tissu résid… |    35   0.15%|
Ux - Secteurs d'activités économiques à vocation commer… |    35   0.15%|
UC - UC                                             |    32   0.14%|
N - N : Zone naturelle et forestière                |    31   0.14%|
U MIXTE 1 - U mixte 1                               |    30   0.13%|
UA - Zone urbaine de bourg et des villages anciens  |    30   0.13%|
A - A : Zone agricole générale                      |    29   0.13%|
UCM1.1.1 - Coeur métropolitain - Centralités        |    29   0.13%|
U ECO 1 - U éco 1 -                                 |    28   0.12%|
UAe3 - Centre-ville  de Marseille en évolution      |    28   0.12%|
UCA1.1 - Villes-centres d'agglomération - Centralités |    28   0.12%|
N - N : zone naturelle                              |    27   0.12%|
N - zone naturelle                                  |    27   0.12%|
UX - UX : Zone urbaine à vocation d'activités économiqu… |    27   0.12%|
Uib - Secteur destiné aux activités et installations po… |    26   0.11%|
UG - Zone urbaine générale                          |    25   0.11%|
UMa - Secteur de développement des centralités actuelle… |    25   0.11%|
<rest>                                              |11,890  52.00%|
```

Why we love the terminal

We do repetitive tasks

And the terminal

- is a shortcut
- is an archive of previous activities
- doesn't change much



```
$ xan help

Usage:
    xan [options] <command> [<args>...]
    xan [options]

Options:
    -h, --help    Display this message
    <command> -h  Display the command help message
    --version     Print version info and exit

Common options:
    -n, --no-headers       Typically used to indicate that input file has no headers.
    -d, --delimiter <arg>  Typically used to indicate a custom delimiter.

    help        Get help regarding the expression language
    --version   Print the tool's version

## Explore & visualize
    count       Count rows in file
    headers (h) Show header names
    view    (v) Preview a CSV file in a human-friendly way
    flatten (f) Display a flattened version of each row of a file
    hist        Print a histogram with rows of CSV file as bars
    plot        Draw a scatter plot or line chart
    heatmap     Draw a heatmap of a CSV matrix
    progress    Display a progress bar while reading CSV data

## Search & filter
    search      Search for (or replace) patterns in CSV data
    grep        Coarse but fast filtering of CSV data
    filter      Only keep some CSV rows based on an evaluated expression
    head        First rows of CSV file
    tail        Last rows of CSV file
    slice       Slice rows of CSV file
    top         Find top rows of a CSV file according to some column
    sample      Randomly sample CSV data

## Sort & deduplicate
    sort        Sort CSV data
    dedup       Deduplicate a CSV file
    shuffle     Shuffle CSV data

## Aggregate
    frequency (freq) Show frequency tables
    groupby          Aggregate data by groups of a CSV file
    stats            Compute basic statistics
    agg              Aggregate data from CSV file
    bins             Dispatch numeric columns into bins
    window           Compute window aggregations (cumsum, rolling mean, lag etc.)

## Combine multiple CSV files
    cat         Concatenate by row or column
```

Why xan is easy to use

Just test a few commands, and you get started

- Don’t read the doc! Read the quick tour!

  https://github.com/medialab/xan?tab=readme-ov-file#quick-tour

Just test a few commands, and you get started

- What commands are the most useful?

- I asked five colleagues to grep “xan” in their bash history and send me the result. I obtained 1636 lines, where I searched for the most frequent xan commands.

Just test a few commands, and you get started

- **xan view** (142 times), alias **xan v** (162 times): shows a tabular file

```
$ xan view fosdem_file.csv

Displaying 5/7 cols from 100 first rows of fosdem_file.csv
```

-	site_id	site_identif_date	site_type	…	site_url	unite_fonciere_surface
0	54425_23576	2022-10-10	mixte	…	NA	10555
1	66136_8117	2022-10-10	inconnu	…	http://fiches-risques.br…	219
2	49099_4048	2022-10-10	inconnu	…	http://fiches-risques.br…	2258
3	49007_10337	2022-10-10	inconnu	…	https://basol.developpem…	132684

Just test a few commands, and you get started

- **xan view** (142 times), alias **xan v** (162 times): shows a tabular file

```
$ xan view fosdem_file.csv

Displaying 5/7 cols from 100 first rows of fosdem_file.csv
```

-	site_id	site_identif_date	site_type	…	site_url	unite_fonciere_surface
0	54425_23576	2022-10-10	mixte	…	NA	10555
1	66136_8117	2022-10-10	inconnu	…	http://fiches-risques.br…	219
2	49099_4048	2022-10-10	inconnu	…	http://fiches-risques.br…	2258
3	49007_10337	2022-10-10	inconnu	…	https://basol.developpem…	132684

- **xan headers** (170 times), alias **xan h** (7 times): shows the file's headers

```
$ xan headers fosdem_file.csv
0   site_id
1   site_identif_date
2   site_type
3   site_adresse
4   urba_zone_lib
5   site_url
6   unite_fonciere_surface
$
```

Just a few commands, and you get started

- xan map (207 times): applies an operation on each row

```
$ xan map 'site_type.replace("inconnu", "") as site_type_clean' fosdem_file.csv | xan v

Displaying 6/8 cols from 100 first rows of <stdin>
```

-	site_id	site_identif_date	site_type	…	site_url	unite_fonciere_surface	site_type_clean
0	54425_23576	2022-10-10	mixte	…	NA	10555	mixte
1	66136_8117	2022-10-10	inconnu	…	http://fic…	219	<empty>
2	49099_4048	2022-10-10	inconnu	…	http://fic…	2258	<empty>

Just a few commands, and you get started

- **xan map** (207 times): applies an operation on each row

```
$ xan map 'site_type.replace("inconnu", "") as site_type_clean' fosdem_file.csv | xan v

Displaying 6/8 cols from 100 first rows of <stdin>
```

-	site_id	site_identif_date	site_type	…	site_url	unite_fonciere_surface	site_type_clean
0	54425_23576	2022-10-10	mixte	…	NA	10555	mixte
1	66136_8117	2022-10-10	inconnu	…	http://fic…	219	<empty>
2	49099_4048	2022-10-10	inconnu	…	http://fic…	2258	<empty>

- **xan search** (176 times): searches a string

```
$ xan search "Zone urbaine" fosdem_file.csv | xan view -s urba_zone_lib

Displaying 1 col from 100 first rows of <stdin>
```

-	urba_zone_lib
0	UB1e - Zone urbaine type faubourg en lien avec etude densification - hauteur 9 - 12
1	UBA1 - Zone urbaine mixte à dominante habitat individuel
2	UXcp - Zone urbaine a vocation commerces et activites de services peripheriques

Just a few commands, and you get started

- **xan map** (207 times): applies an operation on each row

```
$ xan map 'site_type.replace("inconnu", "") as site_type_clean' fosdem_file.csv | xan v

Displaying 6/8 cols from 100 first rows of <stdin>
```

-	site_id	site_identif_date	site_type	…	site_url	unite_fonciere_surface	site_type_clean
0	54425_23576	2022-10-10	mixte	…	NA	10555	mixte
1	66136_8117	2022-10-10	inconnu	…	http://fic…	219	<empty>
2	49099_4048	2022-10-10	inconnu	…	http://fic…	2258	<empty>

- **xan search** (176 times): searches a string

```
$ xan search "Zone urbaine" fosdem_file.csv | xan view -s urba_zone_lib

Displaying 1 col from 100 first rows of <stdin>
```

-	urba_zone_lib
0	UB1e - Zone urbaine type faubourg en lien avec etude densification - hauteur 9 - 12
1	UBA1 - Zone urbaine mixte à dominante habitat individuel
2	UXcp - Zone urbaine a vocation commerces et activites de services peripheriques

- **xan frequency** (1 time), alias **xan freq** (149 times): computes the frequency of each modality

```
$ xan freq -s site_identif_date fosdem_file.csv | xan v

Displaying 3 cols from 11 rows of <stdin>
```

-	field	value	count
0	site_identif_date	2022-10-10	16540
1	site_identif_date	2021-05-10	928
2	site_identif_date	2022-03-01	927

CSV in, CSV out

- You can write the result of your xan command to a new CSV file
- But also: xan is modular, since you can pipe commands one into another

tiborschneider opened yesterday

Currently, to compute an eCDF, I do the following:

```
xan progress --total 206081665 data.csv \
| xan map 'pow(10, round(log10(time_sec) * 10) / 10) as time' \
| xan groupby 'class,time' 'count(class eq "A") as a, count(class eq "B") as b, count(class
| xan sort -s 'time' -N \
| xan window 'cumsum(a) as cum_a, cumsum(b) as cum_b, cumsum(c) as cum_c' \
| xan window 'max(cum_a) as max_a, max(cum_br) as max_b, max(cum_c) as max_c' \
| xan map 'if(class eq "A", cum_a, if(class eq "B", cum_b, cum_c)) as cum_grouped' \
| xan map 'if(class eq "A", max_a, if(class eq "B", max_b, max_c)) as max_grouped' \
| xan map 'cum_grouped / max_grouped * 100 as probability' \
| xan plot 'time' 'probability' --category 'class' --rows 50 --cols 200
```

It would be super nice if there is a built-in command for this, as I assume this is a common use-case (at least for me). If you like this idea, I might try to add a PR eventually (maybe in a couple of weeks, I am currently very busy)

xan is full of useful tricks

- xan select with *

```
$ xan select site* fosdem_file.csv | xan v

Displaying 5 cols from 100 first rows of <stdin>
```

-	site_id	site_identif_date	site_type	site_adresse	site_url
0	54425_23576	2022-10-10	mixte	R GUY MOQUET	NA
1	66136_8117	2022-10-10	inconnu	NA	http://fiches-risques.br…
2	49099_4048	2022-10-10	inconnu	NA	http://fiches-risques.br…

xan is full of useful tricks

- xan select with *

```
$ xan select site* fosdem_file.csv | xan v

Displaying 5 cols from 100 first rows of <stdin>
```

-	site_id	site_identif_date	site_type	site_adresse	site_url
0	54425_23576	2022-10-10	mixte	R GUY MOQUET	NA
1	66136_8117	2022-10-10	inconnu	NA	http://fiches-risques.br…
2	49099_4048	2022-10-10	inconnu	NA	http://fiches-risques.br…

- xan hist with -D/--dates

```
$ xan search -s site_identif_date 2022-10-1 fosdem_file.csv | xan freq -s site_identif_date | xan hist -D

Histogram for site_identif_date (bars: 10, sum: 16,554, max: 16,540):

2022-10-10 |16,540  99.92%|
2022-10-11 |     0   0.00%|
2022-10-12 |     0   0.00%|
2022-10-13 |     0   0.00%|
2022-10-14 |     0   0.00%|
2022-10-15 |     0   0.00%|
2022-10-16 |     0   0.00%|
2022-10-17 |     0   0.00%|
2022-10-18 |     0   0.00%|
2022-10-19 |    14   0.08%|
```

xan is full of useful tricks

- xan select with *
- xan hist with -D/--dates
- xan flatten with -H/--highlight

```
$ xan flatten fosdem_file.csv -H UB -l 3
Row n°0
─────────────────────────────────────────────────────────────────────────────
site_id                 54425_23576
site_identif_date       2022-10-10
site_type               mixte
site_adresse            R GUY MOQUET
urba_zone_lib           UB - UB
site_url                NA
unite_fonciere_surface  10555

Row n°1
─────────────────────────────────────────────────────────────────────────────
site_id                 66136_8117
site_identif_date       2022-10-10
site_type               inconnu
site_adresse            NA
urba_zone_lib           UB1e - Zone urbaine type faubourg en lien avec etude densification - h
uteur 9 - 12
site_url                http://fiches-risques.brgm.fr/georisques/basias-detaillee/LRO6601292
```

xan is full of useful tricks

- xan select with *
- xan hist with -D/--dates
- xan flatten with -H/--highlight
- xan parallel cat with -P/--preprocess

```
2022-08-02.csv  2022-09-16.csv  2022-10-31.csv  2022-12-15.csv  2023-01-29.csv
2022-08-03.csv  2022-09-17.csv  2022-11-01.csv  2022-12-16.csv  2023-01-30.csv
$ xan parallel cat 2022*.csv -P "search Parliament" | xan flatten -l 3 -H Parliament
Row n°0
─────────────────────────────────────────────────────────────────────────────────
id                   1542524494325002245
timestamp_utc        1656601472
local_time           2022-06-30T17:04:32
user_screen_name     tgassilloud
text                 RT @julienstrandt: Great to see one of the French Parliament's foremost experts
n cyber @tgassilloud as the new president of the @AN_Defense. Surely digital conflict won't slip under t
e radar of the Parliament that way!
possibly_sensitive   <empty>
retweet_count        2
```

```
$ xan parallel cat 2023*.csv --progress -P "grep FOSDEM" | xan v
          0 rows of 2023-01-05.csv in 0s (0/s)
          0 rows of 2023-01-04.csv in 0s (0/s)
          0 rows of 2023-01-16.csv in 0s (0/s)
          0 rows of 2023-01-01.csv in 0s (0/s)
          1 rows of 2023-02-06.csv in 0s (19.6031/s)
          0 rows of 2023-01-03.csv in 0s (0/s)
          0 rows of 2023-01-08.csv in 0s (0/s)
          0 rows of 2023-01-14.csv in 0s (0/s)
          1 rows of 2023-02-24.csv in 0s (16.753/s)
          0 rows of 2023-01-10.csv in 0s (0/s)
          0 rows of 2023-01-11.csv in 0s (0/s)
          0 rows of 2023-01-15.csv in 0s (0/s)
          0 rows of 2023-01-12.csv in 0s (0/s)
          0 rows of 2023-01-07.csv in 0s (0/s)
          0 rows of 2023-01-19.csv in 0s (0/s)
          0 rows of 2023-01-02.csv in 0s (0/s)
          0 rows of 2023-02-07.csv in 0s (0/s)
          0 rows of 2023-02-15.csv in 0s (0/s)
          0 rows of 2023-03-05.csv in 0s (0/s)
          0 rows of 2023-02-10.csv in 0s (0/s)
          0 rows of 2023-03-02.csv in 0s (0/s)
          0 rows of 2023-01-06.csv in 0s (0/s)
          0 rows of 2023-02-28.csv in 0s (0/s)
          0 rows of 2023-01-09.csv in 0s (0/s)
          0 rows of 2023-01-28.csv in 0s (0/s)
          0 rows of 2023-02-25.csv in 0s (0/s)
          0 rows of 2023-01-13.csv in 0s (0/s)
          1 rows of 2023-02-08.csv in 0s (20.1743/s)
          0 rows of 2023-02-01.csv in 0s (0/s)
          1 rows of 2023-03-03.csv in 0s (16.7478/s)
          0 rows of 2023-01-17.csv in 0s (0/s)
          0 rows of 2023-01-20.csv in 0s (0/s)
          0 rows of 2023-02-09.csv in 0s (0/s)
          0 rows of 2023-02-16.csv in 0s (0/s)
          0 rows of 2023-01-30.csv in 0s (0/s)
          0 rows of 2023-01-29.csv in 0s (0/s)
          0 rows of 2023-02-26.csv in 0s (0/s)
          0 rows of 2023-02-27.csv in 0s (0/s)
          0 rows of 2023-03-01.csv in 0s (0/s)
          0 rows of 2023-03-04.csv in 0s (0/s)
          0 rows of 2023-03-06.csv in 0s (0/s)
          0 rows of 2023-02-11.csv in 0s (0/s)
          0 rows of 2023-02-03.csv in 0s (0/s)
          0 rows of 2023-02-12.csv in 0s (0/s)
          2 rows of 2023-02-02.csv in 0s (37.0038/s)
```

How fast is xan?

Demo time - Python standard library

frequency_std.py

```
1   import csv
2   from collections import Counter
3   import sys
4
5   filename = sys.argv[1]
6
7   writer = csv.writer(sys.stdout)
8
9   with open(filename) as f:
10      reader = csv.reader(f)
11
12      header = next(reader)
13      retweeted_user_col = header.index("retweeted_user")
14
15      frequency_counter = Counter(row[retweeted_user_col] for row in reader)
16
17  writer.writerow(["retweeted_user", "count"])
18  for row in frequency_counter.most_common():
19      writer.writerow(row)
```

Demo time - Pandas

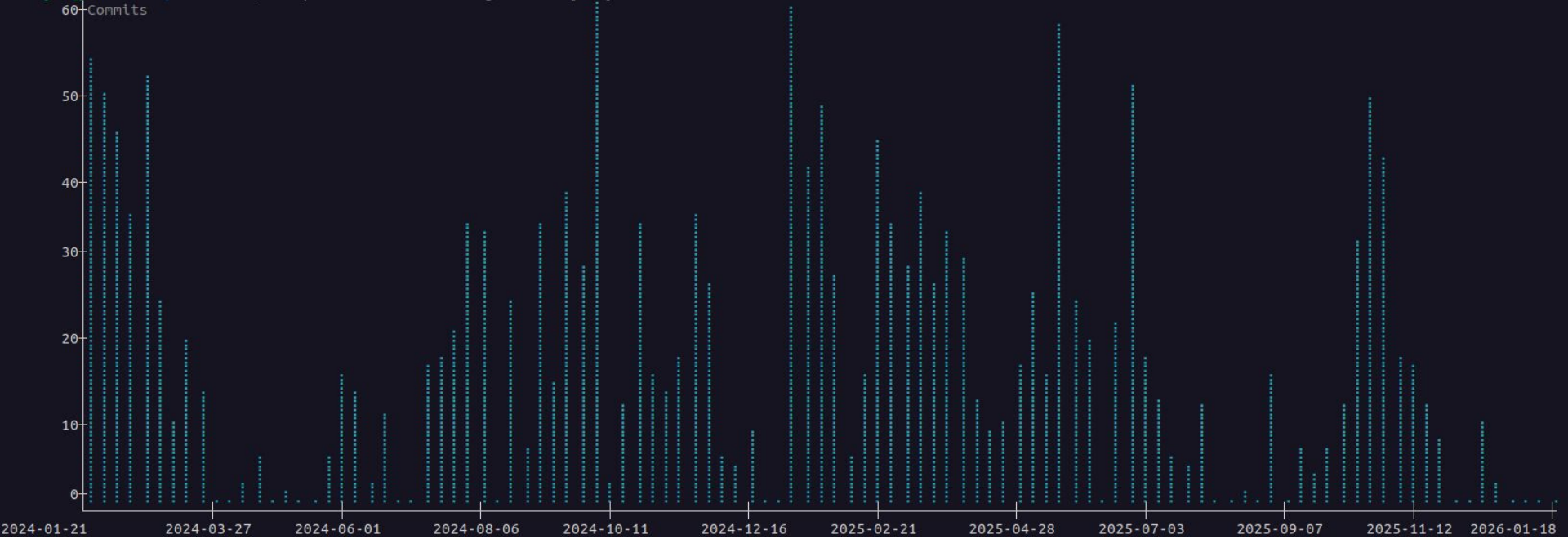
frequency_pandas.py

```
1   import pandas as pd
2   import sys
3
4   filename = sys.argv[1]
5
6   tweets = pd.read_csv(filename)
7
8   frequency_counter = tweets.value_counts("retweeted_user").rename("count")
9
10  frequency_counter.to_csv(sys.stdout, index=True)
11
```

Demo time - Polars

frequency_polars.py

```python
import polars as pl
import sys

filename = sys.argv[1]

p = (
    pl.scan_csv(filename, cache=False)
    .select(pl.col("retweeted_user").value_counts(sort=True))
    .unnest("retweeted_user")
)

p.sink_csv(sys.stdout)

```



Thank you!

https://github.com/medialab/xan