Reddit is a popular website for opinion sharing and news aggregation. The site consists of thousands of user-made forums, called subreddits, which cover a broad range of subjects, including politics, sports, technology, personal hobbies, and self-improvement. Given that most Reddit users contribute to multiple subreddits, one might think of Reddit as being organized into many overlapping communities. Moreover, one might understand the connections among these communities as making up a kind of social structure.

Uncovering a population’s social structure is useful because it tells us something about that population’s identity. In the case of Reddit, this identity could be uncovered by figuring out which subreddits are most central to Reddit’s network of subreddits. We could also study this network at multiple points in time to learn how this identity has evolved and maybe even predict what it’s going to look like in the future.

My goal in this post is to map the social structure of Reddit by measuring the proximity of Reddit communities (subreddits) to each other. I’m operationalizing community proximity as the number of posts to different communities that come from the same user. For example, if a user posts something to subreddit A and posts something else to subreddit B, subreddits A and B are linked by this user. Subreddits connected in this way by many users are closer together than subreddits connected by fewer users. The idea that group networks can be uncovered by studying shared associations among the people that make up those groups goes way back in the field of sociology ([Breiger 1974](https://www.jstor.org/stable/2576011?seq=1#metadata_info_tab_contents)). Hopefully this post will demonstrate the utility of this concept for making sense of data from social media platforms like Reddit.[1](https://datadiarist.github.io/post/mapping-the-underlying-social-structure-of-reddit/#fn1)

**Data**

The data for this post come from an online repository of subreddit submissions and comments that is generously hosted by data scientist Jason Baumgartner. If you plan to download a lot of data from this repository, I implore you to donate a bit of money to keep Baumgartner’s database up and running ([pushshift.io/donations/](https://pushshift.io/donations/)).

Here’s the link to the Reddit submissions data – [files.pushshift.io/reddit/submissions/](http://files.pushshift.io/reddit/submissions/). Each of these files has all Reddit submissions for a given month between June 2005 and May 2019. Files are JSON objects stored in various compression formats that range between .017Mb and 5.77Gb in size. Let’s download something in the middle of this range – a 710Mb file for all Reddit submissions from May 2013. The file is called RS\_2013-05.bz2. You can double-click this file to unzip it, or you can use the following command in the Terminal: bzip2 -d RS\_2013-05.bz2. The file will take a couple of minutes to unzip. Make sure you have enough room to store the unzipped file on your computer – it’s 4.51Gb. Once I have unzipped this file, I load the relevant packages, read the first line of data from the unzipped file, and look at the variable names.

read\_lines("RS\_2013-05", n\_max = 1) %>% fromJSON() %>% names

## [1] "edited" "title"

## [3] "thumbnail" "retrieved\_on"

## [5] "mod\_reports" "selftext\_html"

## [7] "link\_flair\_css\_class" "downs"

## [9] "over\_18" "secure\_media"

## [11] "url" "author\_flair\_css\_class"

## [13] "media" "subreddit"

## [15] "author" "user\_reports"

## [17] "domain" "created\_utc"

## [19] "stickied" "secure\_media\_embed"

## [21] "media\_embed" "ups"

## [23] "distinguished" "selftext"

## [25] "num\_comments" "banned\_by"

## [27] "score" "report\_reasons"

## [29] "id" "gilded"

## [31] "is\_self" "subreddit\_id"

## [33] "link\_flair\_text" "permalink"

## [35] "author\_flair\_text"

For this project, I’m only interested in three of these variables: the user name associated with each submission (author), the subreddit to which a submission has been posted (subreddit), and the time of submission (created\_utc). If we could figure out a way to extract these three pieces of information from each line of JSON we could greatly reduce the size of our data, which would allow us to store multiple months worth of information on our local machine. Jq is a command-line JSON processor that makes this possible.

To install jq on a Mac, you need to make sure you have Homebrew ([brew.sh/](https://brew.sh/)), a package manager that works in the Terminal. Once you have Homebrew, in the Terminal type brew install jq. I’m going to use jq to extract the variables I want from RS\_2015-03 and save the result as a .csv file. To select variables with jq, list the JSON field names that you want like this: [.author, .created\_utc, .subreddit]. I return these as raw output (-r) and render this as a csv file (@csv). Here’s the command that does all this:

jq -r '[.author, .created\_utc, .subreddit] | @csv' RS\_2013-05 > parsed\_json\_to\_csv\_2013\_05

Make sure the Terminal directory is set to wherever RS\_2013-05 is located before running this command. The file that results from this command will be saved as “parsed\_json\_to\_csv\_2013\_05”. This command parses millions of lines of JSON (every Reddit submission from 05-2013), so this process can take a few minutes. In case you’re new to working in the Terminal, if there’s a blank line at the bottom of the Terminal window, that means the process is still running. When the directory name followed by a dollar sign reappears, the process is complete. This file, parsed\_json\_to\_csv\_2013\_05, is about 118Mb, much smaller than 4.5Gb.

Jq is a powerful tool for automating the process of downloading and manipulating data right from your harddrive. I’ve written the a bash script that lets you download multiple files from the Reddit repository, unzip them, extract the relevant fields from the resulting JSON, and delete the unparsed files ([Reddit\_Download\_Script.bash](https://github.com/datadiarist/Reddit-Analysis-Files/blob/master/Reddit_Download_Script.bash)). You can modify this script to pull different fields from the JSON. For instance, if you want to keep the content of Reddit submissions, add .selftext to the fields that are included in the brackets.

Now that I have a reasonably sized .csv file with the fields I want, I am ready to bring the data into R and analyze them as a network.

**Analysis**

Each row of the data currently represents a unique submission to Reddit from a user. I want to turn this into a dataframe where each row represents a link between subreddits through a user. One problem that arises from this kind of data manipulation is that there are more rows in the network form of this data than there are in the current form of the data. To see this, consider a user who has submitted to 10 different subreddits. These submissions would take up ten rows of our dataframe in its current form. However, this data would be represented by 10 choose 2, or 45, rows of data in its network form. This is every combination of 2 subreddits among those to which the user has posted. This number gets exponentially larger as the number of submissions from the same user increases. For this reason, the only way to convert the data into a network form without causing R to crash is to convert the data into a Spark dataframe. Spark is a distributed computing platform that partitions large datasets into smaller chunks and operates on these chunks in parallel. If your computer has a multicore processor, Spark allows you to work with big-ish data on your local machine. I will be using a lot of functions from the sparklyr package, which supplies dplyr backend to Spark. If you’re new to Spark and sparklyr, check out RStudio’s guide for getting started with Spark in R ([spark.rstudio.com/](https://spark.rstudio.com/)).

Once I have Spark configured, I import the data into R as a Spark dataframe.

reddit\_data <- spark\_read\_csv(sc, "parsed\_json\_to\_csv\_2013\_05",

header = FALSE)

To begin, I make a few changes to the data – renaming columns, converting the time variable from utc time to the day of the year, and removing submissions from deleted accounts. I also remove submissions from users who have posted only once – these would contribute nothing to the network data – and submissions from users who have posted 60 or more times – these users are likely bots.

reddit\_data <- reddit\_data %>%

rename(author = V1, created\_utc = V2, subreddit = V3) %>%

mutate(dateRestored = timestamp(created\_utc + 18000)) %>%

mutate(day = dayofyear(dateRestored)) %>%

filter(author != "[deleted]") %>% group\_by(author) %>% mutate(count = count()) %>%

filter(count < 60 & count > 1) %>%

ungroup()

Next, I create a key that gives a numeric id to each subreddit. I add these ids to the data, and select the variables “author”, “day”, “count”, “subreddit”, and “id” from the data. Let’s have a look at the first few rows of the data.

subreddit\_key <- reddit\_data %>% distinct(subreddit) %>% sdf\_with\_sequential\_id()

reddit\_data <- left\_join(reddit\_data, subreddit\_key, by = "subreddit") %>%

select(author, day, count, subreddit, id)

head(reddit\_data)

## # Source: spark [?? x 5]

## author day count subreddit id

##

## 1 Bouda 141 4 100thworldproblems 2342

## 2 timeXalchemist 147 4 100thworldproblems 2342

## 3 babydall1267 144 18 123recipes 2477

## 4 babydall1267 144 18 123recipes 2477

## 5 babydall1267 144 18 123recipes 2477

## 6 babydall1267 144 18 123recipes 2477

We have 5 variables. The count variable shows the number of times a user has posted to Reddit in May 2013, the id variable gives the subreddit’s numeric id, the day variable tells us what day of the year a submission has been posted, and the author and subreddit variables give user and subreddit names. We are now ready to convert this data to network format. The first thing I do is take an “inner\_join” of the data with itself, merging by the “author” variable. For each user, the number of rows this returns will be the square of the number of submissions from that user. I filter this down to “number of submissions choose 2” rows for each user. This takes two steps. First, I remove rows that link subreddits to themselves. Then I remove duplicate rows. For instance, AskReddit-funny is a duplicate of funny-AskReddit. I remove one of these.

The subreddit id variable will prove useful for removing duplicate rows. If we can mutate two id variables into a new variable that gives a unique identifier to each subreddit pair, we can filter duplicates of this identifier. We need a mathematical equation that takes two numbers and returns a unique number (i.e. a number that can only be produced from these two numbers) regardless of number order. One such equation is the Cantor Pairing Function ([wikipedia.org/wiki/Pairing\_function](https://en.wikipedia.org/wiki/Pairing_function)):

Let’s define a function in R that takes a dataframe and two id variables, runs the id variables through Cantor’s Pairing Function and appends this to the dataframe, filters duplicate cantor ids from the dataframe, and returns the result. We’ll call this function cantor\_filter.

cantor\_filter <- function(df, id, id2){

df %>% mutate(id\_pair = .5\*(id + id2)\*(id + id2 + 1) + pmax(id, id2)) %>% group\_by(author, id\_pair) %>%

filter(row\_number(id\_pair) == 1) %>% return()

}

Next, I apply an inner\_join to the Reddit data and apply the filters described above to the resulting dataframe.

reddit\_network\_data <- inner\_join(reddit\_data, reddit\_data %>%

rename(day2 = day, count2 = count,

subreddit2 = subreddit, id2 = id),

by = "author") %>%

filter(subreddit != subreddit2) %>%

group\_by(author, subreddit, subreddit2) %>%

filter(row\_number(author) == 1) %>%

cantor\_filter() %>%

select(author, subreddit, subreddit2, id, id2, day, day2, id\_pair) %>%

ungroup %>% arrange(author)

Let’s take a look at the new data.

reddit\_network\_data

## Warning: `lang\_name()` is deprecated as of rlang 0.2.0.

## Please use `call\_name()` instead.

## This warning is displayed once per session.

## Warning: `lang()` is deprecated as of rlang 0.2.0.

## Please use `call2()` instead.

## This warning is displayed once per session.

## # Source: spark [?? x 8]

## # Ordered by: author

## author subreddit subreddit2 id id2 day day2 id\_pair

##

## 1 --5Dhere depression awakened 7644 29936 135 135 7.06e8

## 2 --Adam-- AskReddit techsupport 15261 28113 135 142 9.41e8

## 3 --Caius-- summonerscho… leagueoflegen… 79 3 124 142 3.48e3

## 4 --Gianni-- AskReddit videos 15261 5042 125 138 2.06e8

## 5 --Gianni-- pics AskReddit 5043 15261 126 125 2.06e8

## 6 --Gianni-- movies pics 20348 5043 124 126 3.22e8

## 7 --Gianni-- gaming videos 10158 5042 131 138 1.16e8

## 8 --Gianni-- gaming pics 10158 5043 131 126 1.16e8

## 9 --Gianni-- movies AskReddit 20348 15261 124 125 6.34e8

## 10 --Gianni-- movies videos 20348 5042 124 138 3.22e8

## # … with more rows

We now have a dataframe where each row represents a link between two subreddits through a distinct user. Many pairs of subreddits are connected by multiple users. We can think of subreddit pairs connected through more users as being more connected than subreddit pairs connected by fewer users. With this in mind, I create a “weight” variable that tallies the number of users connecting each subreddit pair and then filters the dataframe to unique pairs.

reddit\_network\_data <- reddit\_network\_data %>% group\_by(id\_pair) %>%

mutate(weight = n()) %>% filter(row\_number(id\_pair) == 1) %>%

ungroup

Let’s have a look at the data and see how many rows it has.

reddit\_network\_data

## # Source: spark [?? x 9]

## # Ordered by: author

## author subreddit subreddit2 id id2 day day2 id\_pair weight

##

## 1 h3rbivore psytrance DnB 8 2 142 142 63 1

## 2 StRefuge findareddit AlienBlue 23 5 133 134 429 1

## 3 DylanTho blackops2 DnB 28 2 136 138 493 2

## 4 TwoHardCo… bikewrench DnB 30 2 137 135 558 1

## 5 Playbook4… blackops2 AlienBlue 28 5 121 137 589 2

## 6 A\_Jewish\_… atheism blackops2 6 28 139 149 623 14

## 7 SirMechan… Terraria circlejerk 37 7 150 143 1027 2

## 8 Jillatha doctorwho facebookw… 36 9 131 147 1071 2

## 9 MeSire Ebay circlejerk 39 7 132 132 1120 3

## 10 Bluesfan6… SquaredCir… keto 29 18 126 134 1157 2

## # … with more rows

reddit\_network\_data %>% sdf\_nrow

## [1] 744939

We’re down to ~750,000 rows. The weight column shows that many of the subreddit pairs in our data are only connected by 1 or 2 users. We can substantially reduce the size of the data without losing the subreddit pairs we’re interested in by removing these rows. I decided to remove subreddit pairs that are connected by 3 or fewer users. I also opt at this point to stop working with the data as a Spark object and bring the data into the R workspace as a dataframe. The network analytic tools I use next require working on a regular dataframes and our data is now small enough that we can do this without any problems. Because we’re moving into the R workspace, I save this as a new dataframe called reddit\_edgelist.

reddit\_edgelist <- reddit\_network\_data %>% filter(weight > 3) %>%

select(id, id2, weight) %>% arrange(id) %>%

# Bringing the data into the R workspace

dplyr::collect()

Our R dataframe consists of three columns: two id columns that provide information on connections between nodes and a weight column that tells us the strength of each connection. One nice thing to have would be a measure of the relative importance of each subreddit. A simple way to get this would be to count how many times each subreddit appears in the data. I compute this for each subreddit by adding the weight values in the rows where that subreddit appears. I then create a dataframe called subreddit\_imp\_key that lists subreddit ids by subreddit importance.

subreddit\_imp\_key <- full\_join(reddit\_edgelist %>% group\_by(id) %>%

summarise(count = sum(weight)),

reddit\_edgelist %>% group\_by(id2) %>%

summarise(count2 = sum(weight)),

by = c("id" = "id2")) %>%

mutate(count = ifelse([is.na](http://is.na)(count), 0, count)) %>%

mutate(count2 = ifelse([is.na](http://is.na)(count2), 0, count2)) %>%

mutate(id = id, imp = count + count2) %>% select(id, imp)

Let’s see which subreddits are the most popular on Reddit according to the subreddit importance key.

left\_join(subreddit\_imp\_key, subreddit\_key %>% dplyr::collect(), by = "id") %>%

arrange(desc(imp))

## # A tibble: 5,561 x 3

## id imp subreddit

##

## 1 28096 107894 funny

## 2 15261 101239 AskReddit

## 3 20340 81208 AdviceAnimals

## 4 5043 73119 pics

## 5 10158 51314 gaming

## 6 5042 47795 videos

## 7 17856 47378 aww

## 8 2526 37311 WTF

## 9 22888 31702 Music

## 10 5055 26666 todayilearned

## # … with 5,551 more rows

These subreddits are mostly about memes and gaming, which are indeed two things that people commonly associate with Reddit.

Next, I reweight the edge weights in reddit\_edgelist by subreddit importance. The reason I do this is that the number of users connecting subreddits is partially a function of subreddit popularity. Reweighting by subreddit importance, I control for the influence of this confounding variable.

reddit\_edgelist <- left\_join(reddit\_edgelist, subreddit\_imp\_key,

by = c("id" = "id")) %>%

left\_join(., subreddit\_imp\_key %>% rename(imp2 = imp),

by = c("id2" = "id")) %>%

mutate(imp\_fin = (imp + imp2)/2) %>% mutate(weight = weight/imp\_fin) %>%

select(id, id2, weight)

reddit\_edgelist

## # A tibble: 56,257 x 3

## id id2 weight

##

## 1 1 12735 0.0141

## 2 1 10158 0.000311

## 3 1 2601 0.00602

## 4 1 17856 0.000505

## 5 1 22900 0.000488

## 6 1 25542 0.0185

## 7 1 15260 0.00638

## 8 1 20340 0.000320

## 9 2 2770 0.0165

## 10 2 15261 0.000295

## # … with 56,247 more rows

We now have our final edgelist. There are about 56,000 thousand rows in the data, though most edges have very small weights. Next, I use the igraph package to turn this dataframe into a graph object. Graph objects can be analyzed using igraph’s clustering algorithms. Let’s have a look at what this graph object looks like.

reddit\_graph <- graph\_from\_data\_frame(reddit\_edgelist, directed = FALSE)

reddit\_graph

## IGRAPH 2dc5bc4 UNW- 5561 56257 --

## + attr: name (v/c), weight (e/n)

## + edges from 2dc5bc4 (vertex names):

## [1] 1--12735 1--10158 1--2601 1--17856 1--22900 1--25542 1--15260

## [8] 1--20340 2--2770 2--15261 2--18156 2--20378 2--41 2--22888

## [15] 2--28115 2--10172 2--5043 2--28408 2--2553 2--2836 2--28096

## [22] 2--23217 2--17896 2--67 2--23127 2--2530 2--2738 2--7610

## [29] 2--20544 2--25566 2--3 2--7 2--7603 2--12931 2--17860

## [36] 2--6 2--2526 2--5055 2--18253 2--22996 2--25545 2--28189

## [43] 2--10394 2--18234 2--23062 2--25573 3--264 3--2599 3--5196

## [50] 3--7585 3--10166 3--10215 3--12959 3--15293 3--20377 3--20427

## + ... omitted several edges

Here we have a list of all of the edges from the dataframe. I can now use a clustering algorithm to analyze the community structure that underlies this subreddit network. The clustering algorithm I choose to use here is the Louvain algorithm. This algorithm takes a network and groups its nodes into different communities in a way that maximizes the modularity of the resulting network. By maximizing modularity, the Louvain algorithm groups nodes in a way that maximizes the number of within-group ties and minimizes the number of between-group ties.

Let’s apply the algorithm and see if the groupings it produces make sense. I store the results of the algorithm in a tibble with other relevant information. See code annotations for a more in-depth explanation of what I’m doing here.

reddit\_communities <- cluster\_louvain(reddit\_graph, weights = reddit\_edgelist$weight)

subreddit\_by\_comm <- tibble(

# Using map from purrr to extract subreddit ids from reddit\_communities

id = map(reddit\_communities[], as.numeric) %>% unlist,

# Creating a community ids column and using rep function with map to populate

# a column with community ids created by

# Louvain alg

comm = rep(reddit\_communities[] %>%

names, map(reddit\_communities[], length) %>% unlist) %>%

as.numeric) %>%

# Adding subreddit names

left\_join(., subreddit\_key %>% dplyr::collect(), by = "id") %>%

# Keeping subreddit name, subreddit id, community id

select(subreddit, id, comm) %>%

# Adding subreddit importance

left\_join(., subreddit\_imp\_key, by = "id")

Next, I calculate community importance by summing the subreddit importance scores of the subreddits in each community.

subreddit\_by\_comm <- subreddit\_by\_comm %>% group\_by(comm) %>% mutate(comm\_imp = sum(imp)) %>% ungroup

I create a tibble of the 10 most important communities on Reddit according to the subreddit groupings generated by the Louvain algorithm. This tibble displays 10 largest subreddits in each of these communities. Hopefully, these subreddits will be similar enough that we can discern what each community represents.

comm\_ids <- subreddit\_by\_comm %>% group\_by(comm) %>% slice(1) %>% arrange(desc(comm\_imp)) %>% .[["comm"]]

top\_comms <- list()

for(i in 1:10){

top\_comms[[i]] <- subreddit\_by\_comm %>% filter(comm == comm\_ids[i]) %>% arrange(desc(imp)) %>% .[["subreddit"]] %>% .[1:10]

}

comm\_tbl <- tibble(Community = 1:10,

Subreddits = map(top\_comms, ~paste(.x, collapse = " ")) %>% unlist)

Let’s have a look at the 10 largest subreddits in each of the 10 largest communities. These are in descending order of importance.

options(kableExtra.html.bsTable = TRUE)

comm\_tbl %>%

kable("html") %>%

kable\_styling("hover", full\_width = F) %>%

column\_spec(1, bold = T, border\_right = "1px solid #ddd;") %>%

column\_spec(2, width = "30em")

| **Community** | **Subreddits** |
| --- | --- |
| **1** | funny AskReddit AdviceAnimals pics gaming videos aww WTF Music todayilearned |
| **2** | DotA2 tf2 SteamGameSwap starcraft tf2trade Dota2Trade GiftofGames SteamTradingCards Steam vinyl |
| **3** | electronicmusic dubstep WeAreTheMusicMakers futurebeats trap edmproduction electrohouse EDM punk ThisIsOurMusic |
| **4** | hockey fantasybaseball nhl Austin DetroitRedWings sanfrancisco houston leafs BostonBruins mlb |
| **5** | cars motorcycles Autos sysadmin carporn formula1 Jeep subaru Cartalk techsupportgore |
| **6** | web\_design Entrepreneur programming webdev Design windowsphone SEO forhire startups socialmedia |
| **7** | itookapicture EarthPorn AbandonedPorn HistoryPorn photocritique CityPorn MapPorn AnimalPorn SkyPorn Astronomy |
| **8** | wow darksouls Diablo Neverwinter Guildwars2 runescape diablo3 2007scape swtor Smite |
| **9** | blackops2 battlefield3 dayz Eve Planetside aviation airsoft WorldofTanks Warframe CallOfDuty |
| **10** | soccer Seattle Fifa13 Portland MLS Gunners reddevils chelseafc football LiverpoolFC |

The largest community in this table, community 1, happens to contain the ten most popular subreddits on Reddit. Although some of these subreddits are similar in terms of their content – many of them revolve around memes, for example – a couple of them do not (e.g. videos and gaming). One explanation is that this first group of subreddits represents mainstream Reddit. In other words, the people who post to these subreddits are generalist posters – they submit to a broad enough range of subreddits that categorizing these subreddits into any of the other communities would reduce the modularity of the network.

The other 9 communities in the figure are easier to interpret. Each one revolves around a specific topic. Communities 2, 8, and 9 are gaming communities dedicated to specific games; communities 4 and 10 are sports communities; the remaining communities are dedicated to electronic music, cars, web design, and photography.

In sum, we have taken a month worth of Reddit submissions, converted them into a network, and identified subreddit communities from them. How successful were we? On one hand, the Louvain algorithm correctly identified many medium-sized communities revolving around specific topics. It’s easy to imagine that the people who post to these groups of subreddits contribute almost exclusively to them, and that it therefore makes sense to think of them as communities. On the other hand, the largest community has some pretty substantively dissimilar subreddits. These also happen to be the largest subreddits on Reddit. The optimistic interpretation of this grouping is that these subreddits encompass a community of mainstream users. However, the alternative possibly that this community is really just a residual category of subreddits that don’t really belong together but also don’t have any obvious place in the other subreddit communities. Let’s set this issue to the side for now.

In the next section, I visualize these communities as a community network and examine how this network has evolved over time.

**Visualizations**

In the last section, I generated some community groupings of subreddits. While these give us some idea of the social structure of Reddit, one might want to know how these communities are connected to each other. In this section, I take these community groupings and build a community-level network from them. I then create some interactive visualizations that map the social structure of Reddit and show how this structure has evolved over time.

The first thing I want to do is return to the subreddit edgelist, our dataframe of subreddit pairs and the strength of their connections, and merge this with community id variables corresponding to each subreddit. I filter the dataframe to only include unique edges, and add a variable called weight\_fin, which is the average of the subreddit edge weights between each community. I also filter links in the community-level edgelist that connect community to themselves. I realize that there’s a lot going on in the code below. Feel free to contact me if you have any questions about what I’m doing here.

community\_edgelist <- left\_join(reddit\_edgelist, subreddit\_by\_comm %>% select(id, comm), by = "id") %>%

left\_join(., subreddit\_by\_comm %>% select(id, comm) %>% rename(comm2 = comm), by = c("id2"= "id")) %>%

select(comm, comm2, weight) %>%

mutate(id\_pair = .5\*(comm + comm2)\*(comm + comm2 + 1) + pmax(comm,comm2)) %>% group\_by(id\_pair) %>%

mutate(weight\_fin = mean(weight)) %>% slice(1) %>% ungroup %>% select(comm, comm2, weight\_fin) %>%

filter(comm != comm2) %>% filter(comm != comm2) %>%

arrange(desc(weight\_fin))

I now have a community-level edgelist, with which we can visualize a network of subreddit communities. I first modify the edge weight variable to discriminate between communities that are more and less connected. I choose an arbitrary cutoff point (.007) and set all weights below this cutoff to 0. Although doing this creates a risk of imposing structure on the network where there is none, this cutoff will help highlight significant ties between communities.

community\_edgelist\_ab <- community\_edgelist %>%

mutate(weight = ifelse(weight\_fin > .007, weight\_fin, 0)) %>%

filter(weight!=0) %>% mutate(weight = abs(log(weight)))

Visnetwork’s main function, visNetwork, requires two arguments, one for nodes data and one for edges data. These dataframes need to have particular column names for visnetwork to be able to make sense of them. Let’s start with the edges data. The column names for the nodes corresponding to edges in the edgelist need to be called “from” and “to”, and the column name for edge weights needs to be called “weight”. I make these adjustments.

community\_edgelist\_mod <- community\_edgelist\_ab %>%

rename(from = comm, to = comm2) %>% select(from, to, weight)

Also, visnetwork’s default edges are curved. I prefer straight edges. To ensure edges are straight, add a smooth column and set it to FALSE.

community\_edgelist\_mod$smooth <- F

I’m now ready to set up the nodes data. First, I extract all nodes from the community edgelist.

community\_nodes <- c(community\_edgelist\_mod %>% .[["from"]], community\_edgelist\_mod %>% .[["to"]]) %>% unique

Visnetwork has this really cool feature that lets you view node labels by hovering over them with your mouse cursor. I’m going to label each community with the names of the 4 most popular subreddits in that community.

comm\_by\_label <- subreddit\_by\_comm %>% arrange(comm, desc(imp)) %>% group\_by(comm) %>% slice(1:4) %>%

summarise(title = paste(subreddit, collapse = " "))

Next, I put node ids and community labels in a tibble. Note that the label column in this tibble has to be called “title”.

community\_nodes\_fin <- tibble(comm = community\_nodes) %>% left\_join(., comm\_by\_label, by = "comm")

I want the nodes of my network to vary in size based on the size of each community. To do this, I create a community importance key. I’ve already calculated community importance above. I extract this score for each community from the subreddit\_by\_comm dataframe and merge these importance scores with the nodes data. I rename the community importance variable “size” and the community id variable “id”, which are the column names that visnetwork recognizes.

comm\_imp\_key <- subreddit\_by\_comm %>% group\_by(comm) %>% slice(1) %>%

arrange(desc(comm\_imp)) %>% select(comm, comm\_imp)

community\_nodes\_fin <- inner\_join(community\_nodes\_fin, comm\_imp\_key, by = "comm") %>%

rename(size = comm\_imp, id = comm)

One final issue is that my “mainstream Reddit/residual subreddits” community is so much bigger than the other communities that the network visualization will be overtaken by it if I don’t adjust the size variable. I remedy this by raising community size to the .3th power (close to the cube root).

community\_nodes\_fin <- community\_nodes\_fin %>% mutate(size = size^.3)

I can now enter the nodes and edges data into the visNetwork function. I make a few final adjustments to the default parameters. Visnetwork now lets you use layouts from the igraph package. I use visIgraphLayout to set the position of the nodes according to the Fruchterman-Reingold Layout Algorithm (layout\_with\_fr). I also adjust edge widths and set highlightNearest to TRUE. This lets you highlight a node and the nodes it is connected to by clicking on it. Without further ado, let’s have a look at the network.

The communities of Reddit do not appear to be structured into distinct categories. We don’t see a cluster of hobby communities and a different cluster of advice communities, for instance. Instead, we have some evidence to suggest that the strongest ties are among some of the larger subcultures of Reddit. Many of the nodes in the large cluster of communities above are ranked in the 2-30 range in terms of community size. On the other hand, the largest community (mainstream Reddit) is out on a island, with only a few small communities around it. This suggests that the ties between mainstream Reddit and some of Reddit’s more niche communities are weaker than the ties among the latter. In other words, fringe subcultures of Reddit are more connected to each other than they are to Reddit’s mainstream.

The substance of these fringe communities lends credence to this interpretation. Many of the communities in the large cluster are somewhat related in their content. There are a lot of gaming communities, several drug and music communities, a couple of sports communities, and few communities that combine gaming, music, sports, and drugs in different ways. Indeed, most of the communities in this cluster revolve around activities commonly associated with young men. One might even infer from this network that Reddit is organized into two social spheres, one consisting of adolescent men and the other consisting of everybody else. Still, I should caution the reader against extrapolating too much from the network above. These ties are based on 30 days of submissions. It’s possible that something occurred during this period that momentarily brought certain Reddit communities closer together than they would be otherwise. There are links among some nodes in the network that don’t make much logical sense. For instance, the linux/engineering/3D-Printing community (which only sort of makes sense as a community) is linked to a “guns/knives/coins” community. This strikes me as a bit strange, and I wonder if these communities would look the same if I took data from another time period. Still, many of the links here make a lot of sense. For example, the Bitcoin/Conservative/Anarcho\_Capitalism community is tied to the Anarchism/progressive/socialism/occupywallstreet community. The Drugs/bodybuilding community is connected to the MMA/Joe Rogan community. That one makes almost too much sense. Anyway, I encourage you to click on the network nodes to see what you find.

One of the coolest things about the Reddit repository is that it contains temporally precise information on everything that’s happened on Reddit from its inception to only a few months ago. In the final section of this post, I rerun the above analyses on all the Reddit submissions from May 2017 and May 2019. I’m using the bash script I linked to above to do this. Let’s have a look at the community networks from 2017 and 2019 and hopefully gain some insight into how Reddit has evolved over the past several years.

Perhaps owing the substantial growth of Reddit between 2013 and 2017, we start to see a hierarchical structure among the communities that we didn’t see in the previous network. A few of the larger communities now have smaller communities budding off of them. I see four such “parent communities”. One of them is the music community. There’s a musicals/broadway community, a reggae community, an anime music community, and a “deepstyle” (whatever that is) community stemming from this. Another parent community is the sports community, which has a few location-based communities, a lacrosse community, and a Madden community abutting it. The other two parent communities are porn communities. I won’t name the communities stemming from these, but as you might guess many of them revolve around more niche sexual interests.

This brings us to another significant change between this network and the one from 2013: the emergence of porn on Reddit. We now see that two of the largest communities involve porn. We also start to see some differentiation among the porn communities. There is a straight porn community, a gay porn community, and a sex-based kik community (kik is a messenger app). It appears that since 2013 Reddit is increasingly serving some of the same functions as Craigslist, providing users with a place to arrange to meet up, either online or in person, for sex. As we’ll see in the 2019 network, this function has only continued to grow. This is perhaps due to the Trump Administration’s sex trafficking bill and Craigslist’s decision to shutdown its “casual encounters” personal ads in 2018.

Speaking of Donald Trump, where is he in our network? As it turns out, this visualization belies the growing presence of Donald Trump on Reddit between 2013 and 2017. The\_Donald is a subreddit for fans of Donald Trump that quickly became of the most popular subreddits on Reddit during this time. The reason that we don’t see it here is that it falls into the mainstream Reddit community, and despite its popularity it is not one of the four largest subreddits in this community. The placement of The\_Donald in this community was one of the most surprising results of this project. I had expected The\_Donald to fall into a conservative political community. The reason The\_Donald falls into the mainstream community, I believe, is that much of The\_Donald consists of news and memes, the bread and butter of Reddit. Many of the most popular subreddits in the mainstream community are meme subreddits – Showerthoughts, drankmemes, funny – and the overlap between users who post to these subreddits and users who post to The\_Donald is substantial.

That brings us to May 2019. What’s changed from 2017? The network structure is similar – we have two groups, mainstream Reddit and a interconnected cluster of more niche communities. This cluster has the same somewhat hierarchical structure that we saw in the 2017 network, with a couple of large “parent communities” that are porn communities. This network also shows the rise of Bitcoin on Reddit. While Bitcoin was missing from the 2017 network, in 2019 it constitutes one of the largest communities on the entire site. It’s connected to a conspiracy theory community, a porn community, a gaming community, an exmormon/exchristian community, a tmobile/verizon community, and architecture community. While some of these ties may be coincidental, some of them likely reflect real sociocultural overlaps.

**Recap/Next Steps**

That’s all I have for now. My main takeaway from this project is that Reddit consists of two worlds, a “mainstream” Reddit that is comprised of meme and news subreddits and a more fragmented, “fringe” Reddit that is made up of groups of porn, gaming, hobbiest, Bitcoin, sports, and music subreddits. This begs the question of how these divisions map onto real social groups. It appears that the Reddit communities outside the mainstream revolve around topics that are culturally associated with young men (e.g. gaming, vaping, Joe Rogan). Is the reason for this that young men are more likely to post exclusively to a handful of somewhat culturally subversive subreddits that other users are inclined to avoid? Unfortunately, we don’t have the data to answer this question, but this hypothesis is supported by the networks we see here.

The next step to take on this project will be to figure out how to allow for overlap between subreddit communities. As I mentioned, the clustering algorythm I used here forces subreddits into single communities. This distorts how communities on Reddit are really organized. Many subreddits appeal to multiple and distinct interests of Reddit users. For example, many subreddits attract users with a common political identity while also providing users with a news source. City-based subreddits attract fans of cities’ sports teams but also appeal to people who want to know about non-sports-related local events. That subreddits can serve multiple purposes could mean that the algorithm I use here lumped together subreddits that belong in distinct and overlapping communities. It also suggests that my mainstream Reddit community could really be a residual community of liminal subreddits that do not have a clear categorization. A clustering algorithm that allowed for community overlap would elucidate which subreddits span multiple communities. SNAP (Stanford Network Analysis Project) has tools in Python that seem promising for this kind of research. Stay tuned!