Twitter Network Analysis Based on Retweets

🧭 Overview & Setup

This tutorial walks you through on:

1. Extract usernames from retweets using a regular expression
2. How to create a directed graph data structure
3. How to run a network analysis based on the number of **retweets**
4. How to create a graph visualization

📦 Install and load packages

Install `tidyverse` and `igraph` if you do not have them in your R environment.

- `tidyverse` is a collection of R packages for data science
- `igraph` is a collection of network analysis tools

```
# uncomment and run the lines below if you need to install these packages
# install.packages("tidyverse")
# install.packages("igraph")
```

Load packages.

```
library(tidyverse)
library(igraph)
```

🔍 Data Preparation

📄 Read CSV file

Ensure you use a retweets dataset. Your filename should end with `-retweets.csv` (e.g., `Tesla-retweets.csv`).

```
df_retweets = read_csv('Lululemon-retweets.csv')
```

```
## Rows: 2416 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (2): username, text
## dbl (1): id
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_retweets %>% head()
```

id <dbl>	username <chr>
1.603436e+18	lolmayah
1.603436e+18	sendtoHarley
1.603435e+18	HarleyReign_
1.603433e+18	mlplace1012
1.603433e+18	GRANADAPOSTERS
1.603432e+18	K_Filer

6 rows | 1-2 of 3 columns

Print out the number of rows.

```
nrow(df_retweets)
```

```
## [1] 2416
```

📌 Handle usernames

Rename `username` column

Rename `username` column to `retweet_username` . This column contains the usernames who retweeted other tweets.

```
# rename username column to retweet_username
# retweet_username column contains the username who retweeted the original tweet
if ("username" %in% colnames(df_retweets)) {
  df_retweets <- df_retweets %>% rename(
    retweet_username = username
  )
}

df_retweets %>% head()
```

id	retweet_username
<dbl>	<chr>
1.603436e+18	lolmayah
1.603436e+18	sendtoHarley
1.603435e+18	HarleyReign_
1.603433e+18	mlplace1012
1.603433e+18	GRANADAPOSTERS
1.603432e+18	K_Filer

6 rows | 1-2 of 3 columns

Extract original authors

Every retweet starts with “RT @” and is followed by the original author’s username. Here is a sample retweet.

> RT @original_user: Here is a tweet.

Extract usernames using a regular expression.

```
# extract username using regular expression
# "^" refers to the beginning of a line
# "\w+" matches one or more alphanumeric and underscore characters
# "[, 2]" only extracts the username portion
df_retweets$original_username <- str_match(df_retweets$text, "^RT @(\\w+)")[, 2]

# display the mentions column in the first few rows
df_retweets %>% head() %>% select(text, original_username)
```

text
<chr>
RT @gherbo: HEALING IS ALL THAT MATTERS ❣️ ❣️ \n\nSwervin’ Through Stress Day powered by @ @alkemehealth and @Chicagobulls\n\nWe pra…
RT @TheQueenSuperia: I just got my dream puppy so I got him some welcome home gifts then went t & Lululemon 💛 Send £475 to see…
RT @TheQueenSuperia: I just got my dream puppy so I got him some welcome home gifts then went t & Lululemon 💛 Send £475 to see…
RT @nedryun: “For example, nearly $5,000 spent in 2022 at Lululemon, a luxury athletic apparel branc classified as “office expense,” a…
RT @nedryun: “For example, nearly $5,000 spent in 2022 at Lululemon, a luxury athletic apparel branc classified as “office expense,” a…
RT @nedryun: “For example, nearly $5,000 spent in 2022 at Lululemon, a luxury athletic apparel branc classified as “office expense,” a…

6 rows | 1-1 of 2 columns

🚀 Perform network analysis

🐬 Create a DataFrame that describes a directed graph

Each retweet can be represented as a directed edge in a graph that connects *from* the retweeter's username *to* the original author's username.

```
edges <- df_retweets %>%
  select(retweet_username, original_username) %>%
  rename(
    from = retweet_username,
    to = original_username
  )

edges <- unnest(edges, cols=to)

# display 20 first rows
head(edges, n = 20)
```

from <chr>	to <chr>
lolmayah	gherbo
sendtoHarley	TheQueenSuperia
HarleyReign_	TheQueenSuperia
mlplace1012	nedryun
GRANADAPOSTERS	nedryun
K_Filer	nedryun
snowlady09	healybaum
nolimitscat	gherbo
BowtieCarolina	GamecockBourbon
UnaFelixCulpa	ebeth360

1-10 of 20 rows Previous **1** 2 Next

🧪 Create a graph

We created the `edges` DataFrame in one of the previous steps. We can build a graph object using the DataFrame. `directed = TRUE` parameter is used to create a directed graph.

```
graph <- graph_from_data_frame(edges, directed = TRUE)

# print graph
graph
```

```
## IGRAPH f63eff6 DN-- 2474 2416 --
## + attr: name (v/c)
## + edges from f63eff6 (vertex names):
##  [1] lolmayah       ->gherbo          sendtoHarley   ->TheQueenSuperia
##  [3] HarleyReign_   ->TheQueenSuperia mlplace1012    ->nedryun
##  [5] GRANADAPOSTERS ->nedryun         K_Filer        ->nedryun
##  [7] snowlady09     ->healybaum       nolimitscat    ->gherbo
##  [9] BowtieCarolina ->GamecockBourbon UnaFelixCulpa  ->ebeth360
## [11] vaneerdz545    ->magalyyortizz   SteveGrassoSG  ->CNBCFastMoney
## [13] LorrieUScitizen->nedryun         dlc_chamb317   ->nedryun
## [15] jmlac282       ->nedryun         fuzzymcgovern21->nedryun
## + ... omitted several edges
```

🧮 Calculate in-degree centrality

```
# calculate degree centrality
deg <- degree(graph, mode = "in")

# sort by degree centrality in descending order
deg <- deg %>%
  sort(decreasing = TRUE)

deg %>% head()
```

```
## unusual_whales   JackFarley96    shespeaksup        nedryun  MillieParfait
##            403            379            311            144            103
##     hon3ybaby3
##            102
```

Check the number of vertices (i.e., users) in our graph

```
gorder(graph)
```

```
## [1] 2474
```

🧞 Top influencers by number of retweets

Identify the top 20 users by number of retweets.

```
top20 <- deg %>% head(n = 20)
top20 %>% head(20)
```

```
## unusual_whales   JackFarley96    shespeaksup        nedryun  MillieParfait
##            403            379            311            144            103
##     hon3ybaby3         gherbo           CNBC      zerohedge   StockMKTNewz
##            102             67             51             46             31
##    MarketWatch     EconomyApp GoddessPetite2 GoddessEliza_1  alphagirlclub
##             25             23             21             20             20
## IKantEvenToday       iCASH_ca  permanentglue      kvtherync        Reuters
##             19             18             17             17             15
```

`top20` is a named numeric vector. Convert it to a DataFrame. This will allow us to add new columns.

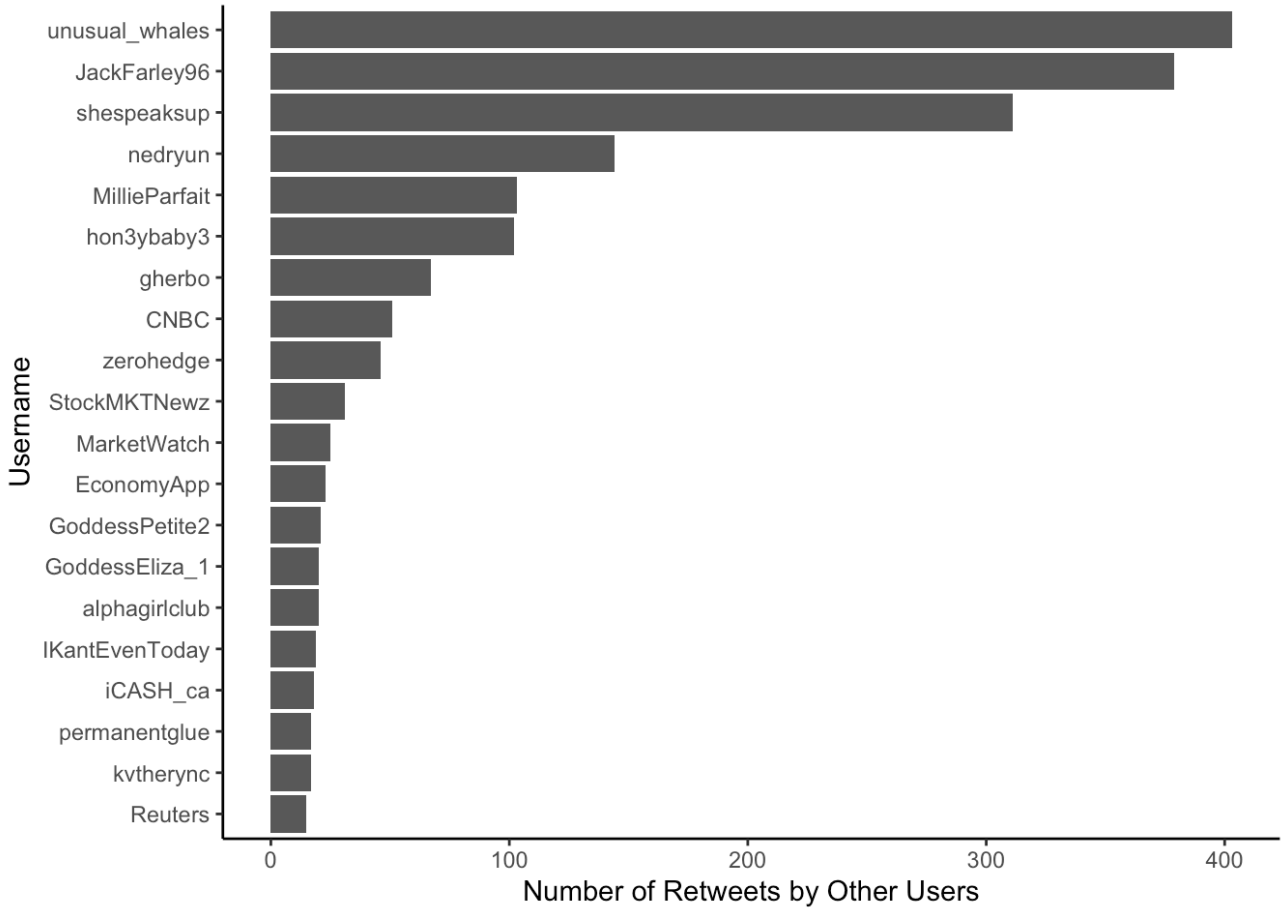
```
top20 <- top20 %>%
  enframe(name = "username", value="retweeted_count")

top20
```

username <chr>	**retweeted_count** <dbl>
unusual_whales	403
JackFarley96	379
shespeaksup	311
nedryun	144
MillieParfait	103
hon3ybaby3	102
gherbo	67
CNBC	51
zerohedge	46
StockMKTNewz	31

1-10 of 20 rows Previous **1** 2 Next

```
ggplot(
  data = head(top20, n = 20),
  aes(x = retweeted_count, y = reorder(username, retweeted_count))
) +
  geom_col() +
  theme_classic() +
  xlab("Number of Retweets by Other Users") +
  ylab("Username")
```

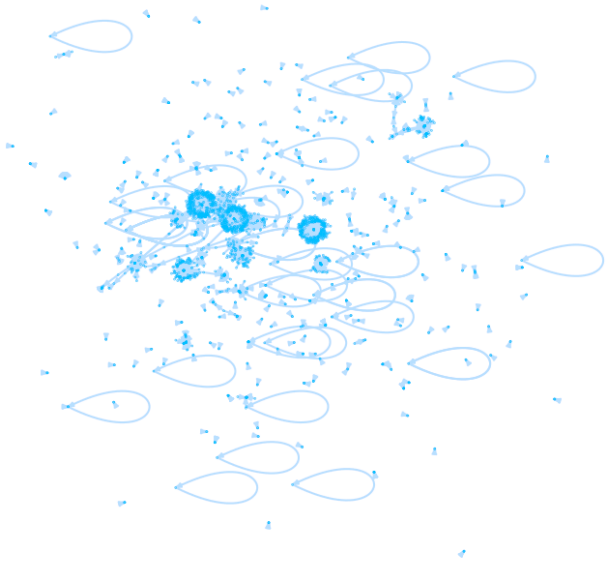


✨ Graph visualization

🔨 Network graph of all users

```
plot(
  graph,
  layout = layout_with_fr(graph),
  main="Retweets network graph of all users",
  edge.arrow.size = 0.15,
  edge.color = "#BBDFFF",
  vertex.label = NA,
  vertex.color = "#20DFFF",
  vertex.frame.color = "#00BFFF",
  vertex.size = 0.2
)
```

Retweets network graph of all users



💎 Top retweeted users within the largest connected component

```
# find the connected components of our graph
gc <- igraph::components(graph)

# delete users that are outside the largest connected component
graph_filtered <- delete_vertices(graph, gc$membership != which.max(gc$csize))

# calculate in-degrees within the filtered graph
filtered_deg_in <- degree(graph_filtered, mode = "in")
vertex_size <- pmax(pmin(filtered_deg_in * 0.08, 6), 0.3)

# find the top 12 retweeted users within the filtered graph
top_retweeted_users <- filtered_deg_in %>%
  sort(decreasing = TRUE) %>%
  head(n = 12) %>%
  names()

plot(
  graph_filtered,
  layout = layout_with_fr(graph_filtered),
  main="Top retweeted users within the largest connected component",
  edge.arrow.size = 0.15,
  edge.color = "#BBDFFF",
  vertex.label = ifelse(
    names(filtered_deg_in) %in% top_retweeted_users,
    V(graph_filtered)$name,
    NA
  ),
  vertex.label.cex = 0.8,
  vertex.label.color = "#000000",
  vertex.color = "#20DFFF",
  vertex.frame.color = "#00BFFF",
  vertex.size = vertex_size
)
```

Top retweeted users within the largest connected component