

FB: Data Preview (csvstat, csvcut, head, cat)

WE'LL COVER THE FOLLOWING ^

- Data download
- Learning objectives
- Dataset Preview
 - How many columns and rows?
 - How the data looks like?
- Do you want to know more?

Now that the Facebook has more than a billion of active users, it really has become a personal, product and corporate branding hub. Companies would like to understand what people think about topics related to their business, so they can make their products and marketing more relevant to their customers. One way to achieve such goal is to analyse company's FB pages which can make marketing content more relevant for marketers.

In this lesson, we're going to mine a dataset generated by using a [Facebook scraper](#) on a particular Facebook page (undisclosed). The goal of this experiment is to find the most vibrant status message on that page, with just one Bash command.

Video Lecture: Facebook data preview

Data download

You should download the data from the [here](#)- Educative's webpage, as we have slightly simplified the data and Let's save the data as: `facebookdata.csv`. Soon we will see that the dataset contains the following attributes: `status_id`, `status_message`, `link_name`, `status_type`, `status_link`, `status_published`, `num_reactions`, `num_comments`, `num_shares`, `num_likes`, `num_loves`, `num_wows`, `num_hahas`, `num_sads`, `num_angrys`. From this data, using Bash we will explore different features and finally find which message was the most vibrant in terms of total number of activities.

Learning objectives

By completing this, you will learn to use the following Bash commands:

- `head` – output the first part of files
- `tail` – opposite to head
- `cat` – concatenate and print files
- `sort` – sort file contents

- `grep` – search the input files for lines containing a match to a given pattern list
- `uniq` – remove duplicate entries
- `awk` – programming language (New!)
- Bash `functions` (New!)

Dataset Preview

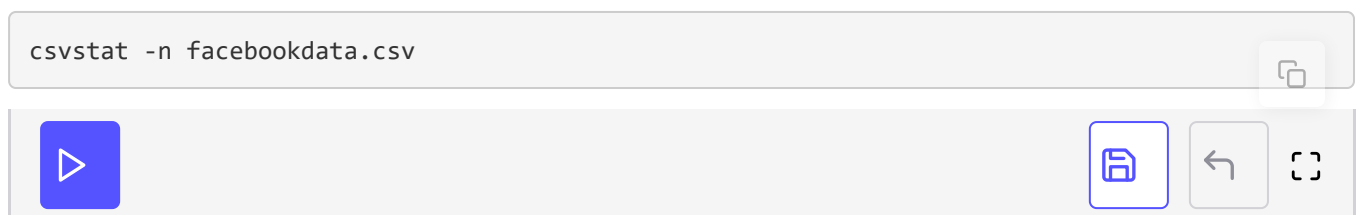
Same as before, this dataset is also small (toy) and we could in principle open it in a text editor or in Excel. However, as mentioned in the first chapter, real-world datasets are often larger and cumbersome to open in their entirety. Instead, let's get a sneak peak of the data.

How many columns and rows?

First, let us, find some stat about the data using `csvstat` tool from the `csvkit`:

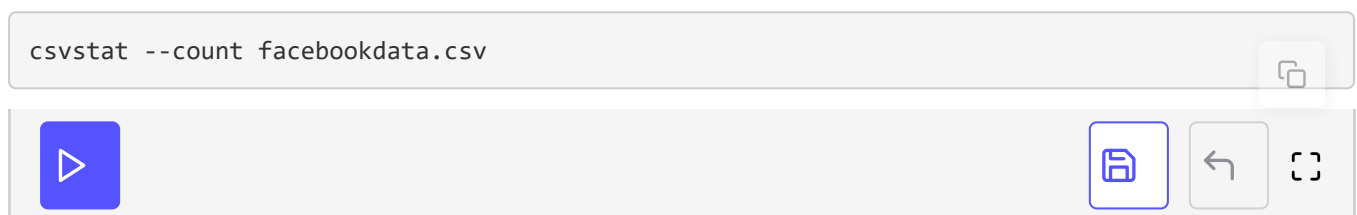
Finding the stat of the columns:

```
csvstat -n facebookdata.csv
```

A terminal window with a light gray background. The command 'csvstat -n facebookdata.csv' is entered in a dark gray input bar. Below the input bar is a row of controls: a blue square button with a white play icon, a blue square button with a white floppy disk icon, a gray square button with a white left arrow, and a gray square button with a white double-outlined square icon. A small copy icon is visible in the top right corner of the input bar.

Finding the `stat` of the rows:

```
csvstat --count facebookdata.csv
```

A terminal window with a light gray background. The command 'csvstat --count facebookdata.csv' is entered in a dark gray input bar. Below the input bar is a row of controls: a blue square button with a white play icon, a blue square button with a white floppy disk icon, a gray square button with a white left arrow, and a gray square button with a white double-outlined square icon. A small copy icon is visible in the top right corner of the input bar.

Final output:

```
facebookdata: bash
hello@igdata:~$ csvstat -n facebookdata.csv
 1: status_id
 2: status_message
 3: link_name
 4: status_type
 5: status_link
 6: status_published
 7: num_reactions
 8: num_comments
 9: num_shares
10: num_likes
11: num_loves
12: num_wows
13: num_hahas
14: num_sods
15: num_angrya
hello@igdata:~$ csvstat --count facebookdata.csv
Row count: 3222
hello@igdata:~$
```

facebookdata.csv stats using csvstat

It looks like that the dataset has a total of 11 columns and 3222 rows.

How the data looks like?

This is often the first thing to do when you get your hands on new data; previewing it is important to get a sense for what it contains, how it is organized, and whether the data makes sense in the first place. To help us get a preview of the data, we can use the command `head`, `csvlook` and `csvcut`:

```
csvcut -c 1,4,7-11 facebookdata.csv | csvlook | head -n 50
```



```
facebookdata: bash
hellobigdata@bash:facebookdata$ csvcut -c 1,4,7-11 facebookdata.csv | csvlook | head -n 50
```

status_id	status_type	num_reactions	num_comments	num_shares	num_likes	num_loves
7331091005_10154123560186006	video	5,565	178	461	5,488	43
7331091005_10154123362896006	video	11,997	1,932	3,158	10,385	96
7331091005_10154123319126006	link	2,063	270	400	1,971	28
7331091005_10154123234521006	photo	116,543	11,811	43,923	107,561	1,202
7331091005_10154123219076006	link	10,475	999	2,978	8,833	28
7331091005_10154123189346006	photo	32,111	897	2,525	29,660	398
7331091005_10154123132896006	video	1,261	628	90	1,173	43
7331091005_10154123136121006	video	9,257	283	819	8,776	348
7331091005_10154123102181006	photo	13,314	147	917	12,771	433
7331091005_10154123021136006	video	43,325	4,447	35,534	41,914	621
7331091005_10154122959911006	link	449	19	33	441	3
7331091005_10154122914261006	video	3,717	38	355	3,580	97
7331091005_10154122756301006	video	131	11	10	126	4
7331091005_10154122866096006	video	3,985	149	366	3,786	82
7331091005_10154122823271006	link	4,468	1,026	1,632	4,190	20
7331091005_10154122765336006	photo	25,707	739	2,413	24,667	774
7331091005_10154122722061006	photo	48,708	1,326	3,840	43,658	503
7331091005_10154122686201006	link	2,655	89	100	2,596	14
7331091005_10154122655576006	video	8,894	395	811	7,685	99
7331091005_10154122459461006	link	3,702	222	578	3,610	23
7331091005_10154122428786006	link	4,955	139	184	4,345	36
7331091005_10154121679506006	link	14,426	1,474	1,204	13,797	162
7331091005_10154121610466006	photo	3,719	156	202	3,323	31
7331091005_10154121541501006	photo	6,940	65	285	6,739	141
7331091005_10154121541286006	link	12,719	338	897	12,035	258
7331091005_10154121438506006	link	4,520	346	517	4,138	42

FB data preview

The `csvcut` command can help us to cut a given set of columns (e.g., `1,4,7-11`). Note that we have not previewed the column numbers `2` and `3` (`status_message`, `link_name`), which are wider columns and wouldn't fit properly into our preview-screen above!

Do you want to know more?

[📎 'csvcut' man page](#) [📄](#) [➡](#)