Pipe delimited format

Ben Klemens

24 July 2017

The norm in tabular text data is *comma-separated values* (CSV), where each row in a table is on a single line, and each value is delimited by commas.

This is not a good idea. Use pipes.

Not commas Commas appear everywhere. In some locales, the comma is the normal decimal separator: *3,1415* and not *3.1415*. So this is already US-centric.

You may know some people with commas in their names, like *Pradeep Smith, Jr.* No problem: we'll protect that with something like *'Pradeep Smith, Jr'*. Now that we have this accommodation, what are we going to do about the street honorifically named after a DC go-go legend, *Li'l Benny Way, NW*? Now we have to provide an accommodation for the accommodation, like *'Liĺ Benny Way, NW'*, and now that that's in place, another policy on accommodating backslashes in the input data.

Will Excel, STATA, et al., read this correctly? Maybe. Should backslashes be treated differently inside a pair of ' ' and outside? Is text allowed to be free of " " constraints? There are a lot of reasonable answers to such questions, each incompatible with the next.

Pipes Nobody on Earth uses pipes in names or numbers. In the U.S., where the government won't even legally grant *María* as a name, pipes aren't coming any time soon.

I recently cleaned up my mp3 collection (prepping for a playlist-making toy[1] I wrote) and found basically every Unicode character represented somewhere in the names—mp3 file naming is still the metaphorical Wild West. Any combination of ' 's and " "s broke on somthing. There was even one single album in the pantheon of music that had filenames with pipes [Kendrick Lamar's *untitled unmastered.*, which, as you can tell by the period in the album name, was going out of its way to be typographically complicated].

So we still need some plan for the day when our data has a pipe in it. This works:

- Read a field until you hit a pipe, then start the next field.
- A newline ends the observation.
- If the pipe or newline is preceded by a backslash, then treat it as part of the input data and not a delimiter.

[1] `https://github.com/b-k/ludwig_van_shuffle`

OK, you're done. You don't even need a backslash in front of backslashes: if the next character is anything but a pipe or a newline, it must not be special. For most data sets, the third step doesn't even come into play.

Also, pipes look nice: with consistent-width data, you see neat columns, which is why they're a nicer delimiter than other infrequently-used charactes like or, I dunno, }.

So the pipe-delimited format (PDF?) makes a lot more sense than the comma, and in the social context of dealing with disparate programs written by disparate authors, is more likely to work everywhere.

[You could implement a similar algorithm with commas: let commas be the delimiter, when the input text has commas put a backslash in front of them, and don't make any other changes to the input data. This is not the norm in CSV files; we can only speculate why.]

Social The big problem is that the default in this world is still commas. If you click *open* in Microsoft Excel, you'll see your file read as comma-delimited UTF-16, which is a correct assumption, to a first approximation, never. [UTF-8 update: it's now 89.5% of the web[2]; UTF-16 is under 0.1%.] With a minute of clicking on dialogue boxes every time you open a text file, you'll eventually get where you want to go—with slightly less frequent clicking if you change the system-wide list delimiter in the Windows control panel(!). Other systems also impose sometimes significant work on people who want to use anything but a comma as a delimiter.

If you're a data producer, think pipes. The very rare incidence of pipes in human-language data or numbers makes almost every text-recording problem evaporate. If you're the author of things that read data, please bear in mind that nobody really wants commas to be the delimiter—they've just always been there—and it should be easy for data users to specify the rules that make sense for their data.

PS: As an experiment, I'm outsourcing commenting to Twitter. Here's the tweet announcing this blog entry; click the date stamp to see the thread and leave comments and replies. If you don't have a Twitter account and post a reply anywhwere else on the Web, please notify me and I'll tweet about your post.

[2] `https://w3techs.com/technologies/overview/character_encoding/all/`