“Social News” web-sites with democratic interfaces - Analysis of one month’s voting from Reddit.com.

Richard Mills

Submitted for the degree of Mres in Applied Social Statistics

September 2009

Lancaster University Postgraduate Statistics Centre

Abstract

This study aims to develop an understanding of the activity taking place on “Social News” websites which use quantitative democratic interfaces. The data analysed were 3,446,522 votes from one such site (Reddit.com) for a single month. Data were analysed in terms of Users, Links, and Sub-Reddits. Exploratory analyses revealed that exponential distributions dominate many facets of activity on the site – conforming to the Power-Law observed by previous research. Users, Links and Sub-Reddits were Partitioned around Medoids to determine if there were different “types” of each. Clusters fitted to Users data suggest that Users tend to take on different roles in the community by prioritising a certain kind of activity (i.e. voting or submitting). A concept of “community involvement” was found useful in describing the different types of User on the site. Furthermore, Row-Column association models suggested that the users who were the most active and involved were the most likely to submit popular content. Latent trajectory analysis was also employed to look for patterns to the temporal distribution of votes on Links. The potential for quantitative democratic interfaces to facilitate communication between large groups of people is discussed; and some proposals for how these systems might be further studied are put forward.

Acknowledgements

I would like to thank Professor Brian Francis for all of the help and supervision he provided throughout this project.

I would also like to thank Christopher Slowe for arranging access to Reddit.com's database – and of course the users of Reddit.com, for populating this database in the first place.

Contents

1 Introduction

1.1 The Internet and social change.

There is a wide variety of literature available about the internet and its potential to change aspects of society. From a sociological perspective a lot of the literature is theoretical in nature (e.g. Hansen, Berente & Lyytinen, 2009); in particular there is a sub-set of literature dealing with how the internet may affect the political process (e.g. Agre, 2002). Some authors have theorized that the internet could involve citizens so much more directly in democratic systems that their nature would change qualitatively (e.g. Grossman, 1995).

There is also a (largely separate) body of literature dealing with empirical research on peoples' participation in online social activities; much of this from a computer science/communications perspective. Most of this empirical research deals with older forms of web communication, like e-mail (e.g. Butler, 2001), Web-pages (Albert, Jeong & Barbarasi, 2004), and Usenet discussion/news groups (Himelboim, 2008). One of the most often-reported findings of these studies is the utility of a Power-Law distribution in describing the data (e.g. Adamic, 2000; Raban & Rabin, 2007). The Power-law distribution describes a trend whereby the majority of content is being produced by a minority of individuals. Another noteworthy concept within this literature (with particular relevance here) is that of Information Overload (e.g. Nye, 2002; Jones, Ravid & Rafaeli, 2002). The essence of this concept is that as the amount of content in a system increases it becomes harder for Users to access the content most suited to them. There are advantages to a system holding a lot of information from many Users - the more information a system holds the more likely it is to contain the information a given user requires; but when the quantity of information is sufficiently large, allocation of Users' attention becomes more important.

While functionality like Web-Pages and E-mail significantly expedited existing forms of communication (e.g. mail and publication) and had significant impact on society; we would argue that they did not themselves offer anything qualitatively new. In the last few years web technology has developed to facilitate forms of communication which do seem qualitatively different to those which existed before the internet. The term "Web 2.0" is frequently used to describe sites offering forms of communication which, at the highest level, could not function offline (see Wikipedia.org; itself an example of a Web 2.0 site).

One particular breed of these Web 2.0 sites which this research will focus on is the "social news" or "social book-marking" site. Since 2005/2006 web-sites have been emerging which are devoted to this kind of interaction, several of these now have large user-bases (e.g. Digg.com, Reddit.com, Delicious.com). Broadly speaking, these sites exist to sort and aggregate external web content; any member can submit content and also rate the content submitted by other users. These ratings are then used to rank all submitted content, and these rankings are used to determine the prominence with which said content will be displayed on the site for other users. More recently (in the last year or so), established sites like Facebook.com and Google.com have begun to integrate this kind of functionality into their existing services.

1.2 What makes these sites relevant to social scientists?

The concept of voting on the internet is not itself entirely new, for many years it has been possible for individuals to vote in on-line opinion polls. The thing which makes social news sites different in this regard is that users' votes have significance beyond the expression of opinion. When a user gives a positive vote to an item of content on these sites; this act makes it more likely that other users of the site will be exposed to this item of content (the converse is true of negative votes). When an individual first visits this site they will generally only see content which has been endorsed by the community of users on the site through its quantitative democratic interface.

We do not mean to say that the way in which users of these sites share news is itself of relevance to social scientists. Rather, the relevance of these sites lies with the quantitative democratic interface itself; and the other purposes a system like this might potentially be used for. These sites seem to offer a way around the previously noted problem of Information Overload; indeed this could be thought of as their *raison d'etre*. Allowing Users to vote on each others' content has the potential to shift some of the burden of sorting through hundreds or thousands of sources to find the most worthwhile - from the Individual to the Group. When an Individual visits the site they see immediately the recent content which the other users in the group have deemed most worthy of their attention. If quantitative democratic "Social News" systems do offer a way around Information Overload to some degree, this in itself would represent a significant development in computer-mediated communication – and consequently would warrant the re-visiting of some sociological theories regarding the Internet.

The democratic nature of these systems also raises interesting questions about the psychological effects of participation. Older forms of online communication (such as Usenet Groups and Bulletin Boards) tend to employ a hierarchy which distinguishes between Members and Moderators (also sometimes Administrators). Most Users of these resources would be classed as Members –defined by their ability to submit "posts". A minority of Users (the moderators) are explicitly given power over the submissions of other users, and charged with monitoring this content to ensure that it is acceptable, appears in the correct location, etc. Newer "Social News" sites buck this trend; every User has the power to both submit their own content and vote on the content submitted by others. The responsibility for moderation has essentially been delegated to the community at large. Social Identity Theory (Tajfael & Turner, 1979) suggests that the performance of this role at the group level will have consequences for the way Users perceive this community – and consequently what their membership and participation means to them. It is probable that the up/down voting (integral to how these sites work) will be an important factor in determining how individuals construct their shared identity as users of a given site. There is also a chance that this identity will be stronger than those associated with older forms of online communication - because members have more means whereby they can participate in the "community", and more power has been placed in the hands of this community.

The democratic interfaces behind "Social News" sites are quite a recent phenomenon; and therefore there are many unanswered questions regarding the uses they could potentially be put to. For example, could an interface like this be used to share and organise ideas or solutions to problems? Would it

work with scientific, political or economic ideas? How would the use of a system like this effect the quality of "content" which is generated in any/each of these areas? Older forms of online communication (bulletin boards, Usenet) have already been applied to most of these fields and so these answers are known to some extent. Quantitative democratic interfaces have however only been deployed in a few different contexts thus far (mostly News-related); therefore much less is known about how they operate and what they can be used for. If we wish to consider what lies ahead for the Internet and Social change – it seems pertinent to look closely at the emerging social groups for whom online democratic activity is an everyday reality.

1.3 Purpose of this research

Quantitative Democratic systems would seem to have the potential to circumvent some of the problems of Information Overload – and consequently may facilitate new forms of communication between larger groups than was previously possible. This potential; and the fact that these systems are largely unknown to the social sciences - provides impetus for the present research. This research will however not address these issues directly because they are much too broad to tackle in the available time. Rather, this piece of research is intended to lay the groundwork for a programme of primary data collection and experimentation. As such, the purpose of the present research is to investigate precisely how the behaviours of individual users combine to perform the functions they have been allocated as a group.

1.4 Why Reddit.com?

To this end, we have approached one of the aforementioned social news sites and requested data to analyse. The site which was chosen is Reddit.com. There are several characteristics which make this site particularly suitable for our purposes. Primarily: the domain of the quantitative democratic system is larger on this site than others. While other sites allow users to vote on submitted content and display this content accordingly, Reddit.com extends this system to cover comments on this content as well.

A visitor to Reddit.com will see the top 25 items of content ordered by recent user voting activity (these can actually be displayed along a variety of user-determined criteria, or viewed in terms of the "Sub-Reddit" they were submitted to). If this visitor views the comments on a given item of content, these comments are also threaded and displayed in accordance with their aggregate positive/negative votes. In fact, there are only three items of content on this site which are not subjected to user voting and ranked accordingly. These are one sponsored link (appearing in blue at the top of the list of ranked items), one advertisement (displayed to the right of these ranked items), and the page footer (which contains links to pages maintained by the Reddit administrators, and also links to affiliate sites).

The other major characteristic of Reddit.com which makes it suitable for this research is that items of content can be simple statements or questions directed at the Reddit.com community (known as **Self** links). This feature was added soon after the site's launch to facilitate the development of a Reddit.com community. This community takes quite an active role in discussing aspects of how the site does, and should operate; informal monitoring of popular content for several months suggests that some of the

more popular Self links submitted are related to how the site operates (ideas for improvements, complaints about problems, etc.).

The site's administrators take very little control of moderating the site's content; until recently there were no known instances of links being manually deleted by administrators. On 20th August 2009 a User submitted a link which exploited a hole in the Sears.com (American department store website) site's code. This exploit allowed users to link to a product and control the product categories that product would appear to belong to. Sears contacted Reddit.com's parent company and asked for the link to be removed, and Reddit's administrators duly complied (Reddit Link – "Where did my post about Sears.com's URL-hackable categories go? Am I actually being censored!?"). This prompted a lot of discussion by Reddit Users about the decision (more than half of the links on the site's front page on the day related to this), with a lot of complaints and some threats to quit using the site among the contributions.

This reaction suggests that many of the site's users feel quite strongly that the determination of prominent links should always be left to the site's community. The strength of this reaction suggests that the control of prominent content granted through the democratic interface might be an important part of what it means to be a member of the Reddit.com community. It should be noted however that the backlash from this decision was directed more at Sears than the Reddit administrators; suggesting a level of trust in the administrators that they would not take such an action unless it was unavoidable.
Reaction from Reddit's users was by no means unanimous however; for all the links complaining about Sears or Reddit's administrators; there were also some prominent links and comments defending the behaviour of both Sears and Reddit's administrators.

Casual observation also suggests that some of the moderation performed by users operates along certain conventions, some of these being quite specific to the community. Comment threads (and even some links) frequently make reference to some new or established pattern of up/down-voting related to particular content (e.g. links beginning with "Hey Reddit" receiving a disproportionate number of votes). The site's community have also developed some of their own words and phrases to describe things related specifically to using the site. For example; "Down-modding" is the term used for voting negatively on a piece of content. "Reddit Rage" is a pattern whereby one user takes offence at the down-voting of their content perceived to be instigated by a second user (usually with a comment) - this user then apparently seeks out links or comments submitted by the "offending" user and down-votes these in retribution. These characteristics all suggest quite a strong sense of community. Furthermore, this is a community which is quite transparent and accessible to "outsiders" because the users discuss its specifics frequently and openly.

Finally, the software behind Reddit.com is Open Source - which has two benefits for this research. It is possible to check specifics of how the software operates where desired, and it will be possible to create copies (or modifications) of this site's infrastructure for experimental purposes later in the project. The one exception to this transparency is the site's anti-cheating/anti-spam code, which is a closely

guarded secret. A link which reaches the Reddit.com front page has been estimated to bring about 75,000 visitors to the site being linked to; there have been several instances of individuals or groups setting up commercial services whereby they receive payment to generate traffic like this through Reddit.com. It is for this reason that the anti-cheating code exists, and the importance of its secrecy is also why we do not have access to it.

1.5 Who uses Reddit.com?

The data which we have obtained from Reddit.com only cover the "link-voting" and "link submission" behaviours of registered members for the month of March (more details about this data in subsequent sections). We have not conducted any analyses of the traffic which this site receives; but basic details about traffic at certain points in time are available elsewhere. The graph below shows "daily unique visitors" for the site between July 2008 and January 2009, and was produced by one of the site's co-founders (Ohanian & Golliher, 2009).

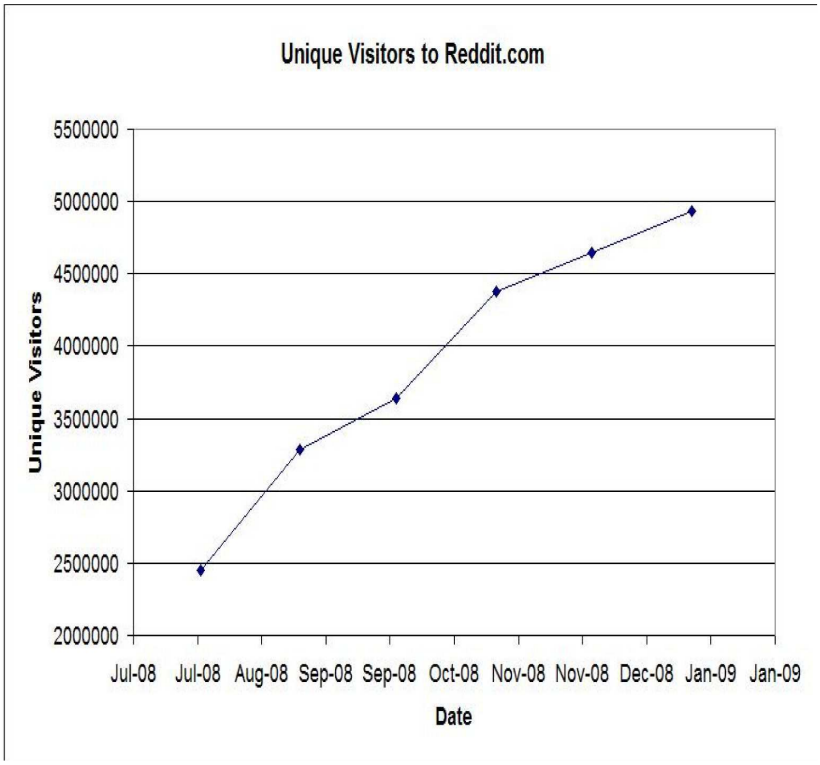


Figure 1.5-1. Showing unique visitors per day on Reddit.com. – from Ohanian & Golliher, 2009.

The above graph suggests that the site's number of "unique users" per day has doubled from 250,000 to 500,000 in the six months to January 2009. It should be noted that this measurement will record a user who visited the site every day in the time period as one unique user per day (i.e. if the site had 250,000 unique users on consecutive days these could conceivably be exactly the same 250,000 users). Private communications with our Reddit contact revealed that the site had 5,664,590 unique visitors (by IP address) in March 2009.

Reddit.com has also been included in an analysis of "social network sites" which offered demographic information on users (Chapelle, 2008). The methods used to produce the following data are however not known, so it may not be reliable. This report suggested that 80% of Reddit's users are Male and

20% Female. Over 75% of the site's users are educated to the level of Bachelor's degree or higher; and about 65% are aged between 18 and 44, with the most common age group being 35-44 (30% of users). Traffic reports from Alexa.com suggest that an average user spends 5-6 minutes on the site per day, visiting between 5-6 links on the site. Alexa.com reports for the last year also show a large surge in Reddit's "global reach" in January 2009, from less than 5% to 10% in one month - presently (Alexa.com - September 8th 2009) between 15% and 20% of global internet users visit Reddit.com.

1.6 Research Questions to be addressed with the Reddit.com data

The purpose of analysing data from Reddit.com is to gain an understanding of how the site's democratic interface is actually being used by its members to regulate the display of content on the site. This broad interest has been broken down into a series of questions which will be addressed here with this data; there follows a brief description of each question and the methods which will be used to address it.

Q1: Can users of Reddit be broken down into different 'types' of user based on the ways in which they used their account in March? Every user account on this site is functionally identical in that each user has the capacity to submit/vote/comment with equal weighting. Given this equality of account status, it will be interesting to see whether users take on different roles in the running of the site by prioritising a certain kind of activity. To address this question; users will be clustered based on the frequency and nature of their behaviours on the site in March (e.g. number of votes, number of link submissions, proportion of positive/negative votes, sub-categories in which behaviours occurred, timing of behaviours, etc.).

Q2: Can submitted links be classified as belonging to different 'types' based on the kind of voting behaviour they elicit from users? Here we are primarily interested in picking out links which are popular/unpopular/controversial. Presumably, each link's individual qualities will be largely responsible for determining the frequency and nature of voting activity it receives from users. Unfortunately, we do not have access to qualitative information about these links; aside from the category they were submitted to and whether they are "self" links. As such, links will be classified according to the kind of voting activity they received and any available indicator variables.

Q3: Can sub-reddits be classified as belonging to different 'types' based on the patterns of activity which occur in them? Sub-Reddit and Link clusters can be produced in combination (e.g. Sub-Reddit type could be used as an indicator when clustering Links) to add another explanatory variable if required.

Q4: Are there any patterns to the temporal distribution of votes? Latent trajectory analysis will be used to look for clusters of links with different voting patterns.

Q5: Do certain 'types' of user submit (or vote on) certain 'types' of link? Our primary focus here will be salient (i.e. popular) links; it is predicted that only a small proportion of submitted links will receive

the kind of voting activity required for them to be widely disseminated (i.e. through the site's front page). We will seek to identify any 'user types' who are disproportionately responsible for submitting certain kinds of link.

2 Overview of the Data

2.1 The Data

The data provided by Reddit cover the month of March 2009; **only data relating to "link votes" has been provided,** so the analyses which follow do not consider the commenting aspect of the site at all. As noted above; a "link" submitted to Reddit can be a bonafide link to an external site, but it can also be a link to an internal page on Reddit where a user has posted a statement or question ('Self' links).

The initial data-set offered was a table of every vote received in March (4,336,406 votes) with an ID number for the link it related to, an ID number for the user who submitted it, the value of the vote (positive or negative), and the date and time when the vote was submitted. This initial data-set included votes which had been rejected by the site's anti-cheating code, which was problematic for our purposes.

Reddit were approached again and asked for a table of votes which excluded those which did not actually count; we also asked for data on an array of other aspects which could be used as explanatory variables (they were kind enough to provide most of these). The new table of link votes which counted contained 3,446,522 entries, revealing that 889,884 votes (about 20%) had been rejected as spam in March. The same details noted above were provided for each of these votes.

The additional data we received at this time concerned the links. Tables were provided which showed the User ID of the member who submitted each link, the date and time of this link's submission, the "sub-reddit" (i.e. sub-category) the link was submitted to, and whether the link was "is_self". Self is the name given to the links mentioned above which do not direct to external pages but to a statement or question submitted directly to Reddit. When a link has a value of 1 for "is_self" it is one of these statements or questions, a value of 0 means it is a proper link to another web-page.

The volume of votes being considered in this research was immediately identified as a problem; most statistical software packages simply refused to open a data-set this large. SPSS was the only package identified which could view the data-set, but it did so by only loading a small percentage of the votes into memory. This made carrying out any calculations or transformations of the data extremely time-consuming, even a simple operation like ordering the votes by a given criterion took as long as 15 minutes. It was clear that an alternative approach to this data was required.

2.2 Handling the data

The only way to handle this amount of data seemed to be to work from the kind of SQL database which it came from in the first place. To this end, a local MySQL database was set up and all the data imported into it. This allowed an operation like sorting the votes by a given criteria to be performed in

a few seconds. MySQL was chosen because a graphical interface is available (phpMyAdmin) which allows tables in the database to be easily created and consulted.

In order to produce data that was amenable to statistical analyses, many transformations were required. The fastest way to achieve this was to write custom programs in a language which could interface with the MySQL database. These programs could perform the counts and calculations required with a high level of efficiency and store the results in new tables of the MySQL database. Python was chosen for this purpose because it has a reputation for flexibility and efficiency, and a module is available which facilitates interfacing between Python and MySQL.

The Python scripts which were written to work with the data are included in **Appendix D**. The details of how these programs worked will largely be omitted here. Instead, we will discuss here the details of the new data-sets which were created for statistical analyses; with reference to the Python scripts used to create them.

2.3 Transforming the data – and summary statistics.

With the data arranged in tables of a MySQL database it was possible to generate a count of the number of positive and negative votes in total. Of the 3,446,522 total votes: 2,635,688 were positive (76.5%), 787,874 were negative (22.9%) and 22,960 (0.5%) were Null votes. Querying our contact at Reddit revealed that a Null vote represented an instance where a user changed their mind about a vote they had previously submitted and cancelled said vote. Instances of this behaviour are uncommon in the data.

2.3.1 User Data

Our primary interest in this data is the users, so this is where the re-coding proper began. A new Users table was created. The User ID for every vote in the votes table was then extracted, and these were inserted into the Users table without duplication (*populate_user_ids.py* -**D1**). This revealed that 102,232 different users registered at least one vote on Reddit in March.

Next a program was written which would cycle through every User ID in the Users table: pull the votes for this user from the Votes table, count the number of positive, negative and null votes; then update the Users table with these variables and the total number of votes (*populate_users_with_vote_nos.py*-**D2**). It should be noted that when a user on the site submits a link, a positive vote from them is automatically attributed to the link, so the act of submitting a link also counts as a vote. This program took about 12 hours to execute because it involved searching through the table of 3.5 million votes 102,232 times. It is this necessity of searching through 3.5 million records which has generally been responsible for the computational intensity of generating useful data.

By way of comparison: a subsequent program that searched through the Link Authors table (370,710 records) - to count the number of times a user had submitted a link and add this count to the Users table - took only one hour to execute (*populate_users_with_sub_nos.py* –**D3**). Once all these counts

were in place, another program was written to add the same information, expressed as percentage of total activity, to the Users table. This program (*add_percentages_to_user_votes.py* **-D4**) expressed a user's positive, negative and null votes; and also their link submissions - as a percentage of their total voting activity in March. The program also subtracted their number of negative votes from their positive votes to give an aggregate positive-negative votes total. All of these variables were submitted to the Users table.

2.3.2 Summary statistics for Users data

Consultating of the Users table at this point revealed a few interesting facts. Of the 102,232 users who voted in March, 33,589 (33%) only registered one vote. 26,190 users (25%) made one link submission and this was their only action on the site in March. This probably reflects the ease with which an individual can sign up on Reddit, and suggests that these 26,190 users may have signed up for the sole purpose of submitting a link.

At the other end of the spectrum; the user with the most votes in March registered 23,776 votes and 95% of these were negative; suggesting that this user has taken on the role of moderating content which they do not consider worthy downwards (known in the community as "down-modding"). 12 of the top 30 voters have negative aggregate scores (i.e. they made more negative than positive votes), but there is considerable variation in the proportions of positive/negative votes submitted by this group of 30 users. There were 171 users who each registered more than 1000 votes, 872 users who registered more than 500 votes each, and 7,757 users made more than 100 votes each.

In terms of link submissions; the user with the most link submissions made 1,246 submissions and this represents 75% of their activity on the site. If we rank users by their number of link submissions; for 11 of the top 30 link submitters this behaviour represents more than 95% of their activity on the site. If we consider again the top 30 voters, none of these had a rate of even 1% link submissions as a proportion of activity.

These summary statistics provide some evidence in support of the hypothesis that users will take on different roles in the running of the site. We have however thus far only considered users at the highest end of the activity scale; and even in this small group there is considerable variation in the degree to which users concentrate on one form of activity.

2.3.3 Links Data

A table displaying information in terms of Links was the next to be created. The process of populating this Links table with initial variables was very similar to that employed for the Users table. The following programs were used to carry out these transformations: (*populate_link_ids.py* **-D5**) & (*populate_links.py* –**D6**). The Links table contained some additional pieces of information taken from the supplementary explanatory variable tables; these were the sub-reddit each link was submitted to, whether the link was a self-post, and also the date and time of the link's submission.

2.3.4 Summary statistics for Links data

There were 370,710 links submitted in March in total; 17,808 (5%) of these were "nullified" by the site's anti-cheating code and therefore have zero votes associated with them. Of the 352,902 links which were not nullified, 167,668 (47.5%) of these only received one vote (i.e. the vote automatically cast when they were submitted). The link with the most votes in March received a total of 5,997 votes, 86% of these being positive.

It is interesting to note that when sorting links by total number of votes, 10 of the top 30 voted-for links are Self-posts. A total of 13,353 links are self-posts, just 3.8% of the total number of non-nullified links. Impressions at this point suggest that the Self-posts may have attracted a disproportionate amount of voting activity.

2.3.5 Sub-Reddit data

A table displaying counts of behaviours in terms of the Sub-Reddit they occurred in was produced at this stage (*populate_SR_ids.py* –**D7** & *populate_sub_reddits.py*-**D8**). Most of the fields in this table were generated from the links table (i.e. number of links and votes per sub-reddit, counts and proportions of positive, negative and null votes). In addition to the variables in User and Links tables, the Sub-Reddits table contains values for the average number of votes per link in the category and also the average aggregate score for a link in the category. 2,184 sub-reddits saw activity in March, but of these 730 (33%) only had one link submitted to them. Examination of the Sub-Reddits table reveals that a single sub-reddit accounts for 150,042 (42.5%) of the links submitted in March; this is the general/default sub-reddit.

2.3.6 Vote timing

We now move to a consideration of temporal factors in the voting activity from Reddit in March. Of all the votes cast on links in March, only 81,630 (2.3%) were cast for links submitted before the 1st March ("Old Links"). This gives an initial impression that the voting activity of users will concentrate on new/fresh links (i.e. whether or not a given link will receive enough votes to reach the front page, will probably be decided within a few days of its submission).

Every vote and link-submission event in the data has a recorded date-time; these were supplied in the format Year-Month-Day Hours: Minutes: Seconds. This format is difficult to work with, so an inbuilt MySQL function was used to convert the times to Unix time. Unix time represents time as the number of seconds which have passed since midnight January 1st 1970; storing times in this format makes it easy to find the number of seconds which passed between any two given points in time.

A program was written to calculate, for each vote, the number of seconds which passed between the link submission time and the time at which the vote was cast (*calc_and_store_secs_since_link_sub.py*-**D9**). This program retrieved the times associated with a given link and every vote attributed to it in Unix time; then subtracted the link-post time from the vote time to produce a measure of "seconds since link submission" for the votes tables. This program also divided the "number of seconds since

link post" value by 60 to generate a "minutes since link post" variable, and this was also stored in the Votes table.

To compliment the measures of real time generated above, it was decided that order variables should also be generated for votes. Generating these vote-order variables required one of the most complex and computationally intensive programs written for this project (*generate_vote_orders.py*-**D10**). This program cycles through every link ID, extracting a list of vote IDs ordered by the seconds_since_link_post variable. An accompanying list of vote orders is generated in synchrony with the extraction of ordered vote IDs. Each vote order value is also divided by the total number of votes for the relevant link, to generate a proportional vote order variable. Then the vote order and proportional vote order values are stored in the Votes table under the appropriate vote IDs, and the program moves on to the next link.

This program took about 40 hours to execute, when it had completed every vote (except those for links submitted before March 1st, as no submission time was available for these) had a vote order value and a proportional vote order value. A vote order of 1 means the vote was automatically generated when the link was submitted; votes with an order of 2 were the first vote to be cast for that link by another user, vote order 3 was the 2nd, etc. etc. Proportional vote order values range from 0.0002 to 1. Lower proportional vote order values mean that the vote was cast early in the "voting lifetime" of the link, the maximum proportional vote order value is 1, and these votes were the last to be cast on a given link in March.

2.3.7 Adding extra variables to the Users and Links data-sets.

The next program to be written calculated the mean of the vote order values (proportional and absolute) for each user and stored these in the Users table (*populate_users_with_avg_vote_orders.py*-**D11**). These measures will be used as a rough guide to whether a given user tends to vote early or late in the voting lifetime of a link (average order proportion); and on average how many votes a link already has when they vote on it (average absolute order).

The program written and executed after this (*add_extra_vars_to_users.py*-**D12**) calculated a series of additional user variables. The first of these, average minutes since link post, is related to the order variables above in that it offers a time-based equivalent; the difference is that this measure excludes automatic link submission votes (which are always 0 minutes after link submission). Looking at all three "temporal" averages in combination will give a quick insight into the number of minutes which pass after link submission before the user votes, the number of pre-existing votes on the link when the user votes, and the sequential position of the user's vote relative to other votes on the link before and after.

The other variables added in this program relate to characteristics of the links a user votes on. These include the number of different sub-reddits a user voted in and their average number of votes per sub-reddit; and also the proportion of a user's votes which were registered for Self posts. These measures

will be considered when investigating whether user voting behaviour is affected by recorded characteristics of the links (e.g. are there users whose voting activity is concentrated on one or two sub-reddits? are there users who vote much more on Self links than other types?).

The next program to be written (*add_controversy_to_links.py*-**D13**) generated two more link measurements which should provide a means of gaining insight into the kind of content a given link represented. These measurements have been termed "controversy"; the basic measurement was calculated simply by comparing the number of positive and negative votes on a link and taking the smaller number. Therefore this measure can be thought of as the number of votes on a link which were cancelled out by votes in the opposite direction. This was represented as an absolute value, and also as a proportion of the larger figure (e.g. 0.1 means 10% of votes were negated by votes in the opposite direction, low controversy; 0.9 means 90% of votes were negated by votes in the opposite direction, high controversy).

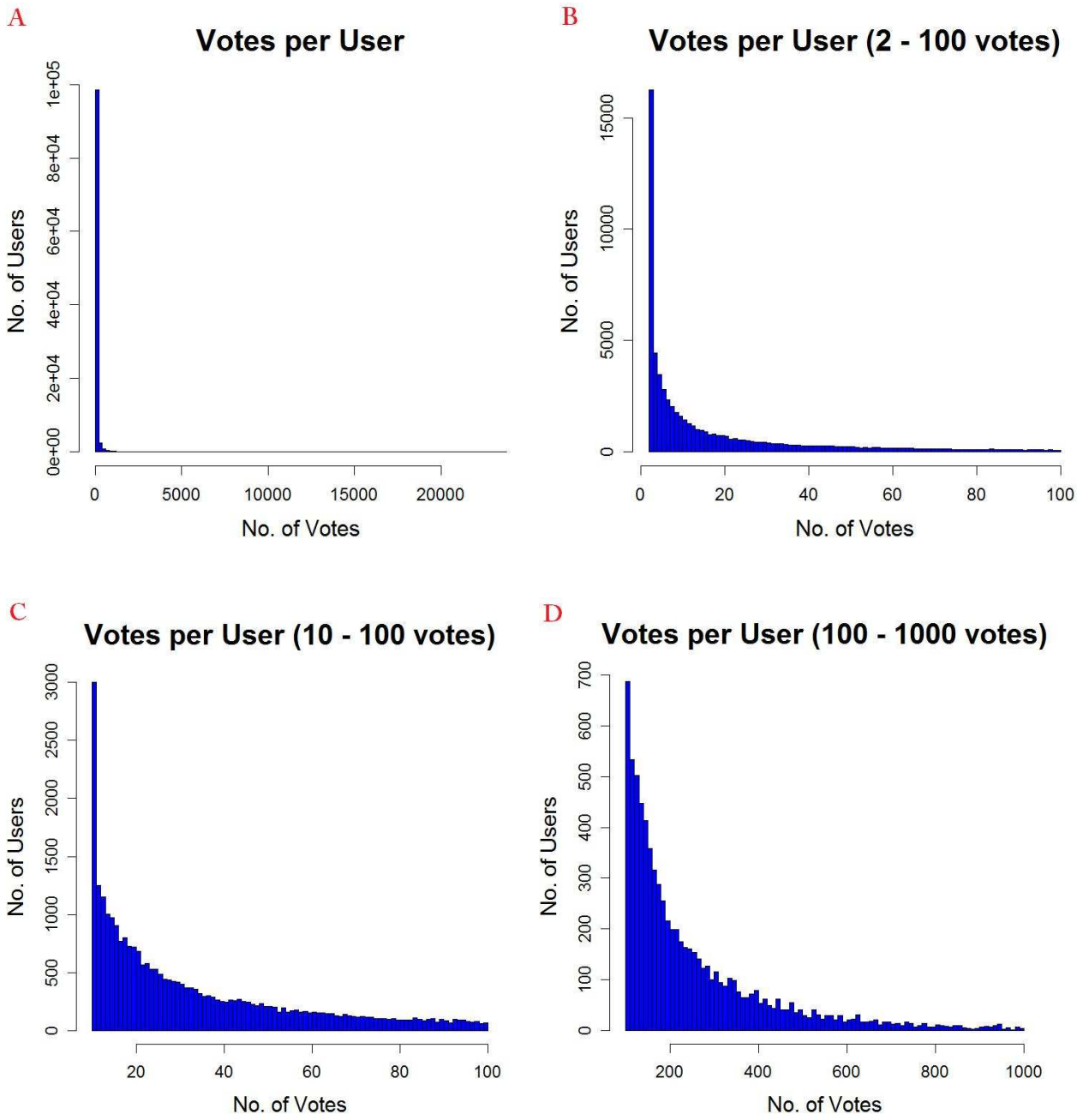
Finally, several scripts were written and executed which added some extra variables to the Users, Links and Votes tables. A script (*generate_user_reg_order.py*-**D14**) was written to rank users' ID numbers; these ID numbers are generated when a user creates their account, so smaller IDs represent older accounts. In the raw data ID numbers range from 77,713 to 5,774,442; these have been ranked from 1 to 102,232 to provide an ordinal measure of how old a user's account on the site is (smaller equals older). The Links and Votes tables also received two additional variables each at this stage; the hour (0-23) and day (1-31) of submission were recorded separately to the full dates and times, for ease of access.

Extracts from the data tables used during the following analyses are included in **Appendix F**.

3 Analysis of Users data

3.1 Descriptive statistics and exploratory analysis

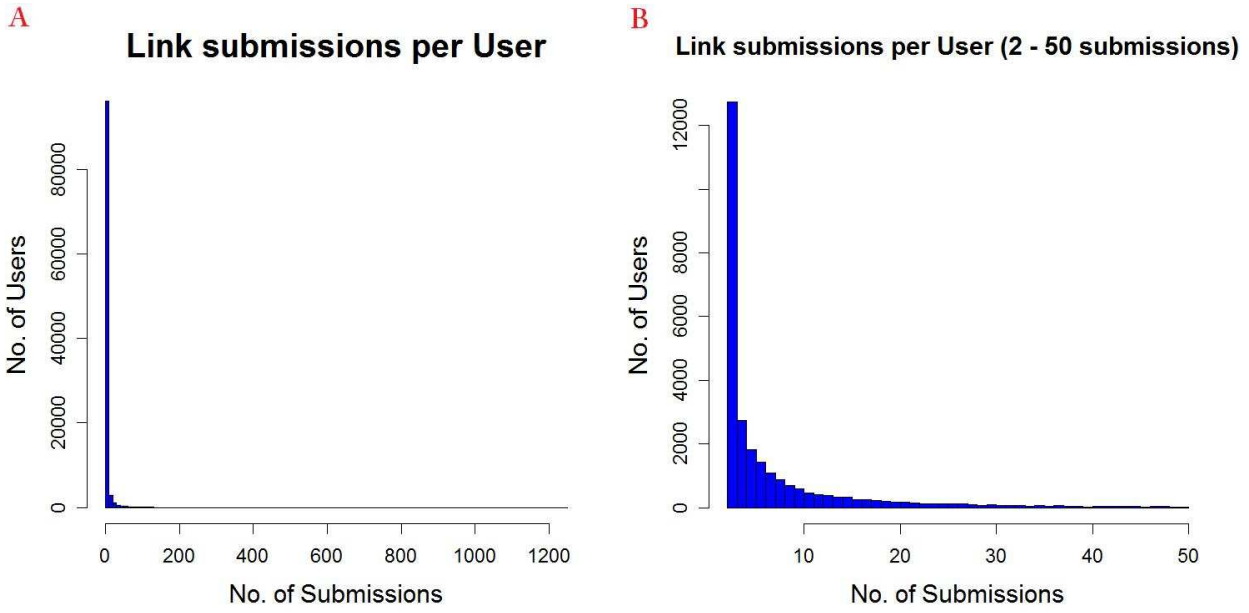
A total of 102,232 users registered at least one vote (which was not rejected as spam) on Reddit in March; therefore the full Users dataset which was produced contained 102,232 cases. This number gives an immediate indication that the vast majority of people who browse Reddit (5,664,590 unique users in March) do not contribute in any way to deciding which content appears on the site. 33,589 of the users who were active in March (33%) only registered one vote; while at the other end of the spectrum the user with the most votes registered 23,776 votes in March. The following histograms (**Figures 3.1-1 A-D**) show that user voting frequency follows an exponential distribution.



Figures 3.1-1: showing (A) The entire range of voting activity (B) Users with between 2 and 100 votes (C) Users with between 10 and 100 votes (D) Users with 100-1000 votes.

Link submission frequencies also follow the exponential distribution (**Figures 3.1-2 below**). These distributions suggest that the activity levels of Reddit's User-base (both voting and submissions) conform to the Power-law cited above as frequently being associated with online communication.

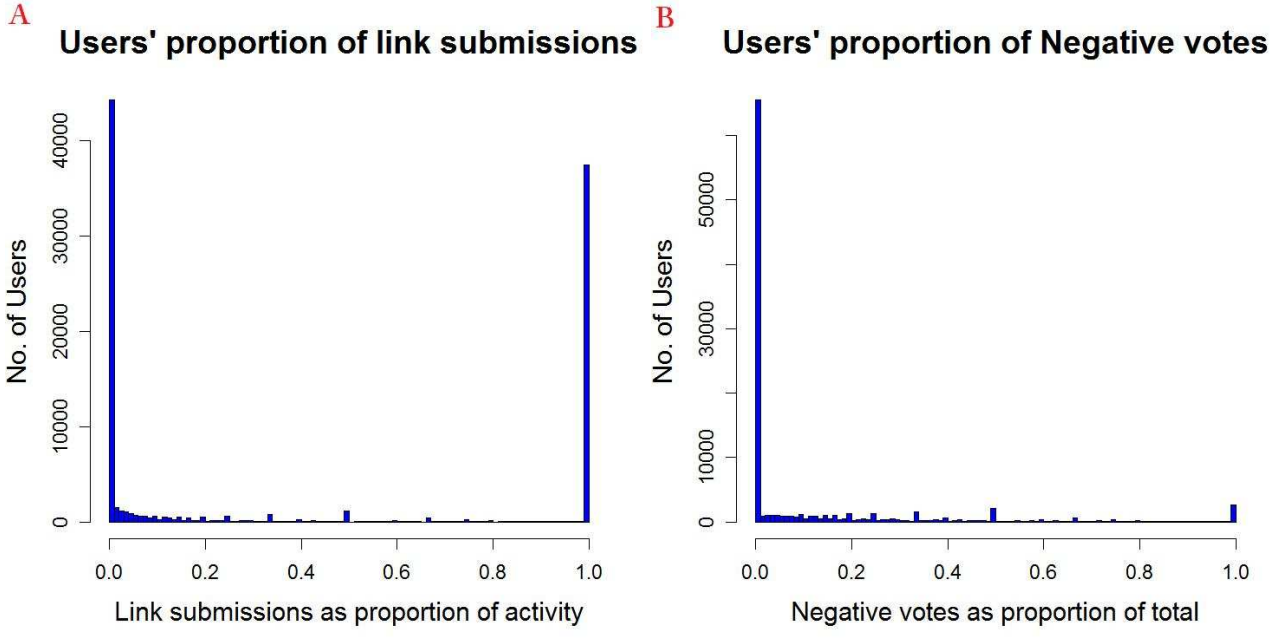
42,788 users (42%) made no link submissions in March and 26,190 users (25%) made just one link submission. The user with the most link submissions made 1,246, and this represented 75% of their activity on the site. Of the top 30 link submitters all submitted more than 430 links; for 28 of these 30 users, link submissions represented more than 50% of their activity in the data; for 11 of these 30 users link submission represents more than 95% of their activity. If we compare these users to the top 30 voters; none of the top voters had a link submission rate representing even 1% of activity.



Figures 3.1-2: showing the number of link submissions per user (A) for the full range of submission behaviour, and (B) for only those Users with between 2-50 link submissions.

If we look at link submission as a proportion of activity on the site (**Figure 3.1-3A below**) we can see that many of the users have a link submission proportion of 0 (42,788 users - 42%) or 1 (37,434 users - 37%). Of the users with less extreme link submission proportions; most have between 0-20% link submissions, but there are also some small peaks at around the 35% and 50% marks.

In terms of voting direction, 62,700 (61%) of users registered only positive votes in March (this includes all users who only submitted links), while 2,542 (2.5%) registered only negative votes. **Figure 3.1-3B** below shows the proportion of users' votes which were negative; there are again a large number of users at the 0% end of the scale, but the number of users at the 100% end of this scale is much smaller than seen with proportional link submissions.



Figures 3.1-3: (A) showing users' link submissions as proportion of activity, and (B) showing users' negative votes as proportion of total votes.

3.2 Clustering Users

Exploratory analyses revealed that the count variables which will form the basis of user clustering (e.g. vote and link submission frequencies) are all exponentially distributed. This means that the bulk of users are situated towards the very low end on scales measuring activity; critical variables to consider if we wish to understand how the site's user-base are moderating its content. Given that the importance of a user's role in the moderating the site's content is in some way proportional to their level of activity, the users which are the most interesting are those who are most active.

3.2.1 Minor issues with the Users data

Some problems with variable values that could negatively effect clustering were noted at this stage. Users' average voting order should have a minimum value of 1, but 417 cases were found with a value of less than 1; similarly users' proportional link submissions should have a maximum value of 1 but 2,665 users were found with values greater than 1. The first issue comes about because the user only registered votes for "old links" (i.e. those submitted prior to the 1st March); these votes were not considered when generating average user voting orders so this is only a problem when the user only made votes on old links (therefore they would have an average voting order of 0). The second issue relates to the site's anti-cheating code, and comes about because some of the users' link submissions were rejected as spam but there are still traces of these in the data. The solutions to these problems were to remove the 417 users with average voting order of less than 1 from the clustering analyses altogether; and to change the proportion of link submissions to 1 for users who had a value greater than 1.

3.2.2 Choosing a clustering algorithm

Clustering on the raw counts was likely to result in the total votes variable dominating the process of cluster formation, with most clusters concentrating on the low end of the scale where most of the data were. This suspicion was confirmed by running several K-means and Partitioning Around Medoids analyses (more specifically: the CLARA (Clustering Large Applications) implementation) in the software package R. Results with the K-means algorithm were highly sensitive to the algorithms starting point due to the large and exponentially distributed nature of the data-set. K-means clusters also tended to concentrate on clusters representing users with relatively small numbers of votes, the cluster with the largest number of votes having a mean of about 50-150. The CLARA algorithm had similar difficulty producing clusters which represented users at the higher end of the activity scale, but faired slightly better here than K-means. CLARA had some other advantages; its basis in medoids rather than means makes it more suited to this kind of data, it produces stable solutions, and analysis of average silhouette widths offers a way of easily comparing the fit with varying numbers of clusters. For these reasons most subsequent clustering analyses have been undertaken by Partitioning Around Medoids (using CLARA).

3.2.3 Transforming the data to make it more suited to clustering

It was clear that the raw count data needed to be transformed prior to clustering if we were to produce useful clusters. Two methods of transforming data were available; standardising all of the variables (so

that the numerical range of each was more similar, thus giving them more equal weight) or manually re-coding them into factors.

The first method of data transformation to be employed was standardisation, and this was done in two ways. Firstly, variables were transformed by subtracting their mean, and dividing by their standard deviation (this approach has the advantage that cluster medoids could be easily back-transformed to yield interpretable values). Average silhouette width for clustering solutions with between 2 and 65 clusters suggested that the optimal solution for this standardised data was just two clusters; one large cluster with a very small activity level and one very small cluster with a medium activity level. The second means of standardisation was employed through the software package SPSS, and resulted in variables which all had a mean of 0 and a standard deviation of 1. The average silhouette width for this approach suggested that either 3 or 4 clusters were optimal, depending on exactly which variables were being considered. These clusters were however difficult to interpret, and it was felt that a detailed summary of user behaviour would need to utilise many more clusters.

At this stage the second method of data transformation was employed; re-coding variables into categories and clustering on these categories. This approach immediately produced more useful clusters which covered the spectrum of user activity levels much more adequately. This approach also had the advantage that specifics of how variables were coded could be tweaked to produce better clustering solutions. This approach appeared to be the most suited to generating useful and interesting clusters; therefore it was decided to focus attention on finding the selection of variables, and means of re-coding these, which would produce the best fit with the data.

3.2.4 Finding the best Users clustering fit

In this process of refining the clusters between 25 and 30 different data-sets were assessed, each with a different combination of indicator variables or different way of factoring these. For each data-set produced: the CLARA function was used to generate clustering solutions with between 2 and 65 clusters; these were then compared by average silhouette width to determine the optimal number of clusters for that data-set (average silhouette width plots for the final version of each data-set clustered are included in **Appendix A**). CLARA was then used to fit this clustering solution (with optimal number of clusters) to the data and the results were inspected to see what the clusters represented and how well the cases in the data fitted the clusters (using silhouette plots and Wk statistics).

Details of all data-sets considered in this process will not be reported here; instead we will concentrate on the final data-set produced for clustering, describing precisely how it was produced then moving on to interpretation of the clusters.

Variables included in the final Users clustering solution were as follows: Total votes, ID age, Average Absolute Voting Order, Proportion of Negative votes, Proportion of Link submissions, and Proportion of votes registered for Self-posts. Total votes is taken as the most important of these variables because it represents a user's total level of activity in the data; it was therefore broken down into nine

categories, while ID age and Average Voting Order were split into five categories each (details below in **table 3.2.4-1**). Proportional variables (Negative votes, Link submissions, Self votes) all have values ranging from 0 to 1; left as they are, these variables will have much less weight in the determination of clusters than the 5/7 point scales. These variables could be transformed such that they take the same range of values as the aforementioned factors; however some of these variables (in particular, proportional link submissions) have the majority of cases situated at the extremes of their scale. Variables like this would tend to take on much more weight in cluster determination than an exponentially distributed variable like total votes if they were represented on the same scale.

It was therefore decided to leave the proportional variables used in clustering on their original 0-1 scale; this would limit the extent to which these variables helped to define clusters, but it would still be possible to look at differences in these proportions between clusters.

Table 3.2.4-1 below shows details of how count variables were re-coded into factors. ID age is a measure of when a user account was created on the site, a value of 1 represents roughly the oldest 20% of accounts, while a value of 5 represents the newest accounts. No additional information about these ID ages is available; they were not explicitly supplied in the received data, instead they have been opportunistically extracted from the data because a measure of account age was sought. ID age re-coding parameters are straightforward (5 categories each representing 20% of users) and therefore are not included in **table 3.2.4-1**.

Total Votes			
Level	Total Votes	No. Users	% of Votes
1	1	33,217	1%
2	2 - 5	24,025	2.2%
3	6 - 10	10,503	2.4%
4	11 - 25	12,584	6%
5	26 - 50	7,810	8.2%
6	51 - 100	5,920	12.2%
7	101 - 200	39,50	16.2%
8	201 - 500	2,869	25.7%
9	500+	867	26%

Average Absolute Voting Order			
Level	Voting order	No. Users	Label
1	1	39,684	Link Submissions
2	1.01 - 10	5,769	Fresh Links
3	10.01 - 100	12,016	Young Links
4	100.01 - 500	30,142	Established Links
5	500.01 +	14,138	Large Links

Table 3.2.4-1: showing factor definitions and frequencies for the re-coded Total Votes and Absolute Voting Order variables.

As previously noted; Total votes has more categories because we wished to bias the clustering solution towards using this important variable, and also towards producing clusters which represent users at the higher end of the activity spectrum. The column in **table 3.2.4-1,** showing the percentage of Votes attributable to users from each Voting activity band, illustrates why it is important to have clusters representing the users at the high end of the total votes measurement. 51.7% of all votes registered on Reddit in March are attributable to just 2.8% of the active users, with 26% of these coming from the top 867 voters. 13.4% of active users in March made 80% of the votes registered.

Average Absolute Voting Order is the mean order of a user's votes, a voting order of 1 means that the user was the first to vote on a link (i.e. they submitted the link), the higher the voting order the more people voted on the link before the user. This variable has been included for two reasons; firstly, it

allows us to differentiate between users who vote on fresh links and those who vote on established links - in doing so it might be possible to identify group(s) of users who are disproportionately responsible for determining which content will become popular (possibly the Users who preferentially vote on fresh links). This variable was split into categories in a way which maximised its usefulness for this purpose; the labels in **table 3.2.4-1** above illustrate what these categories represent.

The second reason for including this variable is that it will add more weight to link submissions in clustering; as a voting order of 1 means the user submitted the link, a mean voting order of 1 means a user's only votes were those which accompanied link submissions. As this value of 1 exists on a 5-point scale it will have more weight than a 1 on the proportion of link submissions variable. This is desirable because whether a user submits links or votes with their account is second in importance only to their level of activity; and exploratory analyses suggest that most users will either always or never submit links, so the clustering solution should be able to reflect this.

There were many other User variables that could have been included in the final data-set to be clustered; the number of variables was deliberately limited to include only those which were good indicators of the most important attributes of user behaviour, and efforts were made to avoid including variables which represented the same information in different forms (e.g. percentage positive and negative votes are inversely related, counts of aggregate votes or link submissions are related to the count of total votes, and average minutes since link post is indirectly related to average voting order).

One variable which seemed to provide a useful insight into the data but was not included in the clustering solution is average proportional voting order. This variable gives an indication of when a user voted in the "voting lifespan" of a link which could be very informative. If this variable was combined with absolute voting order it could be useful in identifying any users with disproportionate influence over which links become popular and receive a lot of votes. This variable was excluded because of its usefulness in associating user types with link success. If clusters are formed with an indicator of link success built in, this could introduce an artificial correlation into the later analyses of relationships between user types and link types. Although this variable was not included in clustering it has been singled out for analysis later in the research (see section **5.4.1**).

The fit of promising looking clustering solutions was also assessed more formally by comparison of their Wk statistics. Wk offers a measure of distance between cluster centres and the cases attributed to each cluster. This was calculated by first finding the raw data variable medians for members of a given cluster, then for every member of that cluster each variable measure was subtracted from its median and the results squared and summed; this was repeated for members of every cluster to produce a total Wk for the whole clustering solution. These values were initially calculated on raw data scores but the differing scales of variables caused problems here. Proportional measures such as proportion of link submissions or negative votes could only be wrong by a maximum of 1; whereas the ID_Age variable ranged from 1 – 100,000 and was uniformly distributed, so it was not uncommon for a case to be off by 10,000 on this measure. For this reason Wk statistics were calculated on standardised data (**sample**

R code in Appendix E), variable means were subtracted from the raw data score and they were then divided by their standard deviation; this made the contributions of different variables to the Wk much more consistent. Wk scores for a range of solutions are reported in **table 3.2-2** below; these include the optimal 58-cluster solution based on variables as categorised in **3.2-1** above.

Clustered on (optimal)	2 clusters	18 clusters	58 clusters
Raw data (18)	576448.4	527513	521260.6
Standardised data (2)	649463.1	533516.3	528605.5
Categorised data (58)	493253.7	379342.8	**318580.2**

Table 3.2-2: showing Wk statistics for nine clustering solutions; clusters formed on raw data, standardised data and categorised data, with the respective optimal number of clusters (determined by silhouette width) for each data type.

3.3 The optimal Users clustering solution

The best clustering solution for this final Users data-set involved fitting 58 clusters. This is quite a lot of clusters to interpret, but we are dealing with a large data-set (101,759 cases after the removal of users who only voted on old links) and each of the six variables being considered could provide a lot of insight into patterns of User activity in the data. Throughout the process of refining the variables for clustering the optimal number of clusters tended to increase with each iteration of the data-set; the composition of these clusters was however quite stable, many clusters persisted through all iterations of the data-set despite variation in the data used and the method of coding it.

These clusters were fitted on six variables (as described above), variables used were limited to those which reflect the nature of a user's actions and available contextual information (as it was at the time a user acted); these clusters were formed on variables that bear no indication of what happened to the link a user submitted or voted on after they made their contribution.

Clusters in **table 3.3-1** (overleaf) have been ordered first by their medoid voting category and then by their size. **Cluster medoids for all interpreted clustering solutions are re-produced in Appendix A**. In **table 3.3-1** clusters have been coloured to indicate their medoid total votes category, as this has been deemed to be a very important characteristic. User clusters with smaller total votes medoids have lighter colours, while those with a lot of votes have darker colours. This same pattern of colouring has been used to represent voting activity levels in other clustering solutions below.

It is immediately apparent that the "link submissions as proportion of activity" variable seems to be useful in determining the types of user these clusters represent. 48 of 58 clusters have a proportion of link submissions which is either 0 or 1, suggesting that most of the clusters represent users who either always or never submit links. Where a user type has a link submission proportion of 1; several columns have been highlighted in blue and the **link submission medoid is in red**. These columns have been highlighted because the medoids they contain are always the same when the proportional link submission is 1; these columns represent the user's average voting order (always 1, because the user only casts votes which are automatically generated with their link submissions), proportion of

negative votes (always 0), and proportion of votes on Self posts (this proportion does not have to be 0, but for these clusters it always is; suggesting that people who only submit links tend not to submit 'Self' links). Where a cluster has a % link submission medoid of 0 its colour is not changed, where a cluster has a % link submission of between 0 and 1 these cells has been highlighted with a white background.

Cluster	Total Votes	ID age	Voting Order	% Negative	% link submission	% self votes	Cluster size
35	1 vote	5	Order 1	0	1	0	**14502**
39	1 vote	4	Order 1	0	1	0	5930
14	1 vote	2	Order 1	0	1	0	2501
6	1 vote	**2**	Order 101 - 500	0	0	0	2313
7	1 vote	3	Order 11 - 100	0	0	0	2077
24	1 vote	3	Order 1	0	1	0	1972
33	1 vote	3	Order 500+	0	0	1	1163
3	1 vote	**1**	**Order 2 - 10**	0	0	0	888
56	1 vote	5	**Order 2 - 10**	0	0	0	858
4	1 vote	**1**	Order 11 - 100	**1**	0	0	855
44	1 vote	**2**	**Order 2 - 10**	**1**	0	0	334
18	1 vote	**1**	Order 101 - 500	**1**	0	1	280
52	2 - 5 votes	5	Order 1	0	1	0	3506
49	2 - 5 votes	4	Order 1	0	1	0	3052
47	2 - 5 votes	4	Order 101 - 500	0.25	0	0	2449
21	2 - 5 votes	**2**	Order 1	0	1	0	2107
25	2 - 5 votes	3	Order 1	0	1	0	1756
16	2 - 5 votes	**1**	Order 11 - 100	0.25	0	0.25	1590
20	2 - 5 votes	**2**	Order 500+	0	0	0.8	1552
22	2 - 5 votes	**2**	Order 101 - 500	0.25	0	0	1506
27	2 - 5 votes	**1**	Order 101 - 500	**0.6**	0	0.2	1500
38	2 - 5 votes	**2**	Order 11 - 100	**1**	0	0	1246
46	2 - 5 votes	5	Order 101 - 500	0	0	0.2	1029
19	2 - 5 votes	**2**	**Order 2 - 10**	**0.6**	**0.4**	0	964
11	6 - 10 votes	3	Order 101 - 500	0.125	0	0.125	2559
13	6 - 10 votes	3	Order 500+	0.3333	0	0.1111	1889
5	6 - 10 votes	**1**	Order 500+	0.3	0	0.1	1736
17	6 - 10 votes	**1**	Order 101 - 500	0.4444	0	0.1111	1403
51	6 - 10 votes	4	Order 11 - 100	0.2857	0	0	1268
55	6 - 10 votes	5	Order 1	0	1	0	1028
54	6 - 10 votes	4	Order 1	0	1	0	930
41	6 - 10 votes	3	Order 1	0	1	0	699
57	6 - 10 votes	5	Order 500+	0.3333	0	0	395
8	11 - 25 votes	**1**	Order 101 - 500	0.48	0	0.04	2395
28	11 - 25 votes	**2**	Order 101 - 500	0.1538	0	0.0769	2258
1	11 - 25 votes	3	Order 1	0	1	0	1757
37	11 - 25 votes	2	Order 500+	0.2308	0	0.2308	1625
2	11 - 25 votes	3	Order 101 - 500	0.3333	0	0	1580
48	11 - 25 votes	5	Order 101 - 500	0.2308	0	0.0769	1484
29	11 - 25 votes	3	Order 500+	0	0	0.25	1146
36	11 - 25 votes	**2**	**Order 2 - 10**	0.1667	**0.3333**	0.5	456
23	26 - 50 votes	**1**	Order 101 - 500	0.0789	0	0	1962
31	26 - 50 votes	**2**	Order 101 - 500	0.2444	0.0889	0.1778	1541
12	26 - 50 votes	3	Order 101 - 500	0.3448	0	0.2069	1411
43	26 - 50 votes	3	Order 11 - 100	0.0526	0.1053	0.1842	1095
32	26 - 50 votes	3	Order 500+	0	0	0.0213	990
53	26 - 50 votes	**1**	**Order 2 - 10**	0.0238	**0.5238**	0.0238	515

58	26 - 50 votes	5	Order 1	0	1	0	205
50	26 - 50 votes	1	Order 2 - 10	0.9655	0	0.3448	82
9	51 - 100 votes	1	Order 101 - 500	0	0.0926	0.4074	1884
45	51 - 100 votes	4	Order 101 - 500	0.1017	0	0.0339	1754
42	51 - 100 votes	2	Order 101 - 500	0.7647	0.0392	0.0588	1631
40	51 - 100 votes	4	Order 1	0	1	0	695
30	101 - 200 votes	2	Order 101 - 500	0.3103	0	0.0966	2567
10	101 - 200 votes	4	Order 101 - 500	0.0724	0	0.125	1339
34	101 - 200 votes	5	Order 101 - 500	0.3354	0.1402	0.1463	216
15	201 - 500 votes	2	Order 101 - 500	0.1874	0.0407	0.1263	2522
26	501+ votes	2	Order 101 - 500	0.001	0.006	0.0379	802

Table 3.3-1. showing cluster centre medoids for the final Users clustering solution.

Colours have also been applied to some clusters' Voting order, proportion negative votes, and ID age medoids. Where a cluster has a voting order medoid of **2 – 10**; this has been highlighted with a **purple background and yellow text**; it has already been suggested that users who vote on fresh links might be noteworthy as fulfilling a particular role on the site. Users with a **percentage of negative votes greater than 50%** have been highlighted using a red **background with bold text.** Users with **ID ages of 1 or 2** are **bold**, these represent older user accounts.

The number and size of clusters representing users with **100% link submissions** tends to decrease as we move down the list and consider the clusters which represent more active users. **74%** of users with 1 vote fall into this category; **47%** of users in the 2-5 votes group, 26% of users in the 6-10 votes group, 14% in the 11-25 votes group, **3%** in the **26-50 votes group** and **12%** in the 51-100 votes group also fell into this category. Of the five clusters representing users with a medoid of more than 100 votes, none represented **users who only submit links**.

Most of the users at the top of the table who don't have a link submission proportion of 100% activity tend to have a proportion of 0%. This is a given for users in the 'one vote' group who must either have a proportion of 0 or 1, but it is surprising that so few of the users in the other 'low votes' groups have been placed in clusters representing a user type that both votes and submits. Of the 41 clusters representing users with a total votes medoid of between 1 - 25, only two represent users that both vote and submit links (representing just 2% of users in these clusters). This is in stark contrast to the lower end of the table representing user clusters with a larger total votes medoid; about half of the clusters representing users with more than 25 votes have link submission proportions of between 0 and 1; with half the users in this group being placed in these clusters.

Considering these "proportion of link submission" medians across the full spectrum of activity levels suggests a number of trends in this data. At the lower end of the activity spectrum (1-25 votes) users are very likely to prioritise one form of activity (i.e. voting or submission) and marginalise or exclude the other. Within this group that either votes or submits, the users with the lowest activity levels of all are most likely to be **'submitters'**; with the chances of someone being a 'voter' increasing with their level of activity. This suggests two main types of infrequent or casual user; those who submit and those who vote. The shift from submission to voting behaviour with an increasing level of activity

could reflect the level of effort involved in each kind of behaviour; casting a vote merely involves clicking an up/down arrow, while making a submission requires an idea for something to submit, a title for the submission, a choice of which Sub-Reddit to submit to, etc.

At the higher end of the activity spectrum; frequent users are more likely to fall into a category which represents a combination of voting and submission behaviours. There are however still quite a few clusters in the 26 – 200 votes group which have a link submission rate medoid of 0 or 1; suggesting that the tendency to use an account for either voting or submitting can also be found among some of the more frequent users of the site.

While we are considering trends which concur with increasing activity levels, let us look at the account age factor. There seems to be a tendency for older user accounts to be more active, 79% of users in clusters with medoid "total votes" of greater than 100, had a medoid "account age" in the oldest 40%. A trend like this is not surprising, it is expected that users who have been active on the site for a number of years will be more active than those who created an account in the last month (it is unfortunate that out measure of account age is not more specific about when accounts were created).
This trend is however not particularly strong for user accounts in the oldest 20%; clusters with this medoid tending to be found more towards the mid-range of the activity spectrum.

Account age would appear to have a stronger relationship with one's proportion of link submissions than with level of activity. Of the clusters representing users that **only submit links**; none have an account age medoid in the oldest 20% and just two of the relatively small clusters have an account age medoid of **2.** This suggests that a **100% link submission rate** is much less likely for older user accounts.

If we consider next a user's average voting order this reveals another possible difference between short and long term users. An **average voting order of 2 – 10**, i.e. someone who tends to vote on fresh links, has previously cited as being of particular interest in these analyses. There are **seven clusters with this medoid**, and six of them have an account age medoid in the **oldest 40%.** This could signify that longer-term users of the site recognise the increased importance of votes cast early in a link's voting lifespan, and are more likely to use their votes in this way. The relatively large "proportion of negative votes" medoids for clusters with this vote order characteristic would tend to support this hypothesis. Any such support is however tentative; because this measure is based on a mean, users who registered even one vote late in the lifespan of a link (i.e. order of over 1000) would be very unlikely to be placed in this average order category. This probably explains why none of the users in the more active clusters have an **average voting order medoid of 2 – 10**; we will return to this issue later in the analyses **(section 5.4.1).**

Looking at the proportion of negative votes medoids reveals that 34 of the clusters represent users with at least one negative vote; eight of these represent users with greater than 50% negative votes, with the remainder having medoids quite evenly distributed between 0 and 50% negative votes. Of the clusters

representing users who never voted negatively; 15 only submitted links (so no chance to vote negatively), leaving 10 types of user who only cast positive votes (by choice).

There are a few clusters with particularly interesting negative votes medoids. Cluster 19 represents 964 users whose activity on the site consists of (40%) link submission and (60%) negative votes cast early in the life of links. These users could be using their account to search out and vote down links which might compete with those submitted by themselves, thereby increasing the chances that their link might be successful; this type of user tends to not be very active however (medoid 2 – 5 votes). Clusters 50 and 42 represent users who are quite active voters (26 – 50 and 51 – 100 votes respectively) and have a strong tendency to vote negatively, cluster 50 is however the smallest of all clusters with just 82 members. All three of these active clusters with a lot of negative votes have an **ID age of 1 or 2**, suggesting that these users tend to have accounts in the **oldest 40%**.

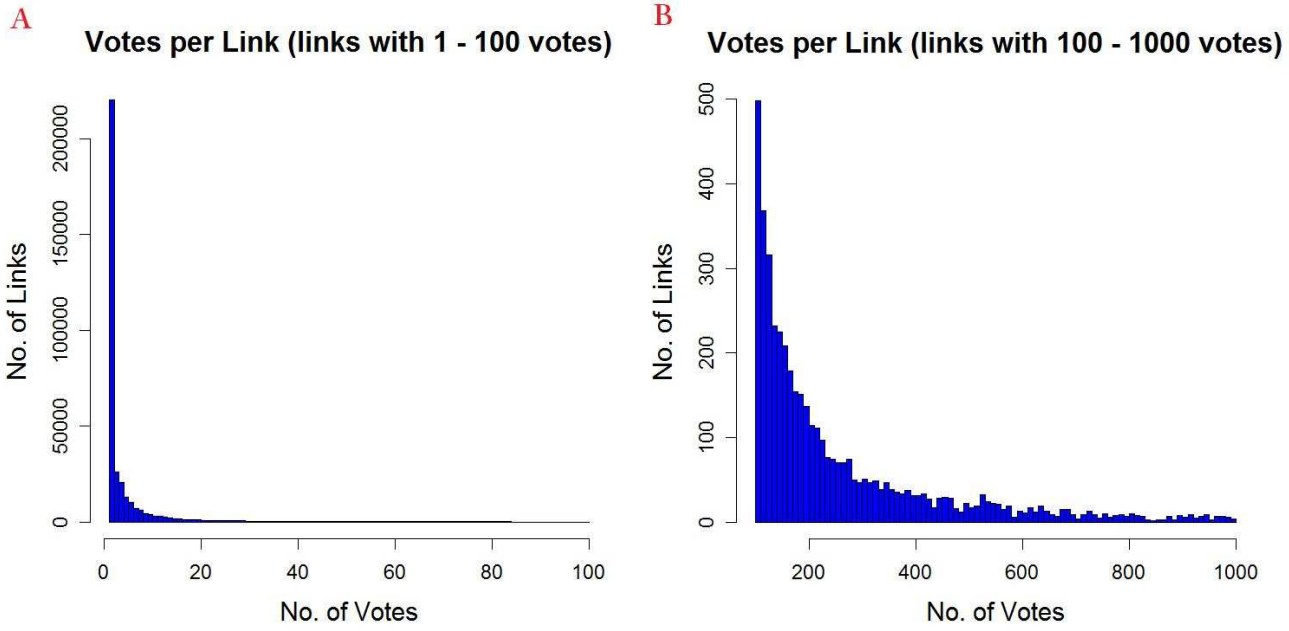
Considering now the proportion of Self votes medoids, we can see that most of the clusters representing more active users indicate some level of activity related to "Self" content. However, all of the user types who only submit links have a proportion of Self votes medoid of 0, suggesting that these users have a very strong tendency to submit only links to external sites. This could be quite a revealing relationship, and highlights an interesting aspect of the "Self" link. As these links do not direct to external sites, the only purpose one could have in submitting a link like this would be related to the reddit.com community (this could be gauging their opinion to something, their responses to a particular question, or to highlight some aspect of the poster's personality to the community or enhance their reputation). While it is possible that a user could submit an external link for similar reasons to those identified for self links; there are another set of possible motivations which could be quite strong here. As noted previously, a link which makes it unto the Reddit front page can bring a lot of traffic to that internet location; this characteristic of the site would be a big attraction to individuals wishing to promote their own website (or being paid to promote someone else's). It is conceivable that some individuals might sign up on Reddit for the sole purpose of using it to promote their own web content; these would most likely be users who only submit links.

If we consider the interactions between proportion of link submissions, proportion of self votes, account age and level of activity in these cluster medoids; we can begin to see the markings of something which could be termed "community involvement". A 100% link submission rate would seem to suggest a user type with low community involvement; these users very rarely submit Self links, so every behaviour they exhibit could be serving a purpose external to Reddit. The fact that these users tend to have newer accounts and aren't usually very active would support the idea that they have low community involvement. Conversely, users who make more votes on other peoples' links, who vote more on Self links, who are more active and who have older accounts (medoids on these variables seem correlated to some degree) are more likely to have a high level of "community involvement". It will be interesting to see whether any evidence will be found to support this hypothesis when we consider the relationship between user types and link types later.

4 Analysis of Links data

4.1 Descriptive statistics and exploratory analysis

There were 370,710 links submitted to Reddit in March; of these 17,808 (5%) were rejected by the site's anti-cheating code and will not be considered further; this leaves 352,902 links for analysis. Of all the votes cast in March, only 81,630 (2.5%) were registered for links submitted before March 1st; suggesting that the turnover of popular content on the site is quite fast.



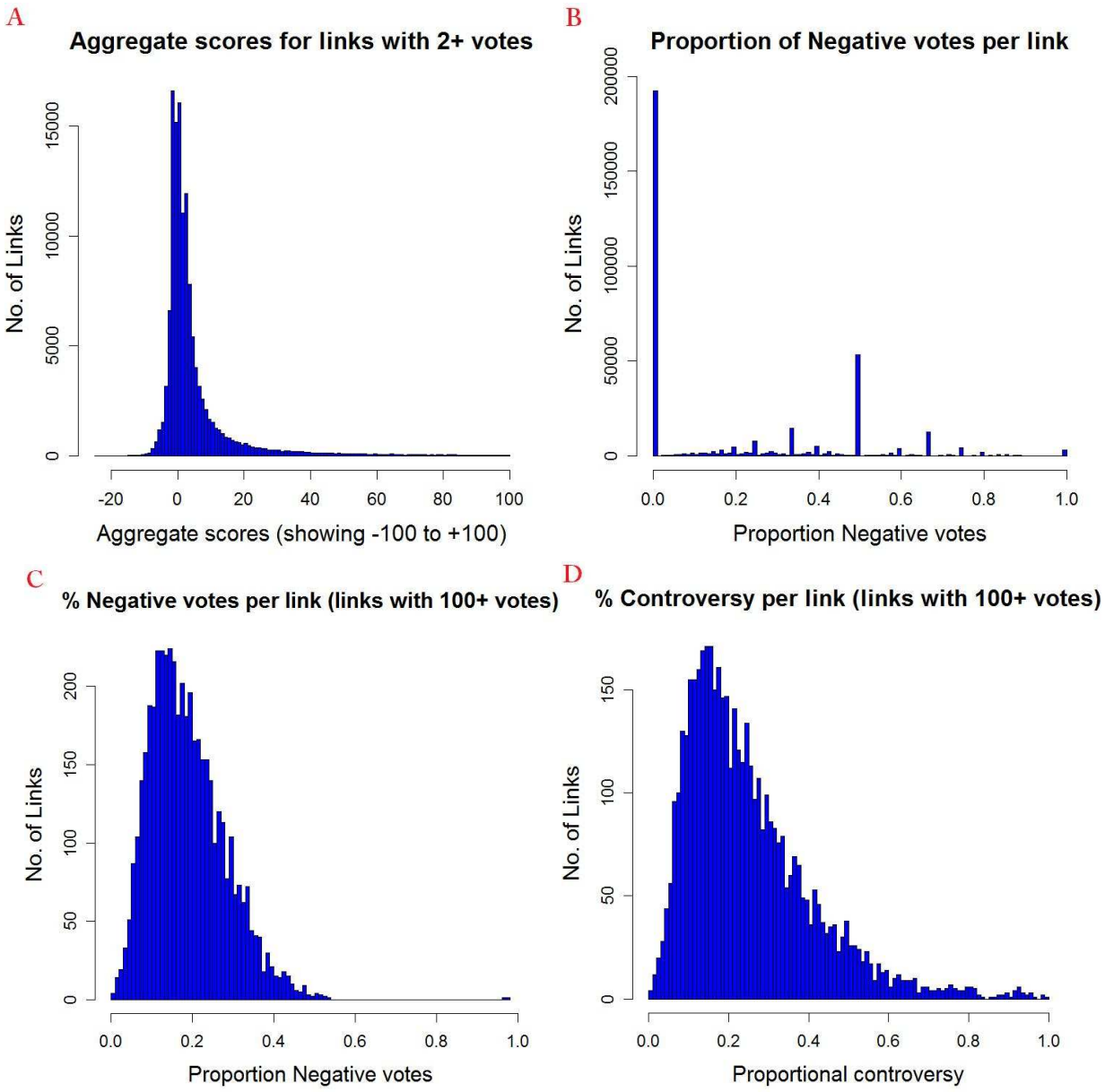
Figures 4.1-1. Histograms showing (A) the number of links which received between 1 and 100 votes and (B) the number of links which received between 100 and 1000 votes.

Figures 4.4-1 A and B above suggest that the number of votes per link follows an exponential distribution. 167,688 of the links not rejected as spam (47.5%) only received one vote, the vote which was automatically generated by the user who submitted them. At the other end of the scale, the link with the most votes in March received 5,997 votes (86% of these being positive).

The direction of votes registered on links can be assessed in several ways. The easiest way to do this is to look at the links positive – negative aggregate score. **Figure 4.1-2A** (below) suggests that these scores are normally distributed in the area immediately around the zero score, but there is a much greater positive tail to these scores. The most negative score produced by any link was -624 but this was somewhat of an outlier, only two links had a score more negative than -25; whereas the link with the most positive score achieved a score of +4,393. This skew to the distribution reflects the mechanics of how the site works; once a link has a negative aggregate score it does not appear in any easily accessible areas of the site, therefore it will not appear for people to vote on presumably unless they are deliberately searching through unpopular content. This aspect of how the site works suggests that if we wish to consider how positively or negatively a link was received the best way to do this is with a proportional measure.

Looking at the proportions of Negative votes per link overleaf (**Figures 4.1-2B+C**), it is clear that the majority of all links only receive positive votes; but this includes a lot of links which only received one vote, from the user who submitted them. There is also a substantial number of links which received

50% negative votes, which would leave their aggregate score at 0. If we consider the sub-set of links with more than 100 votes a much clearer distribution emerges; this is a normal distribution with mean of around 18% and tails which reach the 0% and 50% marks. This abrupt halt at around the 50% negative votes value is another marker of the mechanics of link voting on the site; once a link has a proportion of negative votes greater than 50% it will have a negative aggregate and therefore be unlikely to elicit more voting.



Figures 4.1-2 (A) showing aggregate scores between -100 and 100 for links with more than one vote. (B) showing the proportion of Negative votes received by all links. (C) showing the proportion of Negative votes received by links with more than 100 total votes, and (D) showing the proportional controversy score for this same group.

An alternative measure, proportional link controversy, might also be useful in classifying links based on how they were received by Reddit users. This variable expresses a link's votes as the number of negative votes (or positive, whichever is smaller) divided by the count of the more frequent vote type; this yields a measure reflecting the percentage of votes on a link counteracted by votes in the opposite

direction. A proportional controversy value of 1 therefore means that the link will have an aggregate score of 0 regardless of how many votes it received in total. **Figure 4.1-2** above shows proportional controversy for the sub-set of Links with more than 100 votes in total.

This measure could be more useful in differentiating between links with quite a few votes than the proportion of negative votes; it can take values of between 0 and 1 for any quantity of total votes, as opposed to proportion of negative votes which seems restricted to between 0 and 0.5 for links with a moderate number of votes. The advantage of the proportion of Negative votes variable is that it is possible to identify links which were overwhelmingly disliked, proportional controversy values do not specify which direction an aggregate will be in. These two variables will therefore both have a place in the analyses conducted on Links data.

4.2 Clustering Links

Links were then clustered based on selections of the available variables; this clustering was carried out using a similar approach to that employed with Users clustering. Variables used at this stage all represented either the amount of voting activity received by the link or the direction of these votes; these were clustered as raw, standardised, and categorised values. It quickly became apparent that the range of variables available were not adequate to generate useful clusters; to improve the usefulness of these clusters more information about the nature of the links was required. In the absence of access to the actual qualities of the links, the only available information about what a link might represent was the Sub-Reddit it was submitted to. There were 2,814 active Sub-Reddits in March, so using the Sub-Reddit itself as an indicator variable was not feasible. For this reason it was decided to shift focus to clustering the Sub-Reddits, if this was successful it would be possible to use the Sub-Reddit type as an indicator when analysing links.

4.3 Sub-Reddits on Reddit.com

4.3.1 Descriptive statistics and exploratory analysis

Sub-Reddits on Reddit.com are essentially sub-categories of content. The largest Sub-Reddits are by default displayed as buttons at the top of the site; when a user clicks one of these buttons they are presented with only the links from that Sub-Reddit, these are by default sorted by aggregate scores weighted towards more recent votes. The default Sub-Reddits (on August 20th 2009) included topics ranging from “Politics”, “Technology”, “Science” and “Business” to those labelled “Pic[ture]s”, “Funny”, “Offbeat” and “Videos”. Default Sub-Reddits also include “AskReddit”, devoted to Self posts asking questions of other users; and “Bestof”, dedicated to links to material on Reddit itself (usually comment threads).

In addition to Sub-Reddits which serve particular purposes, there is a General or Main Sub-Reddit; this is the default category for a link to be submitted to, and 150,042 (42.5%) of the links submitted in March were submitted here. The size of this “Sub-Reddit” suggests that it is not a Sub-Reddit in the sense that those serving particular topics are; and therefore that it should be handled separately.

The other function of Sub-Reddits on the site is that users who have created accounts can specify which Sub-Reddits they subscribe to. If a user removes one of the default Sub-Reddits from the list they subscribe to then they will no longer see links from this Sub-Reddit on the site's "front page". The default Sub-Reddits at any given time are determined according to level of activity/subscriptions. This mechanic of how the site works will most likely lead to a sharp divide between the activity levels of Sub-Reddits which are large enough to be included in the default list; and those which are too small to make it unto this default list (material on these Sub-Reddits is therefore only going to be seen by users who have subscribed to them). The choice registered Users make about which Sub-Reddits they subscribe to therefore fulfils two functions. Firstly, it determines the types of content which the user themselves will be exposed to. Secondly, the user-base as a whole, by their individual choices of subscriptions; determine which types of content will be displayed by default (i.e. the types of content which will appear to the many users of the site who have not registered an account). This is significant because it allows the democratic determination of popular content to operate at a second more general level. Surprisingly, this aspect of the site's democratic system receives very little attention. Relative to the prominence of the link and comment up/down voting on the site, this second process is like an obscure footnote to the site's workings.

4.3.2 Clustering Sub-Reddits

The main sub-reddit will not be included in the process of clustering Sub-Reddits; due to its large size this sub-reddit will be treated as its own type. The remaining sub-reddits were clustered according to a variety of variables, usually standardised. Average silhouette width was again used to determine the optimal number of clusters for any given combination of variables. Wk statistics for some of these clustering solutions are included in **Table X** below; these values have been calculated using standardised variables.

Clustered on (optimal)	2 clusters	10 clusters
Raw data (2)	10355.5	8499.39
Standardised data (10)	10355.5	**3369.67**

Table 4.3.2-1. showing Wk statistics for some Sub-Reddit clustering solutions.

The best clustering solution for Sub-Reddits involved fitting 10 clusters on five standardised variables. Variables included in this solution are as follows: Total Links, Total Votes, Aggregate votes per link, average proportional controversy and the proportion of Self links in the Sub-Reddit. These variables were chosen to prioritise the level of activity in the Sub-Reddit (link submissions and voting), also paying attention to how links were generally received by voters in the Sub-Reddit (Aggregates and Proportional controversy), and whether the Sub-Reddit received a lot of Self links.

Cluster	Total Links	Total Votes	Agg per Link	% Controversy	% Self	Cluster N
1	150042	516775	1.93	0.1763	0.02178	1
2	1969	23521	6.51	0.5313	0.0747	18
3	1729	63923	19.76	0.5622	0.0133	12
4	183	9213	35.28	0.2654	0.1967	5
5	82	968	6.93	0.3926	0.0000	191
6	17	44	1.88	0.2157	0.0000	301
7	9	35	3.33	0.0444	0.5556	82
8	2	4	1.00	0.5000	0.0000	169
9	1	2	0.00	1.0000	0.0000	71
10	1	1	1.00	0.0000	1.0000	121
11	1	1	1.00	0.0000	0.0000	1213

Table 4.3.2-2. showing back-transformed cluster medoids for the 10 clusters generated using CLARA (plus the general sub-reddit type); ordered, coloured and re-labelled by total number of links received.

The Sub-Reddit clusters suggest that there are two types of Sub-Reddits (SRs), other than the general one, which receive a lot of link submissions. These two Sub-Reddit types are most likely those whixh weew on the default list of Sub-Reddits in March; unfortunately as this default list is re-generated from day to day it is not possible to confirm this. The first of these (type 2) has 18 members and produced a medoid of 1,969 links; the second (type 3) has 12 members and these see slightly less submission activity but much more voting activity. Links in type 2 SRs achieve an average aggregate of +6.5 votes, while those in type 3 SRs have an average aggregate score of +20. This suggests that links in type 3 SRs see a lot more voting activity on average, but the proportional controversy of links in these SR types is similar.

The fourth SR type is much smaller, representing SRs which receive far fewer links, but where these links do very well on average; links in this SR type have a medoid Average Aggregate of +35.28 and a lower average proportional controversy. This would suggest that the links submitted to these SRs have a higher likelihood of receiving positive votes than those submitted elsewhere. SR types 5 and 6 represent larger clusters of Sub-Reddits which receive a moderate amount of link submission and voting activity. The remainder of SR types represent Sub-reddits that receive very little activity; the largest SR type has 1,213 members; these are characterised by having just one link and one vote in March.

Subsequent contact with Reddit allowed us to put names to some of the Sub-Reddits making up these types. For the two Sub-Reddit types representing large and active Sub-Reddits (types 2 and 3); there is little relationship between the topics of these Sub-Reddits and the Types they have been assigned. Type two includes the following Sub-Reddits: Business, Sports, Gaming, Entertainment, Linux, Videos, AskReddit, Environment, Economics, Music and News. Type three includes the following Sub-Reddits (which tend to receive fewer links but more votes per link): Politics, Science, Pics, Worldnews, Technology, Funny, Bestof, Programming. As expected, Sub-Reddit type five contains

moderately active Sub-Reddits with mostly niche topics likely to appeal to small sub-sets of the User-base. These include the following: StarWars, Germany, Motorcycles, iphone, Survivalist and Hockey.

4.4 Clustering Links re-visited

Sub-Reddit types were added to the Links data-set such that every link was assigned to one of the eleven sub-reddit types based on the sub-reddit it was submitted to. This variable was then included in the clustering of links as a factor. Many clustering solutions were fitted to this new data-set, following the same protocol as earlier clustering attempts. The data were transformed (standardised or categorised) and silhouette width (for between 2 to 65 clusters) was used to determine the optimal number of clusters for each set of variables considered.

Wk statistics were calculated for promising fits and these were used to aid in choosing the best clustering solutions. Sub-Reddit type was not considered in the Wk measures for link clustering solutions as this is a nominal measure. Wks were based on standardised measures of total votes, aggregate votes, proportion negative and proportional controversy; and also on the binary Self variable. Links' Wk statistics were based on a random sample of 35,000 links, due to the large size of the links data-set – i.e. the amount of computation required to generate a Wk statistic encompassing all the Links was too great. Some of these WK statistics are included in **table 4.4-1 below.**

Variables used	Wk
All standardised variables	182263.188239706
Standardised variables without Sub-Reddit type	185516.493598424
Raw variables	185474.580443557
Categorised variables	211980.720449992

Table 4.4-1. showing Wk scores for 47-cluster solutions fitted on different sets of variables.

The above table of Wk scores suggests that adding the Sub-Reddit type factor improves the fit of the clustering solution over that produced without this new factor. Clustering links based on manually categorised values seems to produce a much worse solution for Links.

The clustering solution which will be interpreted and pursued is that which used standardised versions of the Total Votes, Aggregate score, Proportion of Negative votes and Proportional controversy variables. The binary Self variable and pseudo-ordinal Sub-Reddit type were also included. These variables are expected to produce a good clustering solution which prioritises measures of success and activity (aggregate and total votes). Link types will also pay reference to the direction of voting received by the link; and use the type of sub-reddit it was submitted to and whether it was a Self post to provide some context to what the links might represent. The optimal number of clusters for these variables (by average silhouette width) was 47, and the back-transformed medoids for these are included in **Table 4.4-2** below.

The cluster medoids below have been ordered by aggregate score, with the link types receiving the highest aggregate at the top of the list. The Total Votes, Aggregate, and Cluster size medoids have

been coloured according to size, with smaller quantities in light colours and larger quantities in dark colours. This pattern of colouring is similar to that used for User types; except that for Links pink represents a negative aggregate while grey represents links with an aggregate of 1.

Cluster	Aggregate	Total Votes	% Negative	% Controversy	Self	SR type	Cluster N
44	**636**	**938**	0.1578	0.1888	**0**	3	452
42	382	554	0.1534	0.1820	0	1	322
35	262	530	0.2472	0.3333	0	3	338
46	158	316	0.2468	0.3305	0	3	663
32	119	203	0.2020	0.2563	0	3	1659
43	73	264	**0.3561**	**0.5629**	**0**	3	188
18	29	49	0.2041	0.2564	0	3	2969
45	26	27	0.0000	0.0000	0	2	536
20	21	58	**0.3103**	**0.4615**	0	3	1701
15	21	37	0.2162	0.2759	**0**	2	2333
47	18	22	0.0909	0.1000	0	1	720
21	13	17	0.1176	0.1333	0	5	7194
10	9	15	0.2000	0.2500	0	1	3836
7	7	7	0.0000	0.0000	0	5	4823
6	5	25	**0.4000**	**0.6667**	0	3	5870
9	4	16	**0.3750**	**0.6000**	0	5	2519
13	4	11	0.2727	**0.4286**	0	2	5876
12	4	6	0.1667	0.2000	0	2	2763
26	3	3	0.0000	0.0000	0	6	10147
22	2	7	0.2857	**0.5000**	0	5	4398
30	2	6	**0.3333**	**0.5000**	0	3	7111
11	2	4	0.2500	0.3333	0	3	7627
25	2	2	0.0000	0.0000	**1**	2	1810
2	1	11	**0.4545**	**0.8333**	0	2	2600
36	1	5	**0.4000**	**0.6667**	**1**	2	2333
4	1	3	**0.3333**	**0.5000**	0	1	7217
16	1	3	**0.3333**	**0.5000**	0	**8**	3674
1	1	1	0.0000	0.0000	0	1	**110840**
3	1	1	0.0000	0.0000	0	3	32801
5	1	1	0.0000	0.0000	0	2	17718
17	1	1	0.0000	0.0000	0	11	15255
8	0	4	**0.5000**	**1.0000**	0	3	17493
23	0	4	**0.5000**	**1.0000**	0	2	10513
34	0	4	**0.5000**	**1.0000**	0	5	4871
33	0	4	**0.5000**	**1.0000**	0	**8**	2884
19	0	2	**0.5000**	**1.0000**	0	1	17740
39	0	2	**0.5000**	**1.0000**	0	6	2254
24	**-1**	5	**0.6000**	**0.6667**	0	3	3846
27	**-1**	3	**0.6667**	**0.5000**	0	1	6905
29	**-1**	3	**0.6667**	**0.5000**	0	3	5585
31	**-1**	3	**0.6667**	**0.5000**	0	2	2912
37	**-1**	3	**0.6667**	**0.5000**	**1**	3	715
14	**-1**	1	**1.0000**	0.0000	0	1	2094
28	**-1**	1	**1.0000**	0.0000	0	3	1237
41	**-1**	1	**1.0000**	0.0000	0	5	446
38	**-2**	4	**0.7500**	0.3333	**0**	2	1014
40	**-3**	5	**0.8000**	0.2500	0	3	4100

Table 4.4-2. showing back-transformed cluster medoids for the optimal Links solution with 47 clusters.

Sub-Reddit type has been coloured to replicate the colour scheme of the Sub-Reddits table. Red has been used to indicate link types which receive a lot of negative votes **(> 40%)** or have a high proportional controversy **(> 50%)**. For most clusters, there is a positive correlation between negative votes and proportional controversy (with more negative votes equating to more controversy); there is however an interesting group of links at the bottom of the table which received many more negative than positive votes and therefore are unpopular but not very controversial.

The most interesting link types are those at the top of the table; the top six link types by aggregate score will all have stood a good chance of making it unto the Reddit front page. Taking cluster 43 for example; these links received over 250 votes but a lot of these were negative, resulting in a final aggregate score that was quite low. It is possible however that these links received a lot of positive votes in quick succession from readers of the Sub-Reddit it was submitted to; propelling the link unto the front page, where it proceeded to attract a lot of negative voting from the wider Reddit user-base. Clustering cannot reveal the presence of patterns like this directly; latent trajectory analysis will therefore be used to look for patterns such as this later in the research. The five link types above cluster 43 all had an excellent chance of being listed on the site's front page or the front page of a large Sub-Reddit; and therefore can be considered to be relatively successful links

There is an interesting group of three link types near the bottom of **table 4.4-2**; types 14, 28 and 41 have a total votes medoid of 1 but an aggregate score of -1. This pattern can only come about when a link submitting user changes the positive vote automatically generated for their link to a negative one; presumably these users changed their mind about the link they had submitted. There are some other slight inconsistencies in the medoids of link types. For example, link type 45 has medoid total votes of 27, proportion negative votes of 0, but an aggregate score of 26; this is due to one of the voters on this link changing their mind about their vote and nullifying it. The other inconsistencies in medoids are also due to null votes. As noted previously this type of vote is very rare so it has not been dealt with directly in clustering (it is considered indirectly by the combination of total votes, negative votes and aggregate score).

Five of the top six link-types (including the link type with highest total votes) had medoids suggesting they came from Sub-Reddit type three; these were Sub-Reddits which received quite a few links and where on average the links did relatively well in attracting votes. The links with the top medoid aggregate score tended to come from this type of Sub-Reddit, but they tended to receive a much lower proportion of negative votes than the average for links in this type of Sub-Reddit. At the other end of the table there are a lot of link types associated with this type of Sub-Reddit that didn't do so well; there are seven clusters representing links types with medoid SR type 3 and a medoid aggregate score of 1 or less. These clusters have about 48,000 members, suggesting that even links submitted to Sub-Reddits that tend to produce popular links have a very low chance of becoming one of these popular links. It would seem that there are no areas of Reddit where one can guarantee increased success of one's submissions; there may be Sub-Reddits where a link is more likely to be reviewed by other users (e.g. SR type 3) but voting activity seems largely determined by the qualities of the link.

If SR type 3 represents a type of Sub-Reddit where a link is relatively likely to receive at least some voting activity; then the general Sub-Reddit can be thought of as a place where a link is very unlikely to be seen or voted on by other users. Link type **1** represents links submitted to the general Sub-Reddit which received no votes aside from that which was automatically generated; this cluster has **110,840** members. There is also a type of link which is likely to come from the general Sub-Reddit and do well enough to potentially make the front page (link type 42); but this cluster has just 322 members. There are also a considerable number of links (about 20,000) submitted to small Sub-Reddits which only attract one or two votes.

Sub-Reddit types 2 and 5 are both associated with a few link types which received a moderate amount of voting activity. SR type 2 represents sub-reddits which saw quite a few link submissions but had relatively few votes; some of the link types related to this cluster have moderate aggregate scores, and it is likely that some links which performed well came from this type of Sub-Reddit, but have been placed instead in a link cluster with SR type 3.

SR type 5 represents 191 smaller Sub-Reddits which are highly unlikely to be on the default Sub-Reddits list. This SR type is related to a few moderately active link types, but there are reasons to believe that this should be interpreted differently to the moderately active links from SR types 2 or 3. Smaller Sub-Reddits often cater to niche or specialised topics; for example the Maths sub-reddit has about 12,000 subscribers and it is not uncommon for a link with an aggregate of 5 or even less to appear on the front page for this Sub-Reddit. Therefore a link submitted to this type of sub-reddit which received an aggregate of +13 (i.e. **link type 21**) is likely to have reached its target audience (indeed likely to have been quite popular among this group). On the other hand, links submitted to one of the default Sub-Reddits with this kind of aggregate (e.g. **link types 6 and 20**) have likely only been seen (and probably rejected) by a handful of users who devote a lot of time to reviewing links from that particular Sub-Reddit.

Analysis of the proportion of Negative votes and Controversy medoids for links submitted to the general or default sub-reddits reveals an interesting trend. There are five large clusters of links representing links submitted to these Sub-Reddits that only have one vote (i.e. the automatically generated vote from their submitter.) There are 21 link types from the largest SR types (1, 2 & 3) with a medoid total votes of between 2 and 25; of these, 16 have a **proportion negative votes or controversy of greater than 0.4.** In contrast, of the 10 link clusters representing links with a medoid of more than 25 total votes (all associated with SR types 1-3), only two have a **proportion negative votes or controversy of greater than 0.4**

This is suggestive of three stages to the voting lifetime of a link; the first barrier to link success is attracting a second vote from another user (many links don't manage this). At this stage the direction of the next 2-25 votes seems to be critical; if these are largely negative or mixed the link will have little success and is unlikely to receive many more votes. Presumably the links which receive mostly

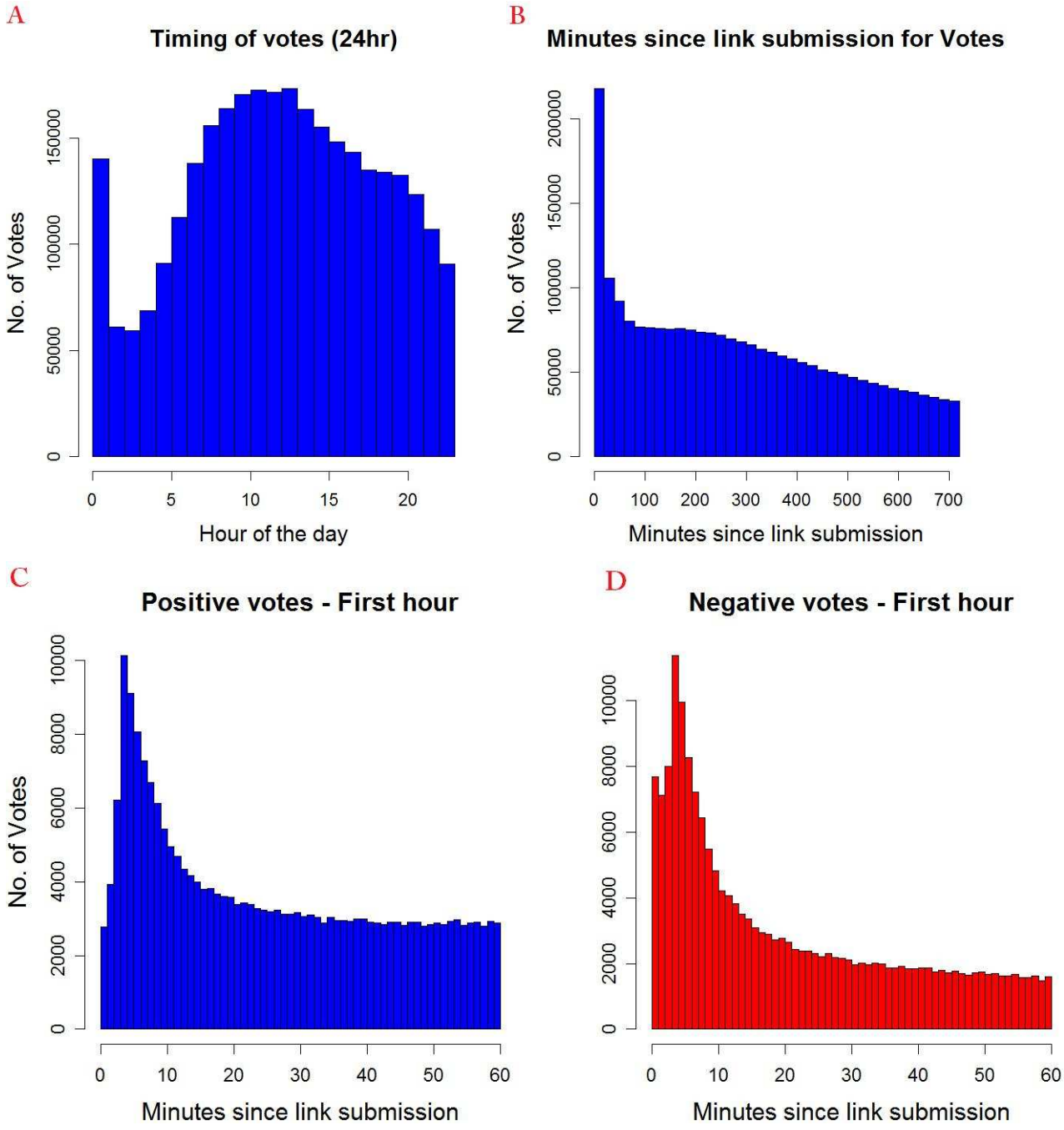
positive votes in this initial period go on to receive a lot more voting activity and some of these go on to become high aggregate successful links. Clustering cannot tell us whether this hypothesis about stages in link voting is correct but it certainly raises the possibility.

In order to get a rough idea of whether this relationship existed and how strong it might be; a data-set containing links with at least two votes was created and the proportion of negative votes in the first 2 - 25 (votes with order 1 were excluded because these are always positive) was calculated for each link. A generalised linear model of the poisson family (with log link) was fitted to see whether total votes could be predicted by the proportion of negative votes in the first 2-25. This model suggests a highly significant effect of proportion negative in votes 2-25 - the main effect of proportion negative suggests that links receiving a lot of negative votes at this stage receive on average only 20% of the total votes a Link receiving mostly positive votes at this stage can expect to receive.

The Self link variable seems to have been the least useful in determining clusters; only three clusters have a medoid of 1 on this binary variable (**types 25, 36 & 37, highlighted in yellow**) and these are all quite small clusters representing low activity links. The three link types with a Self medoid of 1 actually represent groups of links which are 100% self; while most of the other clusters contain at least a small proportion of self links (proportion Self for clusters with a medoid of more than 10% are highlighted in grey). For example, the cluster representing the most active and popular links (type 44) contains 11.7% Self links. When we consider that less than 4% of all links submitted in March were Self, these links seem to do relatively well (the 6 most popular/active link types all contain more than 4% Self submissions). That popular Self links do not have their own clusters in the above solution suggests that there is not enough of a difference between them and non-Self links on the five other measures given priority in forming clusters.
We now move to consider whether there are temporal patterns to the voting on links which might help to distinguish between different types.

4.5 Temporal patterns to voting on links

Exploratory analyses were conducted on both absolute and relative temporal aspects of voting. First, plots were produced showing a breakdown of voting activity at different times of day (**Figure 4.5-1A below**) and on different days. These exploratory analyses suggested that Reddit received quite a lot of voting activity throughout the day, with the lowest levels of activity seen between 1 and 5 am (co-ordinated Universal Time (UTC)).The site is also quite active throughout the month, but more so on week-days than at weekends.



Figures 4.5-1: showing (A) Frequency of voting by hour of day, (B) Frequency of voting by minutes since link submission (first 12 hours), and (C&D) showing positive/negative votes by minutes since link submission (first hour).

Looking at the timing of votes relative to the links they were cast on (**Figures 4.5-1B-D**) reveals a strong tendency for votes to be cast on links soon after they have been submitted. **Figure 4.5-1 B** suggests that many votes are cast within 15-30 minutes of link submission, with voting activity tending to slowly tail off after that for the next 12 hours and beyond. **Figures 4.5-1 C and D** above show a more detailed breakdown of positive (**C**) and negative (**D**) votes cast within the first hour of a link's submission. There seems to be a peak of activity for both positive and negative voting at around 8-10 minutes after link submission. It is interesting to note however that patterns of positive/negative voting are quite different before and after this peak. A lot of negative votes are cast within the first five minutes after link submission, but the number of negative votes tails off dramatically after the 8-10 minute peak. For positive votes there is a lull in activity for the first five minutes before building to the

8-10 minute peak period; after this peak, positive votes do tail off but they seem to stabilise at a rate of about 3500 per minute for the rest of the first hour (and beyond).

This fits with the idea that the first 2-25 votes are critical to a links' success; it would seem that in the first 10 minutes following a link's submission it is slightly more likely to attract negative than positive votes. For links which are still receiving votes after this period their chances of each vote being positive seem to improve with time. This likely reflects the workings of the site's democratic system; if the link receives mostly negative votes at the start of its life it probably won't be receiving any votes after an hour. It seems that this first hour (especially the first 10 minutes) of voting is critical in determining which links will receive no votes, which will receive a negative aggregate, and which will go on to receive many votes. This suggests that, for most of the content submitted to the site, the decision about whether it will be popular or not could be made within its first 10 minutes to an hour.

We will now consider the sub-set of Links which make it through this initial reception phase with a strong aggregate; and which can therefore go on to receive a lot of voting activity. Links belonging to types 44, 42, 35, 46, 32 and 43 (i.e. types with a medoid total votes of greater than 200) were extracted from the Links database and placed in a new data-table. Five new variables were added to this data-table (*generate_5_per_negs_for_links.py*-**D15**); these represented the proportion of negative votes received by the link in 20% increments of the vote order variable. These link types were chosen because they have enough votes that even when these are split into 20% bins each bin will still be large enough to generate a reliable proportion negative (i.e. 40+ cases). A Latent Trajectory model was then fitted to this data to determine whether there are any patterns to the proportion of negative votes received by successful links over their voting lifetime. Clustering suggests that even the most popular link types will receive between 15-25% negative votes; latent trajectory analysis should determine whether links tend to attract a steady rate of negative votes over their lifetime or whether this rate fluctuates.

Latent Trajectory models were fitted (similar to those employed by Nagin, 1999) with the software package Latent Gold 4.5. Solutions with between 1 and 20 latent classes were fitted on the five (proportion negative) indicator variables; with link type included as a co-variate. The fit of these solutions was measured with BIC (Bayes Information Criterion), which is based on log-likelihood penalised for reduced degrees of freedom. The solution with the lowest BIC was chosen for interpretation.

This solution involved fitting 12 latent classes to the data, and produced a BIC of -46576; **Figure 4.5-2** below shows latent trajectory class means for each of these. The largest 10 of the 12 latent classes fitted to this data represent quite a steady proportion of negative votes throughout the lifetime of links assigned to them. This steady proportion of negative votes ranges from 5% to over 30% between these ten classes. There are two latent classes representing a steep increase in the proportion of negative votes during the lifetime of the link; but these are both quite small with case memberships of just under 1% of the total.

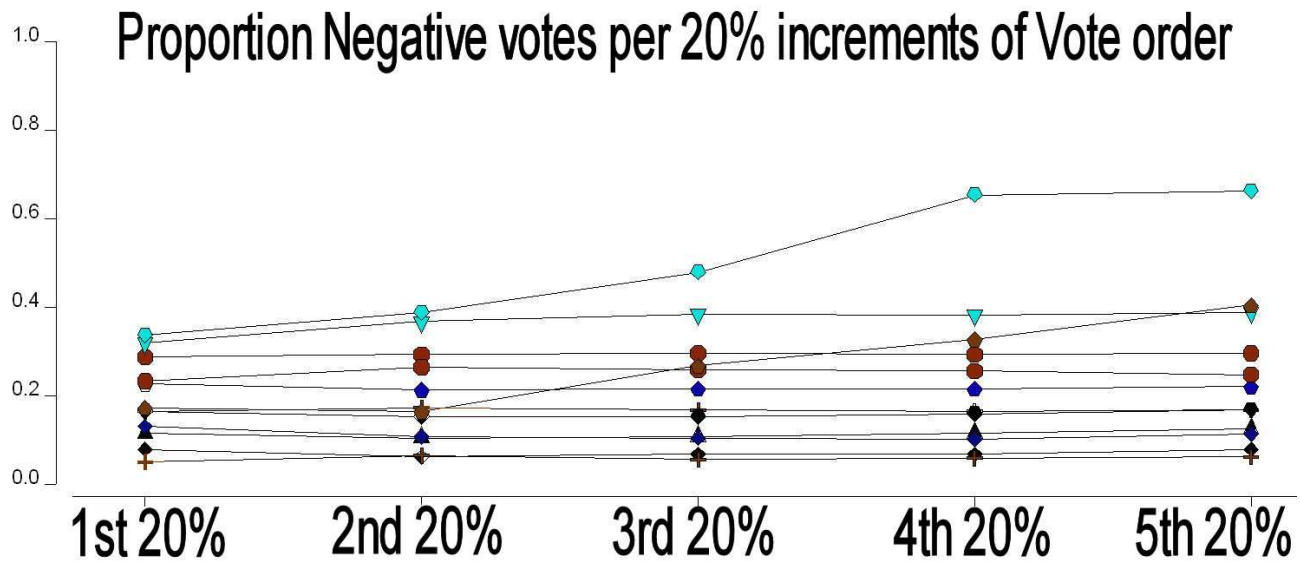


Figure 4.5-2. showing cluster means for latent trajectory analysis of proportion negative votes.

Link type 43, representing controversial links with quite a few votes (medoid total 203), has been very well accounted for by these latent classes, 93% of its members belong (most probably) to the latent class with the largest consistent % negative votes; with the other 7% belonging to the class with the high and steadily increasing % negative votes. The latent classes with the highest proportion negative votes have, unsurprisingly, an extremely low probability of representing link type 44, the link type with the highest total votes medoid. The chance of a latent class being associated with link type 44 does not however increase consistently as we move down the range of steady negative proportions. For example; 22% of type 44 links are represented by latent class 4 with a steady negative proportion of about 20%. On the other hand less than 1% of type 44 links are associated with latent class 8, having a steady negative proportion of about 5%. It would seem that attracting overwhelmingly positive votes is not always enough to propel a link to a very large aggregate score. While some links see a rise in their proportion of negative votes over time it would seem that others see a simple cessation of voting activity (either way, the link's aggregate stops increasing and it will slip down the rankings).

There is one other small trend in these trajectories which may be noteworthy; most of the types have a proportion negative for the first 20% of votes which is slightly lower than at later stages. This fits with the idea that the first 25 or so votes are very important in establishing a link's presence; a better ratio of positive votes at this stage than they would receive later in the process might have helped these links through the initial phase. The latent classes with a higher chance of representing type 44 links actually had a small trend in the opposite direction; they had slightly more negative votes in the first 20% than at later stages. This small trend suggests that although the initial phase is important in establishing a link, the link's ultimate success will be determined by the nature of voting activity in the longer term. In terms of how the site actually works; links with a smaller proportion negative votes in the first 20% are probably those which did slightly better on a Sub-Reddit, than once they reached the front page. The links which are the most active and popular (i.e. type 44) would therefore seem to do slightly better than they did in the Sub-Reddit once they reach the main page.

A second latent trajectory model was fitted to Links data; this time looking at the level of voting activity on popular links (i.e. the same Link types considered above). 24 variables were created; each representing the proportion of a link's votes which were cast in a given hour after submission (*generate_24_hour_votes_for_links.py*-**D16**). Latent trajectory classes were fitted to these 24 hour variables; the BIC criterion suggested the optimal number of classes was 25 (BIC = -477316). A profile plot for these classes is included in **Appendix C-2.** As suggested by the large number of latent classes - there is considerable variation in the patterns of activity seen between different links. Most of the latent classes show a peak period of voting activity lasting between 2 and 5 hours; but there is a lot of variation in where this peak occurs within the first 24 hours and how pronounced it is. The classes which are more likely to have very popular links as members have quite a small peak and instead their voting activity is sustained at a higher level throughout the 24 hour period. There are some classes representing links that get off to a slow start and don't peak until 12-16 hours after they were submitted; and there are others which receive most of their voting activity in the first 5 hours and by the 12th hour have a very low level of activity or none at all (more likely to be smaller link types).

5 Relationships between User types and Link types

Row-column association models (Goodman, 1979) will be used to investigate the relationships between User types and Link types. Cross-tabulations of link submissions will be created such that each cell represents the number of times users of a given type submitted each type of link (a cross-table of votes will be created separately). An independence model will then be fitted to the data in this table to first get an idea of whether there is a relationship between these factors. RC models will then be fitted to determine which cells have a higher or lower frequency count than would be expected if independence were true. The simplest form of RC model (RC1 model) contains one multiplicative factor which can be used to assess the relationship between user types and link types on one dimension.

5.1 Which users submit which links?

At this stage a potential problem with using these clusters with link types was noticed; most of the clusters with a proportional link submission medoid of 0 actually had some members who did submit links. This means that if we simply ignore user types with a link submission percentage of zero when considering link types; we will not be able to include the links they submitted in the analyses. If we were to include all User types in the following analyses, the user types with a link submission medoid of zero would have their relationship with different link types assessed on a very small number of cases; this is likely to reduce the reliability of results generated by these analyses.

5.2 Clustering Users who submit links

For this reason; it was decided to cluster the sub-set of users who submitted at least one link separately to users with no link submissions. The same criteria used to cluster all Users were applied to just the sub-set of users who made at least one link submission; average silhouette width suggested that the optimal number of clusters for this data-set was 60. Cluster medoids for these 60 user types who submitted at least one link can be found in **Table 5.2-1** below.

Cluster	Total Votes	ID age	Voting Order	% Negative	% link sub	% self votes	Cluster N
25	1 vote	5	1	0	1	0	**14018**
27	1 vote	4	1	0	1	0	5762
14	1 vote	3	1	0	1	0	1962
9	1 vote	**2**	1	0	1	0	1433
13	1 vote	**1**	1	0	1	0	1072
55	1 vote	5	1	0	1	1	480
20	1 vote	4	1	0	1	1	179
44	2 - 5 votes	5	1	0	1	0	3412
40	2 - 5 votes	4	1	0	1	0	2838
15	2 - 5 votes	3	1	0	1	0	1643
12	2 - 5 votes	**2**	1	0	1	0	1049
23	2 - 5 votes	**1**	1	0	1	0	805
11	2 - 5 votes	**2**	4	0	0.25	0	594
22	2 - 5 votes	**1**	3	0	0.4	0	557
47	2 - 5 votes	4	**2**	0	0.75	0	477
36	2 - 5 votes	**2**	**2**	0	0.3333	0.3333	476
50	2 - 5 votes	4	4	0.25	0.5	0.25	451
49	2 - 5 votes	5	1	0	1	0	249
43	2 - 5 votes	4	5	0	1	0	192
60	2 - 5 votes	5	5	0	0.5	0	107
53	6 - 10 votes	5	1	0	1	0	928
52	6 - 10 votes	4	1	0	1	0	843
34	6 - 10 votes	3	1	0	1	0	663
31	6 - 10 votes	**1**	4	0.3	0.2	0.1	423
42	6 - 10 votes	4	4	0	0.2857	0.2857	409
7	6 - 10 votes	**2**	4	0.4	0.1	0.1	353
32	6 - 10 votes	**2**	1	0	1	0	324
17	6 - 10 votes	**1**	3	0	0.5	0.3333	287
45	6 - 10 votes	**2**	3	0.4444	0.1111	0.2222	252
57	6 - 10 votes	5	**2**	0	0.1429	0	222
59	6 - 10 votes	5	4	0	0.8571	0	102
1	11 - 25 votes	3	1	0	1	0	1205
29	11 - 25 votes	**1**	4	0.3043	0.087	0.087	1034
2	11 - 25 votes	3	4	0.0556	0.1111	0.2222	1011
18	11 - 25 votes	**2**	4	0.1429	0.2143	0.1429	797
4	11 - 25 votes	**2**	3	0.1667	0.25	0	621
26	11 - 25 votes	**2**	1	0	1	0	444
35	11 - 25 votes	3	3	0.3333	0.1667	0.3333	392
58	11 - 25 votes	4	1	0	1	0	364
51	11 - 25 votes	5	4	0	0.5385	0	357
30	11 - 25 votes	**2**	5	0	0.4545	0.3636	342
54	11 - 25 votes	3	1	0	0.1818	0	40
19	26 - 50 votes	3	4	0.2143	0.2143	0.1071	1227
46	26 - 50 votes	4	1	0	1	0	397
3	26 - 50 votes	**2**	1	0	1	0	305
41	26 - 50 votes	3	1	0	1	0	288
5	51 - 100 votes	**1**	4	0.25	0.0192	0.1538	1358
38	51 - 100 votes	**2**	3	0.0862	0.0517	0.0172	913
8	51 - 100 votes	4	4	0.1806	0.0833	0.3194	574
28	51 - 100 votes	4	1	0	1	0	402
6	101 - 200 votes	3	4	0.1639	0.1475	0.1885	847
39	101 - 200 votes	**2**	4	0.2832	0.0173	0.0347	688
33	101 - 200 votes	**1**	4	0.0841	0.0374	0.1121	670
56	101 - 200 votes	4	4	0.0141	0.6056	0.0423	230
16	201 - 500 votes	**1**	4	0.1152	0.0037	0.0558	763
10	201 - 500 votes	**2**	4	0.1418	0.0073	0.1527	681

37	201 - 500 votes	**2**	3	0.0393	0.0319	0.059	453
24	201 - 500 votes	4	4	0.018	0.006	0.0631	320
48	201 - 500 votes	5	3	0.2189	0.2747	0.0773	98
21	201 - 500 votes	5	4	0.2747	0.0129	0.0815	37

Table 5.2-1. showing 60 cluster medoids for Users who submitted at least one link – ordered by level of activity then cluster size. The same colour scheme used with the previous clustering solution has been applied to this data.

Many of the user types in **table 5.2-1** above are recognisable from the previous Users clustering solution. There are still a lot of User clusters with **a link submission proportion of 100%**; and again the proportion of users belonging to these clusters decreases as we move down the table into clusters representing more active users. There are now two small clusters (types 55 and 20) representing users with 1 link submission which was a Self post; these have account age medoids of 4 and 5 suggesting that people with this pattern of activity are more likely to have registered their account recently. There is also now a type of user (type 13) with the oldest account age and **100% link submissions** but this group is quite small. Looking at the performance of links submitted by these three user types relative to the other user types with just one vote could help to identify an effect of "community involvement" previously hypothesised. There are obviously no longer any user types with a proportion of link submissions equalling zero; there are however quite a few clusters with a very low proportion of link submissions. Presumably many of the users in these clusters with a very low proportion of link submissions had previously been incorrectly assigned to clusters with link submission medoids of zero. There is no longer a User type with medoid total votes of 500+, but there are now six user-types which have been placed in clusters with a 201 – 500 total votes medoid. Of these six; the larger clusters have older account age medoids and a very low link submission proportion.

5.3 Link submitting Users – which users submit which links?

A cross-tabulation of User and Link types was created such that each cell represented the number of links of a given type which were submitted by a given type of user. An independence model was fitted to this table; if the deviance for this model is low it suggests no relationship between user types and link types. Deviance for this model was however very high (152,000 on 2,714 degrees of freedom); providing strong evidence against an independence relationship between user and link types.

A Row-Column (RC) model with 1 multiplicative factor was then fitted to this data table using the GNM package in R. Deviance for this model was 26,428 on 2,610 degrees of freedom, suggesting that it does not offer a significantly good fit with the data. This model does however reduce the deviance by 125,571 with a loss of just 104 degrees of freedom; a dramatic improvement over the independence model. Theoretically, the number of multiplicative factors could be increased to improve this fit; but with a data-set this large an alternative hardware and/or software platform would be required to fit these models. More multiplicative factors would also make interpretation of relationships a lot more difficult.

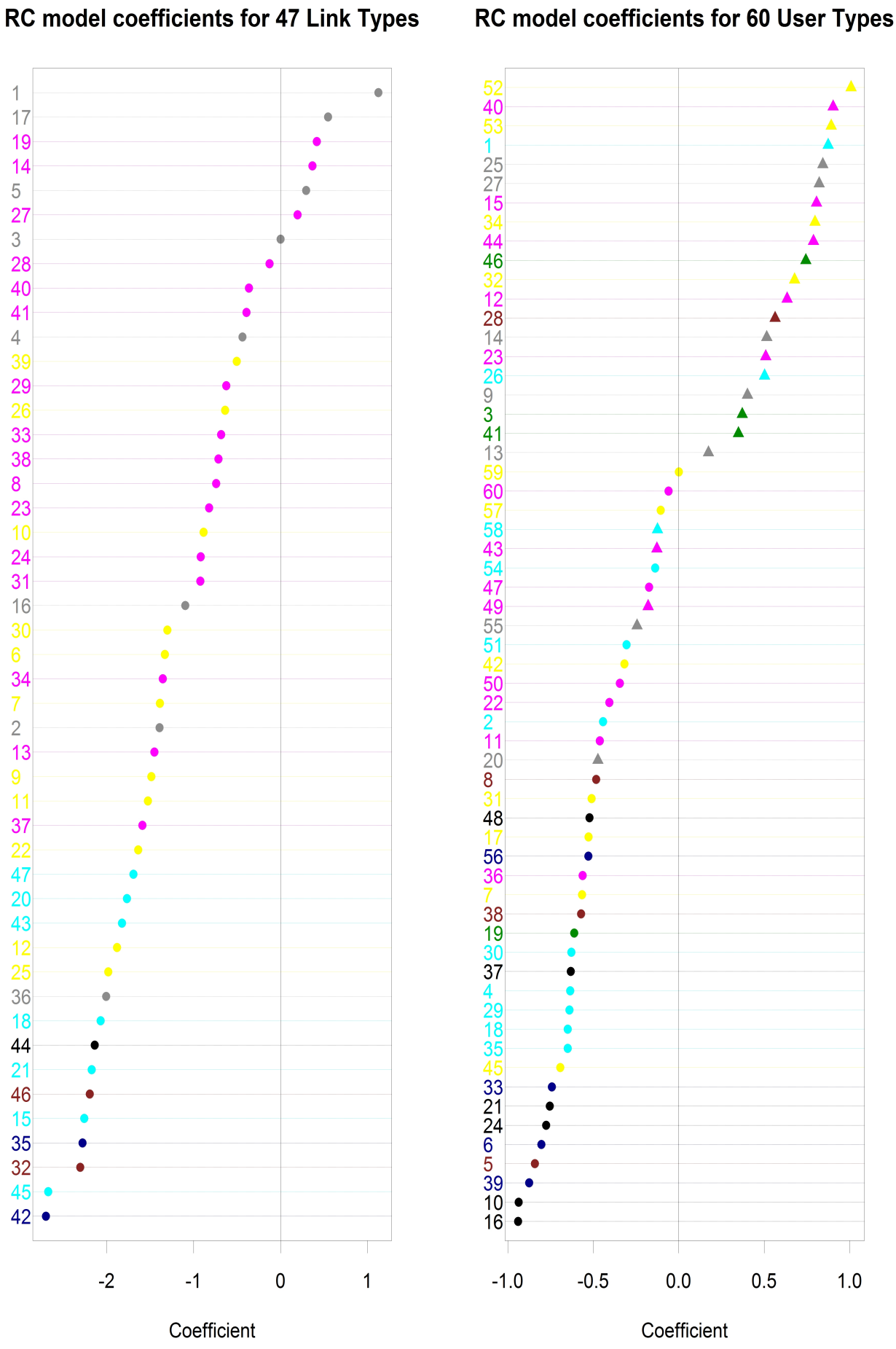


Figure 5.3-1. showing RC coefficients for Links and submitting User types.

With just one multiplicative factor the coefficients produced by our RC model are quite easy to interpret. The key to this interpretation is whether a link or user type's multiplicative coefficient is positive or negative. If a User type's coefficient is positive: it is associated with a greater than expected number of submissions of Link types with positive coefficients; and with a lower than expected number of submissions of Link types with negative coefficients. The converse is true for User types with a negative multiplicative coefficient. Therefore the strongest relationships in this data will be between the User and Link types with the largest positive coefficients; or between the User and Link types with the largest negative coefficients.

Figure 5.3-1 above shows the multiplicative coefficients for each of the 47 link types and 60 user types considered in the RC 1 model. Link and User types have been coloured in accordance with their level of activity in the tables of medoids above. Triangles have been used to represent **users who only submitted links**.

Let us first consider link type coefficients. The link types with positive coefficients all have an aggregate of 1 or less than 1. The link type with the largest positive coefficient is link type 1 (coefficient +1.13); this is the largest link type, representing 110,840 submissions to the general sub-reddit which only received one vote. This indicates that the positive end of the dimension fitted to link types by the RC model represents unsuccessful links.

At the other end of this dimension; the link types with the largest negative coefficients all represent links with a relatively large aggregate score. Nine link types have a negative coefficient stronger than -2, these include the five link types with an aggregate score greater than 100. The link type with the largest negative coefficient is link type 44, with a coefficient of **-2.14.** The relationship between negative coefficients and link aggregates is however not as straightforward as that between positive coefficients and low-scoring links. The link type with the second largest negative coefficient is link type 45; representing links with a medoid aggregate of just 18. Indeed, links with a moderate aggregate of between 10 and 100 are quite common at this end of the scale. The positioning of link type 44 on this dimension is also quite interesting; it is placed towards the "popular" (i.e. negative) end of the scale but the model makes stronger predictions about seven other link types with a lower aggregate score. This suggests that there is more unexplained variation in the types of user which submit this most popular of link types. This could be due to increased importance of link qualities (which aren't considered here) in determining whether a link will be "extremely popular" relative to links which are "very popular".

For the initial interpretation of User type coefficients we can consider positive coefficients to be related to links with one vote or a negative aggregate. We can also consider negative coefficients broadly related to popular links with a large aggregate. It is clear from first glance that the user types with positive coefficients are those who **only submitted links**. There are twenty of these **link submitting user types** with positive coefficients; when links were submitted by users from these types

they were much more likely to receive no votes or a negative aggregate than to receive a good aggregate score. At the other end of the User types dimension, there are very few link submitters. Of the 20 user types with a negative coefficient stronger than -0.5; not one represents users who only submit links.

The two user types with the largest negative coefficients (i.e. the two user types who were most likely to submit successful links) both represent very active users with more than 200 votes in March. The eight user types with the strongest negative coefficients all have medoid total votes of greater than 50. Level of activity certainly seems to be important in determining which users are the most likely to submit the more popular links; but there are obviously other factors effecting this relationship. Not all of the very active users have strong coefficients (although they are all negative).

Also, when we look at more moderate negative coefficients (i.e. -0.6 to -0.4) it is clear that some of the less active user types are associated with a relatively high likelihood of submitting successful links. For example, User type 45 has a remarkably strong relationship with successful links for a group of users with medoid 6-10 votes. Similarly, there are five user types with 11-25 medoid total votes who have a more negative coefficient than the most negative coefficient for user types with 26-50 votes (type 19).

Level of user activity and proportion of link submissions, two of the variables hypothesised to relate to "community involvement", seem to be quite strongly related to the success of submitted links (particularly at the extremes of the dimension fitted by the RC model). This supports the idea that "community involvement" can be inferred from some of the variables used for clustering; in that the links submitted by users who are more "involved" in the community are more likely to be successful. We will now consider whether the other two variables thought to relate to community involvement (Account Age and Proportion of Self activity) can explain patterns in these RC coefficients not accounted for by level of activity and proportion of link submissions.

Let us first consider the relationship between account age and RC coefficients; of the 15 user types with the strongest negative coefficients 12 have account age medoids which put them in the oldest 40% of users. At the other end of this dimension; 7 of the 20 user types with a positive coefficient have an account age medoid in the oldest 40%. This suggests that users with older accounts may generally be more likely to submit links which are successful; but when these users only submit links - they are still likely to submit unsuccessful links. This supports the idea that account age might be related to community involvement, but suggests that it is a less important component than level of activity or proportion of link submissions.

With regards to users' proportion of Self activity; we can first examine the coefficients of the two user types noted above (types 55 and 20) with 1 link submission that was a Self link. These user types have negative coefficients, which is unusual for users who only submit links. In fact; of all the user types who only submit links, these two are the most likely to submit a popular link. Aside from this, there is

very little evidence of a Self effect in the relationship between User types and the performance of the links they submit. Proportion of Self activity was not given much weight in the clustering process for users or links; therefore it is not surprising that its effects are hard to find in the RC coefficients for Link and User types. The relatively strong relationship between user types 55 and 20 and successful links hints at a relationship between Self activity and community involvement. However, based on RC coefficients of the link and user clusters, we would have to conclude that any effect of Self is weaker than the other three aspects of community involvement (at least when considering whether a User will submit successful or unsuccessful links).

There are certainly patterns to the relationship between User types and the types of Link they submit on the dimension fitted by the above RC model. The hypothesised community involvement concept offers a way of accounting for some of these patterns. There are however some user types who have a higher likelihood of submitting popular links than their "community involvement" variables would suggest. The next step in the analysis was to see whether similar patterns existed in the voting relationships between user and link types.

5.4 Do different User types tend to vote on different Link types?

A cross-tabulation of User and Link types was created such that each cell represented the number of users of a given type who voted on each type of link. An independence model was fitted to this table; deviance for this model was very high (1,270,000 on 2,679 degrees of freedom); providing strong evidence against an independence relationship between user and link types. A Row-Column (RC) model with 1 multiplicative factor was then fitted to this data table. Deviance for this model was 83,206 on 2,576 degrees of freedom, suggesting that it does not offer a significantly good fit with the data. This model does however reduce the deviance by 1,186,794 with a loss of just 103 degrees of freedom; a dramatic improvement over the independence model. Coefficients for this RC1 model are included in **Figure 5.4-1** above.

The dimension fitted to Link types by this RC model seems strongly influenced by aggregate score. Positive coefficients are related to links which have a high medoid aggregate score; the strongest coefficient (+6.24) belonging to link type 44, a cluster of links with medoid aggregate of 636. This link type's coefficient is considerably larger than the next largest coefficient. Link types 42 and 35 have coefficients of 4.5 and 4.3 respectively. On the negative side of this dimension are link types with very low numbers of votes. Closer examination of these coefficients suggests that total number of votes might have a stronger effect on this dimension than the aggregate score.

It makes sense that fitting an RC model to voting activity will produce a Links dimension which prioritises the level of voting activity. In the previous table dealing with link submissions, popular links had generally low cell counts because they were rare in the links data-set. For the voting data this trend has been turned on its head; most of the voting activity on the site is focused on popular links, so link types with this characteristic tend to have very high cell counts and might therefore dominate the RC dimension generated.

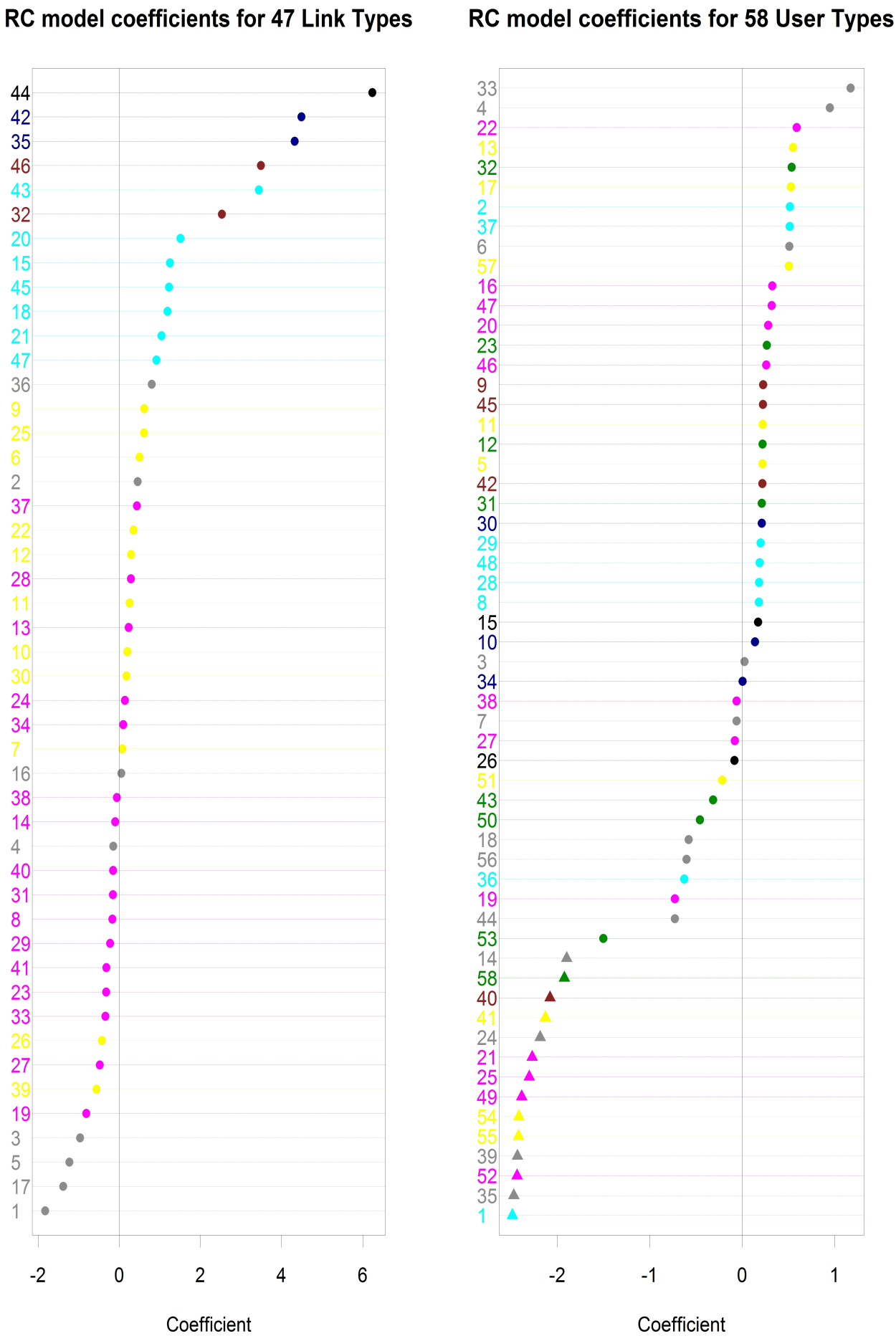


Figure 5.4-1: showing voting RC model coefficients for Link and User types.

Given the relationship between **users with a 100% link submission rate** and unsuccessful links established above in the analysis of link submissions; we would expect to see a lot of "**link submitters**" having negative coefficients in the Votes RC model. If a user only submits links, and these links tend to be unsuccessful, then when we consider voting behaviour these users should have a high likelihood of voting on unsuccessful links. The portion of **figure 5.4-1** showing user type coefficients reveals exactly this kind of pattern. There are a group of 100% link submission users who all have strong negative coefficients (all < -1.9). User type 53 also has a negative coefficient approaching this magnitude; this type represents users with between 26 - 50 votes, just over half of these being link submissions.

There is then a considerable gap to the coefficient of the user type with the next strongest tendency to vote on ultimately unsuccessful links (user type 44, coefficient -0.73). There are a group of user types with similar coefficients here (44, 19, 36, 56); consulting the User type medoids table reveals that these are the user types with an **average voting order of between 2 and 10**. This is the same group which it was hypothesises might hold a disproportionate amount of influence on which links are successful. Users of these types seem most likely to vote on links which do not go on to become popular. Where these votes were negative this could be considered a positive outcome for the user. We will examine the interaction between voting order and the direction of votes below to get a better idea of how much influence the votes of these (and other) user types have on the Link's eventual outcome.

At the other end of the Users dimension; User types 33 and 4 had the strongest positive coefficients (+0.75 and +0.64 respectively), making them the most likely to vote on popular links. User type 33 represents users who voted once positively on an already established link; while user type 4 represents users who voted once negatively on a link with 11-100 previous votes. There are eight user types with a coefficient of around 0.55, all of these representing users who didn't submit any links. There are 30 user types in total with positive coefficients, and 29 of these have a link submission rate of 0. This suggests that users who only vote are more likely to vote on links which (ultimately) have a high aggregate.

For many of the User types considered this relationship is quite weak. In this analysis however, the users with no strong tendency to vote for successful or unsuccessful links are potentially the most interesting types. The users who are most likely to vote on links with a low aggregate are link submitters. The users who are most likely to vote on high aggregate links, judging by average vote order medoids, seem to confine their voting activity to the front page and other areas where prominent content is located (e.g. Sub-Reddit front pages).

This leaves a group of user types in the middle ground with no strong tendency to vote on a particular kind of link. This group includes the five most active user types, who have small coefficients ranging from -0.1 to +0.2. This indicates that users of these types probably tend to vote on a variety of different link types. Three of these types have a link submission rate of less than 1%, so automatic votes on their own submissions cannot account for the lack of a strong positive coefficient. The most plausible way

to account for these User types having such a weak relationship to the links dimension is that these Users cast votes both on established links and also on fresh links submitted by other users.

This suggests that it could be the most active user types who have a disproportionate say in determining which links make it through the seemingly critical 2 – 25 votes period, rather than the **user types with vote order medoids of between 2 – 10** as previously hypothesised. These active users tend to be involved in all aspects of the democratic system on the site to at least some degree. Furthermore, when these active user types submit links these are the most likely links to become successful.

5.4.1 Effects of vote direction by User type

A linear model was fitted at this stage to check whether a linear relationship could be found which roughly quantified the relative influence of votes cast by different user types. This model took proportional vote order as its dependant variable; with user type, vote direction, and the interaction between these as independent variables. Proportional vote order was used because it reflects the order of a user's vote relative to all the other votes on that link. The main effect of proportional order for each user type will therefore reflect whether they tended to vote earlier or later on average in the voting lifespan of a link. The main effect of vote direction reflects whether this order increases or decreases as a function of vote direction (negative votes decrease aggregate so should therefore be associated with a reduction in the number of subsequent votes). The interaction between vote direction and user type is the most interesting parameter; this will give a measure of how much the user type's proportional vote order changes with the direction of their votes (i.e. a crude measure of vote influence).

This model was fitted on a random sample of 500,000 votes. In order to avoid peculiar cases skewing the results only votes cast by a sub-set of user types (those with a link submission rate of less than 100% and medoid total votes of greater than five) on a sub-set of link types (those with medoid total votes greater than 1) were considered. This model reduced deviance by 869 on 55 degrees of freedom, a significant improvement on the null model ($p < 0.001$). The intercept for the model is 0.56, suggesting that the average vote order for the votes being considered was roughly in the middle of the voting lifespan. The main effect of vote direction is -0.01, suggesting that a positive vote tends to be related to a smaller proportional order while a negative vote is related to a larger proportional order. This makes sense because we would expect a negative vote to decrease the probability of subsequent votes slightly (by lowering the link's aggregate), so when a negative vote is registered its proportional order is likely to be larger.

Of the 27 user types being considered here, 22 have vote direction interaction terms which are significantly different to zero; and these are all negative. The two user types with the strongest interaction terms are types 36 and 50; and these are the same two user types with an average voting order medoid of **between 2 and 10.** This suggests that the direction of these user's votes affects their proportional order by an average of 20%; 10% earlier than average for positive votes and 10% later for

negative votes; quite a strong effect.. The five most active user types all have significant interaction terms of between -0.03 and -0.04, so these users' votes seem to have slightly more influence on the link's total than the average user. As these users have a lot of votes; it seems likely that they cast some votes on relatively fresh links but their mean voting order has been increased by votes cast on established links.

The results of this linear model are somewhat limited, but give a rough indication that the effects of votes cast by different users are not necessarily equal. The five user types who do not have significant interaction terms all have total votes medoids of between 6 and 50; it would appear that the votes cast by these users have relatively little influence over the subsequent voting behaviour of other users. If we also considered the user types with less than 6 votes it is likely that many of these would also be found to have a low "voting influence"; these user types were not considered because they mostly represent users who cast votes in only one direction.

5.5 Influential Users

All of this points to a group of between 5000 – 7000 users who have quite a lot of influence in determining what the "hot" links on Reddit will be on any given day of the month. These user types tend to have variable medoids which indicate a high level of what has been termed "community involvement". The results of clustering and RC models so far suggest four reasons why this group of active users might hold a lot of influence on Reddit. **1:** They use their accounts a lot; **2:** They use all the voting/submission features of their accounts to at least some degree; **3:** They use their accounts in an influential way (e.g. sometimes voting early on other users' links); **4:** They tend to have older accounts, so are presumably familiar with the Reddit.com community (i.e. they know the types of link content which other users like and which are generally popular).

6 Discussion and Conclusions

Much of the activity on Reddit is characterised by the exponential distribution and conforms to the Power-law frequently associated with online communication. Reddit had 5,664,590 unique Visitors in March but only 102,232 active Users (as noted, these measures rely on IP addresses and User accounts respectively, so do not equate directly to individual people). If we take these figures literally this means that only about 2% of visitors to the site in March had any involvement in determining which content was displayed there. The activity level of Users who were active in March certainly follows the power-law (13.7% of Users cast 80% of votes); as does the level of Link submissions per User. The activity levels for Links also follow the same kind of exponential distribution.

These distributions suggest that for the millions of people who used Reddit in March, only a very small proportion of these were actively involved in determining what the site was displaying. Of the people who have accounts, their primary defining characteristic is how much they use their account (ranging from 1 vote to 23,776 votes in March). Another important characteristic of Users with accounts is which account features they choose to use; most users prioritise either Voting or Link submitting to a

large degree; often a User account will be used exclusively for voting (37% of Users) or submitting (42% of Users).

Users were clustered along a variety of criteria representing their level of activity, what this activity consisted of, and other available variables like the age of their account. These clusters revealed some interesting User types; and their medoids suggested a relationship between level of activity, proportion of link submissions, proportion of activity on Self links and account age. This relationship has been termed "community involvement". A group of user types with high levels of activity, a low proportion of link submissions, some Self activity, and older accounts, would seem to be those with the highest levels of community involvement. Users with the lowest levels of community involvement tend to only submit links (and none of these are Self links), and have a newer account age. These results answer the question of whether Users can be classified into different types based on how they use their accounts with a firm yes.

Sub-Reddits were clustered to see if they too could be classed as belonging to different types; and so that these types could be used to bolster the available explanatory variables for Links. Sub-Reddit clustering suggests that the Sub-Reddits (presumably) included on the default list can be divided into two types; those with more link submissions and those with a lower number of link submissions but more votes per link. There is no clear difference between the subjects of Sub-Reddits belonging to these types; both types contain Sub-Reddits covering a diverse range of topics. Clustering of Sub-Reddits did however bring to our attention the large number of Sub-Reddits representing niche areas of interest with a moderate level of activity; and closer inspection revealed that these operate on a vastly different scale to the default Sub-Reddits (i.e. an aggregate of 5 could be enough to see a link on a moderately sized Sub-Reddit's front page, but this would be much too low to approach the front page on a "default" Sub-Reddit). It is not uncommon for a Sub-Reddit of this moderate size to have more than 10,000 subscribers; suggesting that quite a lot of the Users spend at least some time browsing content on some of these smaller Sub-Reddits; content that a first-time visitor to the site would be highly unlikely to see.

It is interesting that although all of the content on Reddit is essentially public; there are still pockets of content (i.e. small Sub-Reddits) which are essentially hidden to those who haven't searched them out and/or subscribed to that source. This could potentially offer another indicator of community involvement. As these Sub-Reddits are not on the default list the links submitted here have virtually no chance of being seen by Visitors to the site, no matter how many positive votes they get. A user who votes a lot in these Sub-Reddits is therefore likely to have used the site for long enough that they have their own specific preferences about the type of content they want to be presented with. Furthermore, if they are actually voting in these more niche areas then they consider it worth their while to rate content even when its maximum potential audience is quite small, this would suggest a high level of community involvement.

An RC model was fitted ad hoc to see if a relationship like this between User types and Sub-Reddit types might exist in the data (graphical output for this model is included in **Appendix B**). This model placed Sub-Reddit types 1 and 2 on the positive extreme of the scale, with Sub-Reddit 3 and a lot of smaller Sub-Reddits at the negative extreme. Coefficients for User types suggest that the 14 user types who only submitted links were the most likely types to have voted on the biggest Sub-Reddits. At the other end of the scale for User types, the five most active user types all had negative coefficients; these coefficients are not very strong, so they imply that the votes of the most active users are distributed across both large and small Sub-Reddits. These results provide further support for the utility of a community involvement concept in accounting for the behaviour of Reddit users. This RC model also suggests that links from Sub-Reddits of type 3 are much more likely to be voted on by Users with high community involvement relative to those with lower community involvement.

Links were clustered to determine if they too could be classified as belonging to different types. The defining characteristics of Link types seem to be aggregate score and level of activity. Most of the Link types which received a lot of votes have quite low levels of negative voting and controversy and come from the main Reddit or one of the default Sub-Reddits. Looking at the pattern of Link type medoids suggested several stages to the life-span of a link submitted to the main Reddit or a large Sub-Reddit. The first obstacle which a link must overcome is getting a second vote from another User. If a Link receives this second vote then its direction and the direction of the subsequent 20 or so votes will be critical in determining the Link's ultimate success. If the link receives a lot of positive votes at this stage its aggregate will rise enough that it comes to the attention of more Users and therefore has a good chance of receiving a lot more votes. If the link receives a lot of negative votes at this stage its aggregate will not increase much (or may fall below zero) and this is likely to be the end of its voting lifespan.

Latent trajectories suggest that there may be subsequent stages to voting activity on Links. One of these stages begins at the point when a Link reaches a high enough aggregate to be displayed on one of the main pages; this seems to bring about a large increase in voting activity. The direction of these votes can either see the link quickly removed from this prominent position, or propelled upwards to much larger aggregate scores. The other stage revealed is only experienced by a small sub-set of Links - and is characterised by a steep increase in the Link's proportion of negative votes with time. There are a number of reasons why this might happen to a Link which is initially successful: The content of the Link may have been exposed as fraudulent or inaccurate by other Users; or the Link might have received a large enough aggregate to appear in a location where the majority of users found it to be inappropriate. What is clear is that although the principle of up/down voting is simple - when used by a whole community it results in a plethora of different ways in which a Link can be "received". In other words, this simple tool is quite powerful and adaptable in the hands of Reddit Users; when there are enough people voting on an item of content the desired outcome of the majority is likely what will come to pass.

The relationship between User types and Link types was assessed with separate RC models of Link submissions and Link voting. The RC model of link submissions suggests quite a strong trend whereby the links submitted by active user types (particularly those with high "community involvement") are much more likely to belong to one of the more popular Link types (i.e. they are well received). Users with low community involvement (in particular those who only submit Links to external sites) are more likely to submit a link which doesn't receive a second vote or which ends up with a negative aggregate score. This finding provides strong support for the existence of a factor which we have termed community involvement; the users who are most involved seem to have an advantage when they submit links. It is possible that these users' familiarity with the nuances of the Reddit.com community helps them to decide which content is worth submitting to the site and how this should be presented (i.e. title and Sub-Reddit).

The RC models of votes (and Sub-Reddits) suggest that these more involved users are also more likely to distribute their activity across different types of content (e.g. fresh and established links in large and moderate Sub-Reddits). Users with low community involvement are much more likely to be active largely in prominent areas of the site (i.e. the front pages of the main Reddit and default Sub-Reddits).

These trends in relation to community involvement suggest that there are a group of between 5000-7000 users who are highly involved in the community and who hold a disproportionate influence over which content will be displayed prominently on a given day. This suggests that one of the strategies for dealing with Information Overload and establishing oneself in the community which previous research has suggested (Himmelboim, 2008; responding primarily to content submitted by popular or prominent Users) is being employed to some degree on Reddit.com. These trends are however quite mild in relation to those found on older communication systems like Usenet or Bulletin Boards. There is no single type of extremely active/popular user with a vastly disproportionate level of influence; rather there seems to be a gentle almost linear relationship between Users' involvement and level of influence. Therefore this group of users could not be said to dominate the decision-making process, but it could be said that they hold more sway with each action than a user who isn't so involved in voting on the site.

All of these community involvement trends seem to suggest that casual users of the site don't really contribute much to the site's working; they don't vote much and their links don't tend to do particularly well. There are however plenty of exceptions to these rules of thumb about community involvement. For example; there were 29 Users in the March data who registered less than 10 votes but who submitted a Link which received more than 1000 votes. It has also been noted in the results section that the most popular link types aren't accounted for as well as some other popular link types in the RC model. This suggests a greater random element to the origins of extremely popular content. It also suggests that casual users are of some benefit to the more active community members; every time these users submit a Link there is a small chance that this Link will be very well received by the wider community. Because there are so many of these casual users, there are actually quite a few popular

links attributable to this group; and presumably the site is better and more appealing for the presence of these Links.

With regard to the problems of Information Overload it would seem that Reddit's democratic interface does quite well. The evidence for this comes not from any particular facet of how the site operates; it is based on the large number of people contributing simultaneously to the resource, and the even larger (and growing quickly) number of people to use the resource daily. 102,232 Users have, as a group, determined which of 352,902 Links should be displayed on the site's main page and every Sub-Reddit page for the month of March – and over 5 million unique Visitors (by IP address) have used this resource during this time. The remarkable thing about this is that a lot of the activity has occurred in a single Sub-Reddit. The main sub-Reddit received 150,042 Links in March; these drew 516,775 votes from 68,504 different Users – this is a greater level of activity than seen in the entire data-set considered by Himelboim (2008) comprising of 30 different Usenet discussion groups. This system is not perfect; many of the submissions to this Sub-Reddit never receive a second vote (therefore may never have been seen by another user). There are also patterns in this data whereby more active Users are more likely to be seen and heard, but this trend is not nearly as strong as that seen in older Bulletin board systems (Himelboim, 2008) – there is a larger "random" element to determination of which content will be viewed and voted on.

Of course, the determinants of which Links will be most popular are likely not random but due to individual qualities of the Links themselves (i.e. important, interesting, funny, etc.). One of the biggest weaknesses of the present research is that the qualities of content could not be assessed. This kind of analysis could not be undertaken for two reasons; Reddit did not want us to be able to identify any Users or Links from the data provided, so it was not possible to check the qualities of individual Links which had particularly interesting patterns of voting. Secondly, because there are so many links submitted to Reddit and their content is so varied, it would be very difficult and time-consuming to conduct a detailed qualitative analysis of even a small sub-set of these.

Nevertheless, combining qualitative with quantitative analyses is the most promising avenue down which this research could proceed. This combination of quantitative and qualitative data on Reddit's servers represents a record of everything which has occurred in the Reddit.com community thus far (qualitative). Furthermore, every action and interaction (from the most unremarkable to the most important) which has occurred in Reddit's history to date has already been ranked by the site's users (quantitative) to reflect its relative importance. These rankings provide a shortcut to understanding what the Reddit community is about; what it represents and how it works.

Voting behaviour would also seem to be very high on ecological validity; positive/negative votes have not been cast as part of some experiment or to express an opinion. The people who cast these votes did so to affect the outcome for the particular links they were voting on. The combination of these 3.5 million voting behaviours is not just a data-set to be analysed by social scientists; they also represent

community-level decisions about what content should be prominent on the site at every minute of every day in March.

The importance which "community" seems to have on Reddit highlights the second major drawback of the present research; the absence of commenting data. Commenting and comment-voting actions were originally requested from Reddit but were not provided. Given the problems encountered with the size of the link votes data-set; it seems that analysing a month's worth of comment votes would not have been feasible in any case. Every relatively popular link on Reddit can have hundreds of comments, and each of these can have hundreds of votes. The scale of such a data-set would require drastic sampling (problematic because so much of the content bears some relationship to content elsewhere on the site) or a specialist hardware/software solution.
The quantitative democratic system used by Reddit allows thousands of individuals to communicate in shared spaces – and their interaction produces a resource which is utilised by potentially millions of people. Much more research is required to determine whether and how these systems can be used to deal with the problems of Information Overload in many-to-many communication. The initial findings from this research on voting data suggest that these problems are being dealt with at least in part by sharing the task of finding the best content out among the group members. There is evidence in the voting data of some strategies associated with bulletin board systems being employed whereby more active and involved Users have a higher likelihood of submitting popular content, This trend is however much weaker than that seen in bulletin board systems; these very active Users could not be said to dominate any aspect of proceedings on the site.

To really address the question of how well systems like these can facilitate mass communication, it will be necessary to expand the scope of research. The most promising approach to developing our understanding of these systems is to integrate a qualitative analysis of the content with an understanding of all the quantitative mechanisms of the system. It will also likely prove fruitful to experiment with the uses these interfaces are put to, and specifics of how they work in different contexts. To compliment these studies it would also be beneficial to survey samples of people who use these resources with regard to their feelings and opinions about them.

This research has merely scratched the surface of one prominent Social News site. In doing do we have found some interesting dynamics to activity on the site, and some reasons to believe that systems like these may have the potential to facilitate coherent computer-mediated communications between larger groups than previously possible. It remains to be seen whether Reddit.com's quantitative democratic interface will continue to fulfil its purpose if the site's user-base continues to expand at its current rate. As things stand; an up/down arrow and some addition/subtractions have gone a long way to allowing one online community to sort, organise and distil a very large number of contributions into lists of those which are the "newest and most interesting" (as defined by the collective action of Users).

References

www.reddit.com – Data analysed were generated in March 2009.
Contact for data and questions: **Christopher Slowe** – Senior programmer [E-mail] (Personal communications, April 2009 – August 2009).

Adamic, L. A. (2000). Power-law distribution of the world wide web. *Science*, 287(5461): pp 2115-2115.

Agre, Philip E. (2002). *Real*-Time Politics: The Internet and the Political Process, *The Information Society*, 18 (5). pp 311-331.

Agresti, A., 1990. *Categorical Data Analysis*. New York: Wiley Interscience

Albert, R., Jeong, H. & Barbarasi, A. L., 1999. Diameter of the World-Wide-Web. *Nature*, 401: p130

Alexa.com – The Web Information Company, (2009) *Reddit.com Information* [Online] (updated daily)
Available at: http://www.alexa.com/siteinfo/reddit.com
[Accessed periodically April 2009 – August 2009]

Butler, S., 2001. Membership size, Communication Activity, and Sustainability: A Resource-Based Model of Online Social Structures. *Information Systems Research* 12 (4): pp 346-362.

Chappell, 2008. The 2008 Social Network Analysis Report – Geographic, Demographic and Traffic data revealed. *Ignite Social Media*, [Online] 19th November.
Available at: http://www.ignitesocialmedia.com/2008-social-network-analysis-report/
[Accessed 15th April 2009].

Goodman, L. A., 1979. Simple models for the analysis of association in cross-classifications having ordered categories. *Journal of the American Statistical Association*, 74: pp 537 – 552.

Grossman, L. K., 1995. *The Electronic Republic: re-shaping democracy in the information age.* New York: Viking.

Hansen, S., Berente, N. & Lyytinen, K., 2009. Wikipedia, Critical Social Theory, and the Possibility of Rational Discourse. *The Information Society*, 25: pp 38-59.

Himelboim, I., 2008. Reply distribution in online discussions: A comparative network analysis of political and health newsgroups. *Journal of Computer-mediated communication*, 14: pp 156-177.

Jones, Q., Ravid, G. & Rafaeli, S., 2002. *An Empirical Exploration of Mass Interaction System Dynamics: Individual Information Overload and Usenet Discourse.* Paper presented at the 35th Annual Hawaii conference on System Sciences, 2002.

Ohanian, A. & Golliher, S., 2009. Understanding Social News: A case study using Reddit.com, *Search Engine Marketing Research Journal,* [Online]
Available at: www.**semj**.**org**/documents/VOL2_ISSUE1_Preview.pdf
[Accessed 1st May 2009]

Nagin, D. S., 1999. Analyzing Developmental Trajectories: A semi-parametric, group-based approach. *Psychological Methods,* 4(2): pp 139-157.

Nye, J. S., 2002. The information revolution and American soft power. *Asia Pacific Review* 9(1): pp 60-76.

Raban, A. R. & Rabin, E. (2007). *The power of assuming normality.* Paper presented at the European and Mediterranean conference on Information Systems, 2007.

R Development Core Team (2009). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org

Tajfael, H. & Turner, J. (1979). "An Integrative Theory of Intergroup Conflict". in Austin, William G. & Worchel, Stephen. *The Social Psychology of Intergroup Relations*. California: Brooks-Cole: pp. 94–109.

Turner, H. & Firth, D. (2009). Generalized nonlinear models in R: An overview of the gnm package. (R package version 0.9-9). http://CRAN.R-project.org/package=gnm

Stat references: Goodman, GNM, CLARA, latent trajectory (Nagin)
Find alexis ohanian reference

Equations:
Latent trajectory

RC model

Appendices

Appendix A – Clustering Solution Medoids and Average Silhouette width plots

Figure A-1: Average silhouette width plots for the four data-sets which were clustered.

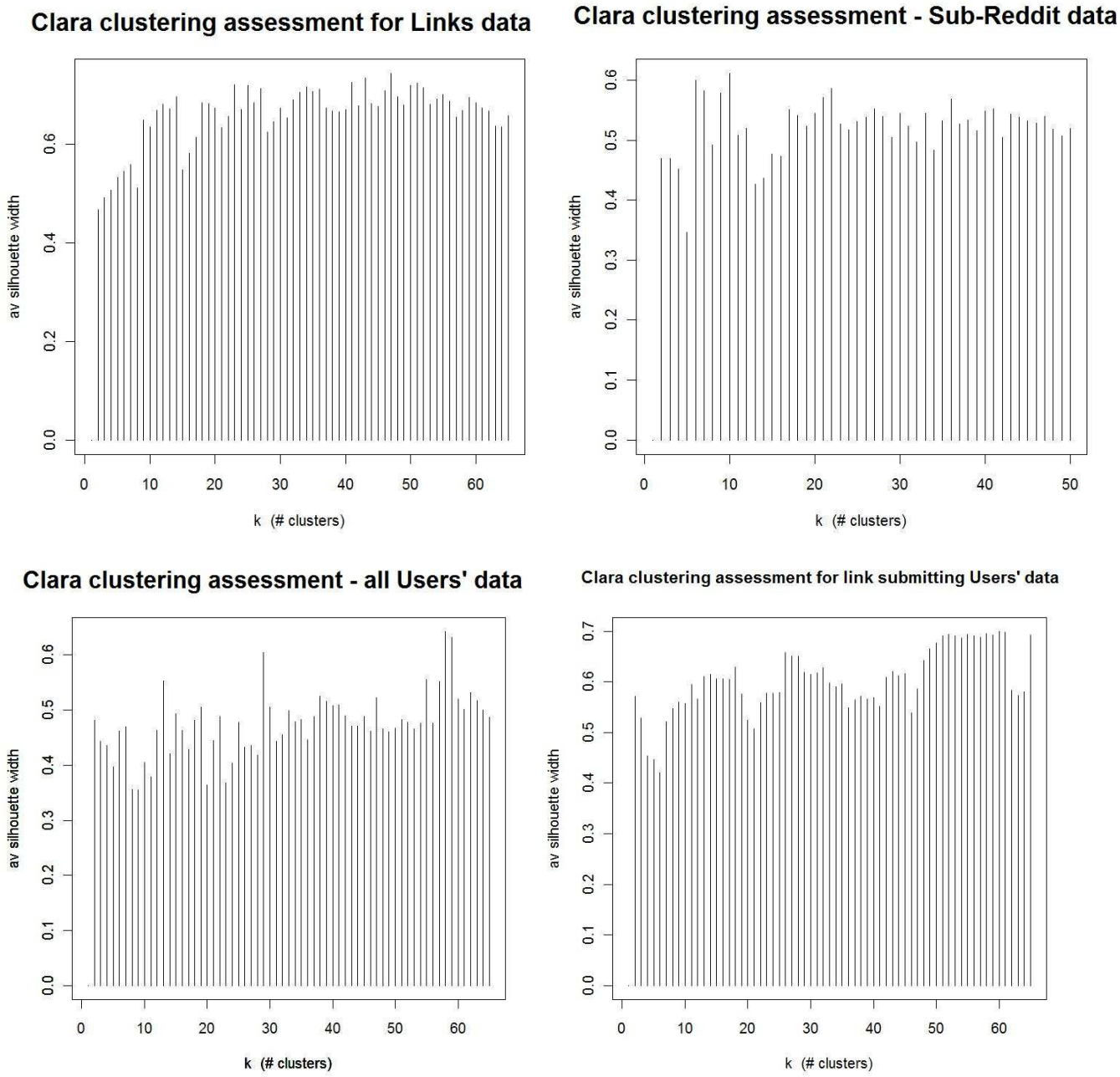


Figure A-2: Back-transformed Cluster medoids for 10 Sub-Reddit types (plus descriptives of the main Reddit – included as No. 1)

Cluster	Total Links	Total Votes	Agg per Link	% Controversy	% Self	Cluster N
1	150042	516775	1.93	0.1763	0.02178	1
2	1969	23521	6.51	0.5313	0.0747	18
3	1729	63923	19.76	0.5622	0.0133	12
4	183	9213	35.28	0.2654	0.1967	5
5	82	968	6.93	0.3926	0.0000	191
6	17	44	1.88	0.2157	0.0000	301
7	9	35	3.33	0.0444	0.5556	82
8	2	4	1.00	0.5000	0.0000	169
9	1	2	0.00	1.0000	0.0000	71
10	1	1	1.00	0.0000	1.0000	121
11	1	1	1.00	0.0000	0.0000	1213

Figure A-3: Back-transformed Cluster medoids for 47 Link types

Cluster	Aggregate	Total Votes	% Negative	% Controversy	Self	SR type	Cluster N
44	**636**	**938**	0.1578	0.1888	**0**	3	452
42	382	554	0.1534	0.1820	0	1	322
35	262	530	0.2472	0.3333	0	3	338
46	158	316	0.2468	0.3305	0	3	663
32	119	203	0.2020	0.2563	0	3	1659
43	73	264	**0.3561**	**0.5629**	**0**	3	188
18	29	49	0.2041	0.2564	0	3	2969
45	26	27	0.0000	0.0000	0	2	536
20	21	58	**0.3103**	**0.4615**	0	3	1701
15	21	37	0.2162	0.2759	**0**	2	2333
47	18	22	0.0909	0.1000	0	1	720
21	13	17	0.1176	0.1333	0	5	7194
10	9	15	0.2000	0.2500	0	1	3836
7	7	7	0.0000	0.0000	0	5	4823
6	5	25	**0.4000**	**0.6667**	0	3	5870
9	4	16	**0.3750**	**0.6000**	0	5	2519
13	4	11	0.2727	**0.4286**	0	2	5876
12	4	6	0.1667	0.2000	0	2	2763
26	3	3	0.0000	0.0000	0	6	10147
22	2	7	0.2857	**0.5000**	0	5	4398
30	2	6	**0.3333**	**0.5000**	0	3	7111
11	2	4	0.2500	0.3333	0	3	7627
25	2	2	0.0000	0.0000	**1**	2	1810
2	1	11	**0.4545**	**0.8333**	0	2	2600
36	1	5	**0.4000**	**0.6667**	**1**	2	2333
4	1	3	**0.3333**	**0.5000**	0	1	7217
16	1	3	**0.3333**	**0.5000**	0	**8**	3674
1	1	1	0.0000	0.0000	0	1	**110840**
3	1	1	0.0000	0.0000	0	3	32801
5	1	1	0.0000	0.0000	0	2	17718
17	1	1	0.0000	0.0000	0	11	15255
8	0	4	**0.5000**	**1.0000**	0	3	17493
23	0	4	**0.5000**	**1.0000**	0	2	10513
34	0	4	**0.5000**	**1.0000**	0	5	4871
33	0	4	**0.5000**	**1.0000**	0	**8**	2884
19	0	2	**0.5000**	**1.0000**	0	1	17740
39	0	2	**0.5000**	**1.0000**	0	6	2254
24	**-1**	5	**0.6000**	**0.6667**	0	3	3846
27	**-1**	3	**0.6667**	**0.5000**	0	1	6905
29	**-1**	3	**0.6667**	**0.5000**	0	3	5585
31	**-1**	3	**0.6667**	**0.5000**	0	2	2912
37	**-1**	3	**0.6667**	**0.5000**	**1**	3	715
14	**-1**	1	**1.0000**	0.0000	0	1	2094
28	**-1**	1	**1.0000**	0.0000	0	3	1237
41	**-1**	1	**1.0000**	0.0000	0	5	446
38	**-2**	4	**0.7500**	0.3333	**0**	2	1014
40	**-3**	5	**0.8000**	0.2500	0	3	4100

Figure A-4 :Cluster Medoids for 58 User types (All Users)

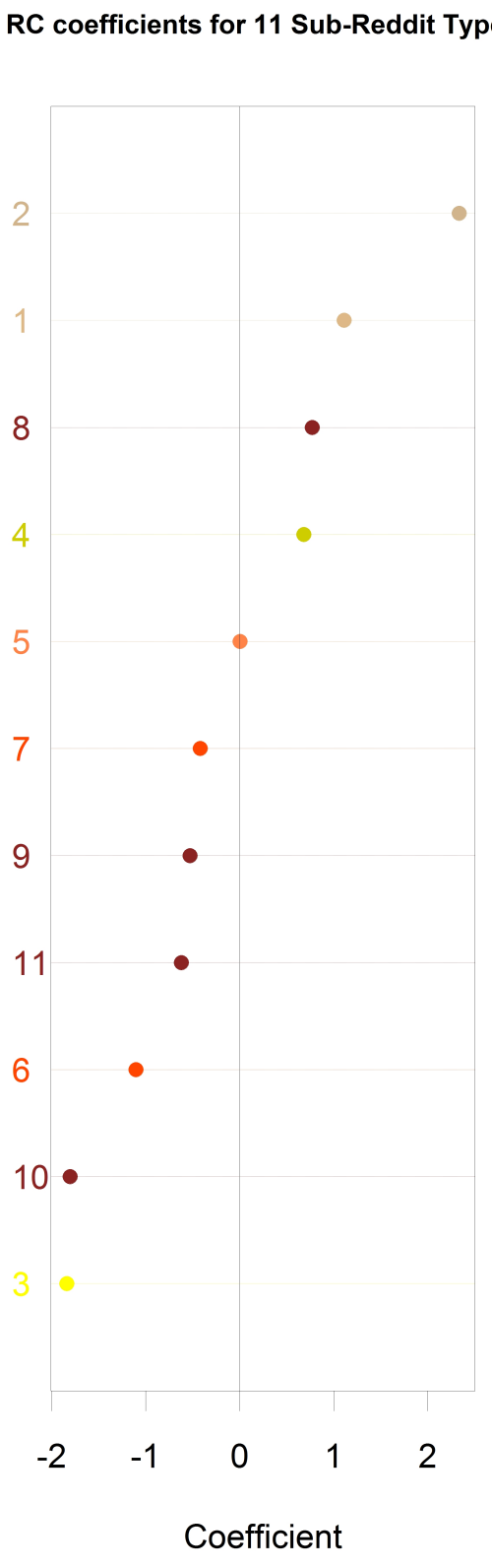
Cluster	Total Votes	ID age	Voting Order	% Negative	% link submission	% self votes	Cluster size
35	1 vote	5	Order 1	0	1	0	**14502**
39	1 vote	4	Order 1	0	1	0	5930
14	1 vote	2	Order 1	0	1	0	2501
6	1 vote	**2**	Order 101 - 500	0	0	0	2313
7	1 vote	3	Order 11 - 100	0	0	0	2077
24	1 vote	3	Order 1	0	1	0	1972
33	1 vote	3	Order 500+	0	0	1	1163
3	1 vote	**1**	**Order 2 - 10**	0	0	0	888
56	1 vote	5	**Order 2 - 10**	0	0	0	858
4	1 vote	**1**	Order 11 - 100	**1**	0	0	855
44	1 vote	**2**	**Order 2 - 10**	**1**	0	0	334
18	1 vote	**1**	Order 101 - 500	**1**	0	1	280
52	2 - 5 votes	5	Order 1	0	1	0	3506
49	2 - 5 votes	4	Order 1	0	1	0	3052
47	2 - 5 votes	4	Order 101 - 500	0.25	0	0	2449
21	2 - 5 votes	**2**	Order 1	0	1	0	2107
25	2 - 5 votes	3	Order 1	0	1	0	1756
16	2 - 5 votes	**1**	Order 11 - 100	0.25	0	0.25	1590
20	2 - 5 votes	**2**	Order 500+	0	0	0.8	1552
22	2 - 5 votes	**2**	Order 101 - 500	0.25	0	0	1506
27	2 - 5 votes	**1**	Order 101 - 500	**0.6**	0	0.2	1500
38	2 - 5 votes	**2**	Order 11 - 100	**1**	0	0	1246
46	2 - 5 votes	5	Order 101 - 500	0	0	0.2	1029
19	2 - 5 votes	**2**	**Order 2 - 10**	**0.6**	**0.4**	0	964
11	6 - 10 votes	3	Order 101 - 500	0.125	0	0.125	2559
13	6 - 10 votes	3	Order 500+	0.3333	0	0.1111	1889
5	6 - 10 votes	**1**	Order 500+	0.3	0	0.1	1736
17	6 - 10 votes	**1**	Order 101 - 500	0.4444	0	0.1111	1403
51	6 - 10 votes	4	Order 11 - 100	0.2857	0	0	1268
55	6 - 10 votes	5	Order 1	0	1	0	1028
54	6 - 10 votes	4	Order 1	0	1	0	930
41	6 - 10 votes	3	Order 1	0	1	0	699
57	6 - 10 votes	5	Order 500+	0.3333	0	0	395
8	11 - 25 votes	**1**	Order 101 - 500	0.48	0	0.04	2395
28	11 - 25 votes	**2**	Order 101 - 500	0.1538	0	0.0769	2258
1	11 - 25 votes	3	Order 1	0	1	0	1757
37	11 - 25 votes	2	Order 500+	0.2308	0	0.2308	1625
2	11 - 25 votes	3	Order 101 - 500	0.3333	0	0	1580
48	11 - 25 votes	5	Order 101 - 500	0.2308	0	0.0769	1484
29	11 - 25 votes	3	Order 500+	0	0	0.25	1146
36	11 - 25 votes	**2**	**Order 2 - 10**	0.1667	**0.3333**	0.5	456
23	26 - 50 votes	**1**	Order 101 - 500	0.0789	0	0	1962
31	26 - 50 votes	**2**	Order 101 - 500	0.2444	0.0889	0.1778	1541
12	26 - 50 votes	3	Order 101 - 500	0.3448	0	0.2069	1411
43	26 - 50 votes	3	Order 11 - 100	0.0526	0.1053	0.1842	1095
32	26 - 50 votes	3	Order 500+	0	0	0.0213	990
53	26 - 50 votes	**1**	**Order 2 - 10**	0.0238	**0.5238**	0.0238	515
58	26 - 50 votes	5	Order 1	0	1	0	205
50	26 - 50 votes	**1**	**Order 2 - 10**	**0.9655**	0	0.3448	82
9	51 - 100 votes	**1**	Order 101 - 500	0	0.0926	0.4074	1884
45	51 - 100 votes	4	Order 101 - 500	0.1017	0	0.0339	1754
42	51 - 100 votes	**2**	Order 101 - 500	**0.7647**	0.0392	0.0588	1631
40	51 - 100 votes	4	Order 1	0	1	0	695
30	101 - 200 votes	**2**	Order 101 - 500	0.3103	0	0.0966	2567
10	101 - 200 votes	4	Order 101 - 500	0.0724	0	0.125	1339
34	101 - 200 votes	5	Order 101 - 500	0.3354	0.1402	0.1463	216
15	201 - 500 votes	**2**	Order 101 - 500	0.1874	0.0407	0.1263	2522
26	501+ votes	**2**	Order 101 - 500	0.001	0.006	0.0379	802

Figure A-5:Cluster Medoids for 60 User types (Link submitting Users only)

Cluster	Total Votes	ID age	Voting Order	% Negative	% link subs	% self votes	Cluster N
25	1 vote	5	1	0	1	0	**14018**
27	1 vote	4	1	0	1	0	5762
14	1 vote	3	1	0	1	0	1962
9	1 vote	**2**	1	0	1	0	1433
13	1 vote	**1**	1	0	1	0	1072
55	1 vote	5	1	0	1	1	480
20	1 vote	4	1	0	1	1	179
44	2 - 5 votes	5	1	0	1	0	3412
40	2 - 5 votes	4	1	0	1	0	2838
15	2 - 5 votes	3	1	0	1	0	1643
12	2 - 5 votes	**2**	1	0	1	0	1049
23	2 - 5 votes	**1**	1	0	1	0	805
11	2 - 5 votes	**2**	4	0	0.25	0	594
22	2 - 5 votes	**1**	3	0	0.4	0	557
47	2 - 5 votes	4	2	0	0.75	0	477
36	2 - 5 votes	**2**	2	0	0.3333	0.3333	476
50	2 - 5 votes	4	4	0.25	0.5	0.25	451
49	2 - 5 votes	5	1	0	1	0	249
43	2 - 5 votes	4	5	0	1	0	192
60	2 - 5 votes	5	5	0	0.5	0	107
53	6 - 10 votes	5	1	0	1	0	928
52	6 - 10 votes	4	1	0	1	0	843
34	6 - 10 votes	3	1	0	1	0	663
31	6 - 10 votes	**1**	4	0.3	0.2	0.1	423
42	6 - 10 votes	4	4	0	0.2857	0.2857	409
7	6 - 10 votes	**2**	4	0.4	0.1	0.1	353
32	6 - 10 votes	**2**	1	0	1	0	324
17	6 - 10 votes	**1**	3	0	0.5	0.3333	287
45	6 - 10 votes	**2**	3	0.4444	0.1111	0.2222	252
57	6 - 10 votes	5	2	0	0.1429	0	222
59	6 - 10 votes	5	4	0	0.8571	0	102
1	11 - 25 votes	3	1	0	1	0	1205
29	11 - 25 votes	**1**	4	0.3043	0.087	0.087	1034
2	11 - 25 votes	3	4	0.0556	0.1111	0.2222	1011
18	11 - 25 votes	**2**	4	0.1429	0.2143	0.1429	797
4	11 - 25 votes	**2**	3	0.1667	0.25	0	621
26	11 - 25 votes	**2**	1	0	1	0	444
35	11 - 25 votes	3	3	0.3333	0.1667	0.3333	392
58	11 - 25 votes	4	1	0	1	0	364
51	11 - 25 votes	5	4	0	0.5385	0	357
30	11 - 25 votes	**2**	5	0	0.4545	0.3636	342
54	11 - 25 votes	3	1	0	0.1818	0	40
19	26 - 50 votes	3	4	0.2143	0.2143	0.1071	1227
46	26 - 50 votes	4	1	0	1	0	397
3	26 - 50 votes	**2**	1	0	1	0	305
41	26 - 50 votes	3	1	0	1	0	288
5	51 - 100 votes	**1**	4	0.25	0.0192	0.1538	1358
38	51 - 100 votes	**2**	3	0.0862	0.0517	0.0172	913
8	51 - 100 votes	4	4	0.1806	0.0833	0.3194	574
28	51 - 100 votes	4	1	0	1	0	402
6	101 - 200 votes	3	4	0.1639	0.1475	0.1885	847
39	101 - 200 votes	**2**	4	0.2832	0.0173	0.0347	688
33	101 - 200 votes	**1**	4	0.0841	0.0374	0.1121	670
56	101 - 200 votes	4	4	0.0141	0.6056	0.0423	230
16	201 - 500 votes	**1**	4	0.1152	0.0037	0.0558	763
10	201 - 500 votes	**2**	4	0.1418	0.0073	0.1527	681
37	201 - 500 votes	**2**	3	0.0393	0.0319	0.059	453
24	201 - 500 votes	4	4	0.018	0.006	0.0631	320
48	201 - 500 votes	5	3	0.2189	0.2747	0.0773	98
21	201 - 500 votes	5	4	0.2747	0.0129	0.0815	37

Appendix B – RC model coefficient plots

Figure B – 1: RC model coefficients for Voting activity by User and Sub-Reddit types.



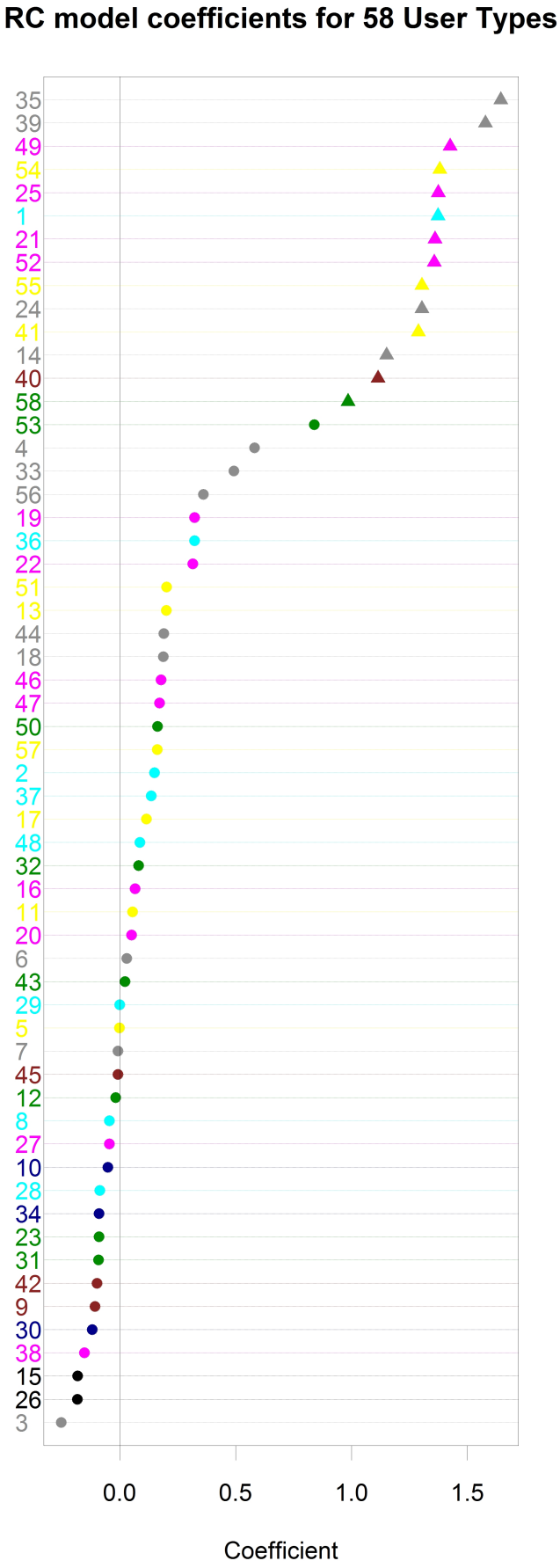


Figure B-2: RC model coefficients for Voting activity by User and Link types.

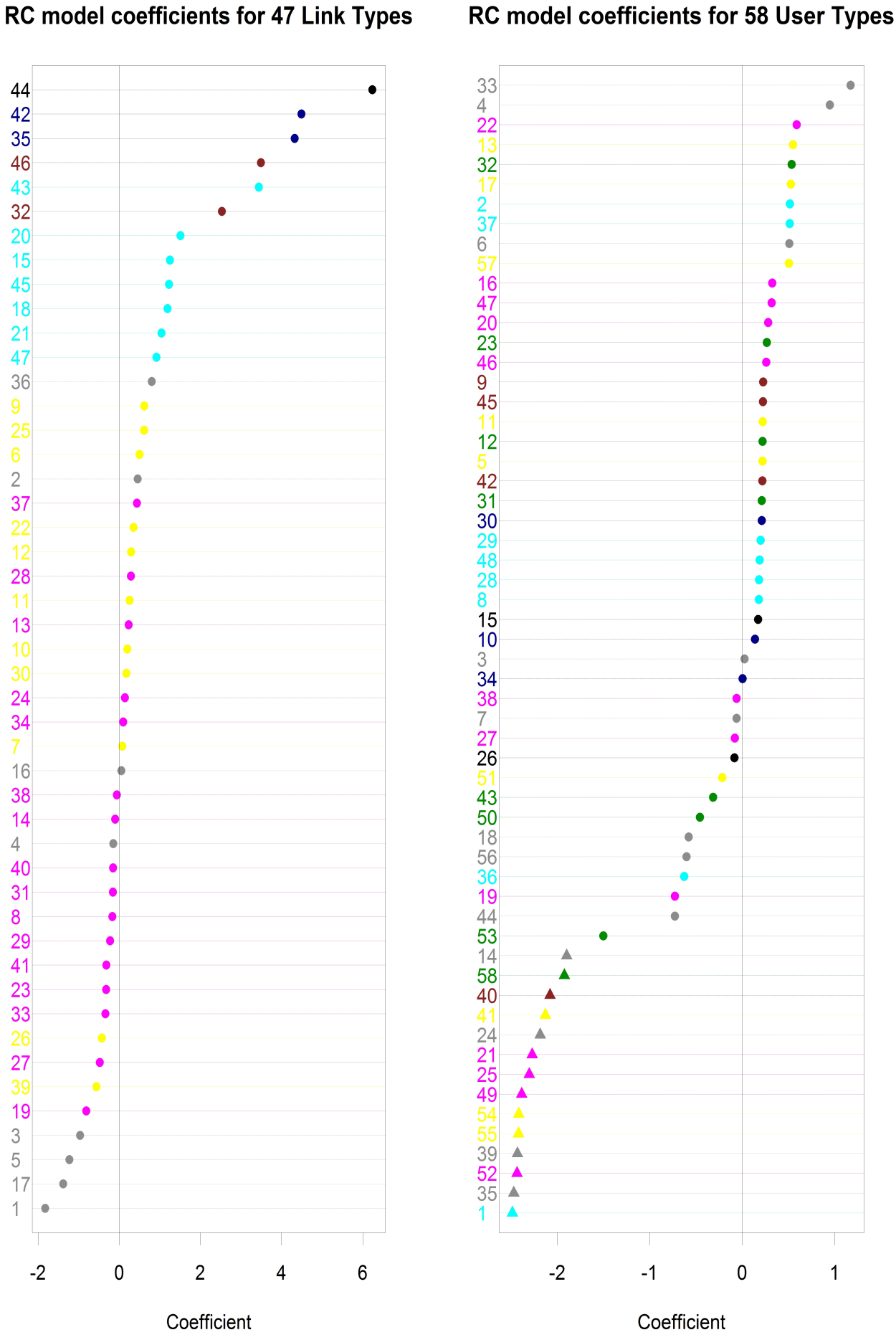
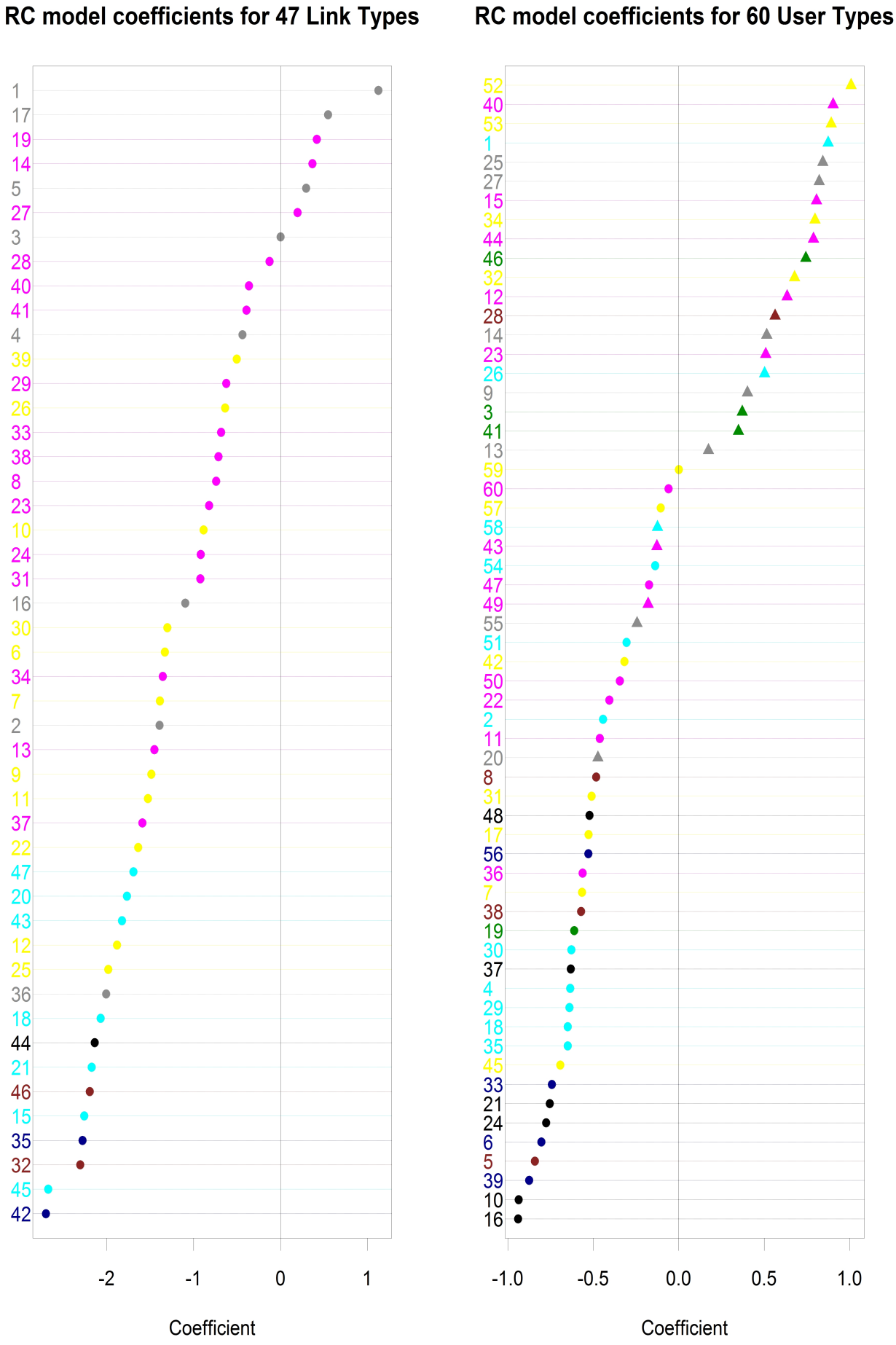


Figure B-3: RC model coefficients for Link submission frequencies by User and Link types. User clusters formed on the sub-set of users with at least 1 link submission.



Appendix C – Latent trajectory model plots

Figure C – 1: Class means for 12 latent classes fitted to Links' proportion of negative votes variable in 20% incremenets.

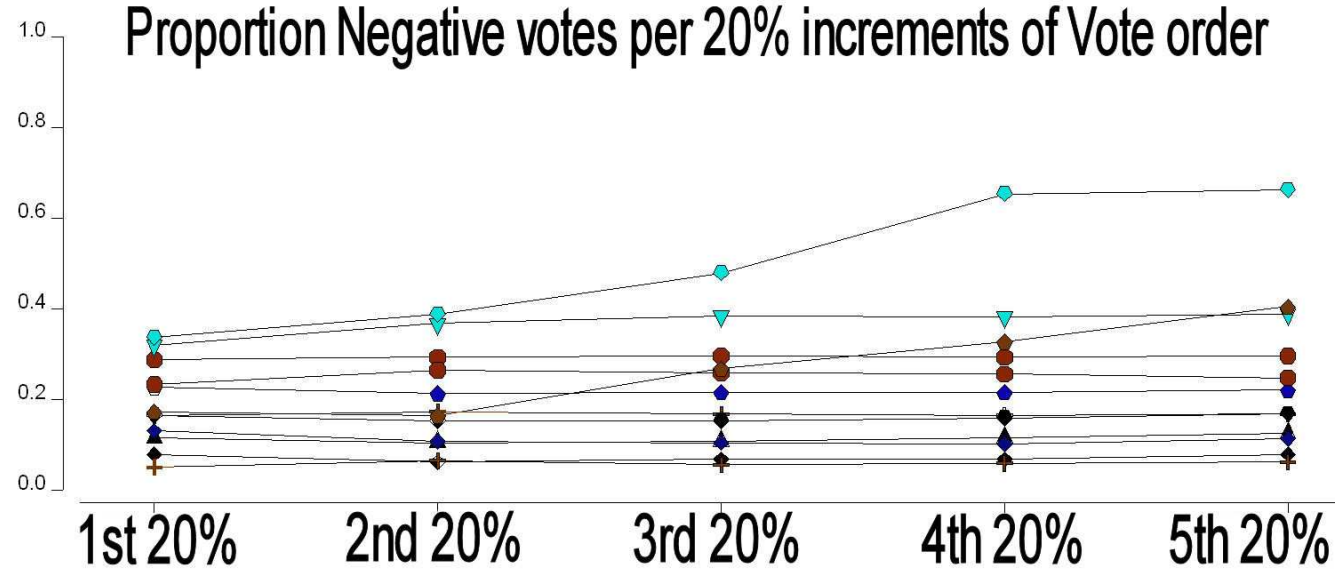
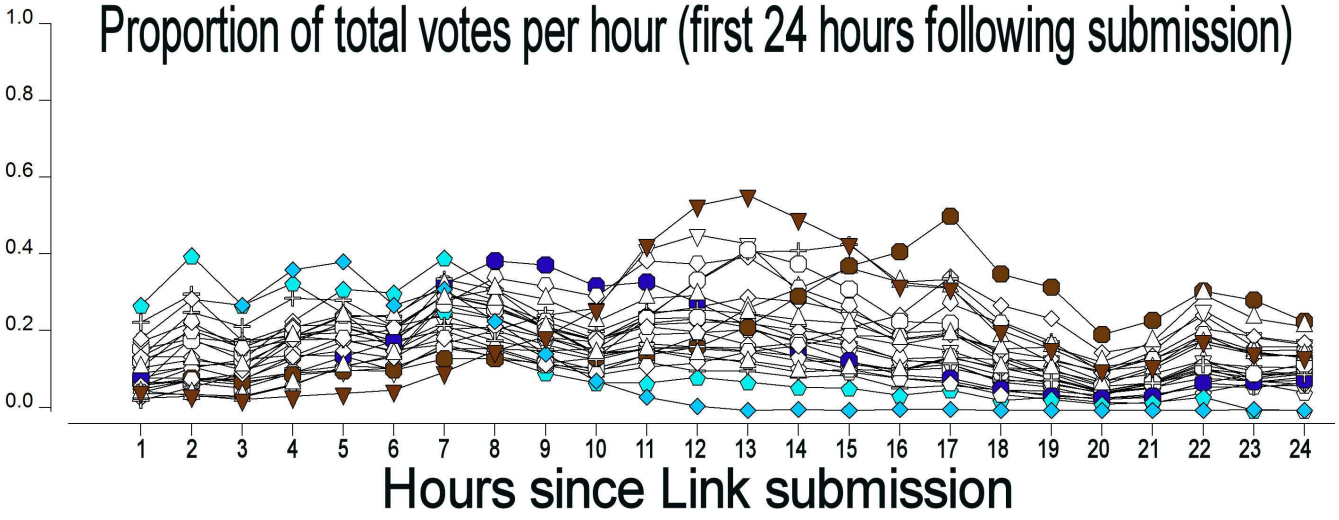


Figure C – 2: Class means for 25 latent classes fitted to Links' proportion of total votes received in each hour following submission (for 24 hours).



Appendix D – Python scripts

D – 1: populate_user_ids.py

```
#pull all user_ids
c.execute("""SELECT user_id FROM votes""")
u1 = []
rows = c.fetchall()
for row in rows:
    for col in row :
         u1.append(int(row[col]))


#make user_ids a set (remove duplicates)
set_users = set(u1)

#put the user_ids in the users table
for i in set_users:
    c.execute("""INSERT INTO users (user_id) VALUES (%s)""",
                  [(i)])
```

D – 2: populate_users_with_vote_nos.py

```
def count_votes(v1):
    pos = 0
    neg = 0
    null = 0
    for a in v1:
        if a == 1:
            pos = pos + 1
            continue
        elif a == -1:
                neg = neg + 1
                continue
        elif a == 0:
                null = null + 1
                continue
    result = [pos, neg, null]
    return result
#pull user_ids
c.execute("""SELECT user_id FROM users""")
u1 = []
rows = c.fetchall()
for row in rows:
    for col in row :
         u1.append(int(row[col]))

#pull votes for user_id
for i in u1:
    ticker = 1
    c.execute("""SELECT vote FROM votes WHERE user_id = %s""",
                  [(i)])
    v1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            v1.append(int(row[col]))
    num_votes = len(v1)
    votes = count_votes(v1)
    pos = votes[0]
    neg = votes[1]
    null = votes[2]
    c.execute("""UPDATE users set tot_votes = %s, pos_votes = %s, neg_votes = %s,
null_votes = %s WHERE user_id = %s""",
              [(num_votes), (pos), (neg), (null), (i)])
    ticker = ticker + 1
    print ticker
```

D – 3: populate_users_with_sub_nos.py

```
#pull user_ids
c.execute("""SELECT user_id FROM users""")
u1 = []
rows = c.fetchall()
for row in rows:
    for col in row :
         u1.append(int(row[col]))

#pull votes for user_id
ticker = 1
for i in u1:
    c.execute("""SELECT link_id FROM link_authors WHERE author_id = %s""",
                  [(i)])
    v1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            v1.append(int(row[col]))
    num_subs = len(v1)
    c.execute("""UPDATE users set link_subs = %s WHERE user_id = %s""",
              [(num_subs), (i)])
    ticker = ticker + 1
    print ticker
```

D – 4: add_percentages_to_user_votes.py

```
#pull user_ids
c.execute("""SELECT user_id FROM users""")
users = []
rows = c.fetchall()
for row in rows:
    for col in row :
         users.append(int(row[col]))

#pull details for user_id
ticker = 1
for i in users:

    #pull tot_votes
    c.execute("""SELECT tot_votes FROM users WHERE user_id = %s""",
                  [(i)])
    rows = c.fetchall()
    for row in rows:
        for col in row :
            tot_votes = (int(row[col]))
    #pull pos_votes
    c.execute("""SELECT pos_votes FROM users WHERE user_id = %s""",
                  [(i)])
    rows = c.fetchall()
    for row in rows:
        for col in row :
            pos_votes = (int(row[col]))

    #pull neg_votes
    c.execute("""SELECT neg_votes FROM users WHERE user_id = %s""",
                  [(i)])
    rows = c.fetchall()
    for row in rows:
        for col in row :
            neg_votes = (int(row[col]))

    #pull null_votes
    c.execute("""SELECT null_votes FROM users WHERE user_id = %s""",
                  [(i)])
    rows = c.fetchall()
    for row in rows:
        for col in row :
```

```
            null_votes = (int(row[col]))


    #pull link_subs

    c.execute("""SELECT link_subs FROM users WHERE user_id = %s""",
                  [(i)])
    rows = c.fetchall()
    for row in rows:
        for col in row :
            link_subs = (int(row[col]))

#calculate percentages and aggregate

    agg_votes = pos_votes - neg_votes
    tot = float(tot_votes)
    pos = float(pos_votes)
    neg = float(neg_votes)
    null = float(null_votes)
    linksub = float(link_subs)
    per_pos = pos/tot
    per_neg = neg/tot
    per_null = null/tot
    per_link_sub = linksub/tot


        #populate the links table
    c.execute("""UPDATE users set per_pos = %s, per_neg = %s, per_null = %s,
per_link_sub = %s, agg_votes = %s WHERE user_id = %s""",
                  [(per_pos), (per_neg), (per_null), (per_link_sub), (agg_votes),
(i)])
    ticker = ticker + 1
    print ticker
    print i
```

D – 5: populate_link_ids.py

```
#pull link_ids
c.execute("""SELECT link_id FROM link_authors""")
links = []
rows = c.fetchall()
for row in rows:
    for col in row :
         links.append(int(row[col]))

ticker = 1
for i in links:
    c.execute("""INSERT INTO links (link_id) VALUES (%s)""",
                  [(i)])
    ticker = ticker + 1
    print ticker
```

D – 6: populate_links.py

```
#pull link_ids
c.execute("""SELECT link_id FROM link_authors""")
links = []
rows = c.fetchall()
for row in rows:
    for col in row :
         links.append(int(row[col]))



#pull details for link_id
ticker = 1
for i in links:
    #pull sr_ids
```

```
    c.execute("""SELECT sr_id FROM link_srs WHERE link_id = %s""",
                  [(i)])
    rows = c.fetchall()
    for row in rows:
        for col in row :
            sr_id =(int(row[col]))

    #pull is_self and set 0 or 1
    c.execute("""SELECT is_self FROM link_is_self WHERE link_id = %s""",
                  [(i)])
    is_seli = []
    is_self = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            is_seli.append(str(row[col]))
    if is_seli == ['t']: is_self = 1
    else: is_self = 0

    #pull votes
    c.execute("""SELECT vote FROM votes WHERE link_id = %s""",
                  [(i)])
    vi = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            vi.append(int(row[col]))
    tot_votes = len(vi)
    #has the link been classed as spam and nullified?
    if tot_votes == 0:
    #populate null_links table
        c.execute("""INSERT INTO null_links (link_id, sr_id, is_self) VALUES (%s, %s,
%s)""",
                  [i, sr_id, is_self])
        ticker = ticker + 1
        print ticker
        print "NULL!!!"
        print i
        continue
    elif tot_votes > 0:
        votes = count_votes(vi)
        pos_votes = votes[0]
        neg_votes = votes[1]
        null_votes = votes[2]
        agg_votes = pos_votes - neg_votes
        tot = float(tot_votes)
        pos = float(pos_votes)
        neg = float(neg_votes)
        null = float(null_votes)
        per_pos = pos/tot
        per_neg = neg/tot
        per_null = null/tot

        #populate the links table
        c.execute("""INSERT INTO links (link_id, tot_votes, sr_id, pos_votes,
neg_votes, null_votes, agg_votes, per_pos, per_neg, per_null, is_self) VALUES (%s,
%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)""",
                      [i, tot_votes, sr_id, pos_votes, neg_votes, null_votes,
agg_votes, per_pos, per_neg, per_null, is_self])
        ticker = ticker + 1
        print ticker
        print i
```

D – 7: populate_SR_ids.py

```
#pull link_ids
c.execute("""SELECT sr_id FROM links""")
```

```
sr_ids = []
rows = c.fetchall()
for row in rows:
    for col in row :
         sr_ids.append(int(row[col]))

sr_set = set(sr_ids)

for i in sr_set:
      c.execute("""INSERT INTO sub_reddits (sr_id) VALUES (%s)""",[(i)])
```

D – 8: populate_sub_reddits.py

```
#pull sr_ids
c.execute("""SELECT sr_id FROM sub_reddits""")
sub_reds = []
rows = c.fetchall()
for row in rows:
    for col in row :
         sub_reds.append(int(row[col]))



#pull details for sr_id
ticker = 1
for i in sub_reds:
    #pull tot_votes and make tot_links and votes_per_link
    c.executemany("""SELECT tot_votes FROM links WHERE sr_id = %s""",
                  [(i)])
    sr_votes = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            sr_votes.append(int(row[col]))
    tot_links = len(sr_votes)
    tot_votes = sum(sr_votes)
    fl_links = float(tot_links)
    fl_votes = float(tot_votes)
    votes_per_link = fl_votes/fl_links



    #pull pos_votes and sum for subreddit
    c.executemany("""SELECT pos_votes FROM links WHERE sr_id = %s""",
                  [(i)])
    sr_pos_votes = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            sr_pos_votes.append(int(row[col]))
    pos_votes = sum(sr_pos_votes)

    #pull neg_votes and sum
    c.executemany("""SELECT neg_votes FROM links WHERE sr_id = %s""",
                  [(i)])
    sr_neg_votes = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            sr_neg_votes.append(int(row[col]))
    neg_votes = sum(sr_neg_votes)

    #pull null_votes and sum
    c.executemany("""SELECT null_votes FROM links WHERE sr_id = %s""",
                  [(i)])
    sr_null_votes = []
    rows = c.fetchall()
    for row in rows:
```

```
        for col in row :
            sr_null_votes.append(int(row[col]))
    null_votes = sum(sr_null_votes)

    #work out remaining variables (agg_per_link, %pos, %neg, %null)
    agg_per_link = (pos_votes - neg_votes)/fl_links
    per_pos = pos_votes/fl_votes
    per_neg = neg_votes/fl_votes
    per_null = null_votes/fl_votes

    #upate the sub_reddits table
    c.execute("""UPDATE sub_reddits SET tot_links = %s, tot_votes = %s,
votes_per_link = %s, agg_per_link = %s, pos_votes = %s, neg_votes = %s, null_votes =
%s, per_pos = %s, per_neg = %s, per_null = %s WHERE sr_id = %s""",
                  [(tot_links), (tot_votes), (votes_per_link), (agg_per_link),
(pos_votes), (neg_votes), (null_votes), (per_pos), (per_neg), (per_null), (i)])
    ticker = ticker + 1
    print ticker
    print i
```

D – 9: calc_and_store_secs_since_link_sub.py

```
ticker = 1
#pull vote_ids
c.execute("""SELECT vote_id FROM votes""")
vote_ids = []
rows = c.fetchall()
for row in rows:
    for col in row :
         vote_ids.append(int(row[col]))

#pull secs for votes, lookup link and pull secs
for i in vote_ids:
    c.execute("""SELECT unix_epoch_secs FROM votes WHERE vote_id = %s""", [(i)])
    vote = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            vote.append(int(row[col]))

    c.execute("""SELECT link_id FROM votes WHERE vote_id = %s""", [(i)])
    link_id = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            link_id.append(int(row[col]))

    c.execute("""SELECT unix_epoch_secs FROM links WHERE link_id = %s""",
[(link_id[0])])
    linkt = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            linkt.append(int(row[col]))

#check if the vote's link_id was found in links table and take appropriate actions
    if linkt == []:
        f.execute("""UPDATE votes SET old_link = 1 WHERE vote_id = %s""", [(i)])
        print "old link"
        ticker = ticker + 1
        print ticker
        print i
        continue
    elif linkt[0] > 1:
        td = vote[0] - linkt[0]
        tdmins = td/60.0
        f.execute("""UPDATE votes SET secs_since_link_post = %s,
mins_since_link_post = %s, old_link = 0 WHERE vote_id = %s""", [(td), (tdmins), (i)])
```

```
        ticker = ticker + 1
        print ticker
        print i
        print "Secs since link post calculated and submitted!!!!!!!!!!!!!!!!!!!!"
        continue
```

D – 10: generate_vote_orders.py

```
#pull link_ids for votes which were made on links subbed in march
c.execute("""SELECT link_id FROM links""")
link_ids = []
rows = c.fetchall()
for row in rows:
    for col in row :
         link_ids.append(int(row[col]))

#pull vote_ids ordered by secs_since_link_post
for i in link_ids:
    c.execute("""SELECT vote_id FROM votes WHERE link_id = %s ORDER BY
secs_since_link_post""", [(i)])
    order = []
    votes = []
    k = 0
    #create lists or vote_ids and orders
    rows = c.fetchall()
    for row in rows:
        for col in row :
            k = k + 1
            votes.append(int(row[col]))
            order.append(k)
    m = 0
    #loop through lists of votes and orders generating proportional orders and
submitting the whole lot to database
    for j in votes:
        upd_order = order[m]
        upd_vote_id = votes[m]
        upd_prop_order = float(order[m])/len(order)
        c.execute("""UPDATE votes SET vote_order = %s, vote_order_proportion = %s
WHERE vote_id = %s""", [(upd_order), (upd_prop_order), (upd_vote_id)])
        m = m + 1
        print 'Done a vote-loop'
        print j
        print upd_vote_id
    print 'Done a link_loop!!!!!!!!!'
    print i
```

D – 11: populate_users_with_avg_vote_orders.py

```
#pull user_ids
c.execute("""SELECT user_id FROM users""")
u1 = []
rows = c.fetchall()
for row in rows:
    for col in row :
         u1.append(int(row[col]))
ticker = 1
#pull vote order proportions for user_id and find the average
for i in u1:
    c.execute("""SELECT vote_order_proportion FROM votes WHERE user_id = %s AND
old_link = 0""",
                  [(i)])
    vop = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            vop.append(float(row[col]))
    c.execute("""SELECT vote_order FROM votes WHERE user_id = %s AND old_link = 0""",
                  [(i)])
```

```
    vo = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            vo.append(int(row[col]))

    num_votes = len(vop)
    proportion = sum(vop)

#fix for the problem of a user with a summed order of zero (happens when a user only
voted for "old" links)
    if proportion == 0:
        print 'old link voter'
        continue
    elif proportion > 0:
        avg_proportion = proportion/num_votes
        tot_order = float(sum(vo))
        avg_order = tot_order/num_votes
        c.execute("""UPDATE users SET avg_vote_order_prop = %s,
avg_absolute_vote_order = %s WHERE user_id = %s""",
                  [(avg_proportion), (avg_order), (i)])
        ticker = ticker + 1
        print ticker
        print i
```

D – 12: add_extra_vars_to_users.py

```
#pull all user_ids
c.execute("""SELECT user_id FROM users""")
users = []
rows = c.fetchall()
for row in rows:
    for col in row :
         users.append(int(row[col]))

ticker = 0

for i in users:
#pull mins since link for votes which were not cast on old links or with link
submissions (these have mins since link values of 0)
    c.execute("""SELECT mins_since_link_post FROM votes WHERE user_id = %s AND
old_link = 0 AND link_sub_vote = 0""",
                  [(i)])
    mins = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            mins.append(float(row[col]))
    tot_votes_a = len(mins)
    #check if the user has any votes matching the description, if they do then
calculate the average
      if tot_votes_a == 0:
        avg_mins_since_link_post = 0
        print 'Null!!!!!!!!!!!!!!'
    elif tot_votes > 0:
        all_mins = sum(mins)
        avg_mins_since_link_post = all_mins/tot_votes_a

#pull sr_ids and Is_selfs

    c.execute("""SELECT sr_id FROM votes WHERE user_id = %s""",
                      [(i)])
    srids = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            srids.append(int(row[col]))
    tot_votes_b = len(srids)
```

```
    tot_SRs = set(srids)
    sub_reddits_voted_in = len(tot_SRs)
    setsr = float(sub_reddits_voted_in)
    votes_per_SR = tot_votes_b/setsr


    c.execute("""SELECT is_self FROM votes WHERE user_id = %s""",
                      [(i)])
    isself = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            isself.append(float(row[col]))
    sum_is_self = sum(isself)
    if sum_is_self == 0:
        proportion_self_votes = 0
    elif sum_is_self > 0:
        proportion_self_votes = sum_is_self/tot_votes_b


#submit values
    c.execute("""UPDATE users SET avg_mins_since_link_post = %s,
sub_reddits_voted_in = %s, votes_per_SR = %s, proportion_self_votes = %s WHERE
user_id = %s""",
                  [(avg_mins_since_link_post), (sub_reddits_voted_in),
(votes_per_SR), (proportion_self_votes), (i)])
    ticker = ticker + 1
    print ticker
    print i
```

D – 13: add_controversy_to_links.py

```
#pull link_ids
c.execute("""SELECT link_id FROM links""")
links = []
rows = c.fetchall()
for row in rows:
    for col in row :
         links.append(int(row[col]))



#pull details for link_id
ticker = 1
for i in links:

    #pull tot_votes
    c.execute("""SELECT tot_votes FROM links WHERE link_id = %s""",
                  [(i)])
    rows = c.fetchall()
    for row in rows:
        for col in row :
            tot_votes = (int(row[col]))
    #pull pos_votes
    c.execute("""SELECT pos_votes FROM links WHERE link_id = %s""",
                  [(i)])
    rows = c.fetchall()
    for row in rows:
        for col in row :
            pos_votes = (int(row[col]))

    #pull neg_votes
    c.execute("""SELECT neg_votes FROM links WHERE link_id = %s""",
                  [(i)])
    rows = c.fetchall()
    for row in rows:
        for col in row :
```

```
            neg_votes = (int(row[col]))



#calculate controversy and proportion

    if pos_votes < neg_votes:
        contrv = pos_votes
        if contrv == 0:
            prop_contrv = 0.0
        elif contrv != 0:
            con = float(contrv)
            prop_contrv = con/neg_votes
    elif neg_votes < pos_votes:
        contrv = neg_votes
        if contrv == 0:
            prop_contrv = 0.0
        elif contrv != 0:
            con = float(contrv)
            prop_contrv = con/pos_votes
    elif neg_votes == pos_votes:
        contrv = pos_votes
        prop_contrv = 1.0




    #populate the links table
    c.execute("""UPDATE links SET controversy = %s, proportional_controversy = %s
WHERE link_id = %s""",
                  [(contrv), (prop_contrv), (i)])
    ticker = ticker + 1
    print ticker
    print i
```

D – 14: generate_user_reg_order.py

```
c.execute("""SELECT user_id FROM users ORDER BY user_id""")
user_order = []
k = 0
user_ids = []
rows = c.fetchall()
for row in rows:
    for col in row :
        k = k + 1
        user_ids.append(int(row[col]))
        user_order.append(k)


m = 0
for i in user_ids:
    id_age = user_order[m]
    c.execute("""UPDATE users SET id_age = %s WHERE user_id = %s""", [(id_age), (i)])
    print m
    print i
    m = m + 1
```

D – 15: generate_5_per_negs_for_links.py

```
import MySQLdb

def count_votes(v1):
    pos = 0
```

```
    neg = 0
    null = 0
    for a in v1:
        if a == 1:
            pos = pos + 1
            continue
        elif a == -1:
                neg = neg + 1
                continue
        elif a == 0:
                null = null + 1
                continue
    result = [pos, neg, null]
    return result


#pull link_ids
c.execute("""SELECT link_id FROM links_plustypes""")
link_ids = []
rows = c.fetchall()
for row in rows:
    for col in row :
         link_ids.append(int(row[col]))


ticker = 1
for i in link_ids:
    #select votes within the required range of proportions.... do 10 times
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
vote_order_proportion <= 0.20""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    #find the proportion of positive votes and subtract from 1 to give proportion of
non-positive votes
    v1 = count_votes(votes1)
    pos1 = float(v1[0])
    if pos1 == 0:
        per_neg1 = 1
    if pos1 > 0:
        per_pos1 = pos1/len(votes1)
        per_neg1 = 1 - per_pos1

    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
vote_order_proportion <= 0.40 AND vote_order_proportion > 0.20""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    v1 = count_votes(votes1)
    pos1 = float(v1[0])
    if pos1 == 0:
        per_neg2 = 1
    if pos1 > 0:
        per_pos1 = pos1/len(votes1)
        per_neg2 = 1 - per_pos1

    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
vote_order_proportion <= 0.60 AND vote_order_proportion > 0.40""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    v1 = count_votes(votes1)
```

```
    pos1 = float(v1[0])
    if pos1 == 0:
        per_neg3 = 1
    if pos1 > 0:
        per_pos1 = pos1/len(votes1)
        per_neg3 = 1 - per_pos1

    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
vote_order_proportion <= 0.80 AND vote_order_proportion > 0.60""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    v1 = count_votes(votes1)
    pos1 = float(v1[0])
    if pos1 == 0:
        per_neg4 = 1
    if pos1 > 0:
        per_pos1 = pos1/len(votes1)
        per_neg4 = 1 - per_pos1

    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
vote_order_proportion > 0.80""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    v1 = count_votes(votes1)
    pos1 = float(v1[0])
    if pos1 == 0:
        per_neg5 = 1
    if pos1 > 0:
        per_pos1 = pos1/len(votes1)
        per_neg5 = 1 - per_pos1

#store results
    c.execute("""UPDATE links_plustypes SET per_negA = %s, per_negB = %s, per_negC =
%s, per_negD = %s, per_negE = %s WHERE link_id = %s""",
                 [(per_neg1), (per_neg2), (per_neg3), (per_neg4), (per_neg5), (i)])
    ticker = ticker + 1
    print i
    print ticker
```

D – 16: generate_24_hour_votes_for_links.py

```
import MySQLdb

def count_votes(v1):
    pos = 0
    neg = 0
    null = 0
    for a in v1:
        if a == 1:
            pos = pos + 1
            continue
        elif a == -1:
                neg = neg + 1
                continue
        elif a == 0:
                null = null + 1
                continue
    result = [pos, neg, null]
    return result

#pull link_ids
```

```
c.execute("""SELECT link_id FROM links_plustypes""")
link_ids = []
rows = c.fetchall()
for row in rows:
    for col in row :
         link_ids.append(int(row[col]))


ticker = 1
for i in link_ids:
    #get total votes
    c.execute("""SELECT tot_votes FROM links_plustypes WHERE link_id = %s""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes = (float(row[col]))


    #select votes within the required range of minutes.... do 24 times
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=60""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v1 = 0.00
    elif votes1 > 0:
        v1 = len(votes1)/votes




    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=120 AND mins_since_link_post >60""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v2 = 0.00
    elif votes1 > 0:
        v2 = len(votes1)/votes

    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=180 AND mins_since_link_post >120""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v3 = 0.00
    elif votes1 > 0:
        v3 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=240 AND mins_since_link_post >180""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v4 = 0.00
    elif votes1 > 0:
```

```
        v4 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=300 AND mins_since_link_post >240""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v5 = 0.00
    elif votes1 > 0:
        v5 = len(votes1)/votes

    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=360 AND mins_since_link_post >300""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v6 = 0.00
    elif votes1 > 0:
        v6 = len(votes1)/votes

    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=420 AND mins_since_link_post >360""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v7 = 0.00
    elif votes1 > 0:
        v7 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=480 AND mins_since_link_post >420""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v8 = 0.00
    elif votes1 > 0:
        v8 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=540 AND mins_since_link_post >480""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v9 = 0.00
    elif votes1 > 0:
        v9 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=600 AND mins_since_link_post >540""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v10 = 0.00
    elif votes1 > 0:
```

```
        v10 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=660 AND mins_since_link_post >600""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v11 = 0.00
    elif votes1 > 0:
        v11 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=720 AND mins_since_link_post >660""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v12 = 0.00
    elif votes1 > 0:
        v12 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=780 AND mins_since_link_post >720""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v13 = 0.00
    elif votes1 > 0:
        v13 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=840 AND mins_since_link_post >780""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v14 = 0.00
    elif votes1 > 0:
        v14 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=900 AND mins_since_link_post >840""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v15 = 0.00
    elif votes1 > 0:
        v15 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=960 AND mins_since_link_post >900""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v16 = 0.00
    elif votes1 > 0:
        v16 = len(votes1)/votes
```

```
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=1020 AND mins_since_link_post >960""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v17 = 0.00
    elif votes1 > 0:
        v17 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=1080 AND mins_since_link_post >1020""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v18 = 0.00
    elif votes1 > 0:
        v18 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=1140 AND mins_since_link_post >1080""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v19 = 0.00
    elif votes1 > 0:
        v19 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=1200 AND mins_since_link_post >1140""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v20 = 0.00
    elif votes1 > 0:
        v20 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=1260 AND mins_since_link_post >1200""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v21 = 0.00
    elif votes1 > 0:
        v21 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=1320 AND mins_since_link_post >1260""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v22 = 0.00
    elif votes1 > 0:
        v22 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=1380 AND mins_since_link_post >1320""", [(i)])
```

```
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v23 = 0.00
    elif votes1 > 0:
        v23 = len(votes1)/votes
    c.execute("""SELECT vote FROM votes_plustypes WHERE link_id = %s AND
mins_since_link_post <=1440 AND mins_since_link_post >1380""", [(i)])
    votes1 = []
    rows = c.fetchall()
    for row in rows:
        for col in row :
            votes1.append(int(row[col]))
    if votes1 == 0:
        v24 = 0.00
    elif votes1 > 0:
        v24 = len(votes1)/votes


#store results
    c.execute("""UPDATE links_plustypes SET v1 = %s, v2 = %s,v3 = %s,v4 = %s,v5 =
%s,v6 = %s,v7 = %s,v8 = %s, v9 = %s, v10 = %s, v11 = %s, v12 = %s, v13 = %s, v14 =
%s, v15 = %s, v16 = %s, v17 = %s, v18 = %s, v19 = %s, v20 = %s, v21 = %s, v22 = %s,
v23 = %s, v24 = %s WHERE link_id = %s""",
                  [(v1), (v2), (v3), (v4), (v5), (v6), (v7), (v8), (v9), (v10), (v11),
(v12), (v13), (v14), (v15), (v16), (v17), (v18), (v19), (v20), (v21), (v22), (v23),
(v24), (i)])
    ticker = ticker + 1
    print i
    print ticker
```

Appendix E – R code example

E-1: Generating 58 User Clusters from categorised data (interpreted solution); and Wk score for this solution. Example also fits clusters on raw and standardised variables and computes WKs (for comparison).

```
library(RMySQL)
library(cluster)
users = dbGetQuery(con, 'select * from users where avg_absolute_vote_order >= 1')

tvabs = users$tot_votes
tvcats = tvabs
tvcats[tvcats < 2] = 1
tvcats[tvcats > 1 & tvcats < 6] = 2
tvcats[tvcats > 5 & tvcats < 11] = 3
tvcats[tvcats > 10 & tvcats < 26] = 4
tvcats[tvcats > 25 & tvcats < 51] = 5
tvcats[tvcats > 50 & tvcats < 101] = 6
tvcats[tvcats > 100 & tvcats < 201] = 7
tvcats[tvcats > 200 & tvcats < 501] = 8
tvcats[tvcats > 501] = 9
tvcatsf = factor(tvcats)

aavo = users$avg_absolute_vote_order
aavo[aavo == 1]= 1
aavo[aavo > 1 & aavo <=10]= 2
aavo[aavo > 10 & aavo <=100]= 3
aavo[aavo > 100 & aavo <=500]= 4
aavo[aavo > 500]= 5
aavof = factor(aavo)

idage = users$id_age
idage[idage <= 20000] = 1
idage[idage >20000 & idage <= 40000] = 2
idage[idage >40000 & idage <= 60000] = 3
idage[idage >60000 & idage <= 80000] = 4
idage[idage >80000] = 5
idagef = factor(idage)

per_link_sub = users$per_link_sub
per_link_sub[per_link_sub > 1] = 1
attach(users)

all_factors = data.frame(tvcatsf, idagef, aavof, per_neg, per_link_sub,
proportion_self_votes)
clara58 = clara(all_factors, 58)

cluster = clara58$clustering


#compute std vars
sd_tv = (tot_votes-mean(tot_votes))/sd(tot_votes)
sd_perneg = (per_neg - mean(per_neg))/sd(per_neg)
sd_age = (id_age - mean(id_age))/sd(id_age)
sd_aavo = (avg_absolute_vote_order -
mean(avg_absolute_vote_order))/sd(avg_absolute_vote_order)
sd_ls = (per_link_sub - mean(per_link_sub))/sd(per_link_sub)
sd_psv = (proportion_self_votes -
mean(proportion_self_votes))/sd(proportion_self_votes)
std_factors = data.frame(user_id, sd_tv, sd_perneg, sd_age, sd_aavo, sd_ls, sd_psv)


#compute wk for the cluster9 solution with 58 clusters
wkdata = data.frame(std_factors, cluster)
detach(users)
attach(wkdata)

wk_58clusters = 0
```

```
for (y in wkdata$user_id){
x = cluster[wkdata$user_id ==y]
wki = sum(((wkdata$sd_tv[wkdata$user_id == y] - (median(wkdata$sd_tv[wkdata$cluster
== x])))^2), ((wkdata$sd_age[wkdata$user_id == y] -
(median(wkdata$sd_age[wkdata$cluster == x])))^2), ((wkdata$sd_aavo[user_id == y] -
(median(wkdata$sd_aavo[wkdata$cluster == x])))^2), ((wkdata$sd_perneg[wkdata$user_id
== y] - (median(wkdata$sd_perneg[wkdata$cluster == x])))^2),
((wkdata$sd_ls[wkdata$user_id == y] - (median(wkdata$sd_ls[wkdata$cluster ==
x])))^2), ((wkdata$sd_psv[wkdata$user_id == y] -
(median(wkdata$sd_psv[wkdata$cluster == x])))^2))

wk_58clusters = wk_58clusters + wki
}

wk_58clusters
write.csv(wk_58clusters, 'wk for 58 clusters.csv')


#clustered on raw data, WK worked out from standardised values.
detach(wkdata)

raw_data = data.frame(users$tot_votes, users$id_age, users$avg_absolute_vote_order,
users$per_neg, users$per_link_sub, users$proportion_self_votes)
clara_raw58 = clara(raw_data, 58)

cluster = clara_raw58$clustering



wkdata2 = data.frame(std_factors, cluster)
attach(wkdata2)

wk_raw58clusters = 0
for (y in wkdata2$user_id){
x = cluster[wkdata2$user_id ==y]
wki = sum(((wkdata2$sd_tv[wkdata2$user_id == y] -
(median(wkdata2$sd_tv[wkdata2$cluster == x])))^2), ((wkdata2$sd_age[wkdata2$user_id
== y] - (median(wkdata2$sd_age[wkdata2$cluster == x])))^2),
((wkdata2$sd_aavo[user_id == y] - (median(wkdata2$sd_aavo[wkdata2$cluster ==
x])))^2), ((wkdata2$sd_perneg[wkdata2$user_id == y] -
(median(wkdata2$sd_perneg[wkdata2$cluster == x])))^2),
((wkdata2$sd_ls[wkdata2$user_id == y] - (median(wkdata2$sd_ls[wkdata2$cluster ==
x])))^2), ((wkdata2$sd_psv[wkdata2$user_id == y] -
(median(wkdata2$sd_psv[wkdata2$cluster == x])))^2))
wk_raw58clusters = wk_raw58clusters + wki
}

wk_raw58clusters
write.csv(wk_raw58clusters, 'wk for raw data 58 clusters.csv')

#clustered on raw data, WK worked out from standardised values.
detach(wkdata2)

std_data = data.frame(users$tot_votes, users$id_age, users$avg_absolute_vote_order,
users$per_neg, users$per_link_sub, users$proportion_self_votes)
clara_std58 = clara(std_factors, 58)

cluster = clara_std58$clustering



wkdata3 = data.frame(std_factors, cluster)
attach(wkdata3)

wk_std58clusters = 0
for (y in wkdata3$user_id){
x = cluster[wkdata3$user_id ==y]
```

```
wki = sum(((wkdata3$sd_tv[wkdata3$user_id == y] -
(median(wkdata3$sd_tv[wkdata3$cluster == x])))^2), ((wkdata3$sd_age[wkdata3$user_id
== y] - (median(wkdata3$sd_age[wkdata3$cluster == x])))^2),
((wkdata3$sd_aavo[user_id == y] - (median(wkdata3$sd_aavo[wkdata3$cluster ==
x])))^2), ((wkdata3$sd_perneg[wkdata3$user_id == y] -
(median(wkdata3$sd_perneg[wkdata3$cluster == x])))^2),
((wkdata3$sd_ls[wkdata3$user_id == y] - (median(wkdata3$sd_ls[wkdata3$cluster ==
x])))^2), ((wkdata3$sd_psv[wkdata3$user_id == y] -
(median(wkdata3$sd_psv[wkdata3$cluster == x])))^2))
wk_std58clusters = wk_std58clusters + wki
}

wk_std58clusters
write.csv(wk_std58clusters, 'wk for std data 58 clusters.csv')
```

Appendix F – Samples from the data-tables analysed

Figure F-1: 30 entries from the Votes table (3,446,522 cases total) – Selected variables

Vote ID	User ID	User Type	Link ID	Link Type	SR ID	Self	vote	day	hour	Seconds since Link post	Mins since Link post	Vote order	Vote order %
****	571*****	48	13853021	20	4594350	0	1	31	15	23060	384.33	110	0.7586
****	571*****	10	13848909	42	1058648	0	1	31	16	48474	807.9	485	0.7326
****	571*****	43	13858037	11	4594350	0	1	31	17	354	5.9	3	0.2143
****	571*****	17	13851248	44	3949442	0	1	31	15	30949	515.82	1455	0.7136
****	571*****	48	13854569	14	6	0	-1	31	15	15805	263.42	7	1
****	571*****	37	13843358	46	4594459	0	0	31	16	91175	1519.58	360	0.9783
****	571*****	10	13852994	44	4594431	0	1	31	16	25167	419.45	380	0.4021
****	571*****	10	13857369	13	4594679	0	-1	31	16	2715	45.25	8	0.2286
****	571*****	10	13851248	44	3949442	0	1	31	16	33943	565.72	1565	0.7675
****	571*****	22	13846141	44	4594348	0	1	31	17	73149	1219.15	1165	0.9388
****	571*****	26	13850051	44	4594537	0	1	31	17	42118	701.97	735	0.7853
****	571*****	9	13851825	44	4594362	0	-1	31	17	33854	564.23	660	0.7621
****	571*****	23	13846420	35	3949442	0	1	31	17	73955	1232.58	434	0.9602
****	571*****	9	13857040	44	4594350	0	1	31	19	14838	247.3	542	0.4991
****	571*****	40	13859219	10	6	0	1	31	19	0	0	1	0.2
****	571*****	40	13859226	3	4594362	0	1	31	19	0	0	1	1
****	571*****	30	13854263	32	4594539	0	1	31	19	30536	508.93	142	0.8068
****	571*****	15	13854345	35	1058648	0	1	31	15	16866	281.1	121	0.2769
****	571*****	10	13846141	44	4594348	0	1	31	15	68670	1144.5	1113	0.8969
****	571*****	43	13848551	11	4594348	0	1	31	15	49249	820.82	11	0.7857
****	571*****	15	13848783	44	4594350	0	1	31	15	47211	786.85	1117	0.8323
****	571*****	21	13857397	29	113928	0	1	31	15	0	0	1	0.3333
****	571*****	26	13857244	10	6	0	1	31	15	1180	19.67	4	0.5
****	571*****	15	13850716	12	4594449	0	1	31	15	33367	556.12	15	0.7895
****	571*****	51	13856633	36	4594374	1	1	31	15	4963	82.72	6	0.4
****	571*****	26	13855171	18	4594350	0	1	31	15	12906	215.1	22	0.6286
****	571*****	32	13855143	44	4594323	0	1	31	15	12975	216.25	324	0.3491
****	571*****	15	13856339	18	4594537	0	1	31	15	6872	114.53	5	0.1087
****	571*****	42	13855143	44	4594323	0	1	31	15	13172	219.53	338	0.3642
****	571*****	26	13857421	23	4594679	0	1	31	15	0	0	1	0.1667

Figure F-2: 30 entries from the Users table (102,232 cases total) – Selected variables

User ID	User type	Sub-User type	ID age	total votes	+ votes	- votes	null votes	link submissions	% positive	% negative	% null	% link submission	Aggregate votes	avg vote order	% self votes
****	24	14	51171	1	1	0	0	1	1	0	0	1	1	0.5	0
****	20	0	30789	4	4	0	0	0	1	0	0	0	4	0.451	0
****	27	17	9122	7	6	1	0	1	0.8571	0.1429	0	0.1429	5	0.252	0.1429
****	24	14	56172	1	1	0	0	1	1	0	0	1	1	1	0
****	28	0	30790	12	12	0	0	0	1	0	0	0	12	0.452	0.25
****	29	0	51173	13	12	1	0	0	0.9231	0.0769	0	0	11	0.413	0.0769
****	11	0	47738	5	5	0	0	0	1	0	0	0	5	0.487	0
****	30	0	19242	112	76	36	0	0	0.6786	0.3214	0	0	40	0.463	0.1429
****	31	18	21710	37	36	1	0	2	0.973	0.027	0	0.0541	35	0.668	0.2973
****	27	0	6121	6	5	1	0	0	0.8333	0.1667	0	0	4	0.32	0
****	13	0	51174	3	2	1	0	0	0.6667	0.3333	0	0	1	0.753	0.3333
****	32	0	51175	34	27	6	1	0	0.7941	0.1765	0.0294	0	21	0.535	0.2353
****	6	0	16084	1	1	0	0	0	1	0	0	0	1	0.624	0
****	25	15	51158	2	2	0	0	2	1	0	0	1	2	1	0
****	33	0	51176	1	1	0	0	0	1	0	0	0	1	0.059	1
****	11	7	21712	8	7	1	0	3	0.875	0.125	0	0.375	6	0.545	0.375
****	24	14	51177	1	1	0	0	1	1	0	0	1	1	1	0
****	17	0	30791	10	8	1	1	0	0.8	0.1	0.1	0	7	0.527	0
****	31	0	30792	45	33	12	0	0	0.7333	0.2667	0	0	21	0.459	0.0222
****	16	0	8147	4	4	0	0	0	1	0	0	0	4	0.557	0
****	24	14	51178	1	1	0	0	1	1	0	0	1	1	1	0
****	9	0	11011	52	42	9	1	0	0.8077	0.1731	0.0192	0	33	0.493	0.1538
****	8	0	12827	19	15	4	0	0	0.7895	0.2105	0	0	11	0.437	0
****	16	0	1283	3	1	2	0	0	0.3333	0.6667	0	0	-1	0.559	0.3333
****	11	0	47073	9	9	0	0	0	1	0	0	0	9	0.569	0
****	7	0	21714	1	1	0	0	0	1	0	0	0	1	0.277	1
****	31	19	30793	46	18	28	0	1	0.3913	0.6087	0	0.0217	-10	0.552	0.1739
****	24	14	51180	1	1	0	0	1	1	0	0	1	1	0.2	1
****	34	21	82588	184	151	25	8	9	0.8207	0.1359	0.0435	0.0489	126	0.564	0.0543
****	19	22	16892	3	3	0	0	2	1	0	0	0.6667	3	0.4	0

Figure F-3: 30 entries from the Links table (352,902 cases total) – Selected variables

Link ID	Link Type	User ID	User type	Sub User type	Sub-Reddit ID	Total votes	+ votes	- votes	Null votes	Aggregate	% pos	% neg	% null	Self	contro versy	% Contro versy
*****	1	****	42	38	6	2	2	0	0	2	1	0	0	0	0	0
*****	1	****	49	40	6	1	1	0	0	1	1	0	0	0	0	0
*****	1	****	39	27	6	1	1	0	0	1	1	0	0	0	0	0
*****	1	****	21	12	6	1	1	0	0	1	1	0	0	0	0	0
*****	2	****	10	56	4594318	5	3	2	0	1	0.6	0.4	0	0	2	0.6667
*****	1	****	1	3	6	3	3	0	0	3	1	0	0	0	0	0
*****	1	****	39	27	6	1	1	0	0	1	1	0	0	0	0	0
*****	1	****	49	40	6	1	1	0	0	1	1	0	0	0	0	0
*****	1	****	1	1	6	1	1	0	0	1	1	0	0	0	0	0
*****	3	****	1	1	4594431	1	1	0	0	1	1	0	0	0	0	0
*****	4	****	26	37	6	3	2	1	0	1	0.6667	0.3333	0	0	1	0.5
*****	1	****	40	46	6	1	1	0	0	1	1	0	0	0	0	0
*****	5	****	39	27	4594370	1	1	0	0	1	1	0	0	0	0	0
*****	6	****	10	56	4594362	5	3	2	0	1	0.6	0.4	0	0	2	0.6667
*****	7	****	26	37	4594495	14	14	0	0	14	1	0	0	0	0	0
*****	8	****	15	16	4594537	2	1	1	0	0	0.5	0.5	0	0	1	1
*****	3	****	42	38	4594362	1	1	0	0	1	1	0	0	0	0	0
*****	9	****	15	10	4594519	18	11	7	0	4	0.6111	0.3889	0	0	7	0.6364
*****	1	****	1	26	6	1	1	0	0	1	1	0	0	0	0	0
*****	10	****	32	19	6	9	7	2	0	5	0.7778	0.2222	0	0	2	0.2857
*****	11	****	45	8	4594350	6	5	1	0	4	0.8333	0.1667	0	0	1	0.2
*****	12	****	26	37	4594539	10	8	2	0	6	0.8	0.2	0	0	2	0.25
*****	13	****	26	16	4594300	18	13	5	0	8	0.7222	0.2778	0	0	5	0.3846
*****	3	****	26	37	4594431	1	1	0	0	1	1	0	0	0	0	0
*****	1	****	30	38	6	6	6	0	0	6	1	0	0	0	0	0
*****	1	****	25	15	6	1	1	0	0	1	1	0	0	0	0	0
*****	1	****	49	40	6	1	1	0	0	1	1	0	0	0	0	0
*****	14	****	39	27	6	6	0	6	0	-6	0	1	0	0	0	0
*****	3	****	58	46	4594362	1	1	0	0	1	1	0	0	0	0	0
*****	1	****	40	28	6	1	1	0	0	1	1	0	0	0	0	0

Figure F-4: 30 entries from the Sub-Reddits table (2,184 cases total) – Selected variables

Sub-Reddit ID	SR type	Total Links	Total Votes	Votes per link	Aggregate per link	positive votes	negative votes	null votes	% positive	% negative	%null	% controversy	% Self
4612097	1	2	2	1	-1	0	2	0	0	1	0	0	0
4603906	2	577	5291	9.17	4.88	4030	1215	46	0.7617	0.2296	0.0087	0.32917	0.0052
4610731	3	27	149	5.52	4.48	135	14	0	0.906	0.094	0	0.11038	0.03704
4595716	1	3	3	1	1	3	0	0	1	0	0	0	0
6	0	150042	516775	3.44	1.93	401412	112008	3355	0.7768	0.2167	0.0065	0.1763	0.02178
4595720	3	8	29	3.63	2.63	25	4	0	0.8621	0.1379	0	0.29166	0.25
4595714	3	33	75	2.27	1.7	65	9	1	0.8667	0.12	0.0133	0.14646	0.06061
4612115	1	1	1	1	1	1	0	0	1	0	0	0	0
4612122	1	16	16	1	1	16	0	0	1	0	0	0	0
4612123	4	2	2	1	0.5	1	0	1	0.5	0	0.5	0.5	0
4595743	1	3	3	1	1	3	0	0	1	0	0	0	0
4601842	1	3	3	1	1	3	0	0	1	0	0	0	0
4603939	3	3	4	1.33	0.67	3	1	0	0.75	0.25	0	0.33333	0
4612132	1	3	3	1	1	3	0	0	1	0	0	0	0
4602545	1	4	4	1	1	4	0	0	1	0	0	0	0
4603946	1	1	2	2	2	2	0	0	1	0	0	0	0
4603947	1	1	1	1	1	1	0	0	1	0	0	0	0
4595758	1	6	12	2	2	12	0	0	1	0	0	0	0
4603951	5	41	43	1.05	0.95	41	2	0	0.9535	0.0465	0	0.04878	0.56098
4612104	3	197	831	4.22	3.15	725	105	1	0.8724	0.1264	0.0012	0.15054	0.0203
4612147	1	1	1	1	1	1	0	0	1	0	0	0	0
4612148	6	1	7	7	-3	2	5	0	0.2857	0.7143	0	0.4	1
4612149	1	31	58	1.87	1.65	54	3	1	0.931	0.0517	0.0172	0.06452	0
4603961	1	14	36	2.57	2.57	36	0	0	1	0	0	0	0
4595772	1	1	1	1	1	1	0	0	1	0	0	0	0
4603967	1	7	14	2	2	14	0	0	1	0	0	0	0
4612160	1	1	1	1	1	1	0	0	1	0	0	0	0
4595778	1	1	1	1	1	1	0	0	1	0	0	0	0
4603973	7	1	2	2	0	1	1	0	0.5	0.5	0	1	0
4612167	1	7	7	1	1	7	0	0	1	0	0	0	0