Building a graph from raw data

CASE STUDIES: NETWORK ANALYSIS IN R

Edmund Hart
Instructor

Exploring the data

- Data is several days of all the tweets mentioning #rstats
- Key attributes for building a graph are:
    - screen name
    - raw text of the tweet

Anatomy of a tweet

1. *ReecheshJC*: "Hey #rstats, how do I do fct_lump but where I lump based on count values in a column?"
2. *kom_256*: "RT @elenagbg: Retweeted R-Ladies Madrid (@RLadiesMAD):\n\nEn el #OCSummit17... Fast Talks sobre #rstats organizado por... **https://t.co/CKY5aG..."**

```r
library(igraph)
library(stringr)
raw_tweets <- read.csv("datasets/rstatstweets.csv",
  stringsAsFactors = FALSE)
```

Data sample, single row

```
user_name:    Karen Millidine
screen_name:    KJMillidine
tweet_tex:t    RT @Rbloggers: RStudio v1.1 Released
https://t.co/kCMHc689nY #rstats #DataScience
favorites:    0
retweets:    96
location:    None
expanded_url:    https://wp.me/pMm6L-ExV
in_reply_to_tweet_id:    NA
in_reply_to_user_id:    NA
dt:    10/10/17
```

Building the graph

```r
## Get all the screen names
all_sn <- unique(raw_tweets$screen_name)

## Create graph
retweet_graph <- graph.empty()

## Add screen names as vertices
retweet_graph <- retweet_graph + vertices(all_sn)
```

Building the graph

```
## Extract name and add edges
for(i in 1:dim(raw_tweets)[1]){
  # Extract retweet name
  rt_name <- find_rt(raw_tweets$tweet_text[i])
  # If there is a name add an edge
   if(!is.null(rt_name)){
      # Check to make sure the vertex exists, if not, add it
      if(!rt_name %in% all_sn){
        retweet_graph <- retweet_graph + vertices(rt_name)
      }
     # add the edge
     retweet_graph <- retweet_graph +
       edges(c(raw_tweets$screen_name[i], rt_name))
   }
}
```

Cleaning the graph

```r
## Size the number of degree 0 vertices
sum(degree(retweet_graph) == 0)


## Trim and simplify
retweet_graph <- simplify(retweet_graph)
retweet_graph <- delete.vertices(retweet_graph,
  degree(retweet_graph) == 0)
```

Let's practice!

CASE STUDIES: NETWORK ANALYSIS IN R

Building a mentions graph

CASE STUDIES: NETWORK ANALYSIS IN R

Edmund Hart
Instructor

Recall tweet anatomy

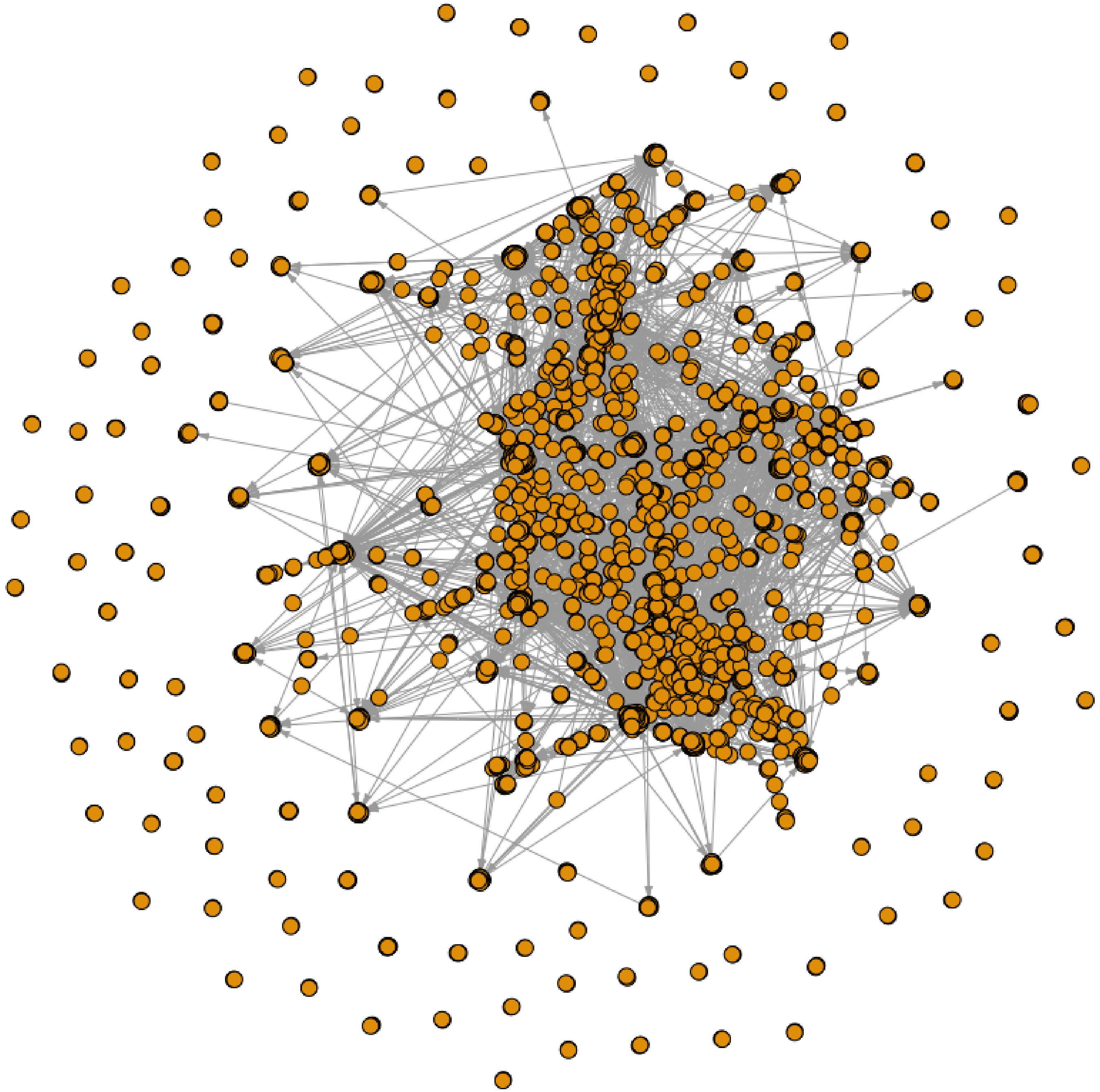
AlexisAchim: "@LAStools @Lees_Sandbox @jhollist @LeahAWasser LidR is also available directly on CRAN #rstats"

timelyportfolio: "just might have a demo of @emeeks new #reactjs/#d3js semiotic in #rstats in the works"

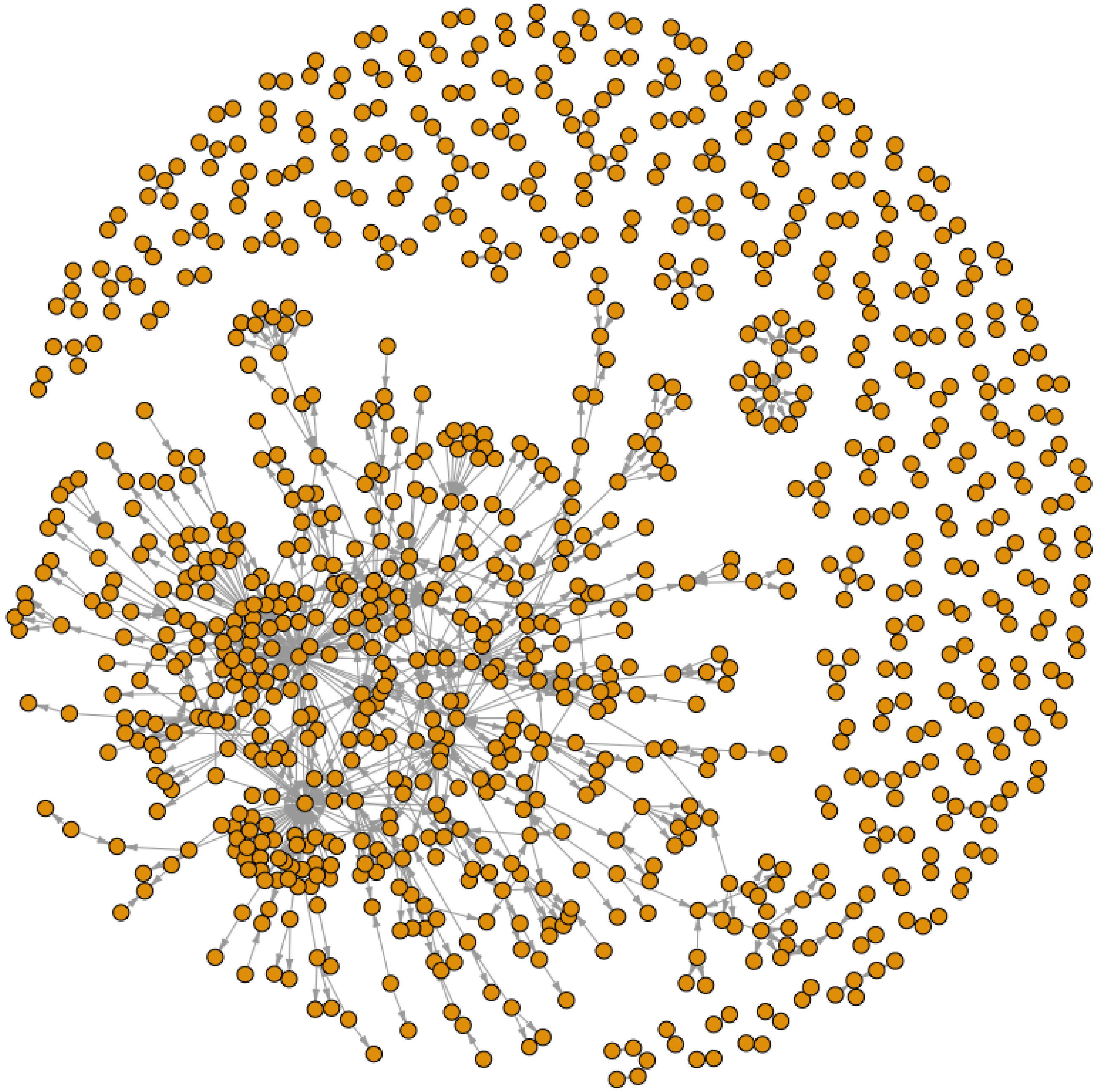
Build your mentions graph

```r
ment_g <- graph.empty()
ment_g <- ment_g + vertices(all_sn)
for(i in 1:dim(raw_tweets)[1]) {
  ment_name <- mention_ext(raw_tweets$tweet_text[i])
  if(length(ment_name) > 0 ) {
    # Add the edge(s)
    for(j in ment_name) {
      # Check to make sure the vertex exists, if not, add it
      if(!j %in% all_sn) {
        ment_g <- ment_g + vertices(j)  }
      ment_g <- ment_g + edges(c(raw_tweets$screen_name[i], j))
    }
  }
}
ment_g <- simplify(ment_g)
ment_g <- delete.vertices(ment_g, degree(ment_g) == 0)
```

Retweet Graph



Mentions Graph



Let's practice!

CASE STUDIES: NETWORK ANALYSIS IN R

Finding communities

CASE STUDIES: NETWORK ANALYSIS IN R

Edmund Hart
Instructor

Three different communities

```
undirected_ment_g <- as.undirected(ment_g)


ment_edg <- cluster_edge_betweenness(undirected_ment_g)
ment_eigen <- cluster_leading_eigen(undirected_ment_g)
ment_lp <- cluster_label_prop(undirected_ment_g)
```

Sizing the communities

```
length(ment_edg)
length(ment_eigen)
length(ment_lp)
```

```
173
168
212
```

```r
table(sizes(ment_edg))
```

```
  2   3   4   5   6   7   8   9  11  12  18  19  20  23  24  26  28
103  21  14   7   3   3   1   2   1   2   2   1   1   1   1   2   1
 31  33  38  40  41  52  58
  1   1   1   1   1   1   1
```

```r
table(sizes(ment_eigen))
```

```
  2   3   4   5   6   7   9  10  12  18  23  26  29  30  32  34  35  58
103  22  14   7   4   3   1   1   1   1   1   1   1   1   1   1   1   1
64  66 101
 1   1   1
```

```r
table(sizes(ment_lp))
```

```
  2   3   4   5   6   7   8   9  10  11  12  13  16  25  26  67  70
103  32  22  19   8   5   4   3   5   1   2   3   1   1   1   1   1
```

Comparing communities

```r
compare(ment_edg, ment_eigen, method = 'vi')
```

```
0.9761792
```

```r
compare(ment_eigen, ment_lp, method = 'vi')
```

```
1.192238
```

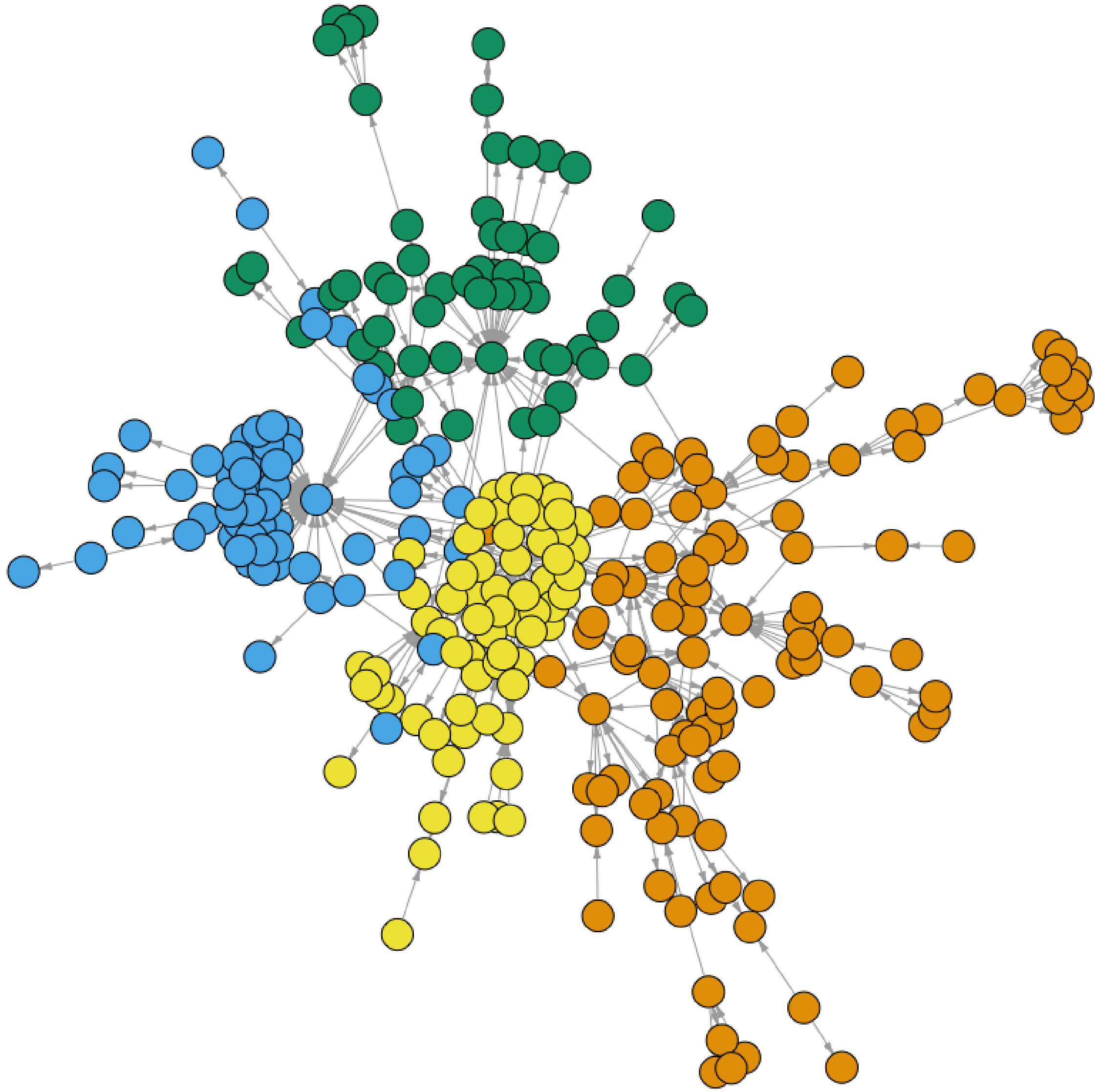
```r
compare(ment_lp, ment_edg, method = 'vi')
```

```
0.9631608
```

Plotting community structure

```
lrg_eigen <- as.numeric(
                names(ment_eigen[which(sizes(ment_eigen) > 45)])
             )
eigen_sg <- induced.subgraph(ment_g,
      V(ment_g)[ eigen %in% lrg_eigen])
plot(eigen_sg, vertex.label = NA, edge.arrow.width = .8,
     edge.arrow.size = 0.2,
     coords = layout_with_fr(ment_sg), margin = 0,
     vertex.size = 6, vertex.color =
     as.numeric(as.factor(V(eigen_sg)$eigen)))
```

Mentions subgraph communities



Let's practice!

CASE STUDIES: NETWORK ANALYSIS IN R