**Typology of repos**

**What we’ve done**: counts of event types for each repo and a few other variables – number of distinct pushers and that sort of thing. Also latent class analysis which classifies repos based on these event counts.

**What to do**: I think we should work more on this, there are a number of types of repo which I think it would be useful to classify and which can be classified using variables I’ve already created or know how to create. We could then look at the types of repo which individuals or organisations tend to “own” – and more broadly at whether organisations tend to own different types of repo to those which don’t have an organisation.

This kind of repo typology is what I’ve been working on this week and I’ll attach a separate file with details of that.

**Organisations**

**What we’ve done:** Built the organisations\_megatale\_ultimate – which brings together counts of events which happened on the organisation’s repos, information on the number of contributors, lifespan of the organisation’s activity, variables representing whether the org’s repos were linked to by stackoverflow posts, information on the first 20 events for the org (their type and how long it took for the org to accrue 20 events).

Also, the manual coding of 5/600 organisations into the 4-level typology – and my tags representing the organisation’s “domain”.

**Work to do:** I think the next step was to feed the orgs megatable into machine learning and use the manual classifications as a training set – to see if it can correctly classify other orgs based on the variables in the megatable.

**Organisational structure of github**

We’ve looked at this in terms of the repos which organisations fork. I think it’s also important to ask whether there’s a difference in the “behaviour” or profile of organisations as opposed to simple repository owners who haven’t made an organisation.

I think we could do more on the links between repos themselves by 1) looking at the network of users based on whether they have pushed to the same repo(s), and 2) combining this with information on pull request relationships between repos.

**The “domain” of repositories**

**What we’ve done:** Topic modelling using repo descriptions and readme files.

We could maybe re-visit this and see if the tags I added to organisations could be used as a training set. However, I think that the documents themselves are just not really suited to this task.

We could also look into going through a 3rd party site which classifies websites – feeding in the urls of the organisations or which are attached to individual repositories.**Github in the news**

**To do:** Look at the level of news coverage featuring github, particularly in the lead-up to its IPO and particularly stories about its diversity and utility in a broad range of contexts.

**Comparison of github and sourceforge (or other open source coding sites)**

Matt is working on a comparison of github and sourceforge. We may also be able to look at some sourceforge data and explore the differences that way.

**Pull requests**

What we’ve done: Looking at the pull request relationships between repos – whether these are short or long term and whether a user can pull request their way into a project - become a contributor by virtue of having demonstrated their usefulness through pull requests.

Also looked briefly at pull request “networks”.

**Significant Events**

This is something we haven’t really explored much. I’ve written that script which shows, for a specified repo, events on github and mentions on Stackoverflow on a weekly basis – we could probably do more with the data for that. One aspect which could be improved is matching posts on stackoverflow more broadly – at the moment I’m only registering a match when the first URL in a post links to the repo on github. We could also use tags, and maybe even the name of the repo.

I also think there’s something to be gained by looking at repo events over time. So, for example, we could look at repos created within the timeline data’s span and at whether these went on to be “successful” (I’d take number of forks or watchers as the success criterion) – then analyse the first 100 events for the repo, or all the events in its first 3 months, to see whether there’s a difference which relates to the ultimate success or failure of the repo.

**Power Laws**

Most of what I’ve written above can apply here too… in this case we’d be interested in the differences between repos which exist in the low/medium/high regions of the events distribution.

**Repo Turnover**

I’ve looked at the lifespan of repos and specifically at repos created before the timeline data begins to see how many were inactive by this point.

**Repo Naming Conventions**

I’ve made a table showing the most popular duplicate names which aren’t due to forking. Most of the popular names indicate that they were created by people as part of learning to use github – and could probably be traced back to a specific tutorial which instructs the learner to create a new repository.

**Non-software repos**

I’ve done some work on this by looking at repos with no language – where these have lots of push events they tend to be pushed to by a bot. They hold things like log files or copies of a daily newspaper’s front page, some of them track or mirror the activity occurring in another repo.

I think if we want to pursue this further we should do so by finding non-software repos ourselves and then seeing what these look like on github.