PLSC 504: Introduction to Network Analysis

November 13, 2024

Network Theory

From our friends at
`https://en.wikipedia.org/wiki/Network_theory`:

> **Network theory** *is the study of graphs as a representation of either symmetric relations or asymmetric relations between discrete objects. In computer science and network science, network theory is a part of graph theory: a network can be defined as a graph in which nodes and/or edges have attributes (e.g. names).*

Networks: Terms

Network *components*:

- *Nodes*: The units that are (potentially) connected in the network (a.k.a. *actor*, *vertex*).
- *Edges*: The connections between the nodes that form the network (a.k.a. *tie*, *link*).

Network *characteristics*:

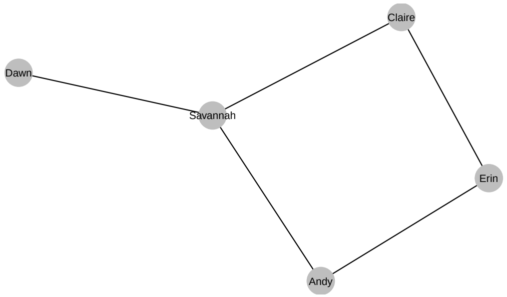
- *Undirected* vs. *Directed*
- *Size*: The number of nodes in the network.
- *Diameter*: The largest shortest path between any two nodes in the network.
- *Density*: The proportion of edges to possible edges.
- *Geodesic Distance*: The shortest path between two nodes
- *Average Path Length*: The average (arithmetic mean) of the shortest path lengths between nodes.

Networks: Types

Networks can be:

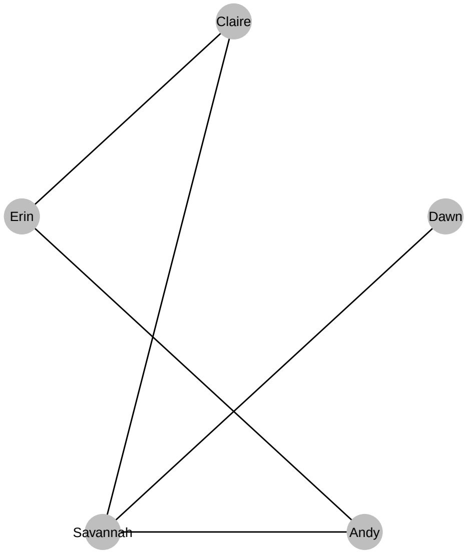
- **Directed** (*digraph*) vs. **undirected**
- **Connected** vs. **disconnected**
  - **Strongly** vs. **weakly** connected
  - **Vertex cuts** and **vertex connectivity**
  - Superconnectivity, hyperconnectivity, etc...
- **Bipartite**: Two types of nodes ($\rightarrow$ multimodal)
- **Others**: Dynamic, multilayer, etc.

A Really Basic (Non-Directed) Network

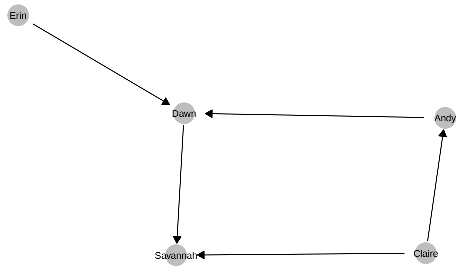


```
> N5[1:5,1:5]
         Claire Dawn Andy Savannah Erin
Claire        0    0    0        1    1
Dawn          0    0    0        1    0
Andy          0    0    0        1    1
Savannah      1    1    1        0    0
Erin          1    0    1        0    0
```

Same Network, Different (Circle) Plot

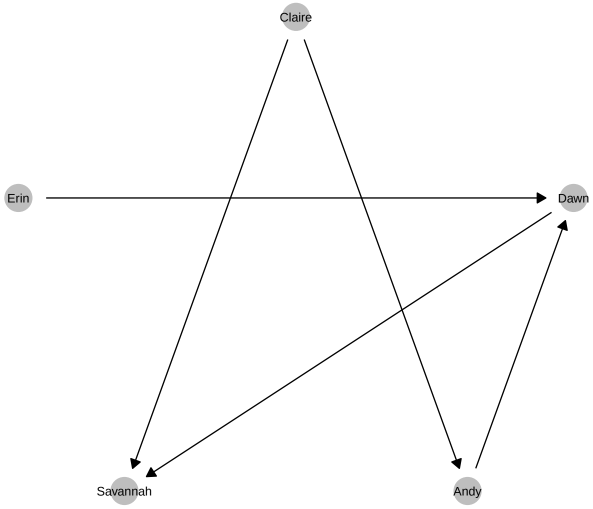


A Really Basic Directed Network

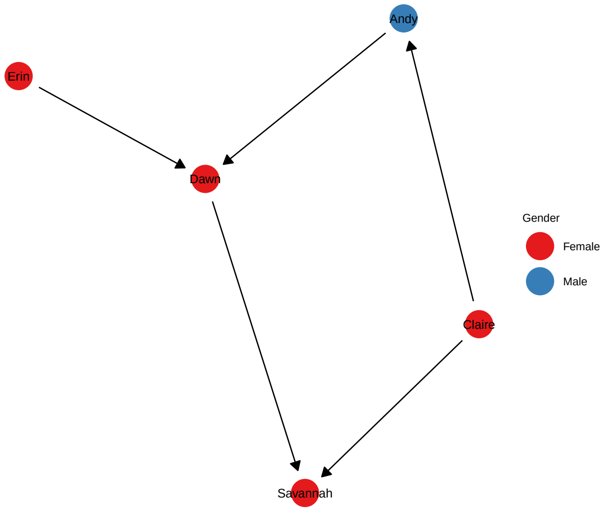


```
> N5D[1:5,1:5]
         Claire Dawn Andy Savannah Erin
Claire        0    0    1        1    0
Dawn          0    0    0        1    0
Andy          0    1    0        0    0
Savannah      0    0    0        0    0
Erin          0    1    0        0    0
```

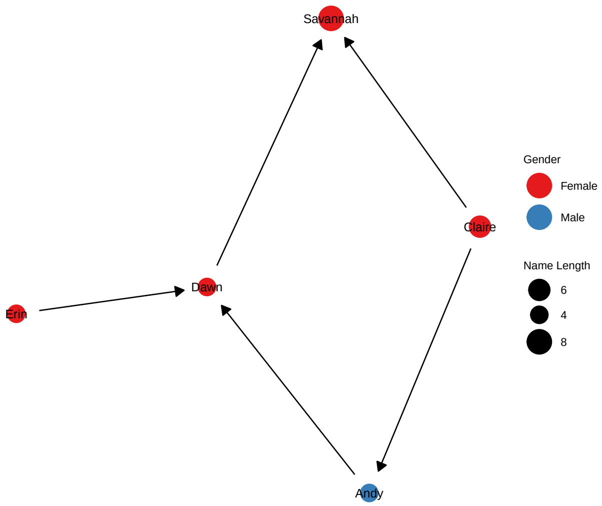
Same Network, Circle Plot



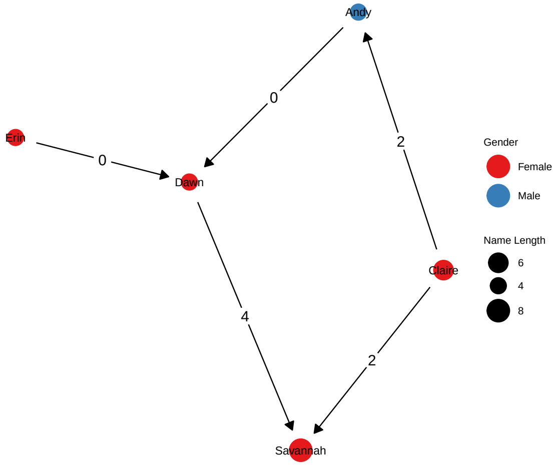
Add Node Characteristics



More Node Characteristics (Name Length)



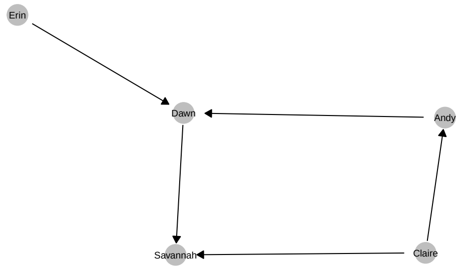
Adding Edge Characteristics (Differences in Name Length)



Basic Statistics On Networks

Summarizing network features:

- Node-level:
  - Centrality
  - Local transitivity, etc.
- Edge-level (e.g., relatedness)
- Network-level
  - Reciprocity
  - Global transitivity



Centrality

What makes a node "central"?

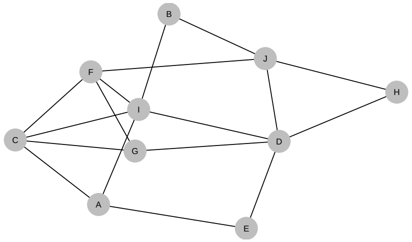
- Importance / Influence?
- Connectedness?
- Vulnerability to contagion?
- "Anchor points"?

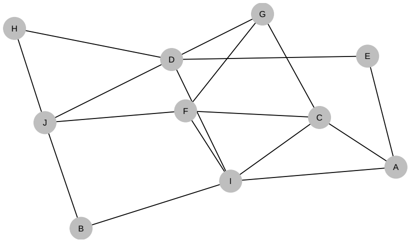
Aspects of Centrality:

- Node Positioning:
    - Radial (paths originate / terminate at a node)
    - Medial (paths pass through a node)
- Properties of Paths:
    - Volume (paths contribute according to their numbers/frequency)
    - Length (paths contribute according to their length)

The (Potential) Importance of Visualization

Two views of the same network:



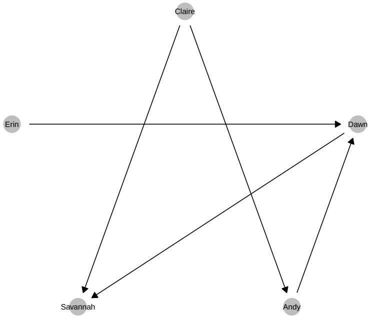


Node Centrality: Degree

Node Properties: *Degree*

- The number of edges adjacent to that node (*total* degree)
- In directed graphs: *indegree* and *outdegree* (+ total)
- A node with a degree of zero is an *isolated node*
- A node with a degree of one is a *leaf* (or *end*) node

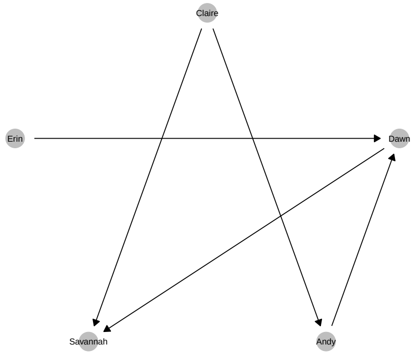
```
> cbind(network.vertex.names(N5D),sna::degree(N5D),
+       sna::degree(N5D,cmode="indegree"),
+       sna::degree(N5D,cmode="outdegree"))
     [,1]       [,2] [,3] [,4]
[1,] "Claire"   "2"  "0"  "2"
[2,] "Dawn"     "3"  "2"  "1"
[3,] "Andy"     "2"  "1"  "1"
[4,] "Savannah" "2"  "2"  "0"
[5,] "Erin"     "1"  "0"  "1"
```



Node Centrality: Betweenness

The **betweenness** of a node is (proportional to) the extent to which that node lies on paths between other vertices.

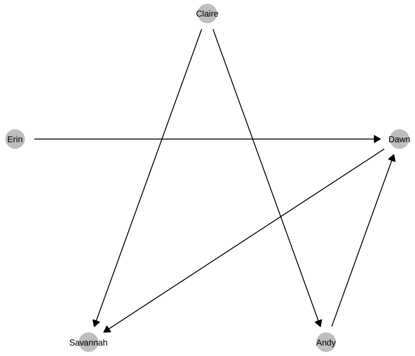
```
> cbind(network.vertex.names(N5D),sna::betweenness(N5D
     [,1]       [,2]
[1,] "Claire"   "0"
[2,] "Dawn"     "2"
[3,] "Andy"     "1"
[4,] "Savannah" "0"
[5,] "Erin"     "0"
```



Node Centrality: Closeness

The **closeness** of a node *i* is (proportional to) the number of steps that are required to access every other node from that given node *i*.

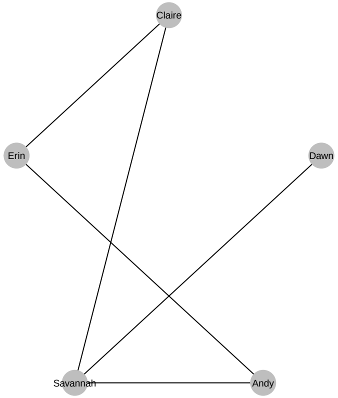
```
> cbind(network.vertex.names(N5D),
+       sna::closeness(N5D),
+       igraph::closeness(asIgraph(N5D)))
     [,1]       [,2] [,3]
[1,] "Claire"   "0"  "0.111111111111111"
[2,] "Dawn"     "0"  "0.0625"
[3,] "Andy"     "0"  "0.0769230769230769"
[4,] "Savannah" "0"  "0.05"
[5,] "Erin"     "0"  "0.0769230769230769"
```



Eigenvector Centrality

- "How well-connected a given node is to well-connected nodes."
- A node's eigenvector centrality is the *i*th component of the first eigenvector of the network's adjacency matrix.
- Only defined for undirected networks...

```
> EC<-igraph::eigen_centrality(asIgraph(N5))
> cbind(network.vertex.names(N5),
+       EC$vector)
     [,1]       [,2]
[1,] "Claire"   "0.83378300630386"
[2,] "Dawn"     "0.468213192462135"
[3,] "Andy"     "0.83378300630386"
[4,] "Savannah" "1"
[5,] "Erin"     "0.780776406404414"
```



Choosing Among Centralities

Intuition:

- Degree: basic “connectedness;” akin to “popularity”
- Betweenness: information flow / loss of information if node is removed
- Closeness: proximity; “influencers”
- Eigenvector: “embeddedness”

Centrality: Why We Care

- Measure of node “importance”:
    - Closeness = proximity
    - Betweenness = information loss if the node is removed
    - Eigenvector = “significance”

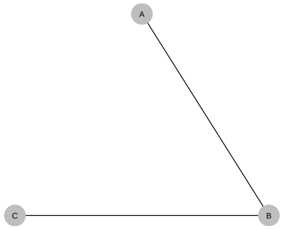
- Can serve as weights in other analyses

- Play an important role in regression-like models on networks...

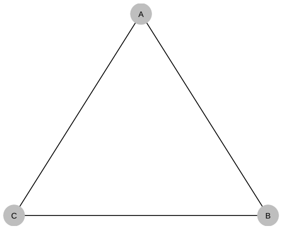
Transitivity

Transitivity is the tendency of the last two nodes of a two-path to receive an edge from the first node.

- Undirected networks: “closure” (or “the friend of my friend is also my friend”)
- Directed networks: directionality (transitivity vs. cyclicality)



“Two star”

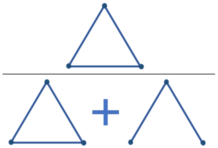


“Triplet”

Transitivity (continued)

What's the point of transitivity?

- It measures the extent of "clustering" in the network
- Practical / substantive interpretations:
  - Cohesiveness / clustering of nodes
  - Information redundancies
  - Polarization (few edges between tight clusters of nodes)
- Node-level measure: *clustering coefficient* (the proportion of a node's neighbors that are tied)
- Network-level measure: *transitivity coefficient* T
  - Essentially $T = \frac{\text{Transitive triples}}{\text{Triples with 0 or 1 ``missing'' edges}}$
  - That is:



Why Model Networks?

- Account for network structure
- Predict edges (connections)
- Test hypotheses about network structures
- Deal with higher order dependencies
- Assess the (commonly-made) conditional independence assumption...

ERGMs / GERGMs

- “(Generalized) Exponential Random Graph Models”
- Regression-like models for nodes and edges in a network
- Originally for binary (present/absent) edges $\rightarrow$ “valued” networks $\rightarrow$ continuous edge values
- Common structure / form with GLMs / “exponential family” regression models

ERGM Details

Start with an N-node network defined by:

$$\underset{N\times N}{\mathbf{Y}} = \begin{cases} Y_{ij} = 1 \text{ if } i, j \text{ have a tie} \\ Y_{ij} = 0 \text{ otherwise} \end{cases}$$

and $\mathbf{X}$ is a matrix of covariates (structural network traits + node/edge-level predictors).

An ERGM is:

$$\Pr(\mathbf{Y}|\mathbf{X}) = \frac{\exp[\theta' g(\mathbf{Y}, \mathbf{X})]}{\sum \exp[\theta, g(\mathcal{Y})]}$$

where:

- θ is a vector of parameters,
- $g(\mathbf{Y}, \mathbf{X})$ is a set of sufficient statistics, and
- $g(\mathcal{Y})$ is the set of all possible networks with N nodes

ERGM: Computation

Note that $g(\mathcal{Y})$ is typically *very* large...

- For N nodes, there are $2^{\frac{N(N-1)}{2}}$ possible labeled, undirected networks
- E.g., for $N = 10$, that's 35,184,372,088,832 possible networks
- Calculating likelihoods for each of them is... unwise.

Alternatives:

- **MPLE**:

$$\Pr(\mathbf{Y}|\mathbf{X}) \approx \prod \Pr(Y_{ij}|\mathbf{Y}_{-ij}, \mathbf{X}_{-ij})$$

  ...which has issues (see e.g. Schmid & Desmarais 2017)

- More common: Sampling from posterior via **MCMC**

ERGMs: SCOTUS Example, again...

SCOTUS data:

- Votes – *liberal* (=1) or *conservative* (=0) on $\approx$ 1400 cases, OT1994-2005
- Network structure: degree of *agreement* among justices' votes...
  - Always vote "opposite" = 0% agreement
  - Always vote the same way = 100% agreement
- Agreement *network*:
  - Undirected
  - 1 = 50 percent or greater voting agreement
  - 0 = less than 50 percent voting agreement
- Node characteristics: Justices' Segal-Cover (liberalism) scores + appointment by GOP president

SCOTUS: Agreement matrix...

Agreement matrix:

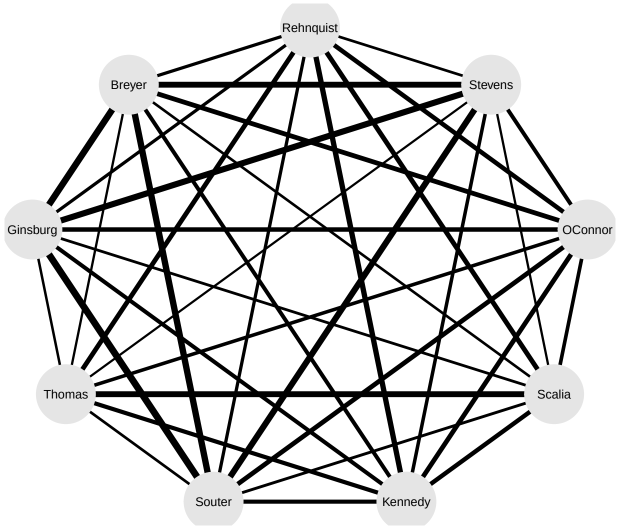
```
> SCAgree

          Rehnquist Stevens OConnor Scalia Kennedy Souter Thomas Ginsburg Breyer
Rehnquist              0.3400  0.5738 0.6094  0.6698 0.4132 0.5996   0.4076 0.4062
Stevens      0.3400            0.5000 0.2905  0.4431 0.7646 0.2803   0.7731 0.7400
OConnor      0.5738  0.5000           0.4592  0.5964 0.5952 0.4484   0.5648 0.6017
Scalia       0.6094  0.2905  0.4592           0.5596 0.3657 0.7425   0.3514 0.3322
Kennedy      0.6698  0.4431  0.5964 0.5596         0.5188 0.5302   0.5090 0.5029
Souter       0.4132  0.7646  0.5952 0.3657  0.5188        0.3410   0.8587 0.8002
Thomas       0.5996  0.2803  0.4484 0.7425  0.5302 0.3410          0.3180 0.2934
Ginsburg     0.4076  0.7731  0.5648 0.3514  0.5090 0.8587 0.3180          0.8199
Breyer       0.4062  0.7400  0.6017 0.3322  0.5029 0.8002 0.2934   0.8199
```

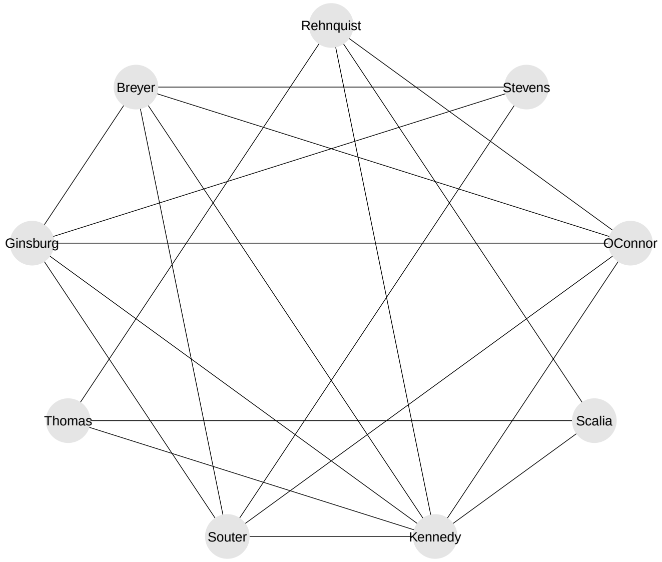
→ Network structure:

```
> SCA
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
 [1,]    1    0    1    1    1    0    1    0    0
 [2,]    0    1    0    0    0    1    0    1    1
 [3,]    1    0    1    0    1    1    0    1    1
 [4,]    1    0    0    1    1    0    1    0    0
 [5,]    1    0    1    1    1    1    1    1    1
 [6,]    0    1    1    0    1    1    0    1    1
 [7,]    1    0    0    1    1    0    1    0    0
 [8,]    0    1    1    0    1    1    0    1    1
 [9,]    0    1    1    0    1    1    0    1    1
```

SCOTUS Agreement



SCOTUS Agreement Network (binary plot)



What Are We Working With?

```
> summary(SCN)
Network attributes:
  vertices = 9
  directed = FALSE
  hyper = FALSE
  loops = FALSE
  multiple = FALSE
  bipartite = FALSE
 total edges = 20
   missing edges = 0
   non-missing edges = 20
 density = 0.5555556

Vertex attributes:

 GOPPres:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  1.0000  1.0000  0.7778  1.0000  1.0000
.
.
.
```

What Are We Working With (cont'd)?

```
.
.
.
 SegalCover:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.1600  0.3250  0.3017  0.4150  0.6800
  vertex.names:
   character valued attribute
   9 valid vertex names

No edge attributes

Network adjacency matrix:
          Rehnquist Stevens OConnor Scalia Kennedy Souter Thomas Ginsburg Breyer
Rehnquist         0       0       1      1       1      0      1        0      0
Stevens           0       0       0      0       0      1      0        1      1
OConnor           1       0       0      0       1      1      0        1      1
Scalia            1       0       0      0       1      0      1        0      0
Kennedy           1       0       1      1       0      1      1        1      1
Souter            0       1       1      0       1      0      0        1      1
Thomas            1       0       0      1       1      0      0        0      0
Ginsburg          0       1       1      0       1      1      0        0      1
Breyer            0       1       1      0       1      1      0        1      0
```

Simple ERGM: Edges Only

```
> fitE<-ergm(SCN~edges,estimate="MPLE")
Starting maximum pseudolikelihood estimation (MPLE):
Obtaining the responsible dyads.
Evaluating the predictor and response matrix.
Maximizing the pseudolikelihood.
Finished MPLE.
Evaluating log-likelihood at the estimate.

> summary(fitE)

Call:
ergm(formula = SCN ~ edges, estimate = "MPLE")

Maximum Likelihood Results:

      Estimate Std. Error MCMC % z value Pr(>|z|)
edges   0.2231     0.3354      0   0.665    0.506

For this model, the pseudolikelihood is the same as the likelihood.

     Null Deviance: 49.91  on 36  degrees of freedom
 Residual Deviance: 49.46  on 35  degrees of freedom

AIC: 51.46  BIC: 53.04  (Smaller is better. MC Std. Err. = 0)


> # Same as density / proportion of edges:
>
> plogis(fitE$coefficients[1])
    edges
0.5555556
```

Edges + Triangles

```
> fitET<-ergm(SCN~edges+triangle,estimate="MPLE",
+            control=control.ergm(MPLE.covariance.method='Godambe'))

Starting maximum pseudolikelihood estimation (MPLE):
Obtaining the responsible dyads.
Evaluating the predictor and response matrix.
Maximizing the pseudolikelihood.
Estimating Godambe Matrix using 500 simulated networks.
Finished MPLE.
Evaluating log-likelihood at the estimate.

> summary(fitET)

Call:
ergm(formula = SCN ~ edges + triangle, estimate = "MPLE",
  control = control.ergm(MPLE.covariance.method = "Godambe"))

Maximum Pseudolikelihood Results:

         Estimate Std. Error MCMC % z value Pr(>|z|)
edges     -3.1780     0.3212      0  -9.893   <1e-04 ***
triangle   1.6651     0.1702      0   9.783   <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 49.91  on 36  degrees of freedom
 Residual Deviance: 33.92  on 34  degrees of freedom

AIC: 37.92  BIC: 41.09  (Smaller is better. MC Std. Err. = 0)
```

Node Covariates: Party ID

Do justices appointed by the same party president tend to vote similarly to each other?

```
> fit1<-ergm(SCN~nodematch("GOPPres")+edges+triangle,estimate="MPLE",
+       control=control.ergm(MPLE.covariance.method='Godambe'))


> summary(fit1)

Call:
ergm(formula = SCN ~ nodematch("GOPPres") + edges + triangle,
    estimate = "MPLE", control = control.ergm(MPLE.covariance.method = "Godambe"))

Maximum Pseudolikelihood Results:

                  Estimate Std. Error MCMC % z value Pr(>|z|)
nodematch.GOPPres  0.19403    0.03452      0   5.621   <1e-04 ***
edges             -3.30202    0.29653      0 -11.135   <1e-04 ***
triangle           1.66714    0.14461      0  11.529   <1e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 49.91  on 36  degrees of freedom
 Residual Deviance: 33.87  on 33  degrees of freedom

AIC: 39.87  BIC: 44.62  (Smaller is better. MC Std. Err. = 0)
```

Node Covariates: Justice Ideology

Do ideologically similar justices tend to vote similarly to each other?

```
> fit2<-ergm(SCN~absdiff("SegalCover")+edges+triangle,estimate="MPLE",
+            control=control.ergm(MPLE.covariance.method='Godambe'))
```

```
> summary(fit2)

Call:
ergm(formula = SCN ~ absdiff("SegalCover") + edges + triangle,
    estimate = "MPLE", control = control.ergm(MPLE.covariance.method = "Godambe"))

Maximum Pseudolikelihood Results:

                   Estimate Std. Error MCMC % z value Pr(>|z|)
absdiff.SegalCover  -4.4516     0.5028      0  -8.853  < 1e-04 ***
edges               -1.4978     0.5057      0  -2.962  0.00306 **
triangle             1.4324     0.2130      0   6.726  < 1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 49.91  on 36  degrees of freedom
 Residual Deviance: 31.57  on 33  degrees of freedom

AIC: 37.57  BIC: 42.32  (Smaller is better. MC Std. Err. = 0)
```

Both Covariates

```
> fit3<-ergm(SCN~nodematch("GOPPres")+absdiff("SegalCover")+edges+triangle,
+            estimate="MPLE",control=control.ergm(MPLE.covariance.method='Godambe'))


> summary(fit3)

Call:
ergm(formula = SCN ~ nodematch("GOPPres") + absdiff("SegalCover") +
    edges + triangle, estimate = "MPLE", control = control.ergm(MPLE.covariance.method = "Godambe"))

Maximum Pseudolikelihood Results:

                   Estimate Std. Error MCMC % z value Pr(>|z|)
nodematch.GOPPres   -0.8751     0.1904      0  -4.596  < 1e-04 ***
absdiff.SegalCover  -6.0001     0.9837      0  -6.099  < 1e-04 ***
edges               -0.3395     1.0356      0  -0.328 0.743004
triangle             1.3484     0.3756      0   3.590 0.000331 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

     Null Deviance: 49.91  on 36  degrees of freedom
 Residual Deviance: 30.98  on 32  degrees of freedom

AIC: 38.98  BIC: 45.31  (Smaller is better. MC Std. Err. = 0)
```

Software (/ R) Things...

Packages:

- `network` (the OG)
- `sna` (does not play well with `igraph`)
- `igraph` (does not play well with `sna`)
- `GGally`, `ggraph` (`ggplot2`-compatible network graphics)
- The `statnet` suite: `network`, `ergm`, `tergm`, `btergm`etc.
- `GERGM` (currently old / broken)
- `ggnet2` (`ggplot`-compatible graphics for network data...)
- Many others (`keyplayer`, `RSiena`,...)

Note: `sna` and `igraph` have many of the same command names (e.g., `closeness()`) but implement them differently. Be careful when both packages are loaded.

Extensions $\leftrightarrow$ Extensions

A partial list:

- Different node / edge types (bipartite, tripartite, etc.)
- Hierarchical / multilevel / "nested" networks
- Spatial / spatially-referenced networks
- Dynamic / temporal networks
- Latent / unobserved networks
- Others...

References!

- Menczer, F., S. Fortunato, and C.A. Davis. 2020. *A First Course in Network Science*. New York: Cambridge University Press.
- Robins, G., P. Pattison, Y. Kalish, and D. Lusher. 2007. "An Introduction to Exponential Random Graph Models for Social Networks." *Social Networks* 29:173-191.
- Lusher, Dean, Johan Koskinen, and Garry Robins. 2012. *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*. New York: Cambridge University Press.
- Hunter, David, M. Handcock, C. Butts, S. Goodreau, and M. Morris. 2008. "`ergm`: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks." *Journal of Statistical Software* 24(3):1-29.
- Cranmer, Skyler J., Philip Leifeld, Scott D. McClurg, and Meredith Rolfe. 2017. "Navigating the Range of Statistical Tools for Inferential Network Analysis." *American Journal of Political Science* 61:237-251.
- Wilson, James D., Matthew J. Denny, Shankar Bhamidi, Skyler J. Cranmer, and Bruce A. Desmarais. 2017. "Stochastic Weighted Graphs: Flexible Model Specification and Simulation." *Social Networks* 49:37-47.

Other Things To Look At

- The StatNet Project (ERGMs and their variants / extensions).
- The CRAN Task View for Social Sciences has the best collection of network-related R things.
- Matt Denny's webpage (among others), for tutorials, etc.
- APSA's Organized Section on Political Networks.