

## Readin and Writin CSV files

[Cross-posted from ICanHazDataScience]

Google should add Lolcatz to translate.google.com...

But seriously, if you want to do data science, you're going to need data. Which means being able to access data: data in streams (like Twitter), data online (like websites) and data in files. We'll start with the files.

Development agencies, bless them, have a major love affair going on with Excel files. Open any [development data](#) site (or crawl it for datafiles) and you'll see overwhelming numbers of reports (we'll get to those later), .xls and .csv files.

Let's start with the easier filetype: CSV. CSV ([Comma-Separated Values](#)) is as it sounds: a file containing readable text, usually separated with commas, but sometimes instead with semicolons ";" tabs, colons ":", pipes "|" or anything else that the person writing the file thought was appropriate ([data.un.org](#), for example, gives you the choice of comma, semicolon or pipe). Here's an example, seen from Microsoft Excel:

CSV file as seen in Microsoft Excel

If you open this file in a text editor (I'm using Notepad++ here - which is heartily recommended for looking at webpage files too), it looks like this:

CSV file as seen in Notepad++

There are two things to note here: first, that the columns in the Excel view correspond to the commas in the Notepad++ one. And second, that the comma in Excel (field C3) doesn't create a new column in the CSV file because Excel wraps the whole field in punctuation marks ("This is a comment, that includes a comma.").

Yes, it's that simple a file format. And there's a friendly Python library to make reading and writing CSV files even easier. It's called "[CSV](#)". Here are some simple bits of code that use it to read a CSV file, print out its contents, and write them out to another CSV file.