Please download Gephi (www.gephi.org)

We will use this software today!

Network Analysis

Dr. Shaina Race
Institute for Advanced Analytics

Course Overview

Day 1: Visualization/Terminology basics

Day 2: Descriptive Statistics

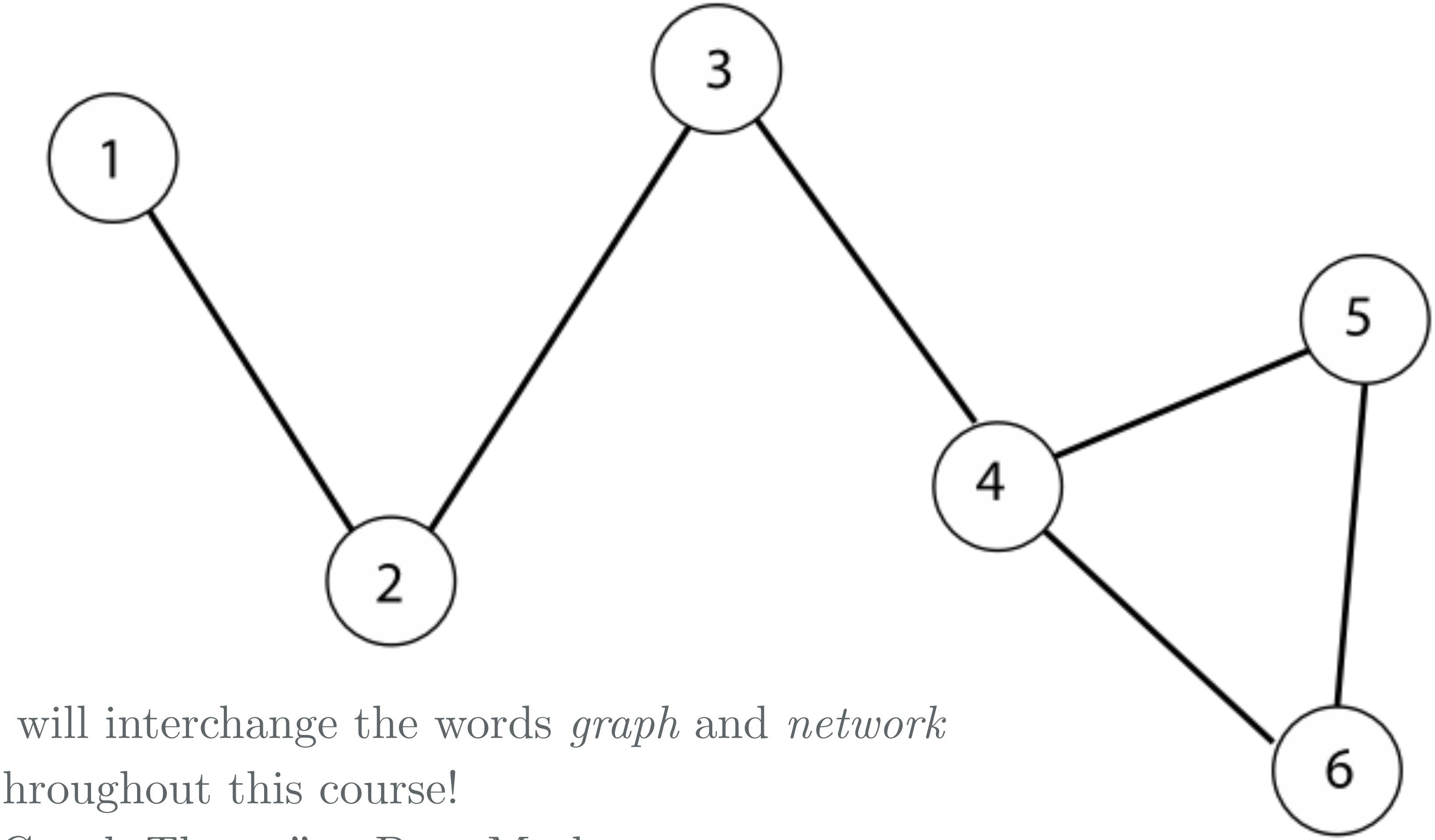
Day 3: Measures of Influence/Importance in a network

Day 4: Community Detection (clustering)

Day 5: Hypothesis Testing on networks

Day 6: Test or Project – up to you!

What is a graph/network?

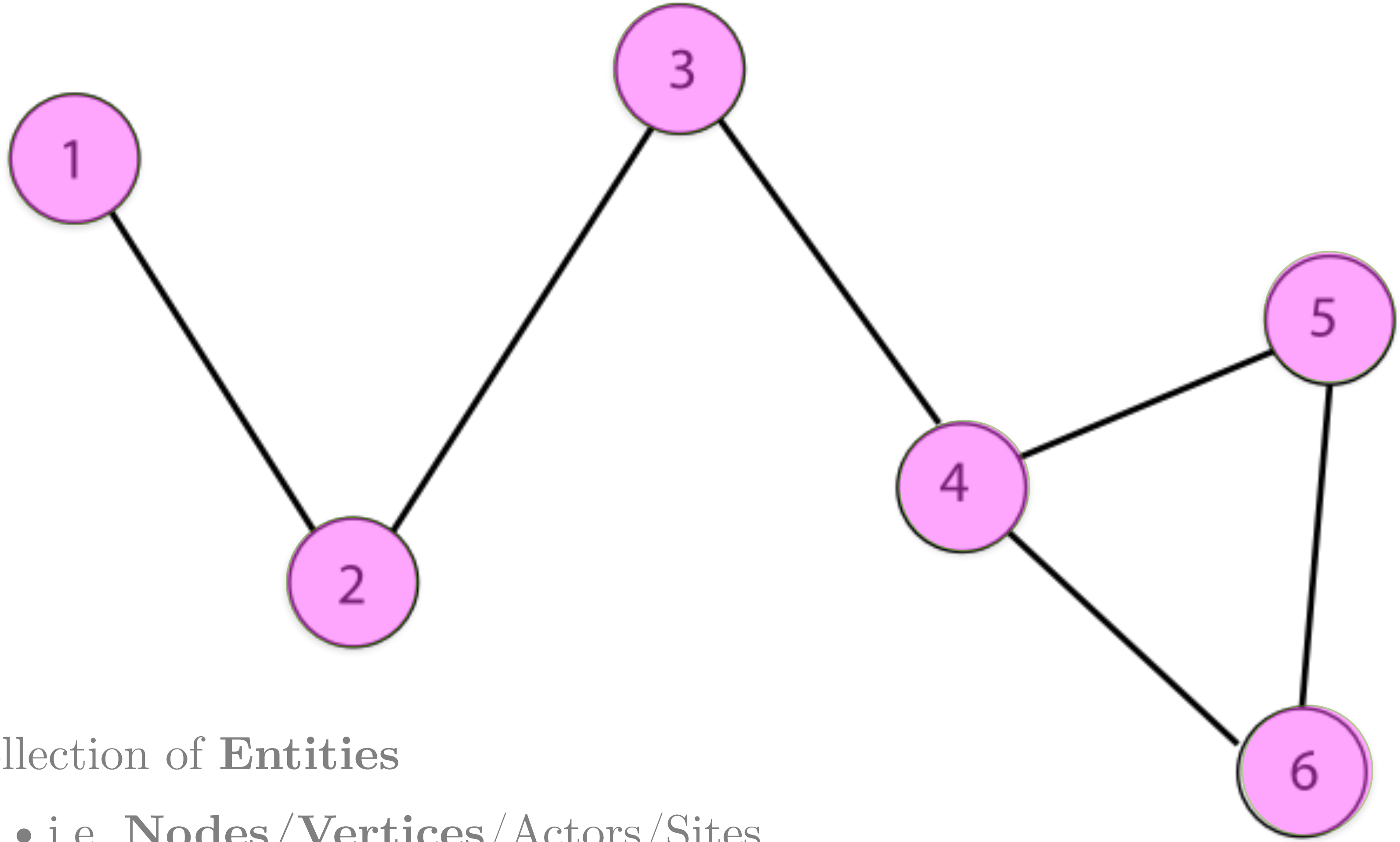


I will interchange the words *graph* and *network* throughout this course!

"Graph Theory" = Pure Math

"Network Analysis" = Applied Math

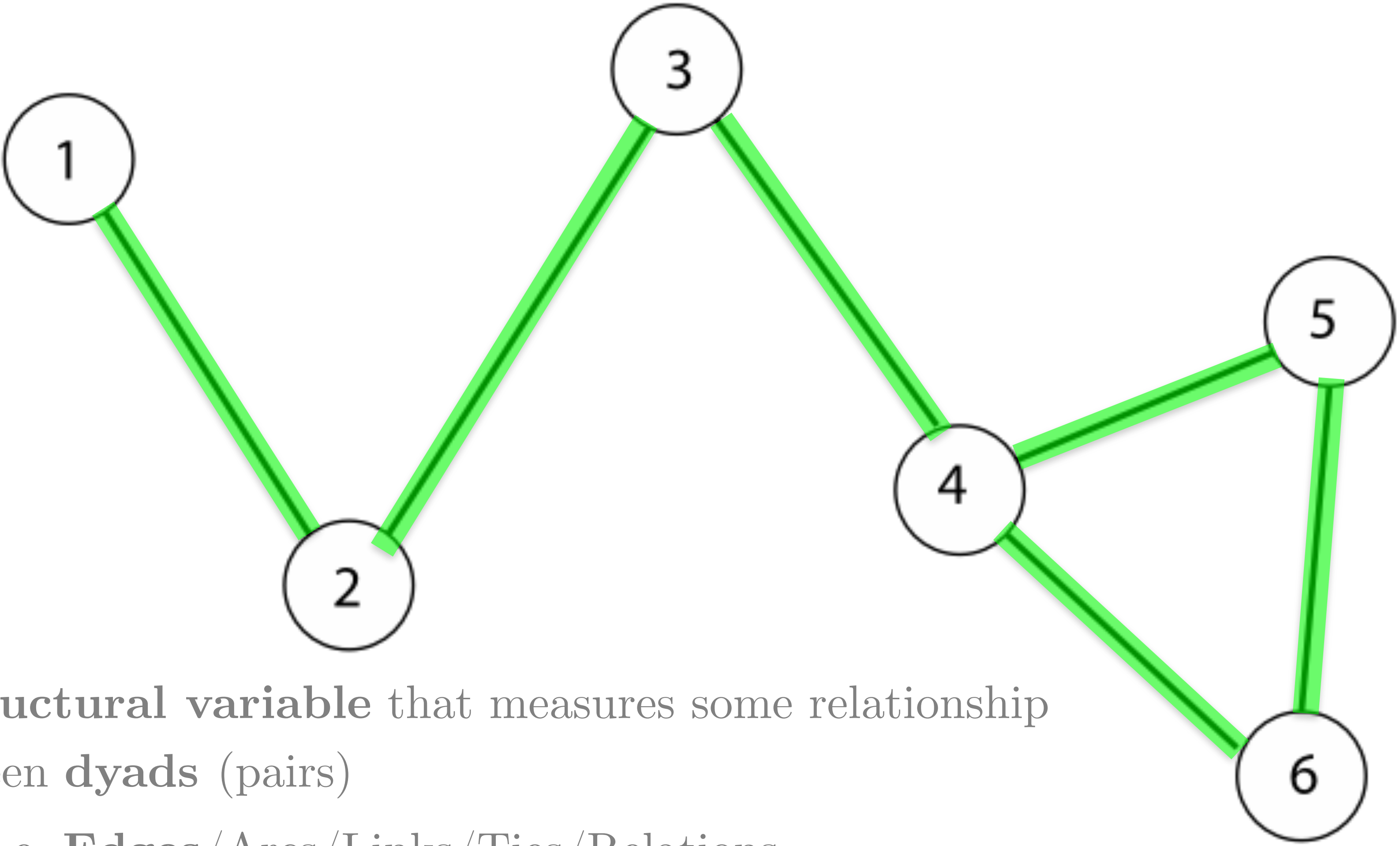
What is a graph/network?



Collection of **Entities**

- i.e. **Nodes**/**Vertices**/Actors/Sites
- could be individuals, organizations, places, objects

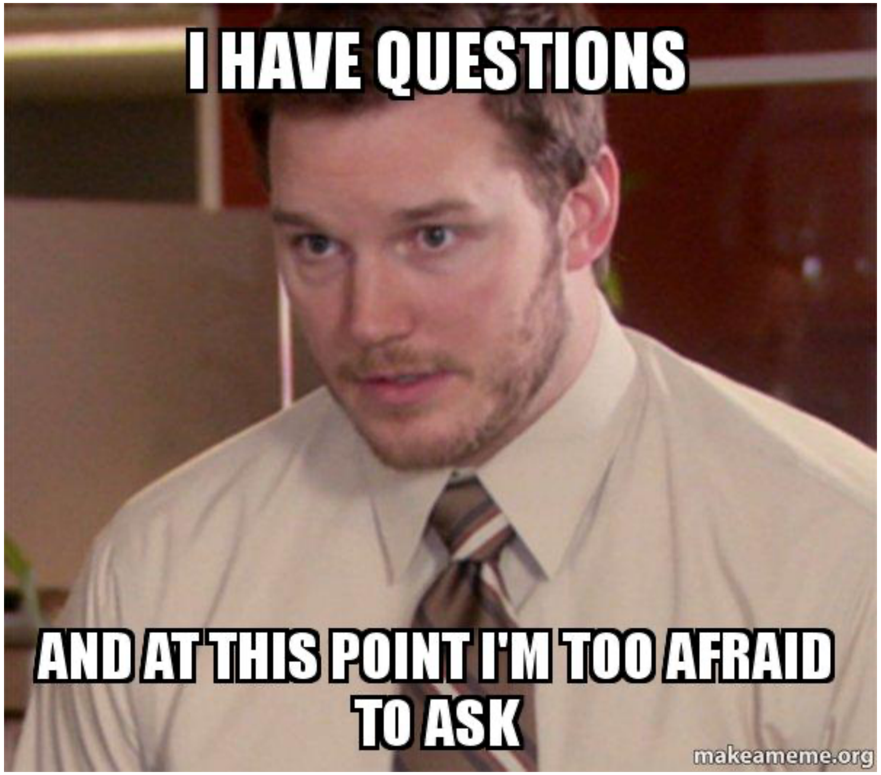
What is a graph/network?



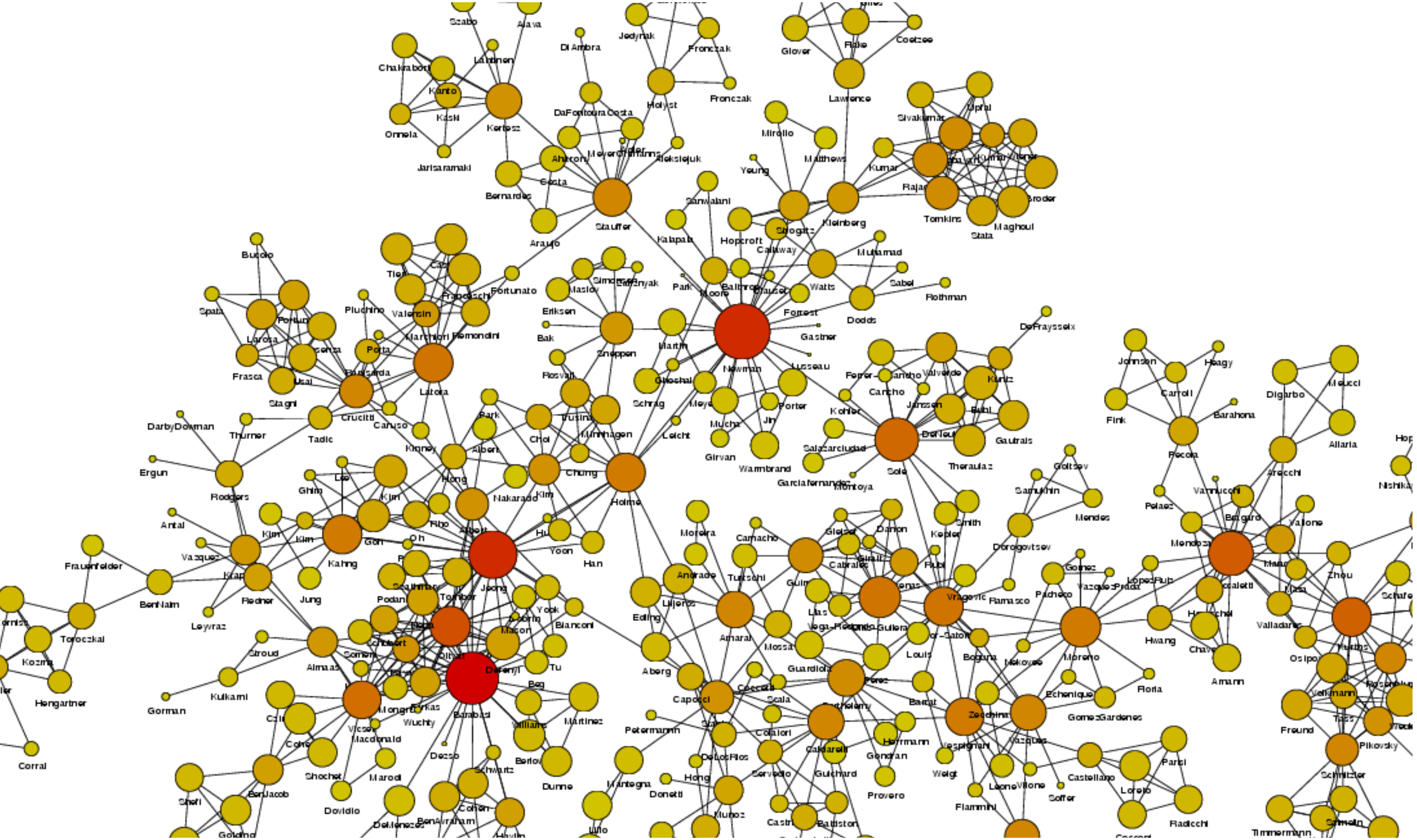
A **structural variable** that measures some relationship between **dyads** (pairs)

- i.e. **Edges**/Arcs/Links/Ties/Relations
- Friendship, distance, similarity, trade, management

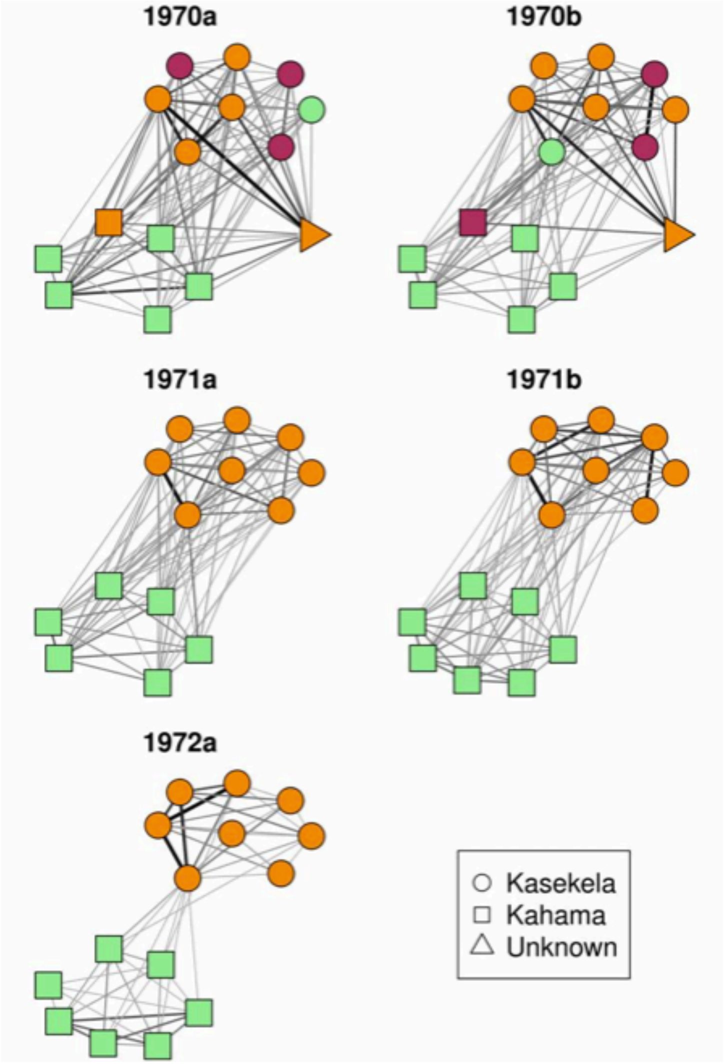
Questions this course might answer



Who is the **most important** or **influential** person in this network?



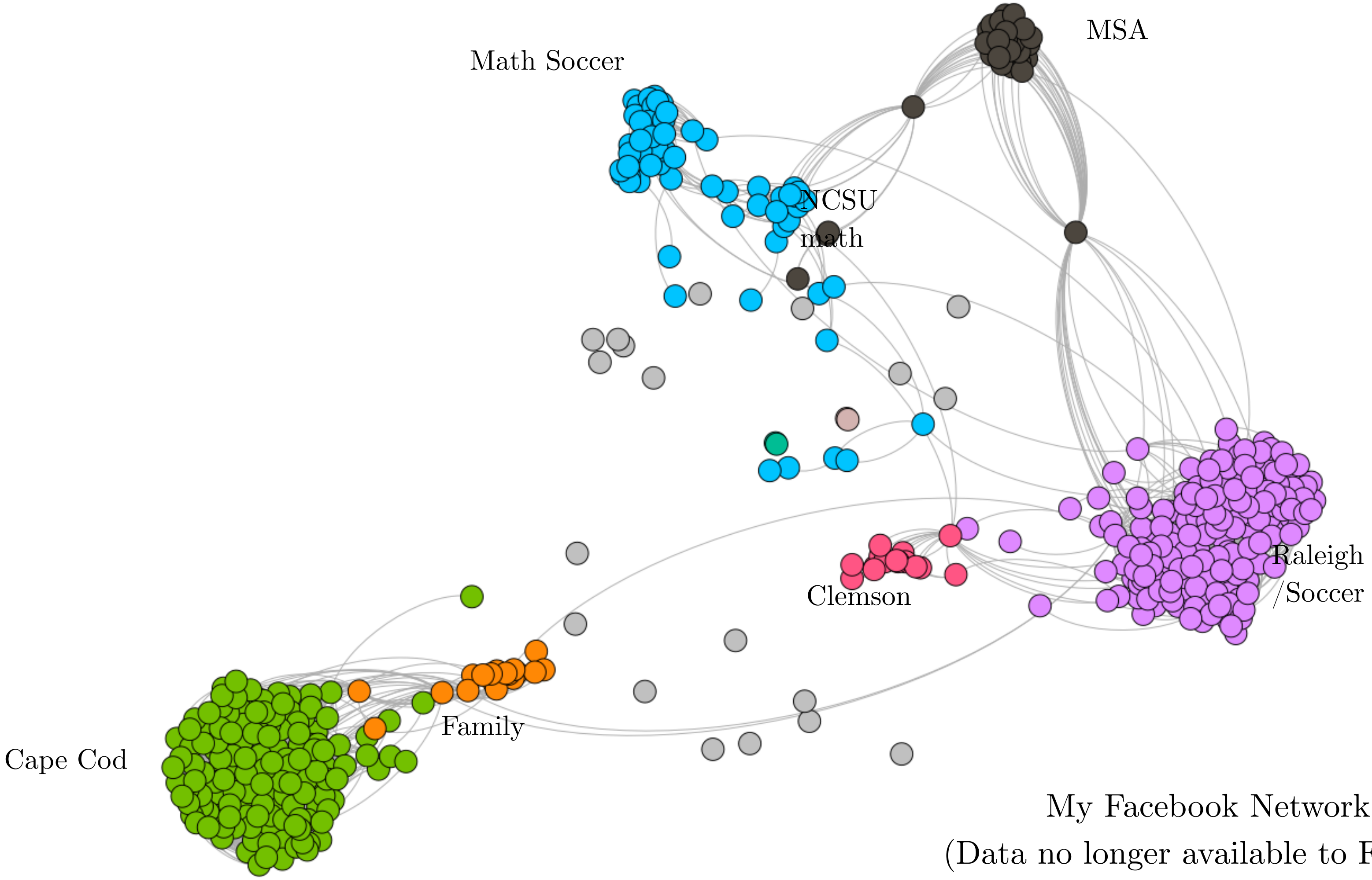
Is there any community structure or clustering apparent? Who is forming social groups? Who is NOT fitting in with the community structure?



"...statistical tests revealed clusters of males that grew more distinct over time."

Link:
https://today.duke.edu/2018/03/
how-infighting-turns-toxic-chimpanzees

What is driving the social atmosphere in our organization? What factors explain the network?

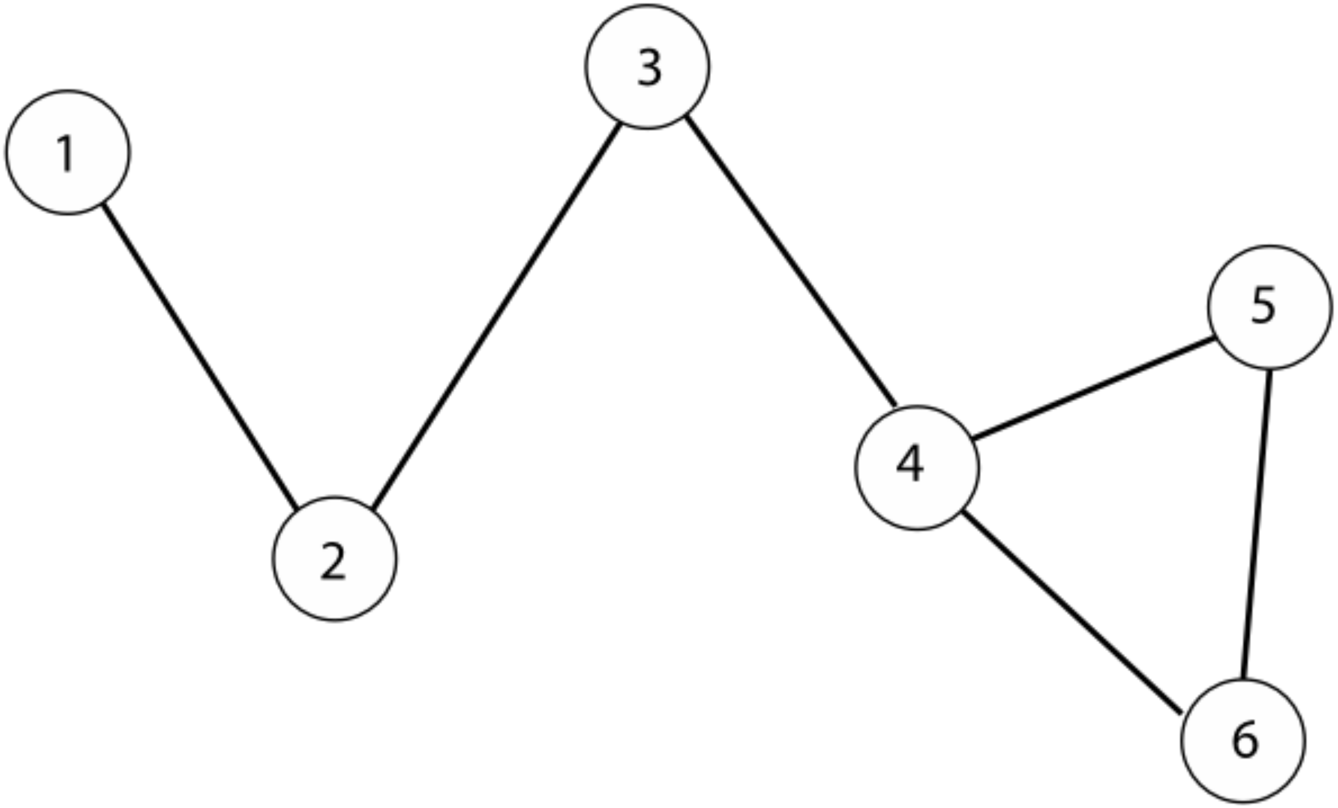


My Facebook Network in 2014
(Data no longer available to Facebook users)

Introduction to Network Data

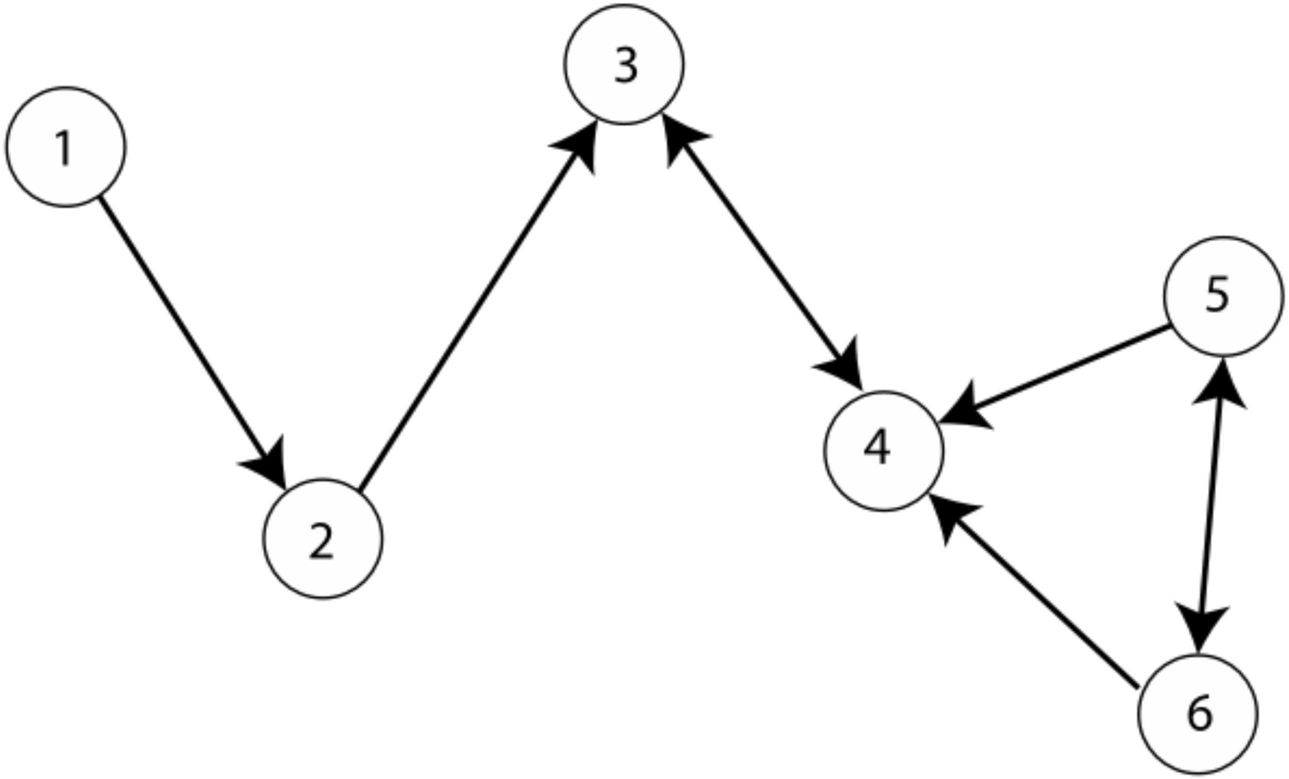
Nodes, Edges, Weights, Directions

Binary/Unweighted Network



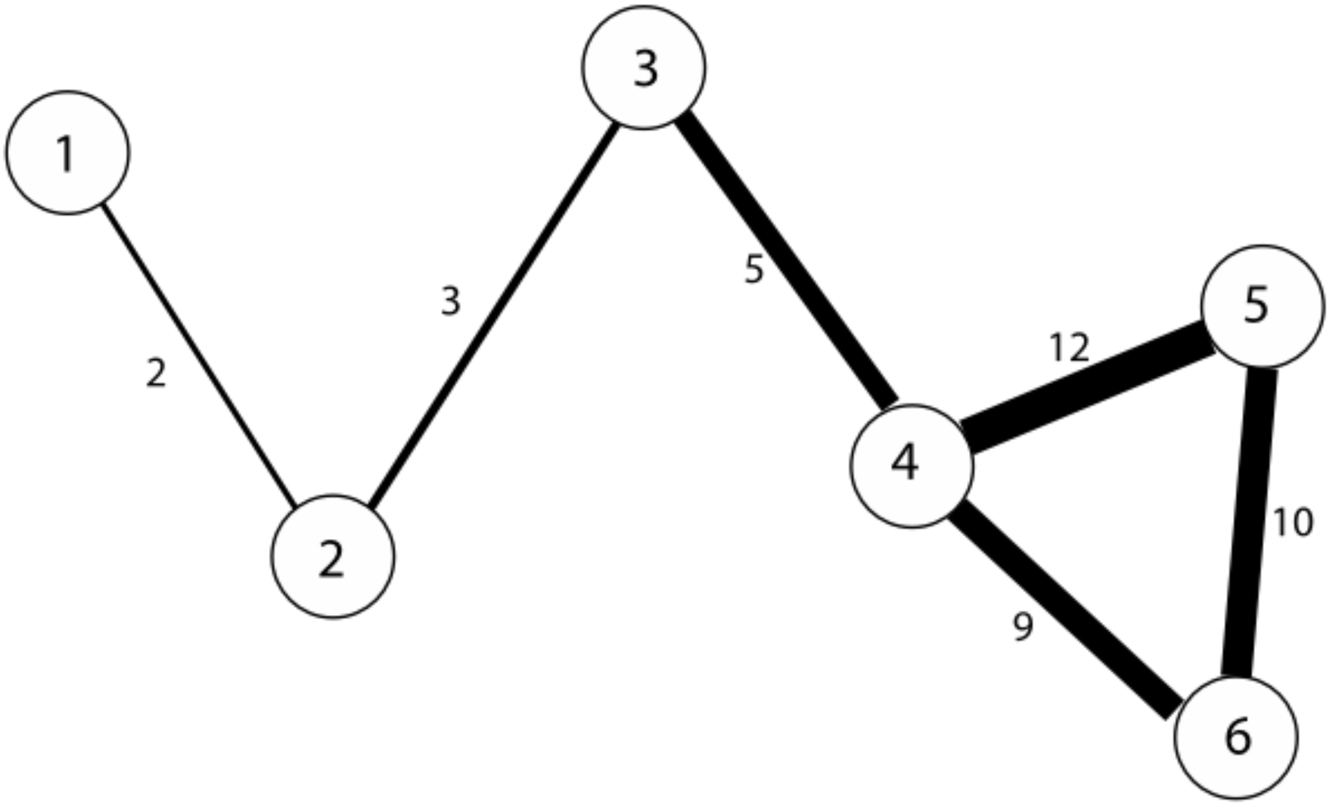
Edges are binary variables indicating presence/absence of relationship

Directed Binary Network



Edges are binary variables indicating presence/absence of *directed* relationship.
(Ex: “is the son/daughter of...”)

Weighted Network



Edges are ordinal or continuous variables indicating strength of relationship (Ex: "$ amount of trade between")

Adjacency Matrix

Generally, all of the mathematical analysis is going to involve the **adjacency matrix**, $\mathbf{A}$.

$$\mathbf{A}_{ij} = \text{weight of edge between vertex } i \text{ and vertex } j$$

Quiz

Which of the following words accurately describes the adjacency matrix for an undirected graph?

A. Diagonal

B. Symmetric

C. Identity

D. Negative

E. Rectangular

F. Sassy

Examples of Networks

Social

- Personal affinity (friendship, respect, readership)
- Interaction (phone records, email records, etc)
- Organizational (managerial networks, organizational maps)
- Affiliation (links to social events, clubs, or organizations)
- Genealogy (family trees, kinship)

Financial/Political/Economic

- Trade networks
- Business transactions, lending

Transportation/Logistics

- Manufacturing, warehousing, and retail networks
- Highway and road systems

Similarity

- *Impose* a network on tabular data using measures of similarity.
- Ex: let edges represent cosine similarity between documents

Quiz!!

Are the following networks weighted or unweighted?

A. A graph that links people together if they have the same birthday

B. A graph that links people together by the number of mutual friends they have

Is the following network directed or undirected?

A. A network that shows how universities transfer students to other universities

Statistical Considerations for Network data

• • •

Sampling, Independence

Sampling

- Suppose you’re studying a massive network.
- How do you collect data to support your analysis?

Snowball sampling

Data often collected via **snowball sampling**.

- Select few individuals
- Follow their ties/links to new individuals
- Follow ties/links of new individuals
- ...Not so random...

Advantages

- Great for hidden/marginalized populations
  - Example: IV drug users or sex workers
- Or when trust needed to collect info

Disadvantages

- Bias! towards initial set of individuals.
  - **Not necessarily representative of entire network distributions**
  - Ex: start NCSU network snowball with the football team vs. the chess team
- Potential for inaccurate referrals - snowball gets lost in left field.

Independence in Networks

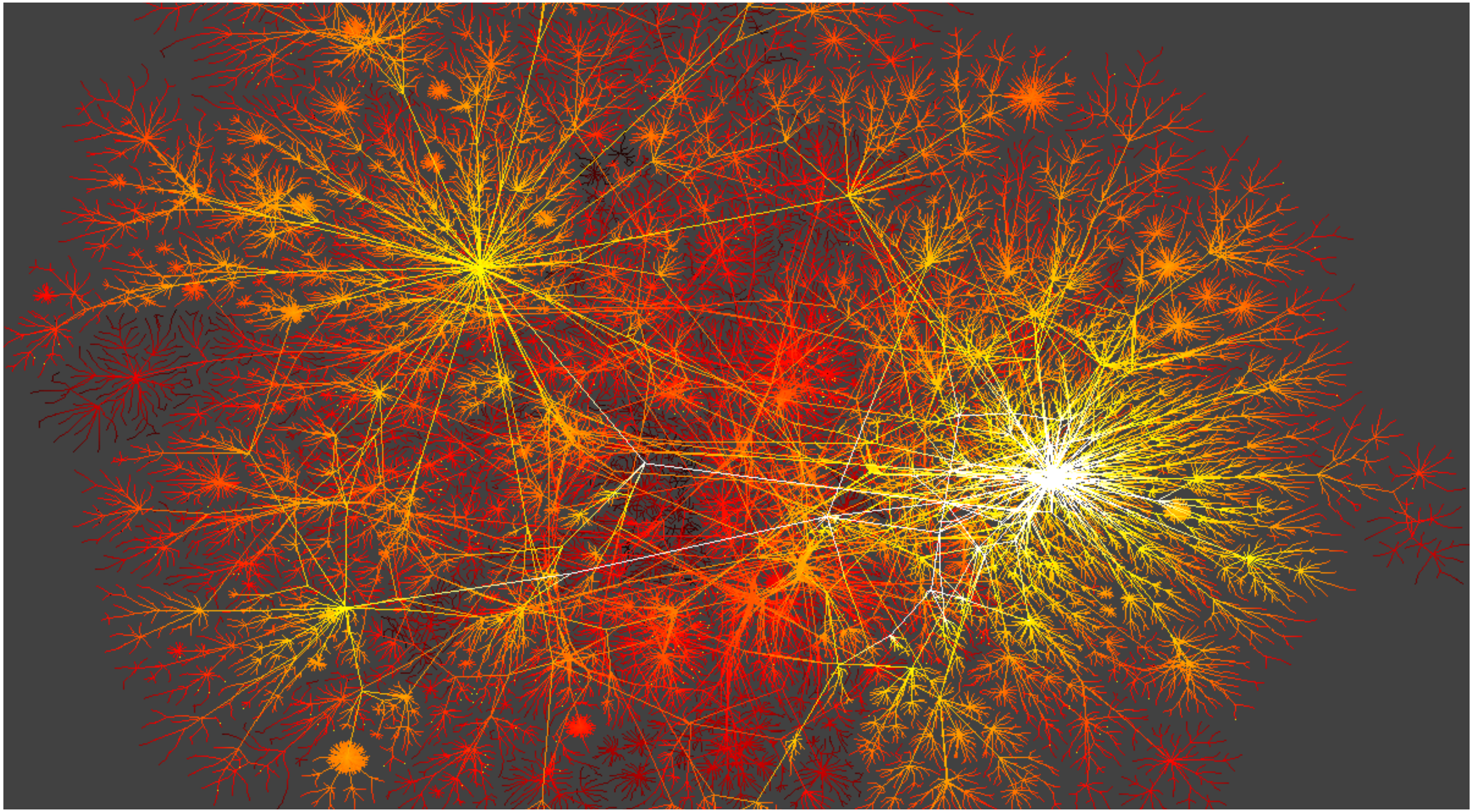
What independence??

Don’t ever use anything that assumes independence!

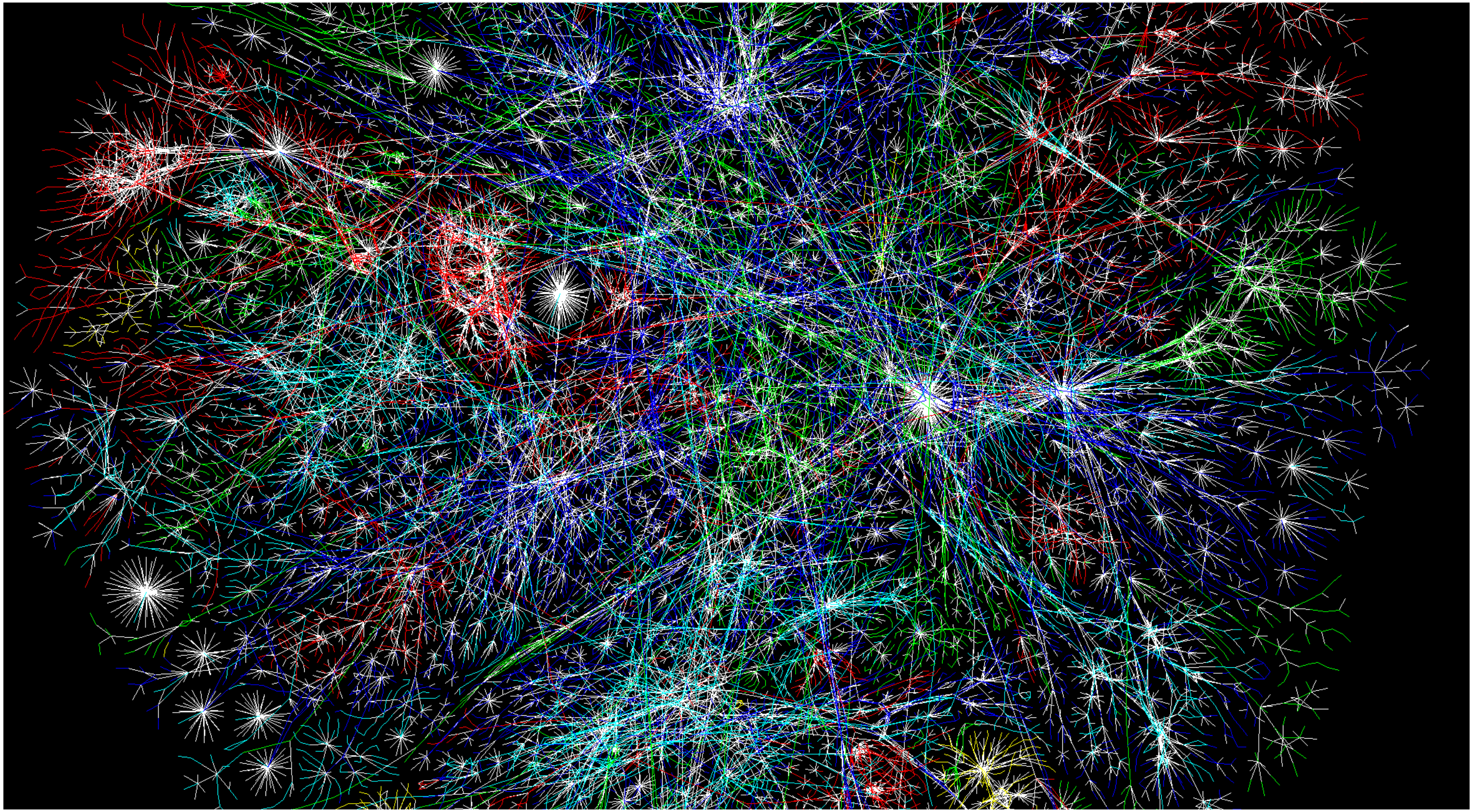
Network Visualization

One of the primary types of network analyses!

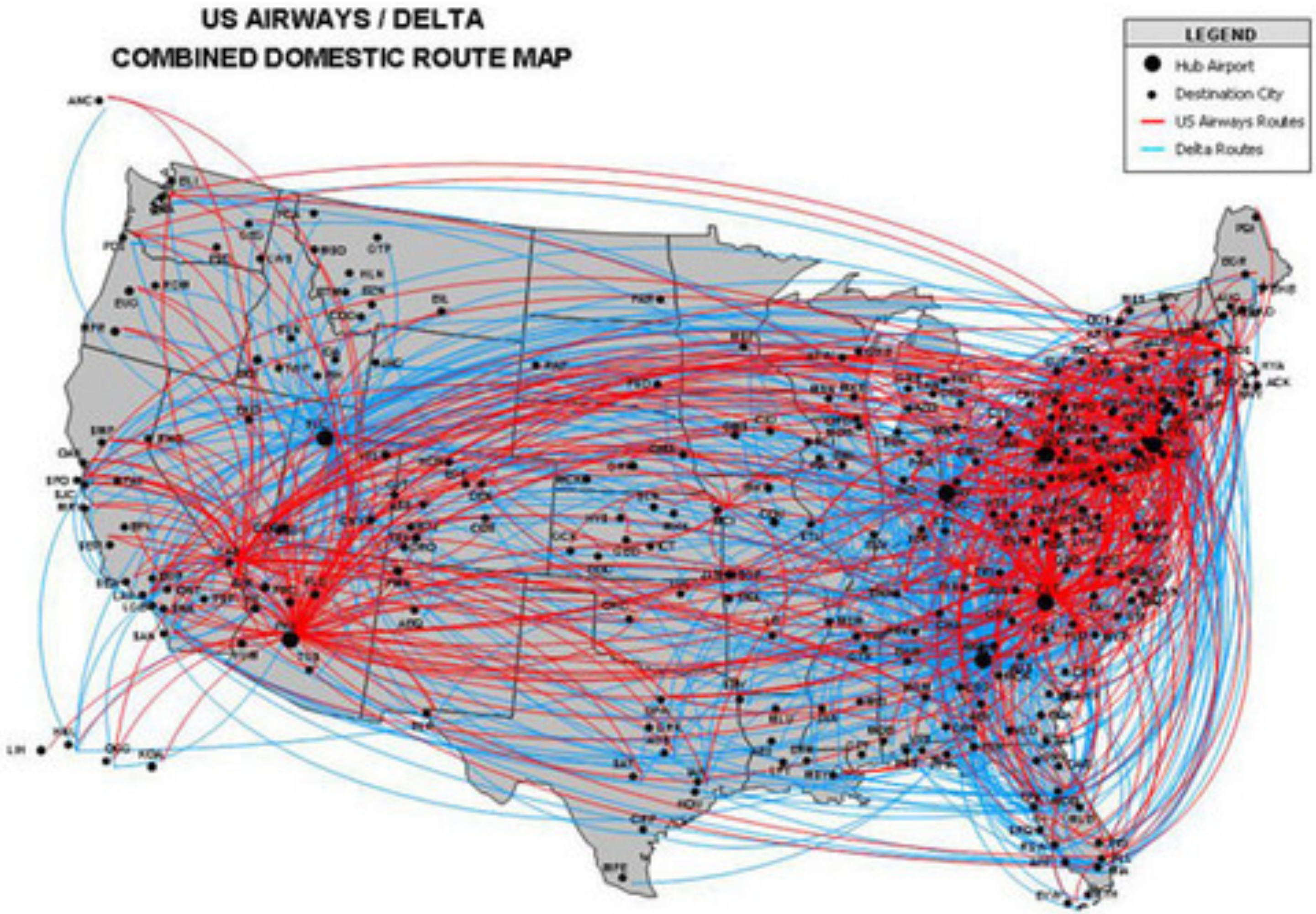
Sample of The Internet



Sample of The Internet

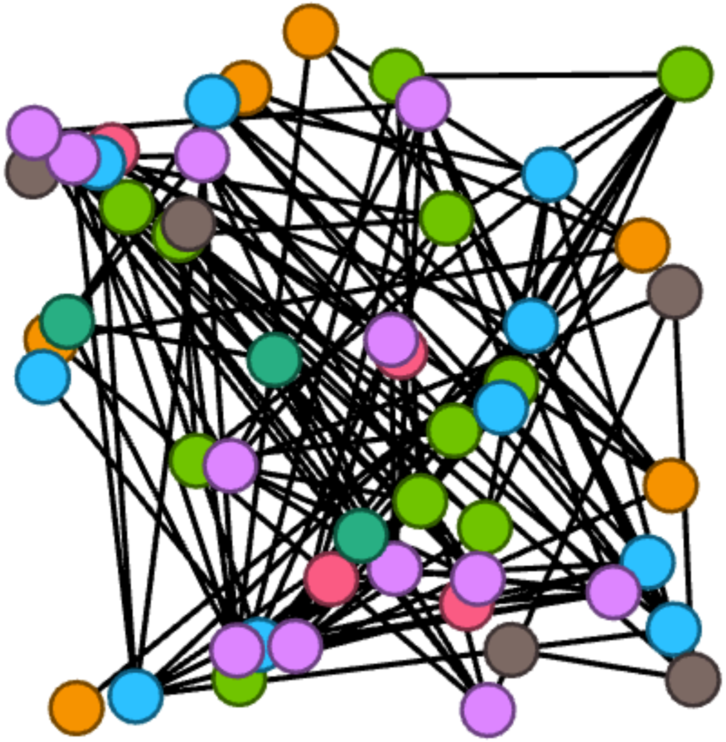


Air Transportation

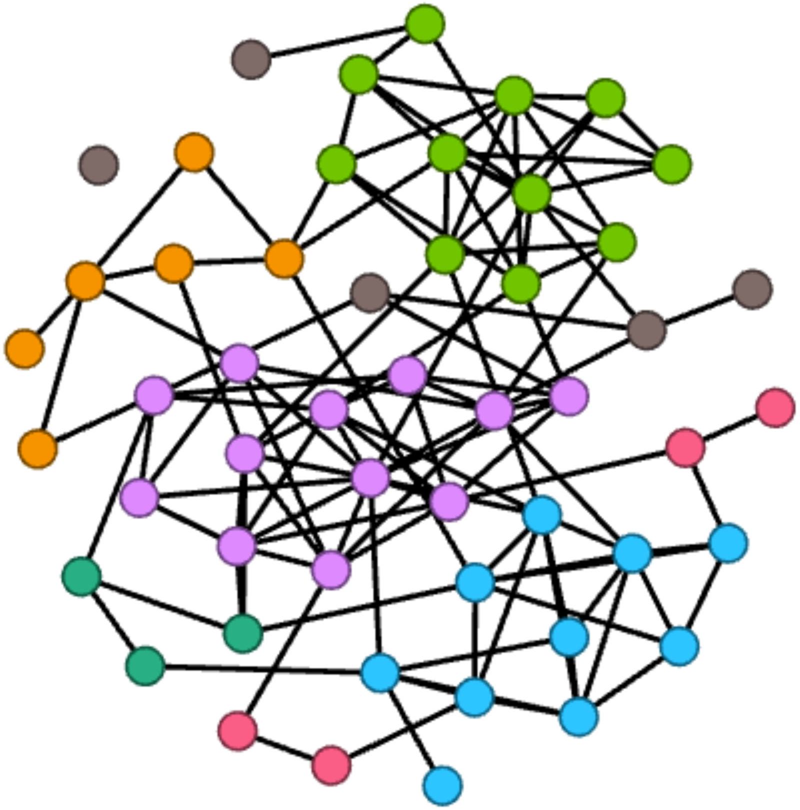


Layout is important!

Two drawings of the same graph!

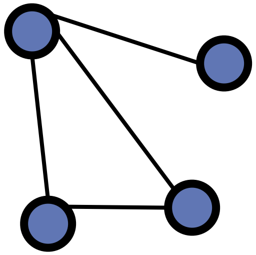


vs.

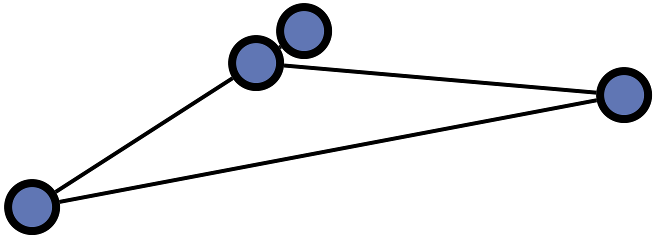


Considerations for Network Layout

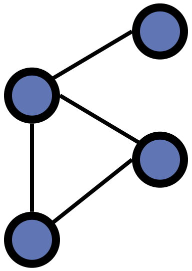
Edge lengths



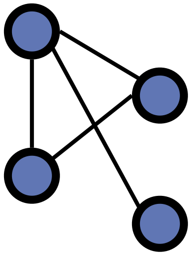
vs.



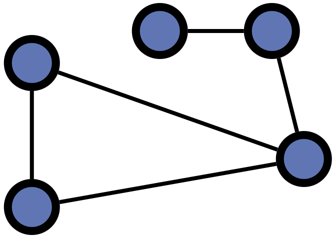
Edge crossings



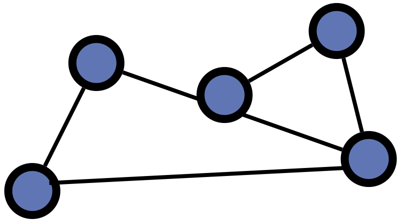
vs.



Node-edge overlap



vs.



Force-Directed Graph Drawing

- Family of algorithms designed to layout graphs

- Some attractive force is placed on adjacent vertices while a repulsive force placed on all vertices. Then the *energy* of the system is minimized.

Force-Directed Graph Drawing

Advantages

- Quality (great results)
- Versatility (adapt to all types of networks)
- Simplicity (to explain)

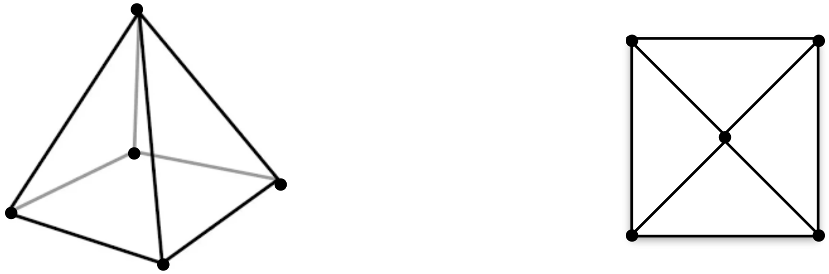
Disadvantages

- **Long** Run-time ~O(n^3) (Problem for large graphs)
  - Standard algorithm: Hundreds-Thousand nodes
  - Barnes-Hut Simulation O(n log(n)): ~100K nodes
  - Multi-level approaches: ~ million of nodes
- Some algorithms find poor local minima

Other Graph Drawings

Multidimensional Scaling

- Class of methods to best represent pairwise distances in 2-space.
- The distances between points in the resulting 2-dimensional approximation are as close as possible to the distances between points in the original space.



- Gives (x,y) coordinates based on the SVD/PCA of a similarity matrix
- For a network, the distance between 2 nodes is the length of shortest path between them.

Visualization Software

Gephi (Point/Click/GUI)

R

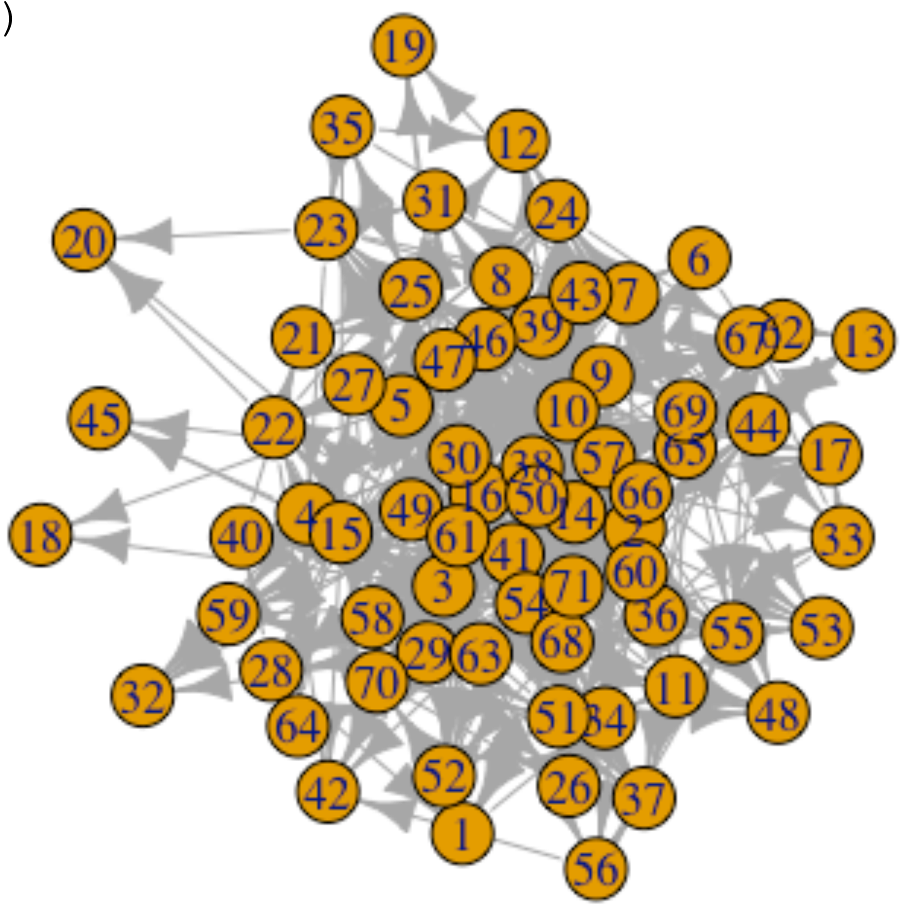
- igraph
- network
- networkD3

Python

- networkX

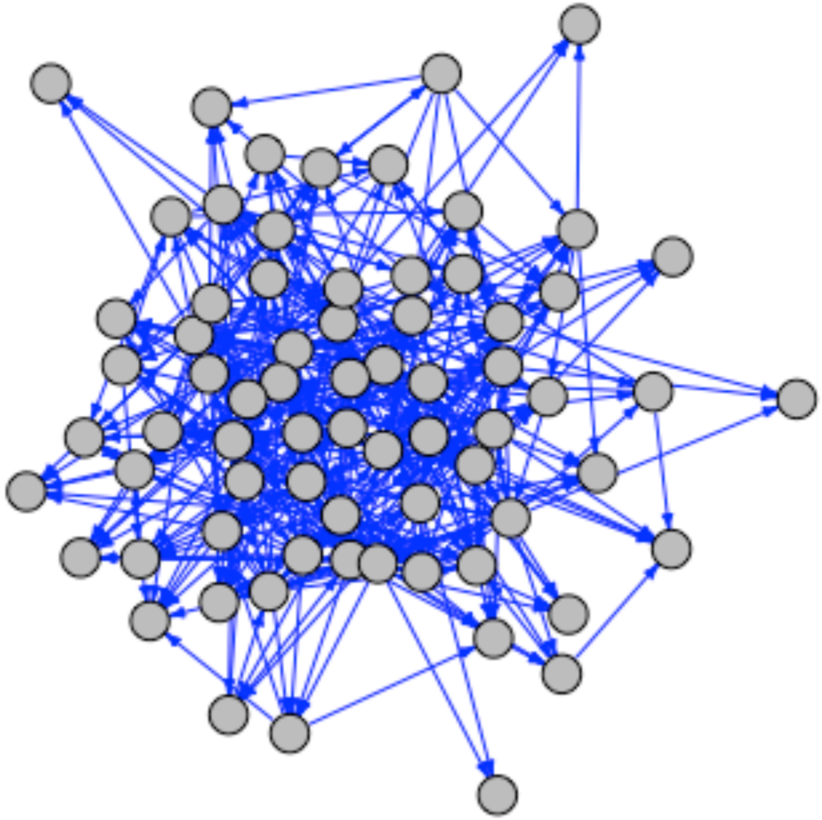
Default iGraph Plot

```
load("FIgraph.RData")
head(vertex_attr(FIgraph)) # Lists all the vertex data
plot(FIgraph)
```



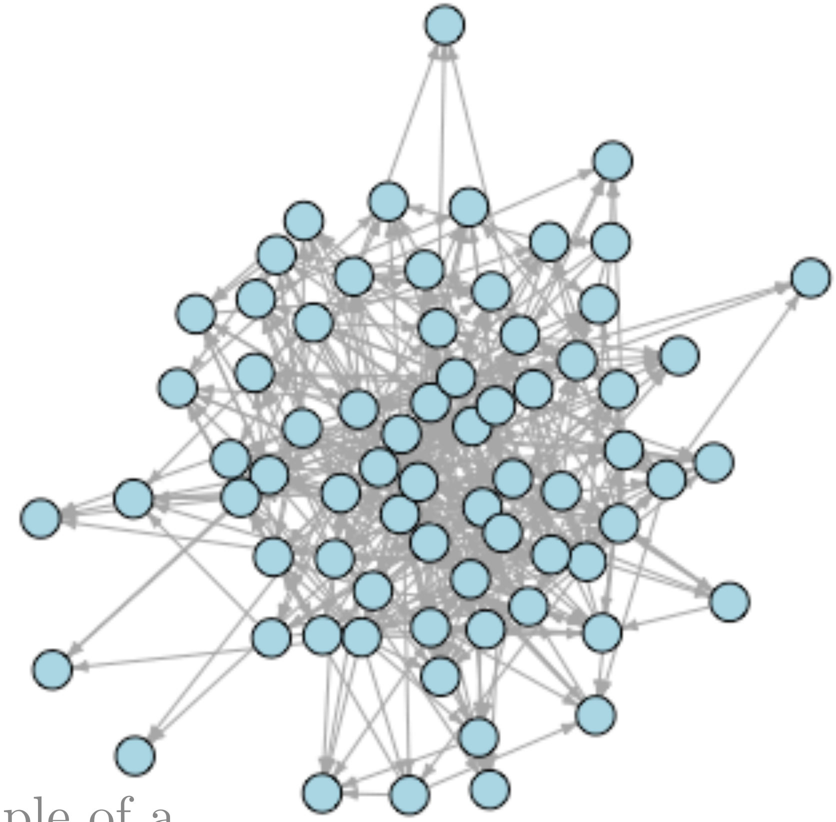
Manual Changes to Appearance

```
# Change the colors, suppress the labels,
# add arrows to directed edges
plot(FIgraph,
     edge.arrow.size = .3,
     vertex.label=NA,
     vertex.size=10,
     vertex.color='gray',
     edge.color='blue')
```



Change Layout of Nodes

```
# Change the layout of nodes
plot(FIgraph,
     edge.arrow.size = .3,
     vertex.label=NA,
     vertex.size=10,
     vertex.color='lightblue',
     layout=layout_with_fr)
```

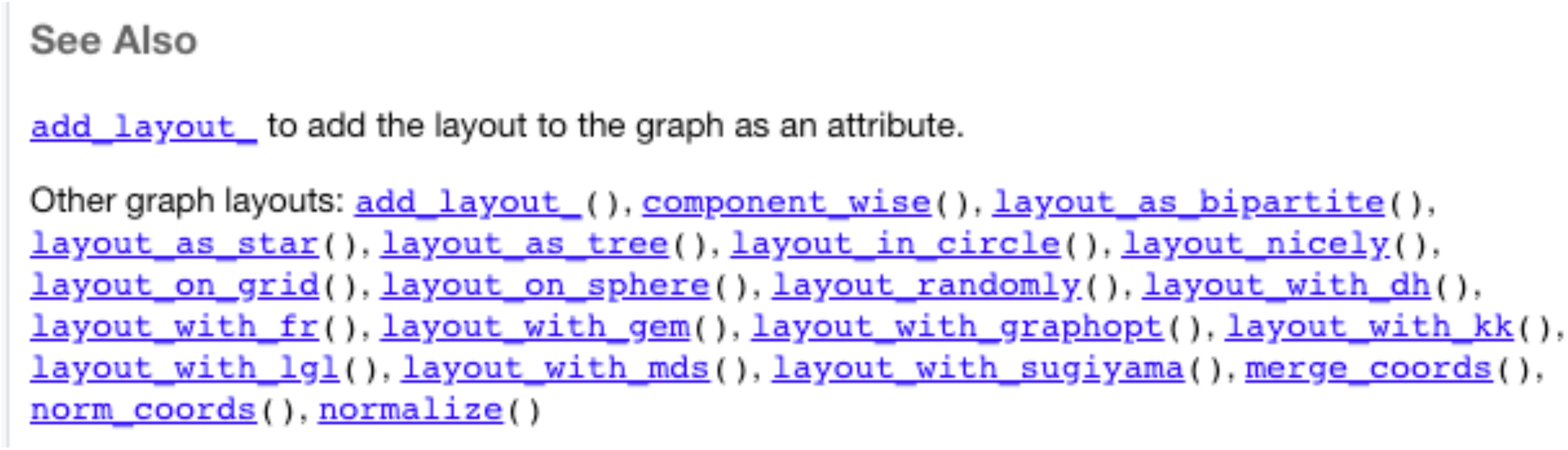


Fruchterman-Reingold is one example of a force directed layout

Save layout coordinates

This way the layout is not recomputed each time you change something like color or edge weight. Much faster.

```
?layout
```



See Also

add_layout_ to add the layout to the graph as an attribute.

Other graph layouts: add_layout_(), component_wise(), layout_as_bipartite(), layout_as_star(), layout_as_tree(), layout_in_circle(), layout_nicely(), layout_on_grid(), layout_on_sphere(), layout_randomly(), layout_with_dh(), layout_with_fr(), layout_with_gem(), layout_with_graphopt(), layout_with_kk(), layout_with_lgl(), layout_with_mds(), layout_with_sugiyama(), merge_coords(), norm_coords(), normalize()

```
l = layout_on_sphere(FIgraph)
l2 = layout_nicely(FIgraph)
l3 = layout_with_mds(FIgraph)
l4 =  layout_with_graphopt(FIgraph)
```

Save layout coordinates

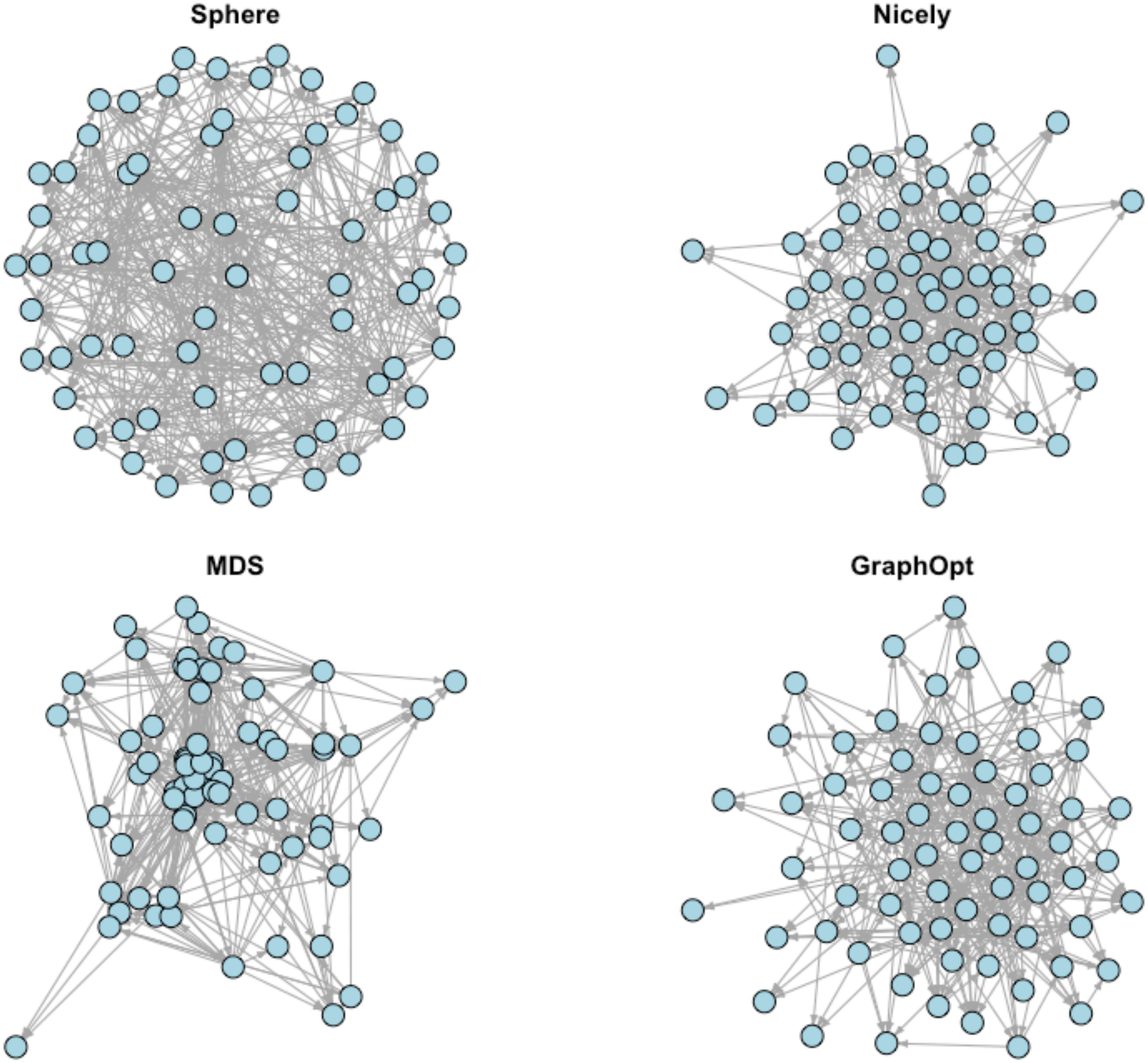
```
l = layout_on_sphere(slack)
l2 = layout_nicely(slack)
l3 = layout_with_mds(slack)
l4 =  layout_with_graphopt(slack)
par(mfrow=c(2,2),mar=c(1,1,1,1))

# Tells the graphic window to use the
# following plots to fill out a 2x2 grid with margins of 1 unit
# on each side. Must reset these options with dev.off() when done!

plot(slack, edge.arrow.size = .3, vertex.label=NA,vertex.size=10,
     vertex.color='lightblue', layout=l,main="Sphere")
plot(slack, edge.arrow.size = .3, vertex.label=NA,vertex.size=10,
     vertex.color='lightblue', layout=l2,main="Nicely")
plot(slack, edge.arrow.size = .3, vertex.label=NA,vertex.size=10,
     vertex.color='lightblue', layout=l3,main="MDS")
plot(slack, edge.arrow.size = .3, vertex.label=NA,vertex.size=10,
     vertex.color='lightblue', layout=l4,main = "GraphOpt")

dev.off() #resets the graphic window options.
```

Examples of Layouts



Sphere

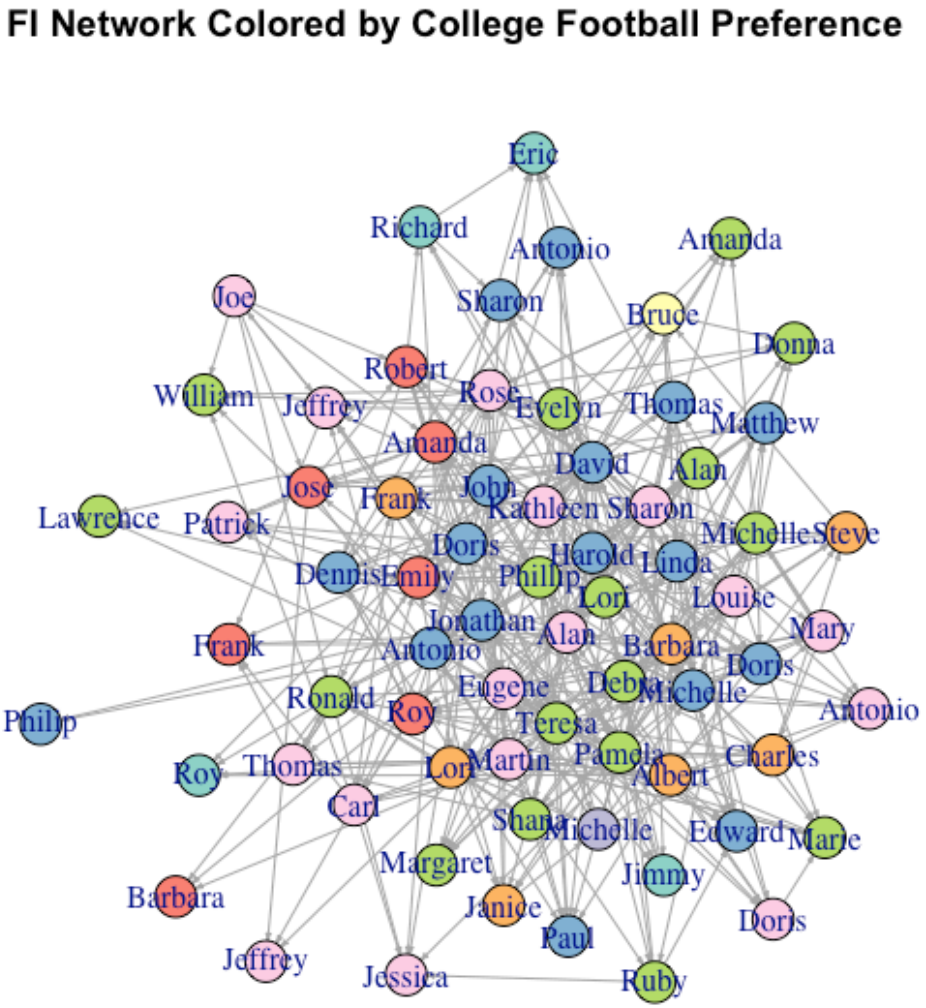
Nicely

MDS

GraphOpt

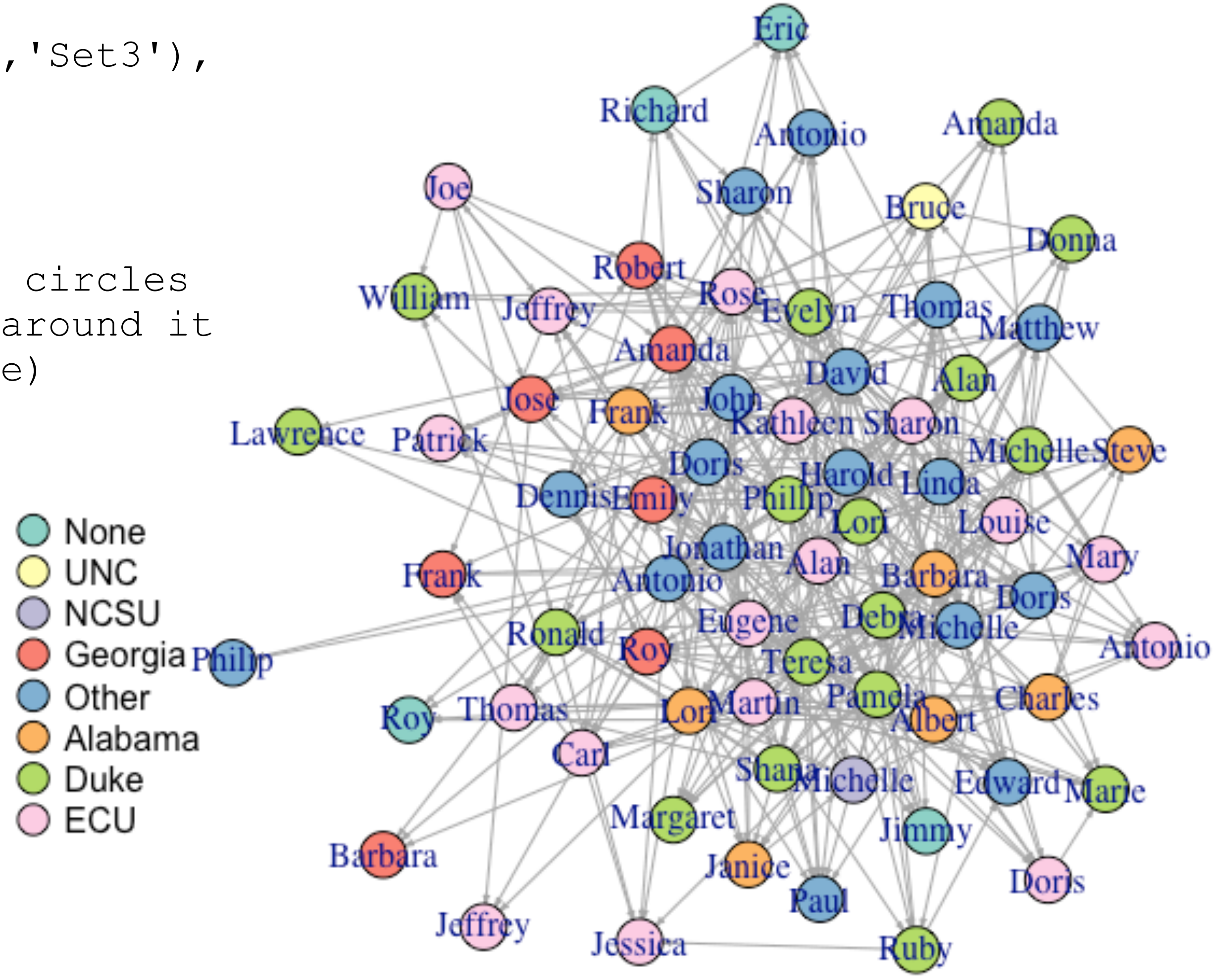
```
plot(FIgraph, edge.arrow.size = .2,
     vertex.label=V(FIgraph)$First,
     vertex.size=10,
     color=brewer.pal(8,'Set3')[as.factor(V(FIgraph)$CollegeFootball)],
     layout=l4,
     main = "FI Network Colored by College Football Preference")
```

Add Attribute information in label or color or size


FI Network Colored by College Football Preference

Add Legend

```
legend(x=-1.5,
       y=0,
       unique(V(FIgraph)$CollegeFootball),
       pch=21,
       pt.bg=brewer.pal(8,'Set3'),
       pt.cex=2,
       bty="n",
       ncol=1)
# pch =21 makes circles
# pt.cex controls size of circles
# bty="n" means no frame around it
# (switch to "y" for frame)
```



Create an igraph object from data frames

```
slack = graph_from_data_frame(SlackNetwork,
                              directed = TRUE,
                              vertices = users)
```

Package NetworkD3: BLUF

http://birch.iaa.ncsu.edu/~slrace/SNA2021/FIgraph.html

Data prep for NetworkD3

```
# This package insists that the label names (indices)
# of your nodes start from zero.

# To use this package, you need a data frame containing
# the edge list and a data frame containing the node data.

# Start by deleting any vertices that have no edges attached.

FIgraph=delete.vertices(FIgraph,degree(slack)==0)

# we need to create ID attribute taking values 0,..,103

nodes=data.frame(vertex_attr(FIgraph))
nodes$ID=0:(vcount(FIgraph)-1)

# data frame with edge list

edges=data.frame(get.edgelist(FIgraph))
colnames(edges)=c("source","target")
edges = edges-1
```

Package NetworkD3

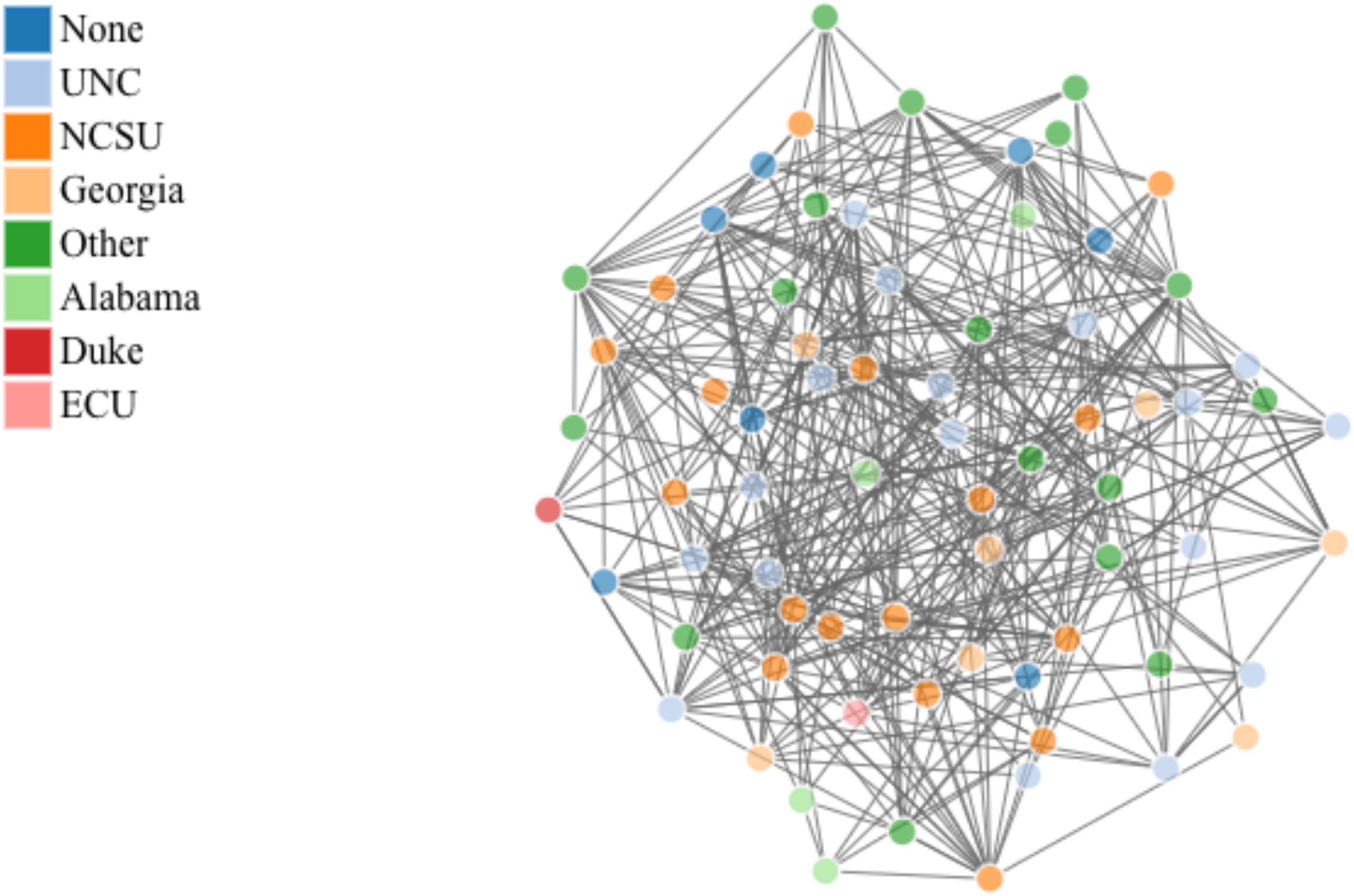
```
# data frame with node data
# reorder nodes table to match the 0,...,104 identifiers in
# edges table
nodes=nodes[order(nodes$ID),]

forceNetwork(Links=edges,
             Nodes=nodes,
             Source = "source",
             Target = "target",
             NodeID="First",
             Group="CollegeFootball",
             fontSize=12,
             opacity = 0.8,
             zoom=T,
             legend=T,
             charge = -100)
```

Save interactive html file

```
N =forceNetwork(Links=edges,
             Nodes=nodes,
             Source = "source",
             Target = "target",
             NodeID="First",
             Group="CollegeFootball",
             fontSize=12,
             opacity = 0.8,
             zoom=T,
             legend=T,
             charge = -100)
saveNetwork(N, file = ‘FI.html')
```

http://birch.iaa.ncsu.edu/~slrace/SNA2021/FIgraph.html



Your slack network pushes limit of Javascript for visualization.

http://birch.iaa.ncsu.edu/~slrace/SNA2021/Slack2021.html

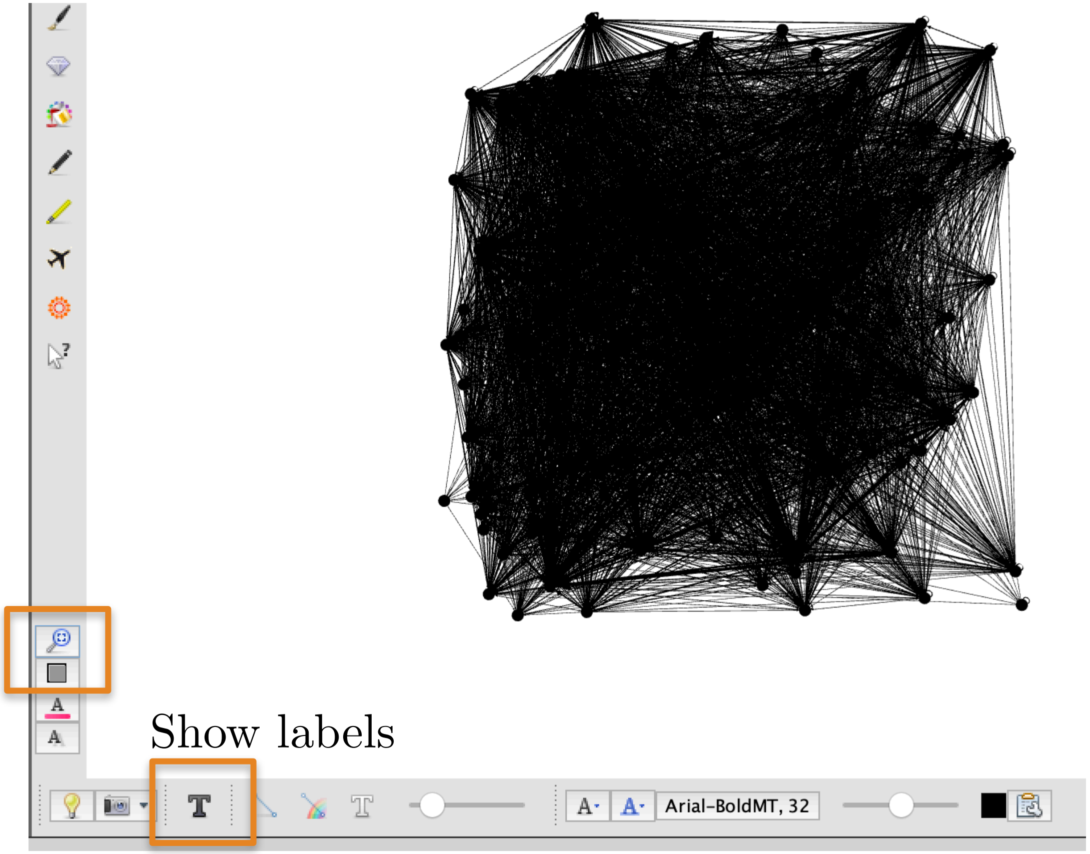
Write igraph object to .gml

```
write_graph(slack, file="MYPATH/slack2021.gml", format="gml")
```

Now you can open it in Gephi's opening prompt

Gephi: important things

- Scroll to zoom in/out
- RIGHT click and drag to pan



Arial-BoldMT, 32

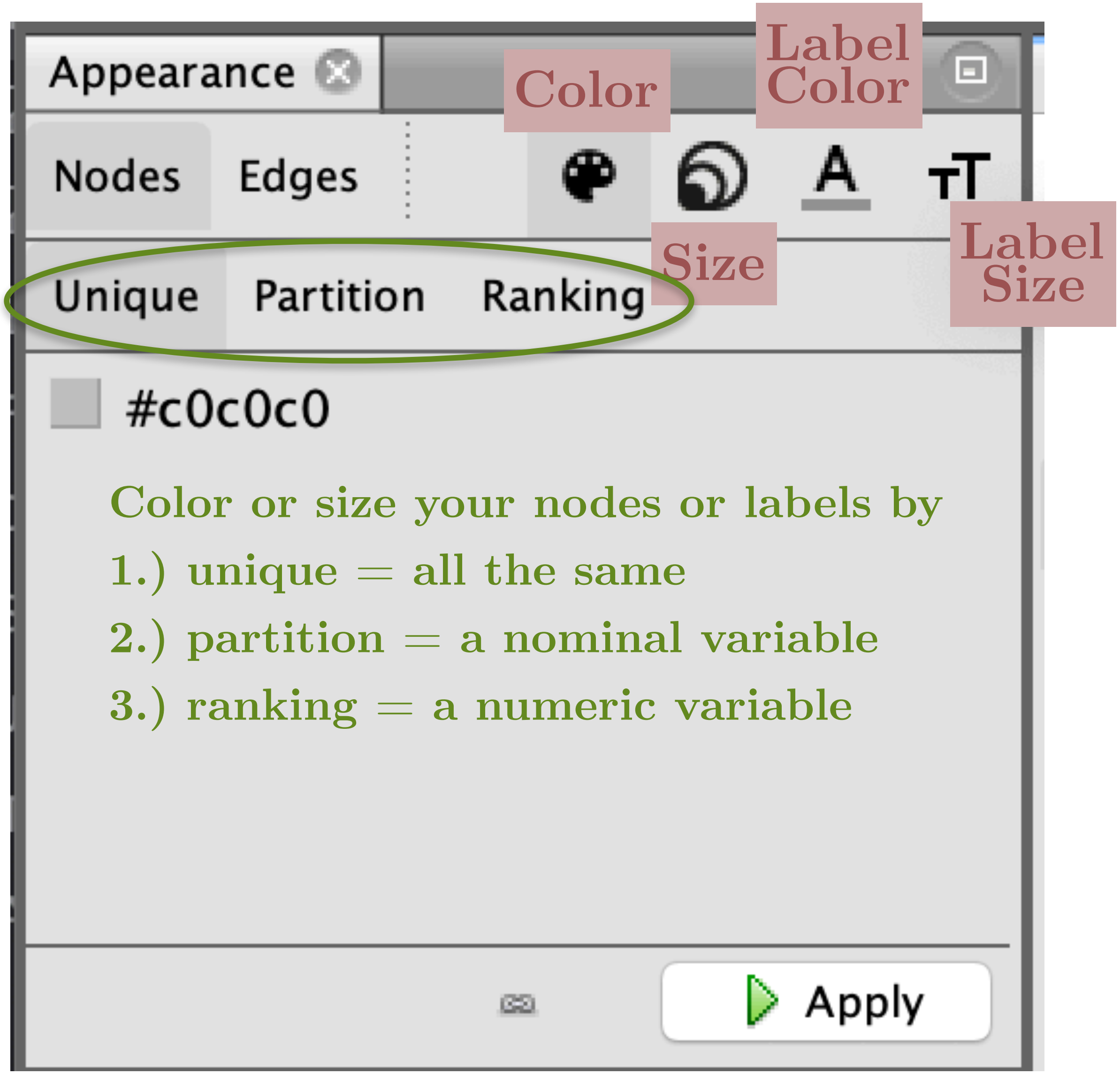
center on screen

(if you lose the graph)

Show labels

Sliders control weight of lines and labels

Gephi: Appearance Panel



#c0c0c0

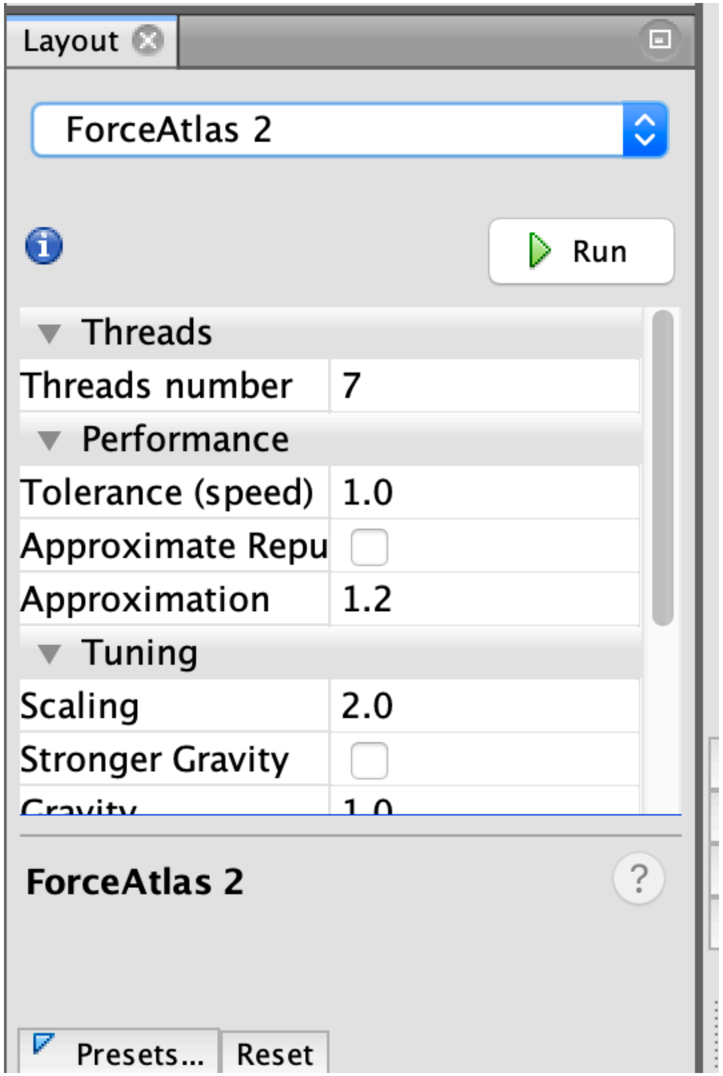
Color or size your nodes or labels by

1.) unique = all the same

2.) partition = a nominal variable

3.) ranking = a numeric variable

Gephi: Layouts



Layout ⊗

ForceAtlas 2

Run

Threads

Threads number	7

Performance

Tolerance (speed)	1.0
Approximate Repu	□
Approximation	1.2

Tuning

Scaling	2.0
Stronger Gravity	□
Gravity	1.0

ForceAtlas 2

Presets... Reset

Try them all and play around with the parameters to achieve what you want.

ForceAtlas 2 is fast.

Expansion and Contraction keep the layout but spread everything out. Very useful!

Data Formats

• • •

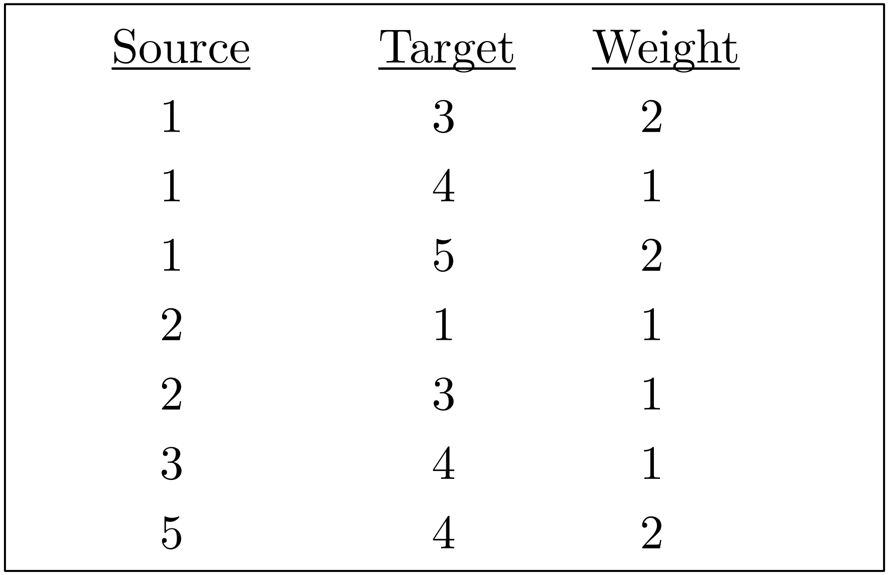
Edgelists, Adjacency Matrices, Lists of Neighbors

Graph Data Formats

- Many possible formats (.gml, .csv, .gephi, etc)
- Many include built-in ways to include individual variables (e.g. gender, age)
  - Node table
  - Node attributes
- All have some way to list edges (structural variables) - sometimes with attributes (for example, messages in ‘general’ vs. messages in ‘linear-algebra’.)
  - Edge list
  - List of neighbors
  - Adjacency Matrix

Edge List Format

- Lists the source and target of each edge. Numbering of nodes dependent on software. (python: 0…n) (R: 1…n package dependent)
- Some formats allow edge attributes (type/weight)



Source	Target	Weight
1	3	2
1	4	1
1	5	2
2	1	1
2	3	1
3	4	1
5	4	2

Neighbors List Format

- Best choice for large graphs
- 1 data line for each node, listing all of its neighbors:

```
1:   3,4,5
2:   1,3
3:   4
4:
5:   4
```

- Cannot easily incorporate edge attributes to this data structure.