**Repo Turnover**

**Two ways to look at this:**

1 - Average duration of repos which are fully tracked on timeline (i.e. created after its start date), can also look at repos which were created before timeline but show up (although we’re dealing with a partial history in that case).

2- History of activity for repos which pre-date timeline and which show up in a few rows having minWatchers greater than 0 or some specified cut-off. 2012-03-11 06:36:14 is the first day of timeline.

**1-** Looking at repo duration for the bigquery data-set. Only looking at repos which have a creation date after the start of timeline (2012-03-11 06:36:14) – this excludes 376,944 repos which were created before timeline data begins.

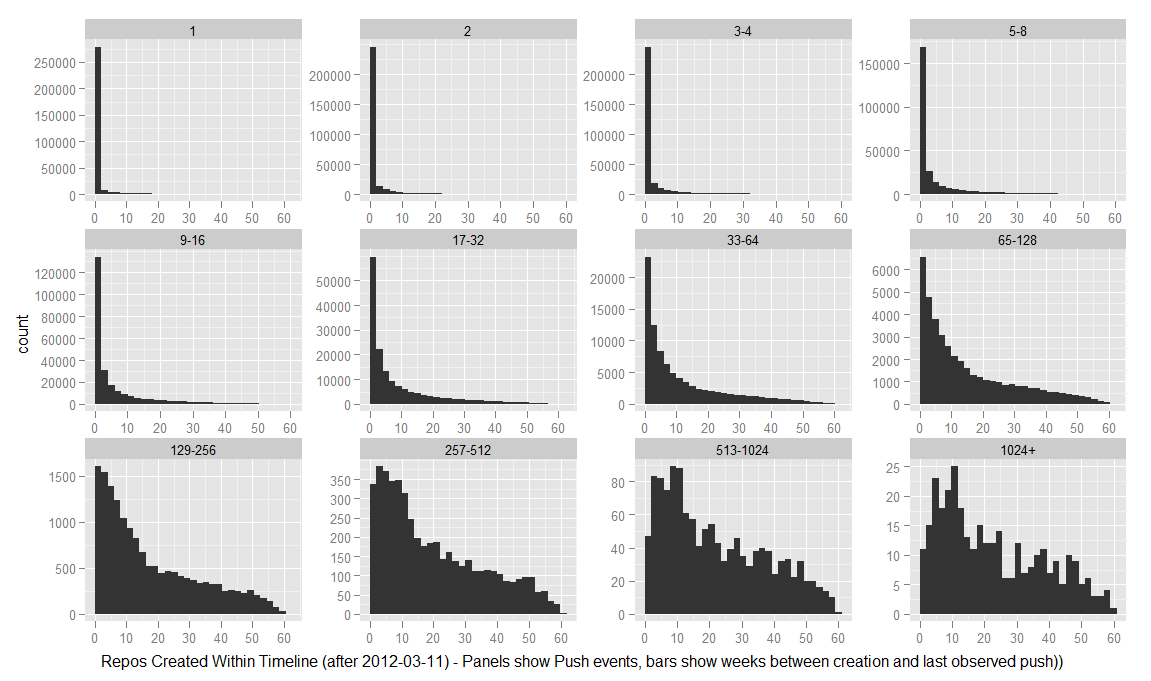
One issue here is that the data is ongoing so first need to pick a criterion which will define when a repo is no longer active. The data-set I'm using stops on 7th June 2013, any repo which has no no pushes after 1st May 2013 will be deemed inactive. There are 2874449 repos in total and 2368819 are no longer active by this criterion.

Binned repos by number of pushevents and looked at median duration in weeks (low and high relate to push events and are the bounds of a bin)

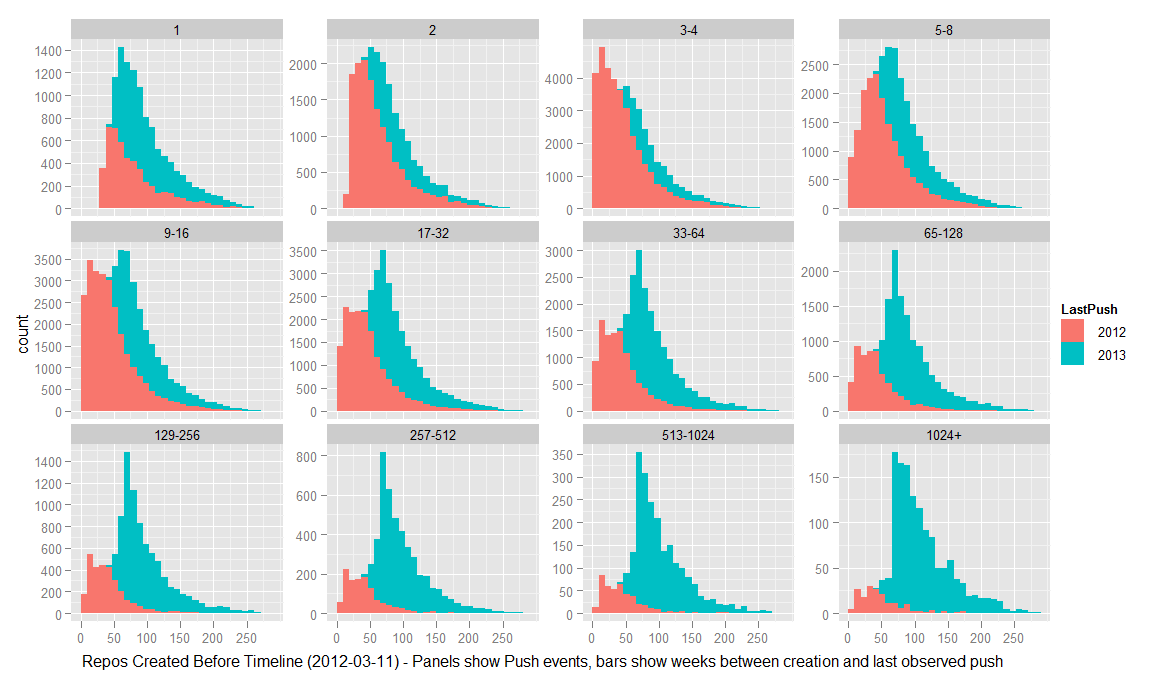


Most repos are short-lived – repos with less than 9 push events tend to be active for less than one week. The graph below shows the repos which were created within timeline span and had no pushing activity in the last 5 weeks of data.

As the graph shows, there are a lot of short-lived repos at all activity levels.



We can also look at the repos which were created before timeline data began (2012-03-11) and which show up in the timeline data. We don't have a full history of push events for these repos so I'm going to be more relaxed with what counts as “active” and go with 1st January 2013 as the date – this means repos with any pushes in the last 6 months of data will be counted as active. There are 135527 repos created pre-timeline with a last push in 2012 and 106512 with a last push in 2013.



**2-** I’m still collecting data on the repos which showed up with 1 event in the census and which had a minimum of 5 watchers (indicating previous activity). There are a lot of repos in this set with no owner name (I mentioned them in the census) and I think they’ve been deleted or it’s a bug with github timeline, anyway I’m excluding these for now.

So far I have data on 4675 ‘single-event’ repos, I’m just counting the number of pre and post timeline commits and recording the dates of the first and last. There are 621 repos which had more than 100 commits before timeline began and have none since then, and 2519 with 10 or more pre-timeline commits and none after.

Strangely, there are 341 repos in the database which have more than one post-timeline commit. Either these have been ‘resurrected’ after the census data-set was isolated or there’s a way for a repo to have commits that don’t result in PushEvents on timeline.

This is really the beginning of investigation into old repos through the API (its fairly slow to gather the data this way) but there’s a good deal more I can do along these lines if we’re sufficiently interested.

**Summary of work on base/head relationships by user**

One of the primary goals here was to look for users who pull-requested themselves unto a project – users who first interact with a base repo through pull requests and then go on to become members of the base repo and make pushes.

Started with 200k pairs of base/head repos that had at least 4 pull request events between them.

Got a list of all users who pushed to each base and head repo, and a list of all pull requests by user and base/head repo. Augmented this second table with variables about pushes to base and head repos.

The 'pull request users' table has 196,437 rows – representing 99,274 distinct users (some were involved in pull requests for multiple base/head pairs).

There are 25,678 cases where the user has at least one push on the base and head repo – but its not clear that all of these users became base contributors AFTER they had made a pull request.

By the strictest criteria I can think of (sequence where first head push precedes first pull request precedes being added as a contributor to base) there are 3,416 users who appear to have received contributor rights as a result of making pull requests.

However, the limiting factor here is being added as a member of the base repo... these events are sparse in the data-set (only 5,538 users have a recorded add memberevent, whereas 25,678 users have at least one push to head and base repo – and therefore must have had contributor rights to the base repo). I'm either missing some events where users were added as contributors, or these events occurred before the timeline data began.

So, if we relax the criteria a bit (a sequence where first head push precedes first pull request precedes first base push) there are 14,348 users who meet it. These users made a median of 18 pushes to the head repo and 8 to the base repo, but the variance on this is huge (its roughly power law).

Once these users had the ability to make pushes on the base repo they no longer needed to make pull requests, technically at least. 5,244 stopped making pull requests once they had made their first push to the base repo, for the remainder there is an overlap.

The major takeaway message from all this is that the situation is messier than expected regarding pull requests. It seems fairly common for people who have contributor rights on a repo (and have exercised these to make pushes directly) to also make pull requests (when they could presumably have achieved the same thing by pushing their changes directly).

Some possibilities: 1) they use pushes for a certain type of change and pull requests for other types 2) there are other contributors to the head repo, and so the pull requests represent a kind of management (external workers change the head repo, the head repo's owner is also a base repo contributor and manages the merging of these changes through pull requests).

**Notes on base/head relationships between repos themselves**

Based on the table of 200k base/head pairs having at least 4 PR events.

There are around 26k base repos which have a 'relationship' with more than one head repo – some have relationships with many heads (e.g. mxcl/homebrew has 1345 different heads, rails/rails has 691).

There are also around 4k head repos which have a 'relationship' with more than one base repo, a fair number of these are related to 'liferay-portal', not sure how many exactly.

There are at least a few 'networks' of repos, where a group of smaller repos share pull requests and are also related to a large repo through pull requests.

It also seems that pull requests can 'flow' in more than one direction (reciprocal pull requests between base/head repos where each acts as both base and head) – have yet to put a number on this.

Liferay-portal repos seem to be exactly this kind of network. There are 877 'pairs' of repos involving 195 distinct repos, with 98 different repos serving as a base and 190 unique head repos.

I've done some inspection of the 'liferay-portal' repos and they're actually fairly interesting. There are 28 repos which served as the base in at least 10 pull request pairings (receiving a total of 49k pull requests between them!). The repo which was presumably the 'original' one (liferay/liferay-portal) isn't actually the 'hub' for pull requests – brianchandotcom/liferay-portal was the base in 68 pairs and received 24k pull requests, it was also the head in 2 pairs and made 10 pull requests in this fashion... I was expecting that one of these pairs would involve liferay/liferay-portal (and that brianchandotcom would turn out to be a working repository with liferay being reserved for stable release versions or somesuch) but there's actually no direct link between liferay and brianchandotcom through pull requests. The liferay/liferay-portal repository itself was the base in 46 pairings, receiving a total of 456 pull requests.

I'm kind of suspicious of the very high levels of pull requesting going on between all these liferay repos... but the project itself seems genuine, and if all the activity is also genuine then this would make for a very interesting (and somewhat unique) example. At this stage I could fairly easily find out anything we'd want to know about the liferay-portal 'network', between the pull request data-sets and the census stuff I've already got the data to answer any number of questions about it. I also feel like I've got a good enough command of bigquery and the github API to get and use just about any kind of data that's available to us for this kind of analysis, so that's some kind of progress I guess!