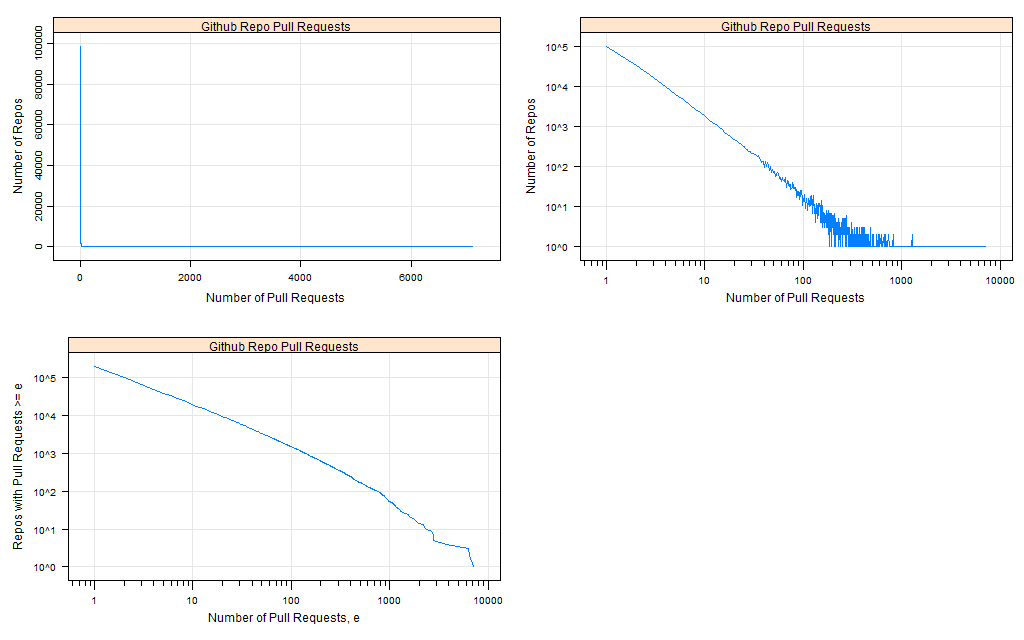
**Pull requests by repo.**

A total of 195509 repos have at least one pull request event in the timeline data.

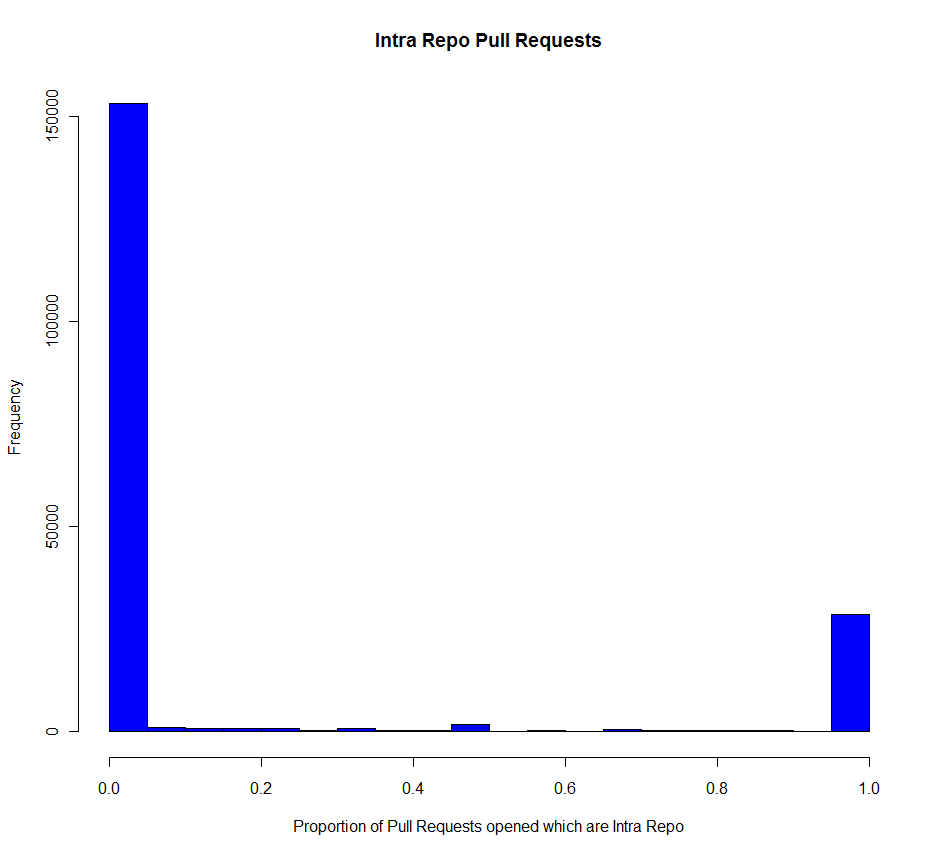
The distribution of pull requests between repos is highly skewed again.



This data is saying 1257009 distinct pull requests in total. 1005607 is 80%

Repos with 5 or more pull requests (20%) account for 80% of all pull requests.

**‘Internal use’**



The first type of pull requests I'm looking at are 'Internal Use' (I.e. Intra-repo). This graph shows that in the full data-set repos tend to either never or always have intra-repo pull requests. In total there are 28351 repos which ONLY have intra-repo pull requests, so 14.5% of all the repos which have at least one pull request. However, most of these (12656 or 44%) are repos which only have a single pull request – so these are probably better interpreted as people testing the feature rather than adopting pull requests as a form of code review.

For now, I'm going to go with 3 classifications related to IntraRepo Pull-requests.

1-1: Single Pull Request which is intra-repo: 12,656 repos

1-2: Repo has more than 1 pull request and 95% or greater pull requests are intra-repo (purity of practice): 15,916 repos

*Example:* [*https://api.github.com/repos/dotCMS/dotCMS*](https://api.github.com/repos/dotCMS/dotCMS) *- has 1758 PullRequest OpenEvents, of which 1742 are intra-repo and 1604 are merged.*

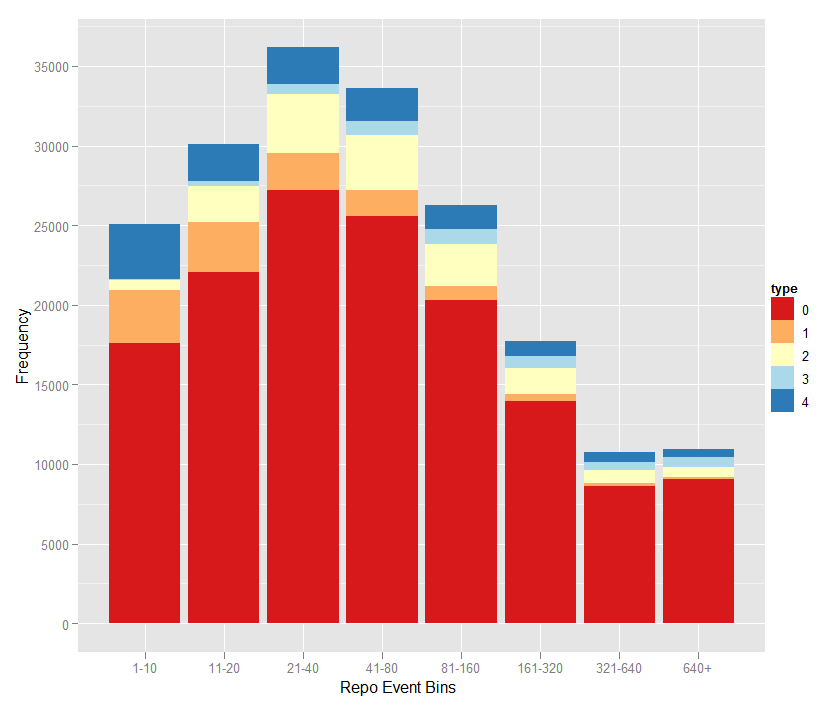
1-3: Repo has more than 100 Intra-Repo pull requests OR a proportion which is 50% or greater (mixed practice) – 4683 repos

*Example:* [*https://api.github.com/repos/mozilla/browserid*](https://api.github.com/repos/mozilla/browserid) *- has 2768 PullRequest OpenEvents, of which 870 are intra-repo.* We spoke about ‘purity of practice’ here, but in a sense this is out of the repo’s own hands – type 2 and 3 repos both use IntraRepo Pull Requests, the difference is whether they also receive external pull requests, which they cannot control.

There are also pull requests which were merged by the same user who instigated them, so this would be another form of 'internal use' of pull requests. As you might imagine there's a lot of overlap between 'intra-repo' pull requests and those merged by the user who created them. I'm adding a fourth type which is secondary to those concerning intra-repo pull requests.

1-4: Repo has 50% or greater pull requests which were merged by the same user who instigated them. In total there are 30619 repos which meet this criterion – 14235 are classified as type 4 (the remainder meet one of the intra-repo criteria)

*Example:* [*https://api.github.com/repos/angular/angular.js*](https://api.github.com/repos/angular/angular.js) *- this one is interesting because it seems to receive a lot of pull requests from external repos but the only merges are ‘self’ merges. 937 Distinct Pull Requests from 433 Distinct Head Repos – 169 are merged, and 163 of these originated with the same user who accepted the merge (but are not intra-repo).*



I’ve matched each of the repos up to the census data-set to see if there’s any relationship between repo size or activity level and the ‘type’ of pull request usage. As you can see there’s not too much of a pattern – Repos which have 95%+ intra-repo pull requests (type 2), which have 50%+ or 100+ intra-repo pull requests (type 3), and which have 50%+ self-merge (type 4) are all represented at every activity level.

**‘Opt-out’ or ‘offline merge’**

These are two types of repo which we discussed – they either ignore/reject pull requests or they merge them manually offline when accepted. I haven’t done any SHA matching so for now these two cases would look the same in the data.

What we can look at now is the number of pull requests which are merged or closed. Again I’ve come up with four classifications, and again I’ve opted to treat the single pull request repos separately.

Type 2-1: Repos with a single pull request which has been merged (These are not relevant to the concept of opting out but its useful to set them aside at this stage): 44,258 repos

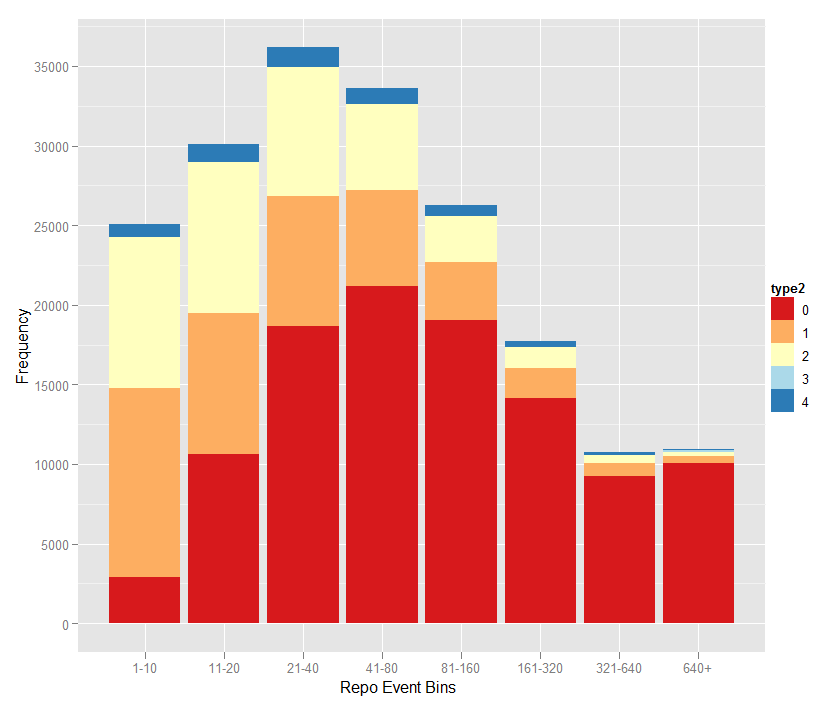
Type 2-2: Repos with a single pull request which has not been merged: 38,970

Type 2-3: Repos with more than one Pull Request and less than 10% of pull requests are merged: 7,709

*Example:* [*https://api.github.com/repos/mxcl/homebrew*](https://api.github.com/repos/mxcl/homebrew) *- 6358 Distinct pull Requests from 2484 Distinct Head Repos – of which all are closed but only 3 resulted in merges.*

Type 2-4: Repos with more than one Pull Request and less than 10% of pull requests are closed (takes precedence over type 3): 5,674

*Example:* [*https://api.github.com/repos/Hexlet/osx-project-2*](https://api.github.com/repos/Hexlet/osx-project-2) *- 136 Distinct pull Requests from 133 Distinct Head Repos, only 3 are closed.*



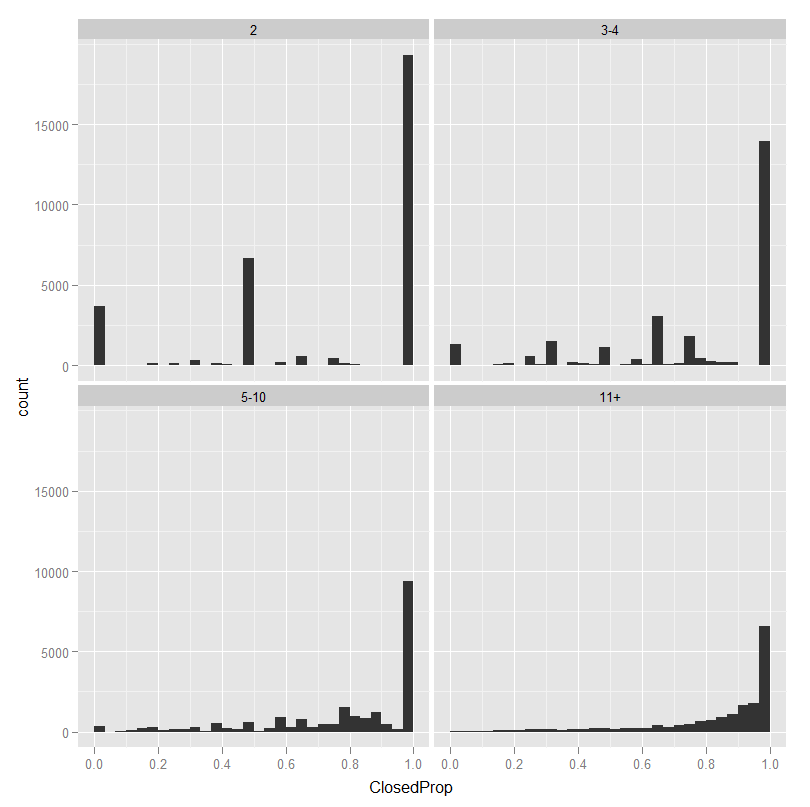
Here there’s more of a trend whereby more active repos are less likely to ‘opt out’ of pull requests.

In general I’m fairly disappointed by the results with this approach of classifying repos by their use of pull requests. It took quite a long time to come up with and implement these different types and after all that we’re still in the dark about a lot of repos. So, I’m going to see what I can come up with to develop a more general understanding of how pull requests are used on github. This will involve quite a few graphs, and I’m going to paste a lot of them here because in combination they should tell us something about the data.

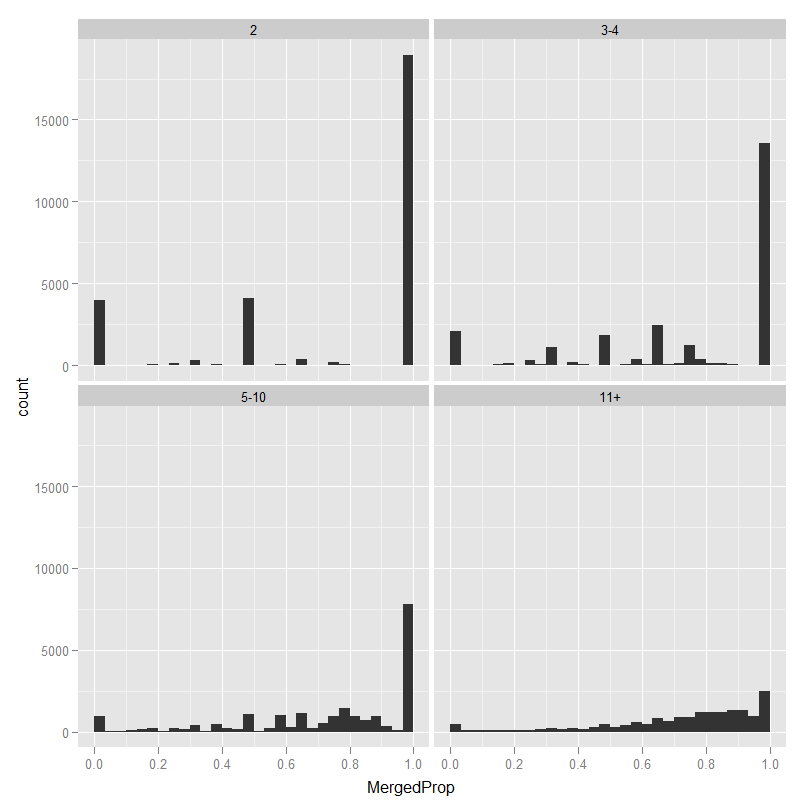
**All repos which have more than one pull request.**

I’m excluding repos with only one pull request because for those the types above actually do tell us prettymuch all there is to know. I’m also excluding repos for which I didn’t have a match with the census data-set (4,893). So, from here on I’m dealing with a data-set containing 96,604 repos with more than one pull request each.

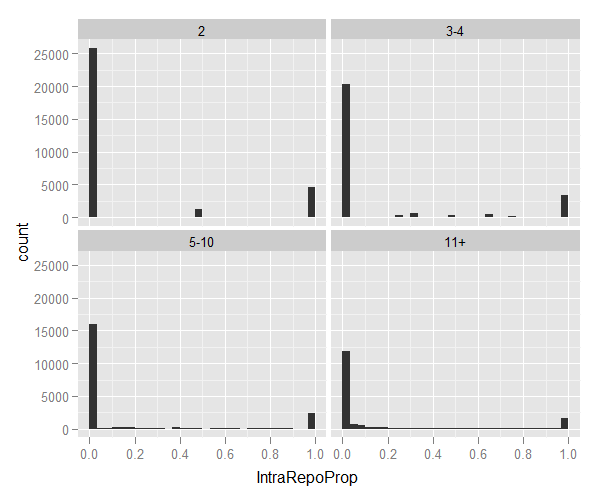
Firstly, let’s look at the repos based on their number of distinct pull requests, I’ve split them into 4 bins. Here are graphs showing the proportion of pull requests which are: closed, merged, Intra-repo and Self-merged respectively. The panels show repos by number of pull requests.



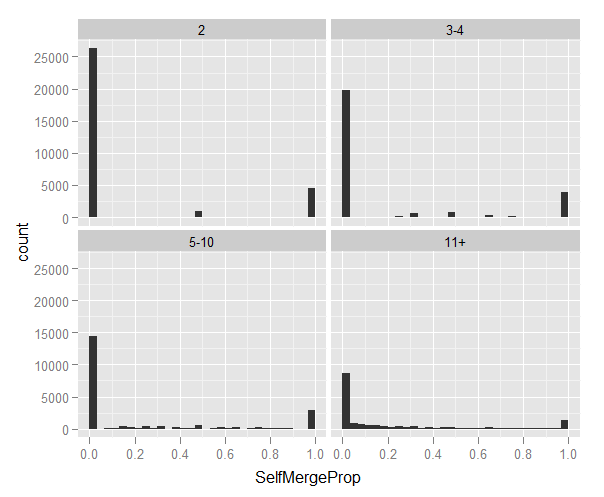
Its common for repos to deal with all or almost all of the pull requests they receive – for repos with at least 5 pull requests it is rare for them to have a low proportion of closes.



Repos which receive a lot of pull requests are less likely to merge them *all* but still tend to merge a high proportion.



Proportion of Pull Request open events which are intra-repo - tends to be all-or-nothing regardless of the number of pull requests.

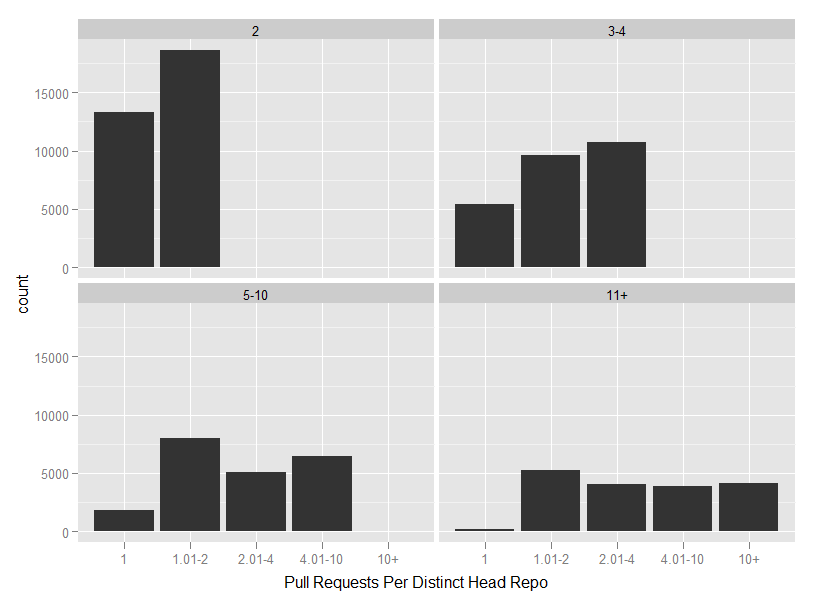


Proportion of merges which are 'self' merges (the same person who instigated the pull request merged it).

**Repeated use of the same head repo**

I noticed in the data that the same head repo can appear in multiple pull requests for a given base repo. This is interesting because it suggests an ongoing relationship between the base and head repo. This kind ofrelationship could be the same individual using multiple repos, or could represent sustained interaction from an individual external to the repo – it would be possible to determine which with a bit of digging on bigquery.

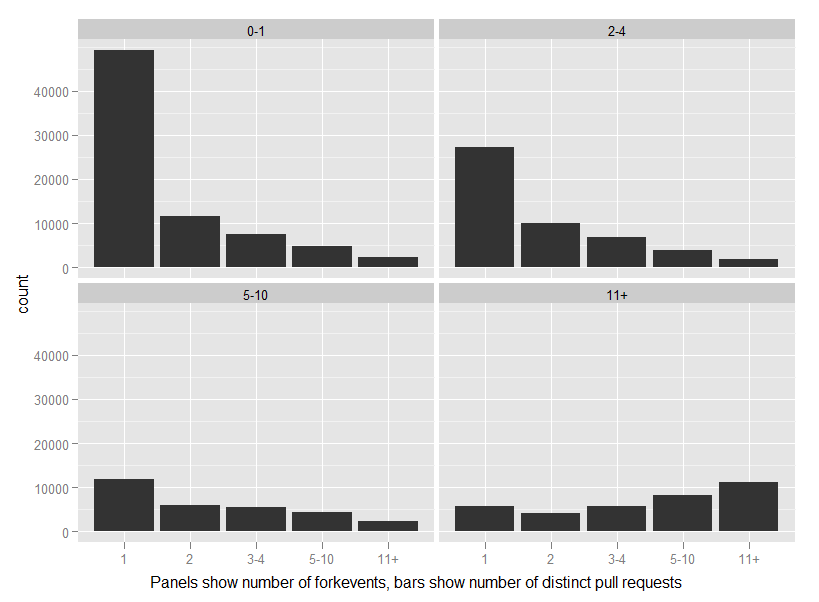
To get a handle on this I’ve created a Pull Requests per Distinct Head Repo variable by dividing the number of distinct pull requests by the number of distinct head repos. This variable is highly skewed (by repos which received a lot of pull requests from a single head repo) so I’ve converted it into a 5-level ordinal variable. The graph below shows that its very common for repos to receive more than one pull request from the same head repo. The panels show repos with a certain number of pullrequests



**Relationship between Fork Events and Pull Requests**

One of the things we might want to do is to consider how often a repo is forked as compared to the number of pull requests it receives. With the data-set I currently have its possible to do some very rough exploration of this. I’m using the number of distinct pull requests and the number of fork events (from the census data-set) – alarm bells start ringing immediately because there are some repos that have pull requests (which are not intra-repo) but no fork events. One possibility is that the relevant forks were created before timeline begins and the pull requests were made afterwards, but I suspect there are other possibilities for this discontinuity as well.

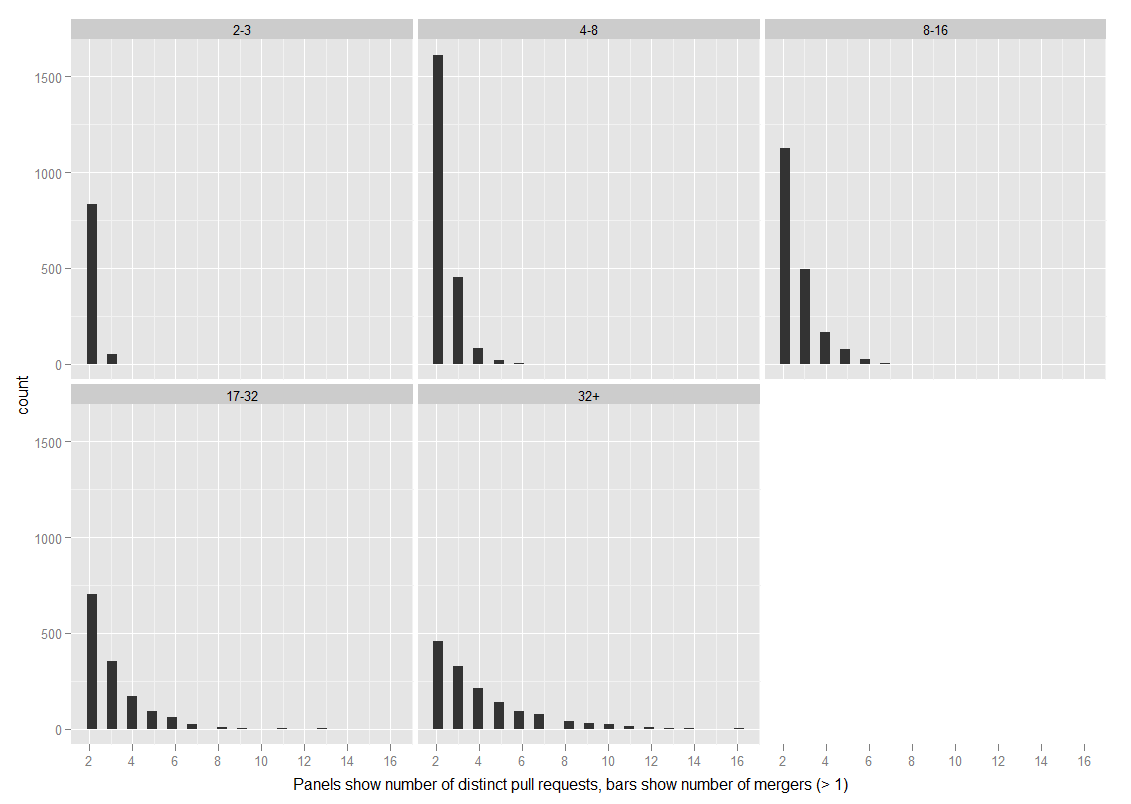
Setting this aside for now, the following graph (I’ve included the previously excluded repos with a single pull request) suggests that there is a relationship between number of fork events and pull requests… but that it is also possible for a repo that has very few fork events to have a lot of pull requests and vice versa. Of course there are also going to be a lot of repos that have fork events but no pull requests, but these won’t feature in the current data-set.



**Number of users who accept merges**

The dominant pattern is for only a single individual to accept merges for a repo – in the full data-set 135,316 repos (69%) - if we only count repos with more than one distinct pull request and at least one merge then its 66,967 (still around 69%).

The below graph has histograms of all the repos with at least 2 different users who merged pull requests – the single-merger repos would obscure everything else.



The following table accounts for all but 200 repos.

|  |  |
| --- | --- |
| Number of Mergers | Repos |
| 0 | 56768 |
| 1 | 121670 |
| 2 | 10932 |
| 3 | 3109 |
| 4 | 1331 |
| 5 | 647 |
| 6 | 357 |
| 7 | 209 |
| 8 | 134 |
| 9 | 87 |
| 10 | 65 |