Writeup

Jacob Martin and Toshi Piazza worked on this together having never worked together before.

All our files for viewing are the html pages in the folder streamgraph. If you have any issues viewing them, open the site in firefox instead, or view the html files in a small python webserver.

Our dataset was the rails and django repositories. We thought it would be interesting to compare the number of lines changed for different filetypes between the two major web frameworks over time.

After trying our hand at pygit and realizing how complex our script would have to be, we quickly grabbed the statistics of commits grouped by months with a simple rewritten bash script. https://gist.github.com/rasmuslp/4598869

From there, we parsed the output with a python script grouping all the files changed by their extension, and outputting to a compatible csv.

We saw multiple issues with the parser initially. Certain edge cases broke it. For example, there can be changed files with 0 lines changed. This would remove the normal +++— you would see therefore breaking the parsing.

Also when we opened the csv in the default xubuntu csv reader, the file was too large to load the whole csv in and caused multiple issues including false positives of bad parsing.

With this we thought our code would work, but the csv had to first be organized by date and then by file extension. Also, in order to get the data to play nicely with d3.js, we had to manually delete some entries that had few to no commits (by percent) and also insert some entries for filetypes that were very substantial in terms of commit percentage by month, but had incomplete entries (we put 0.000% for these entries for that time period). Finally, even though this all checked out in the end, d3 still likes the data to be continuous by day, and so throws exceptions (silently) whenever a mouseover event occurs on an "in between" day (not the tenth day of that month).

Finally we got it working and outputted our stream graphs. Just as you would expect, the main filetype for the two frameworks were the 2 main filetypes for the underlying languages. PY and RB.

Both repositories saw their lines changed per month decrease over time but django's was much more obvious. We had 2 interesting findings. The .po extension for django dropped off considerably after the initial development period and Rails had a very weird balloon expansion during a 4 month period before going back to it's normal downward trend. We hypothesized this might be a release of a new rails version, or the aftermath of that release. Afterwards we found that Rails 4.2.4 was released August 24, 2015. The amount of .md files was also surprising

In looking at the stream graphs versus the stacked bar chart, both conveyed the same information, but the streamgraph helped convey the proportionality of individual file types much better (and it was interactive), but the bar chart had a better legend (in terms of files listed along the right side). Specifically, the trends we see mirror each other exactly, though the spikes seem more drastic in the bar graph even though it is harder to see the demarcation between individual filetypes. Of course, this means that although the trends show more clearly in the bar graph, and better numerical statistics can be taken from it, at first glance it might be more misleading to the user (the sequential colorscheme doesn't help either... Excel would be much happier with a diverging colorscheme). Meanwhile, it was extremely easy to set up the excel stacked bar graph, whereas d3.js expected too much of the data and its contents that it was almost unusable.