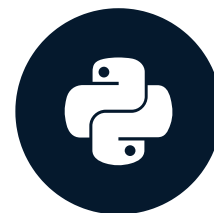


# Twitter networks

ANALYZING SOCIAL MEDIA DATA IN PYTHON

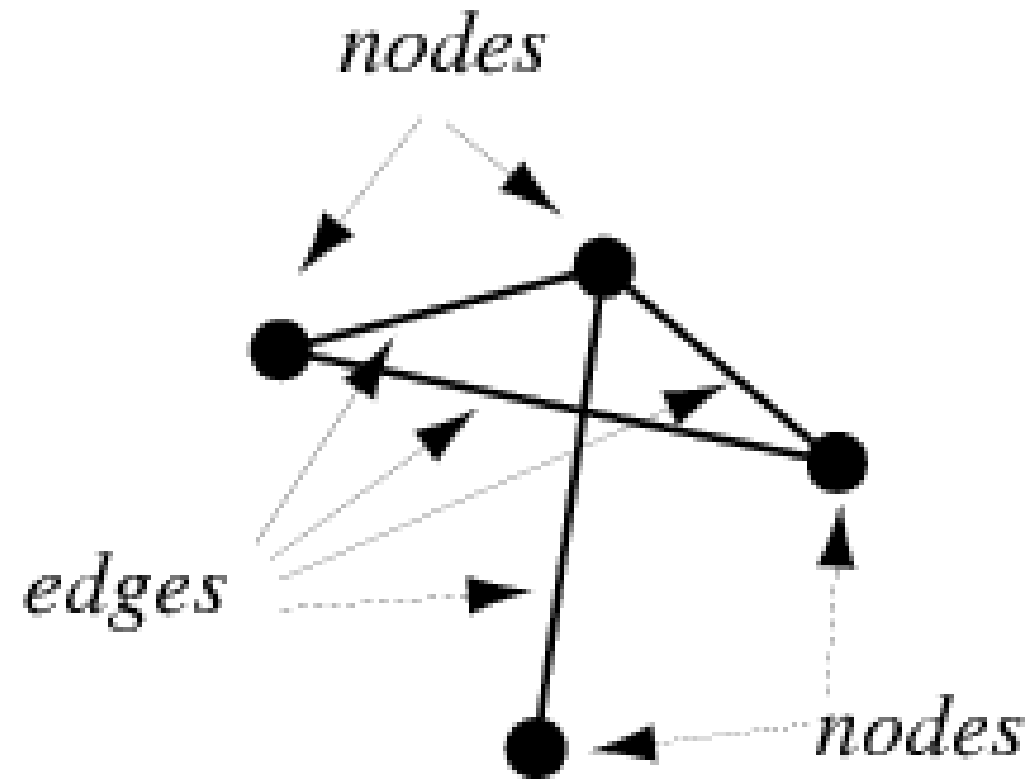


**Alex Hanna**

Computational Social Scientist



# Network analysis: terms



- Directed networks
  - Relationships are not mutual
- Source node
  - Where the arrow starts
- Target node
  - Where the arrow ends

<sup>1</sup> <http://mathworld.wolfram.com/GraphEdge.html>

# Types of Twitter network ties

- Twitter networks
  - Retweets
  - Quotes
  - Replies

# Retweet networks

↻ DataCamp Retweeted

DATIO

DATIO

@datiobd · Jun 12

How can spreadsheet workflows be incorporated into more general #datascience flows in sustainable and healthy ways? by @JennyBryan ow.ly/f4Pa30ksiuR #statistics via @DataCamp



Spreadsheets in Data Science

How can spreadsheet workflows be incorporated into more general data science flows in sustainable and healthy ways?

datacamp.com

💬

↻ 4

❤️ 4

✉️

A diagram illustrating a retweet relationship. On the left is a green circular node containing a black icon of a stylized 'd' and 'c' (DataCamp logo). On the right is a blue circular node containing the word 'DATIO' in white capital letters. A horizontal arrow points from the green node to the blue node. Below the arrow, the text 'retweets a tweet by' is written.

 datacamp

ANALYZING SOCIAL MEDIA DATA IN PYTHON

# Quote networks



# Reply networks

**Mar Hicks**   
@histoftech

Following

Listened to this episode today and it restored my faith in the state of our field.

**Lady Science** @ladyxscience

What are the ethics and methods for examining #LGBTQ histories of science? When can we call a person from the past trans or queer? Is it possible to “out” a historical figure?...

12:29 AM - 15 Jun 2018

4 Retweets 11 Likes



 1  4  11 



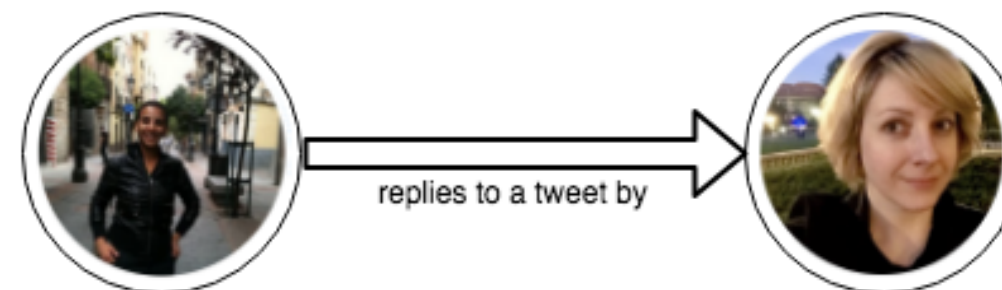
Tweet your reply

**Timnit Gebru** @timnitGebru · Jun 15

Replying to @histoftech

Will listen. Thanks for sharing!

   1 



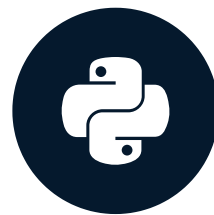
# Let's practice!

ANALYZING SOCIAL MEDIA DATA IN PYTHON



# Importing and visualizing Twitter networks

ANALYZING SOCIAL MEDIA DATA IN PYTHON



**Alex Hanna**

Computational Social Scientist

# Edge Lists

```
BethMohn      ChristianMohn
ASilNY        LarrySchweikart
mattg444      WhiteHouse
hlthiskrieger aravosis
Herky86       SenJeffMerkley
PatrickParsons9 TwitterGov
New_Narrative CFR_org
dddlor        roywoodjr
scrivener50   michaelischerer
ChiefsHeadCoach johnpavlovitz
```

# Importing a retweet network

```
import networkx as nx
## ... flatten and convert JSON
G_rt = nx.from_pandas_edgelist(
    tweets,
    source = 'user-screen_name',
    target = 'retweeted_status-user-screen_name',
    create_using = nx.DiGraph())
```

# Importing a quoted network

```
import networkx as nx

## ... flatten and convert JSON
G_quote = nx.from_pandas_edgelist(
    tweets,
    source = 'user-screen_name',
    target = 'quoted_status-user-screen_name',
    create_using = nx.DiGraph())
```

# Importing a reply network

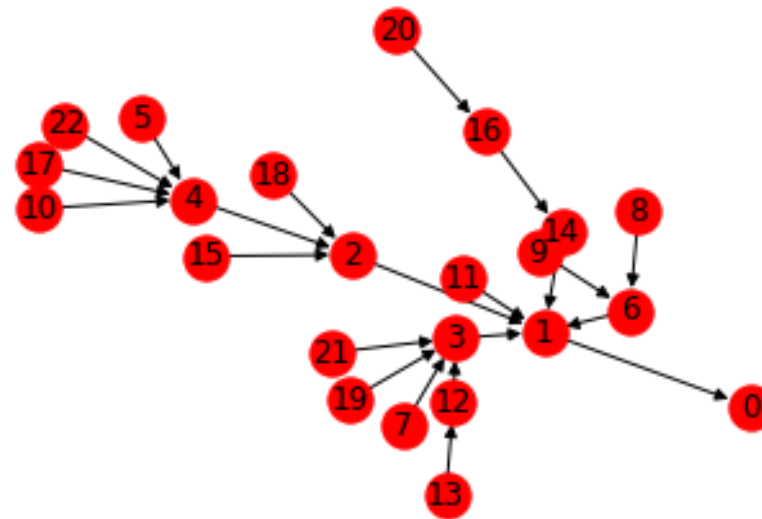
```
import networkx as nx

## ... flatten and convert JSON

G_reply = nx.from_pandas_edgelist(
    tweets,
    source = 'user-screen_name',
    target = 'in_reply_to_screen_name'
    create_using = nx.DiGraph())
```

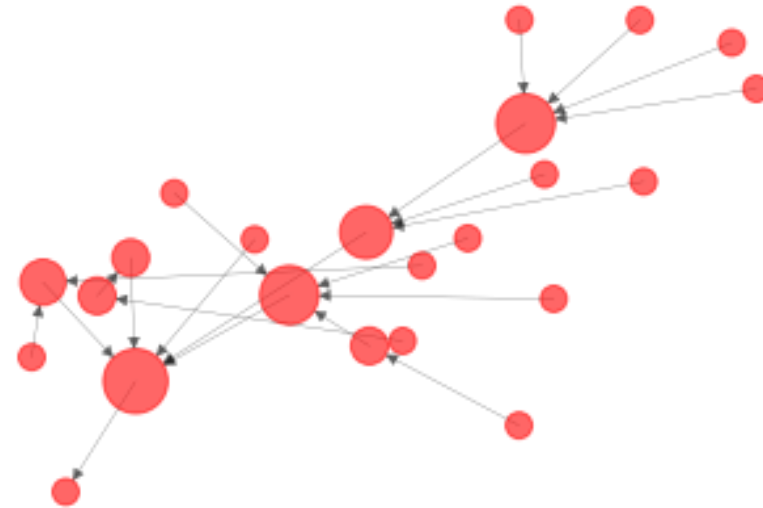
# Visualization

```
nx.draw_networkx(T)  
plt.axis('off')
```



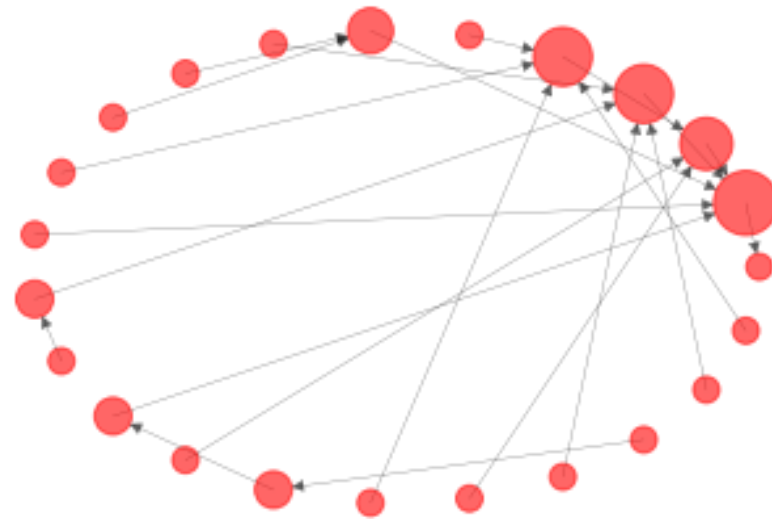
# Visualization options

```
sizes =  
    [x[1]*100 for x in T.degree()]  
nx.draw_networkx(T,  
    node_size = sizes,  
    with_labels = False,  
    alpha = 0.6,  
    width = 0.3)  
plt.axis('off')
```



# Circular layout

```
circle_pos =  
    nx.circular_layout(T)  
nx.draw_networkx(T,  
    pos = circle_pos,  
    node_size = sizes,  
    with_labels = False,  
    alpha = 0.6,  
    width = 0.3)  
plt.axis('off')
```



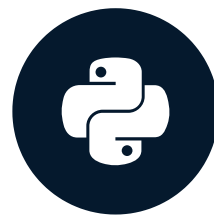


# Let's practice!

ANALYZING SOCIAL MEDIA DATA IN PYTHON

# Node-level metrics

ANALYZING SOCIAL MEDIA DATA IN PYTHON



**Alex Hanna**

Computational Social Scientist

# Centrality: node importance

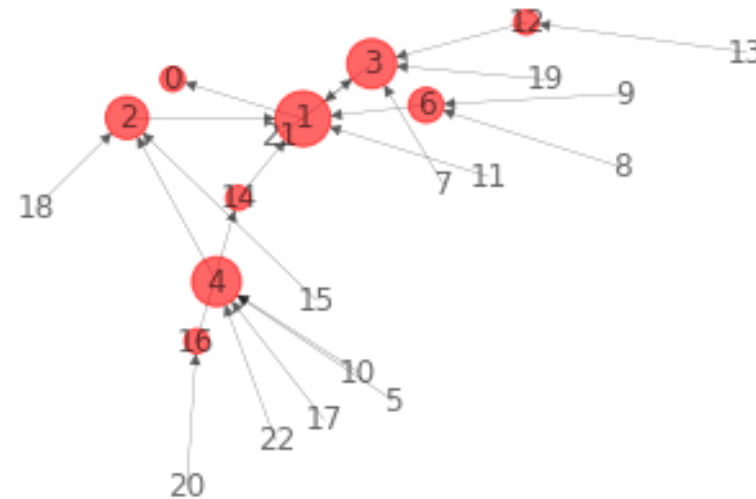
- Centrality
  - Measures of importance of a node in a network
  - Several different ideas of "importance"

# Degree centrality

# Degree

- Number of edges that are connected to node
- Two types of degrees in a directed network
  - In-degree - edge going **into** node
  - Out-degree - edge going **out of** a node

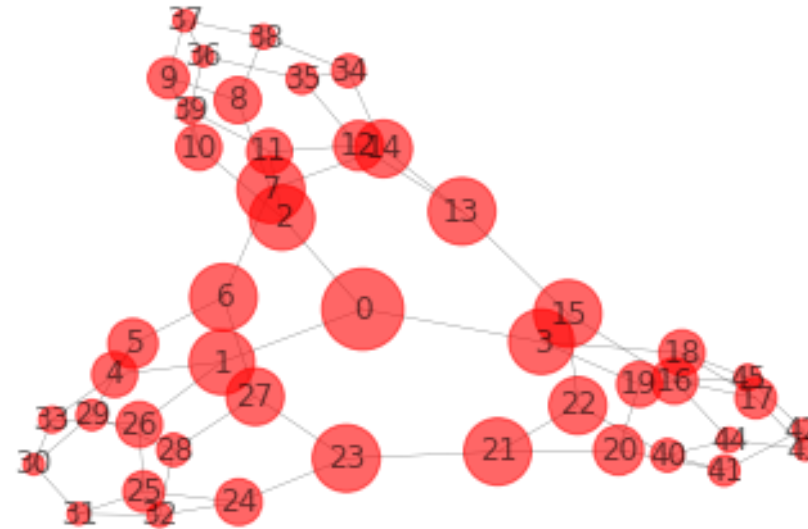
```
nx.in_degree_centrality(T)
nx.out_degree_centrality(T)
```



# Betweenness centrality

- How many shortest paths between two nodes pass through this node
- Importance as a network broker

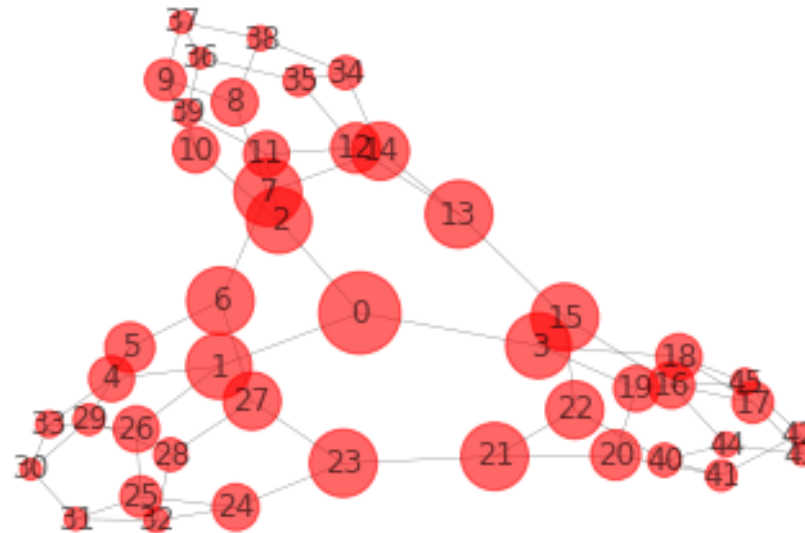
```
nx.betweenness centrality(T)
```



# Printing highest centrality

```
bc = nx.betweenness centrality(T)
betweenness = pd.DataFrame(
    list(bc.items()),
    columns = ['Name', 'Cent'])
print(betweenness.sort_values(
    'Cent',
    ascending = False).head())
```

	Name	Centrality
0	0	0.232540
23	23	0.158514
7	7	0.158514
15	15	0.158514
21	21	0.157588



# Centrality in different networks

		Centrality		
		In-Degree	Out-Degree	Betweenness
Network Type	Retweets	Gets retweets	Shares retweets	Bridges different topic/ideology communities
	Replies	Gets most replies	Participates in many conversations	Bridges different topic/ideology discussions

# The ratio

```
degree_rt = pd.DataFrame(list(G_rt.in_degree()),
                           columns = ['screen_name', 'degree'])
degree_reply = pd.DataFrame(list(G_reply.in_degree()),
                              columns = ['screen_name', 'degree'])
ratio = degree_rt.merge(degree_reply,
                        on = 'screen_name',
                        suffixes = ('_rt', '_reply'))
ratio['ratio'] = ratio['degree_reply'] / ratio['degree_rt']
```



# Let's practice!

ANALYZING SOCIAL MEDIA DATA IN PYTHON