Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Intro to Social Network Analysis with **R**

`https://github.com/ljasny/IntroToSNA`

Lorien Jasny[1]

[1]University of Exeter

`L.Jasny@exeter.ac.uk`

Smart Skills Workshop

16 March 2023

Course Materials

- `https://github.com/ljasny/IntroToSNA`
- Right click on green ‘code’ button
- Download files as a zipped folder
- Make sure to unzip the folder!

Course outline

- Introduce social networks
- World's shortest intro to R
- Handling SNA data in R using `Statnet` and `igraph`
- Different descriptive measures for social networks
- Simulation and network generation

Think Formally

A network is not just a metaphor: it is a precise, mathematical construct

Intro

R Intro

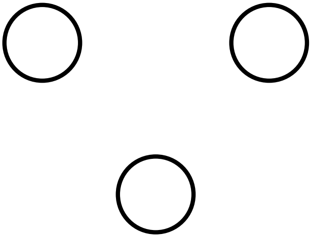
Basic SNA Measures

Graph Level Indices

Simulation

Think Formally

A network is not just a metaphor: it is a precise, mathematical construct of nodes N



Intro

R Intro

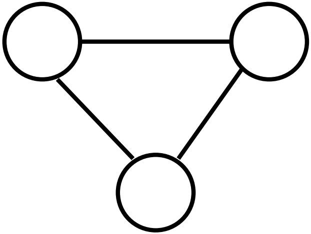
Basic SNA Measures

Graph Level Indices

Simulation

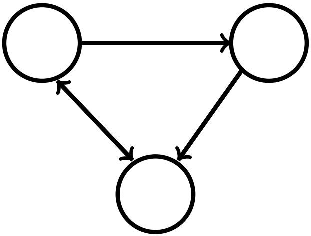
Think Formally

A network is not just a metaphor: it is a precise, mathematical construct of nodes N and edges E



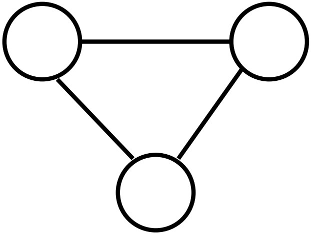
Think Formally

A network is not just a metaphor: it is a precise, mathematical construct of nodes N and edges E that can be directed



Think Formally

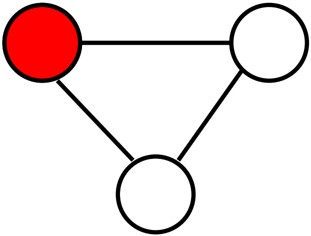
A network is not just a metaphor: it is a precise, mathematical construct of nodes N and edges E that can be directed or undirected.



Intro
R Intro
Basic SNA Measures
Graph Level Indices
Simulation

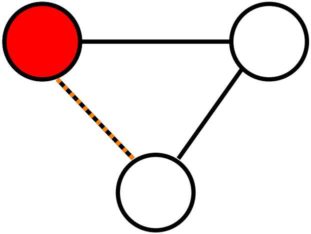
Think Formally

A network is not just a metaphor: it is a precise, mathematical construct of nodes N and edges E that can be directed or undirected. We can include information on the nodes

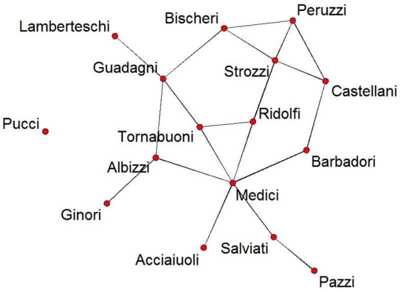


Think Formally

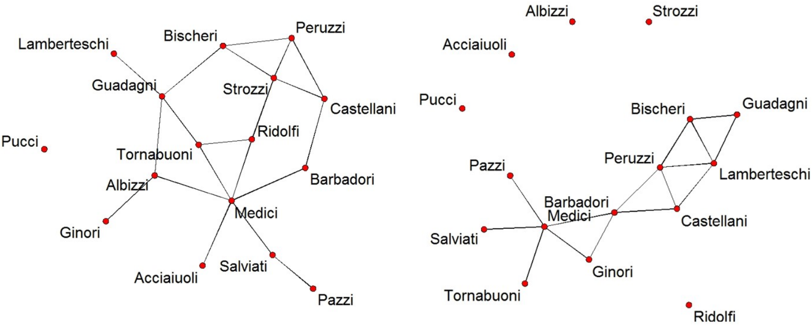
A network is not just a metaphor: it is a precise, mathematical construct of nodes N and edges E that can be directed or undirected. We can include information on the nodes as well as the edges.



Network Intuition



Network Intuition



Why network methods

Why network methods

- Cannot simply use existing statistical methods

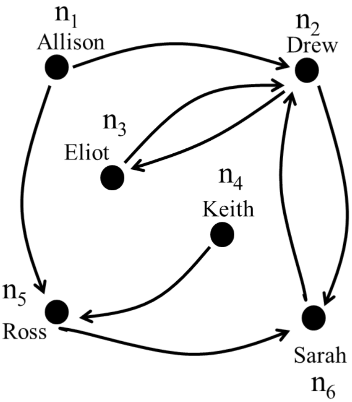
Why network methods

- Cannot simply use existing statistical methods
- The whole point is that observations are interdependent

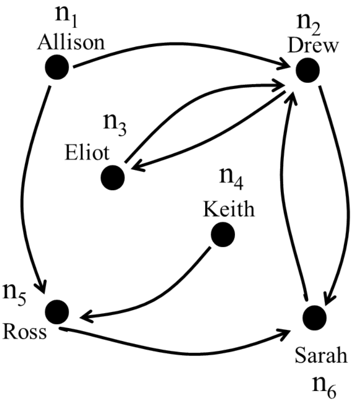
Why network methods

- Cannot simply use existing statistical methods
- The whole point is that observations are interdependent
- Want to explicitly model these interdependencies

Data Structures

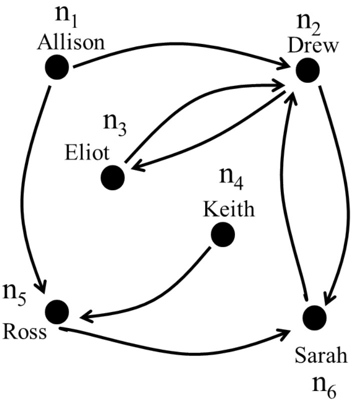


Data Structures



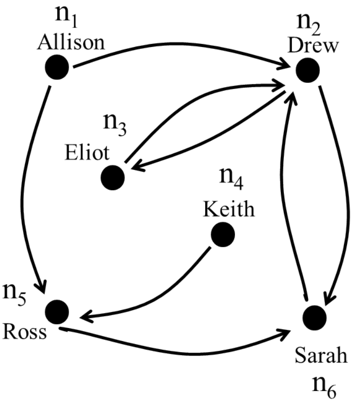
	n_1	n_2	n_3	n_4	n_5	n_6
n_1						
n_2						
n_3						
n_4						
n_5						
n_6						

Data Structures



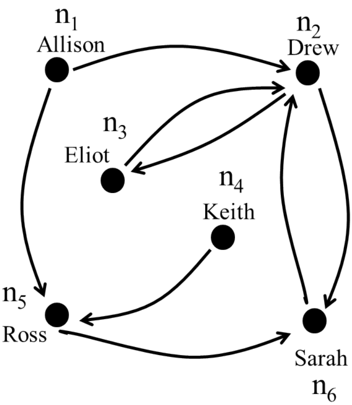
	n_1	n_2	n_3	n_4	n_5	n_6
n_1		1				
n_2						
n_3						
n_4						
n_5						
n_6						

Data Structures



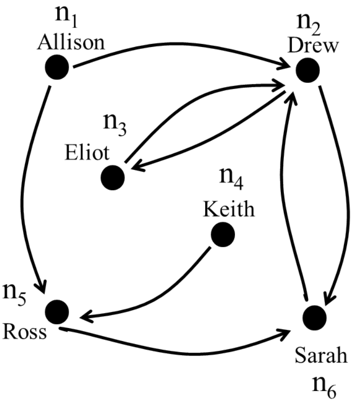
	n_1	n_2	n_3	n_4	n_5	n_6
n_1		1			1	
n_2						
n_3						
n_4						
n_5						
n_6						

Data Structures



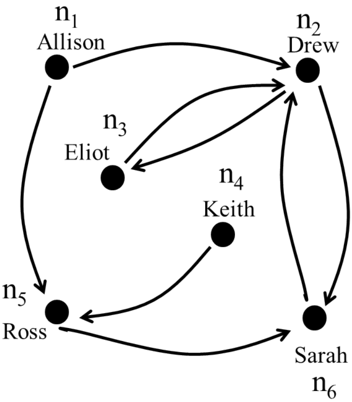
	n_1	n_2	n_3	n_4	n_5	n_6
n_1		1			1	
n_2			1	1		
n_3		1				
n_4					1	
n_5						1
n_6		1				

Data Structures



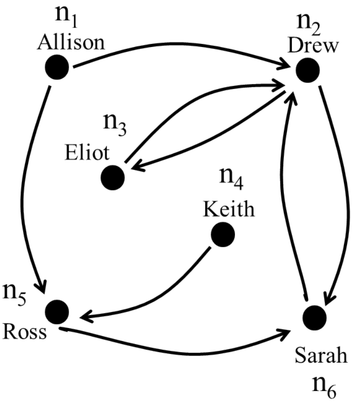
	n_1	n_2	n_3	n_4	n_5	n_6
n_1	-	1			1	
n_2		-	1	1		
n_3		1	-			
n_4				-	1	
n_5					-	1
n_6		1				-

Data Structures

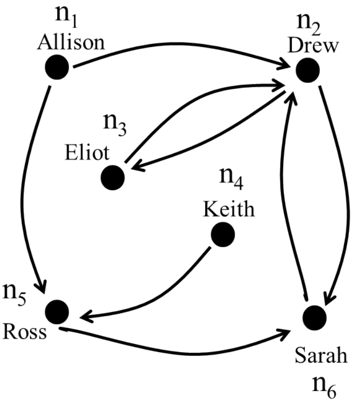


	n_1	n_2	n_3	n_4	n_5	n_6
n_1	-	1	0	0	1	0
n_2	0	-	1	1	0	0
n_3	0	1	-	0	0	0
n_4	0	0	0	-	1	0
n_5	0	0	0	0	-	1
n_6	0	1	0	0	0	-

Data Structures

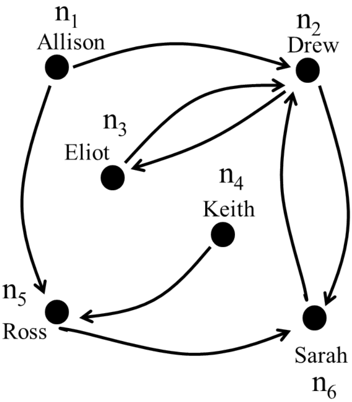


Data Structures



Sender	Receiver	Weight

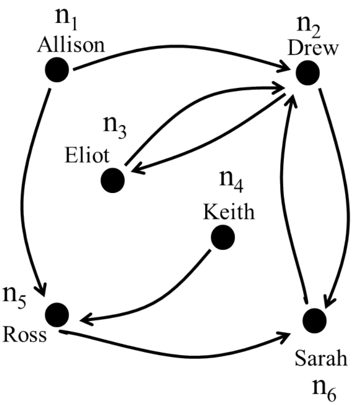
Data Structures



Sender	Receiver	Weight
n_1	n_2	1

Intro
R Intro
Basic SNA Measures
Graph Level Indices
Simulation

Data Structures



Sender	Receiver	Weight
n_1	n_2	1
n_1	n_5	1
n_2	n_3	1
n_2	n_6	1
n_3	n_2	1
n_4	n_5	1
n_5	n_6	1
n_6	n_2	1

The **R** Environment

The **R** Environment

- **R** is both a language and a software platform

The **R** Environment

- **R** is both a language and a software platform
- **R** software is open-source, cross-platform, and free

Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

The **R** Environment

- **R** is both a language and a software platform
- **R** software is open-source, cross-platform, and free
- Its home on the web: http://www.r-project.org

Basics

Basics

- **R** uses a command-like environment, like stata or sas

Basics

- **R** uses a command-like environment, like stata or sas
- **R** is highly extendable

Basics

- **R** uses a command-like environment, like stata or sas
- **R** is highly extendable
  - You can write your own custom functions

Basics

- **R** uses a command-like environment, like stata or sas
- **R** is highly extendable
  - You can write your own custom functions
  - There are thousands of free add-on packages

Basics

- **R** uses a command-like environment, like stata or sas
- **R** is highly extendable
  - You can write your own custom functions
  - There are thousands of free add-on packages
- Generally good at reading in/writing out other file formats

Basics

- **R** uses a command-like environment, like stata or sas
- **R** is highly extendable
  - You can write your own custom functions
  - There are thousands of free add-on packages
- Generally good at reading in/writing out other file formats
- Everything in **R** is an object – data, functions, everything

Fundamentals

Fundamentals

- When you type commands at the prompt '>' and hit *ENTER*

Fundamentals

- When you type commands at the prompt '>' and hit *ENTER*
    - **R** tries to interpret what you've asked it to do (evaluation)

Fundamentals

- When you type commands at the prompt '>' and hit *ENTER*
  - **R** tries to interpret what you've asked it to do (evaluation)
  - If it understands what you've written, it does it (execution)

Fundamentals

- When you type commands at the prompt '>' and hit *ENTER*
  - **R** tries to interpret what you've asked it to do (evaluation)
  - If it understands what you've written, it does it (execution)
  - If it doesn't, it will likely give you an error or a warning

Fundamentals

- When you type commands at the prompt '>' and hit *ENTER*
    - **R** tries to interpret what you've asked it to do (evaluation)
    - If it understands what you've written, it does it (execution)
    - If it doesn't, it will likely give you an error or a warning
- Some commands trigger **R** to print to the screen, others don't

Fundamentals

- When you type commands at the prompt '>' and hit *ENTER*
  - **R** tries to interpret what you've asked it to do (evaluation)
  - If it understands what you've written, it does it (execution)
  - If it doesn't, it will likely give you an error or a warning
- Some commands trigger **R** to print to the screen, others don't
- If you type an incomplete command, **R** will usually respond by changing the command prompt to the '+' character

Fundamentals

- When you type commands at the prompt '>' and hit *ENTER*
  - **R** tries to interpret what you've asked it to do (evaluation)
  - If it understands what you've written, it does it (execution)
  - If it doesn't, it will likely give you an error or a warning
- Some commands trigger **R** to print to the screen, others don't
- If you type an incomplete command, **R** will usually respond by changing the command prompt to the '+' character
  - Hit *ESC* on a Mac to cancel
  - Type in *Ctrl* + *C* on Windows and Linux to cancel

Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Data Structures in **R**

Data Structures in **R**

- **R** has several built-in data types and structures

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)
    - 12
    - 3.14

Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)
    - 12
    - 3.14
  - Strings (alphanumeric characters in quotation marks)

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)
    - 12
    - 3.14
  - Strings (alphanumeric characters in quotation marks)
    - "hello"
    - "3.14"

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)
    - 12
    - 3.14
  - Strings (alphanumeric characters in quotation marks)
    - "hello"
    - "3.14"

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)
    - 12
    - 3.14
  - Strings (alphanumeric characters in quotation marks)
    - "hello"
    - "3.14"
- Common data structures

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)
    - 12
    - 3.14
  - Strings (alphanumeric characters in quotation marks)
    - "hello"
    - "3.14"
- Common data structures
  - Vectors

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)
    - 12
    - 3.14
  - Strings (alphanumeric characters in quotation marks)
    - "hello"
    - "3.14"
- Common data structures
  - Vectors
  - Matrices

Data Structures in **R**

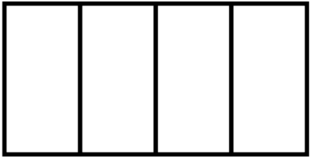
- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)
    - 12
    - 3.14
  - Strings (alphanumeric characters in quotation marks)
    - “hello”
    - “3.14”
- Common data structures
  - Vectors
  - Matrices
  - Data Frames

Data Structures in **R**

- **R** has several built-in data types and structures
- Common data types:
  - Numeric (integers, numbers, etc.)
    - 12
    - 3.14
  - Strings (alphanumeric characters in quotation marks)
    - "hello"
    - "3.14"
- Common data structures
  - Vectors
  - Matrices
  - Data Frames
  - Network objects

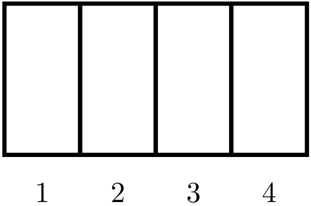
Vectors

A vector is a one-dimensional data structure



Vectors

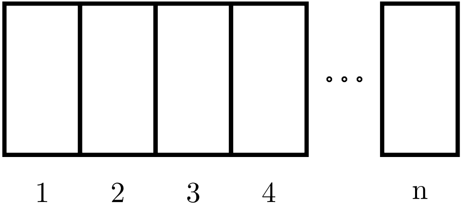
A vector is a one-dimensional data structure



- Vectors are indexed starting at 1

Vectors

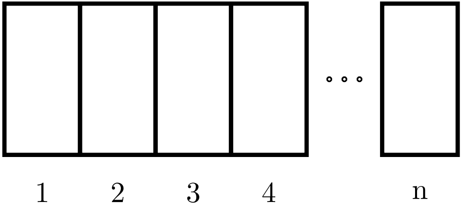
A vector is a one-dimensional data structure



- Vectors are indexed starting at 1
- A vector of length n has n cells

Vectors

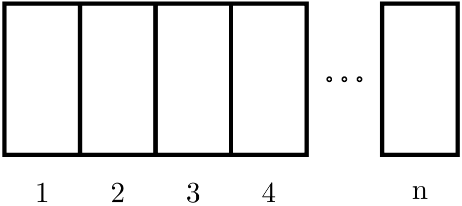
A vector is a one-dimensional data structure



- Vectors are indexed starting at 1
- A vector of length n has n cells
- Each cell can hold a single value

Vectors

A vector is a one-dimensional data structure



- Vectors are indexed starting at 1
- A vector of length n has n cells
- Each cell can hold a single value
- Vectors can only store data of the same type – either all strings or all numerical but not both

Working with Vectors in **R**

Working with Vectors in **R**

- We index vectors in **R** using “square bracket notation”

Working with Vectors in **R**

- We index vectors in **R** using “square bracket notation”
- Example:
  - you have a vector of numeric values called `testScores`

Working with Vectors in **R**

- We index vectors in **R** using "square bracket notation"
- Example:
  - you have a vector of numeric values called `testScores`
  - To retrieve the value in the third cell, type `testScores[3]`

Working with Vectors in **R**

- We index vectors in **R** using "square bracket notation"
- Example:
  - you have a vector of numeric values called `testScores`
  - To retrieve the value in the third cell, type `testScores[3]`
  - To retrieve all BUT the third value, type `testScoress[-3]`

Two-dimensional data in **R**

- Most (all?) of us are familiar with two-dimensional data like that in spreadsheets

Two-dimensional data in **R**

- Most (all?) of us are familiar with two-dimensional data like that in spreadsheets
- **R** has two built-in data structures for storing two-dimensional data

Two-dimensional data in **R**

- Most (all?) of us are familiar with two-dimensional data like that in spreadsheets
- **R** has two built-in data structures for storing two-dimensional data
  - Matrices

Two-dimensional data in **R**

- Most (all?) of us are familiar with two-dimensional data like that in spreadsheets
- **R** has two built-in data structures for storing two-dimensional data
  - Matrices
  - Data Frames

Two-dimensional data in **R**

- Most (all?) of us are familiar with two-dimensional data like that in spreadsheets
- **R** has two built-in data structures for storing two-dimensional data
    - Matrices
    - Data Frames
- In most instances, they behave the same

Two-dimensional data in **R**

- Most (all?) of us are familiar with two-dimensional data like that in spreadsheets
- **R** has two built-in data structures for storing two-dimensional data
  - Matrices
  - Data Frames
- In most instances, they behave the same
- Most functions will accept either a matrix or a data frame

Matrices versus data frames in **R**

Matrices versus data frames in **R**

- Matrices can only store data of one type

Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Matrices versus data frames in **R**

- Matrices can only store data of one type
    - Either all strings or all numbers, but not both

Matrices versus data frames in **R**

- Matrices can only store data of one type
    - Either all strings or all numbers, but not both
    - If you try to give it multiple types, **R** converts everything to string format

Matrices versus data frames in **R**

- Matrices can only store data of one type
  - Either all strings or all numbers, but not both
  - If you try to give it multiple types, **R** converts everything to string format
- Data frames can store data of multiple types

Matrices versus data frames in **R**

- Matrices can only store data of one type
    - Either all strings or all numbers, but not both
    - If you try to give it multiple types, **R** converts everything to string format
- Data frames can store data of multiple types
    - Ideal for classical data analysis where you might have a mix of numerical and string data

Working with matrices

id	name	age	sex	handed	lastDocVisit
5012	Danielle	44	F	R	2012
2331	Josh	44	M	R	2008
1989	Mark	40	M	R	2010
2217	Emma	32	F	L	2012
2912	Sarah	33	F	R	2011

Working with matrices

id	name	age	sex	handed	lastDocVisit
5012	Danielle	44	F	R	2012
2331	Josh	44	M	R	2008
1989	Mark	40	M	R	2010
2217	Emma	32	F	L	2012
2912	Sarah	33	F	R	2011

- Can also use “square bracket notation”

Working with matrices

id	name	age	sex	handed	lastDocVisit
5012	Danielle	44	F	R	2012
2331	Josh	44	M	R	2008
1989	Mark	40	M	R	2010
2217	Emma	32	F	L	2012
2912	Sarah	33	F	R	2011

- Can also use “square bracket notation”
- Inside the square brackets, the first position refers to the row(s) and the second to the column(s)

Working with matrices

id	name	age	sex	handed	lastDocVisit
5012	Danielle	44	F	R	2012
2331	Josh	44	M	R	2008
1989	Mark	40	M	R	2010
2217	Emma	32	F	L	2012
2912	Sarah	33	F	R	2011

- Can also use "square bracket notation"
- Inside the square brackets, the first position refers to the row(s) and the second to the column(s)
- If this matrix is called `friendSurvey`, the command to retrieve Josh's age is `friendSurvey[2,3]`

Working with data frames

Working with data frames

- Square bracket notation works for data frames as well

Working with data frames

- Square bracket notation works for data frames as well
- Data frames provide another option: dollar sign notation

Working with data frames

id	name	age	sex	handed	lastDocVisit
5012	Danielle	44	F	R	2012
2331	Josh	44	M	R	2008
1989	Mark	40	M	R	2010
2217	Emma	32	F	L	2012
2912	Sarah	33	F	R	2011

Working with data frames

id	name	age	sex	handed	lastDocVisit
5012	Danielle	44	F	R	2012
2331	Josh	44	M	R	2008
1989	Mark	40	M	R	2010
2217	Emma	32	F	L	2012
2912	Sarah	33	F	R	2011

- To retrieve the 'sex' column as a vector, use `friendSurvey$sex`

Working with data frames

id	name	age	sex	handed	lastDocVisit
5012	Danielle	44	F	R	2012
2331	Josh	44	M	R	2008
1989	Mark	40	M	R	2010
2217	Emma	32	F	L	2012
2912	Sarah	33	F	R	2011

- To retrieve the ‘sex’ column as a vector, use `friendSurvey$sex`
- To retrieve Josh’s age, use `friendSurvey$age[2]`

Basic SNA Measures

Networks in R

- Many different packages in R to visualize and analyse networks

Networks in R

- Many different packages in R to visualize and analyse networks
- We'll focus on the two most used: `Statnet` and `igraph`

Networks in R

- Many different packages in R to visualize and analyse networks
- We'll focus on the two most used: `Statnet` and `igraph`
- Both have pluses and minuses reflecting the interests of the authors

Networks in R

- Many different packages in R to visualize and analyse networks
- We'll focus on the two most used: `Statnet` and `igraph`
- Both have pluses and minuses reflecting the interests of the authors
- `Statnet` is written towards doing exponential random graph models

Networks in R

- Many different packages in R to visualize and analyse networks
- We'll focus on the two most used: `Statnet` and `igraph`
- Both have pluses and minuses reflecting the interests of the authors
- `Statnet` is written towards doing exponential random graph models
- `igraph` has better visuals and has routines for community detection

Networks in R

- Many different packages in R to visualize and analyse networks
- We'll focus on the two most used: `Statnet` and `igraph`
- Both have pluses and minuses reflecting the interests of the authors
- `Statnet` is written towards doing exponential random graph models
- `igraph` has better visuals and has routines for community detection
- convert between the two using the library `intergraph`

Networks in R

- Many different packages in R to visualize and analyse networks
- We'll focus on the two most used: `Statnet` and `igraph`
- Both have pluses and minuses reflecting the interests of the authors
- `Statnet` is written towards doing exponential random graph models
- `igraph` has better visuals and has routines for community detection
- convert between the two using the library `intergraph`
- But otherwise, they often don't play well loaded simultaneously

Networks in `R`

- Many different packages in `R` to visualize and analyse networks
- We'll focus on the two most used: `Statnet` and `igraph`
- Both have pluses and minuses reflecting the interests of the authors
- `Statnet` is written towards doing exponential random graph models
- `igraph` has better visuals and has routines for community detection
- convert between the two using the library `intergraph`
- But otherwise, they often don't play well loaded simultaneously
- Different structures and conventions

Network Objects

Network Objects

- stores an adjacency matrix or an edgelist as well as metadata

Network Objects

- stores an adjacency matrix or an edgelist as well as metadata
  - vertex, edge, and network attributes

Network Objects

- stores an adjacency matrix or an edgelist as well as metadata
  - vertex, edge, and network attributes
- can use square-bracket notation just like a matrix

Network Objects

- stores an adjacency matrix or an edgelist as well as metadata
  - vertex, edge, and network attributes
- can use square-bracket notation just like a matrix
  - if we have a network *dataNet*, then `dataNet[,]` returns the adjacency matrix

Network Objects

- stores an adjacency matrix or an edgelist as well as metadata
    - vertex, edge, and network attributes
- can use square-bracket notation just like a matrix
    - if we have a network *dataNet*, then `dataNet[,]` returns the adjacency matrix
- Different notation for working with attribute data

Network Objects

- stores an adjacency matrix or an edgelist as well as metadata
  - vertex, edge, and network attributes
- can use square-bracket notation just like a matrix
  - if we have a network *dataNet*, then `dataNet[,]` returns the adjacency matrix
- Different notation for working with attribute data
  - For vertex attributes, `get.vertex.attributes`, `set.vertex.attributes`, `list.vertex.attributes`, etc or `%v%` for shorthand

Network Objects

- stores an adjacency matrix or an edgelist as well as metadata
  - vertex, edge, and network attributes
- can use square-bracket notation just like a matrix
  - if we have a network *dataNet*, then `dataNet[,]` returns the adjacency matrix
- Different notation for working with attribute data
  - For vertex attributes, `get.vertex.attributes`, `set.vertex.attributes`, `list.vertex.attributes`, etc or `%v%` for shorthand
  - Similarly for edge attributes (`%e%`)

Network Objects

- stores an adjacency matrix or an edgelist as well as metadata
    - vertex, edge, and network attributes
- can use square-bracket notation just like a matrix
    - if we have a network *dataNet*, then `dataNet[,]` returns the adjacency matrix
- Different notation for working with attribute data
    - For vertex attributes, `get.vertex.attributes`, `set.vertex.attributes`, `list.vertex.attributes`, etc or `%v%` for shorthand
    - Similarly for edge attributes (`%e%`)
    - And network level attributes (`%n%`)

Network Objects

```
Network attributes:
  vertices = 18
  directed = TRUE
  hyper = FALSE
  loops = FALSE
  multiple = FALSE
  bipartite = FALSE
  total edges= 54
    missing edges= 0
    non-missing edges= 54

 Vertex attribute names:
    Group vertex.names

 Edge attribute names:
    Order
> |
```

Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Code Time

- Sections 1 and 2

Igraph Objects

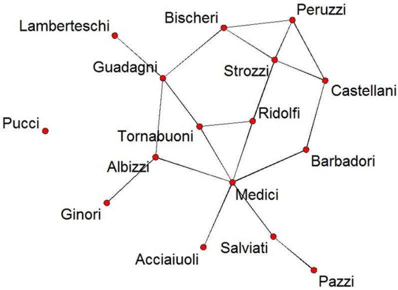
```
net
```

```
## IGRAPH DNW- 17 49 --
## + attr: name (v/c), media (v/c), media.type (v/n), type.label
## | (v/c), audience.size (v/n), type (e/c), weight (e/n)
## + edges (vertex names):
##  [1] s01->s02 s01->s03 s01->s04 s01->s15 s02->s01 s02->s03 s02->s09
##  [8] s02->s10 s03->s01 s03->s04 s03->s05 s03->s08 s03->s10 s03->s11
## [15] s03->s12 s04->s03 s04->s06 s04->s11 s04->s12 s04->s17 s05->s01
## [22] s05->s02 s05->s09 s05->s15 s06->s06 s06->s16 s06->s17 s07->s03
## [29] s07->s08 s07->s10 s07->s14 s08->s03 s08->s07 s08->s09 s09->s10
## [36] s10->s03 s12->s06 s12->s13 s12->s14 s13->s12 s13->s17 s14->s11
## [43] s14->s13 s15->s01 s15->s04 s15->s06 s16->s06 s16->s17 s17->s04
```

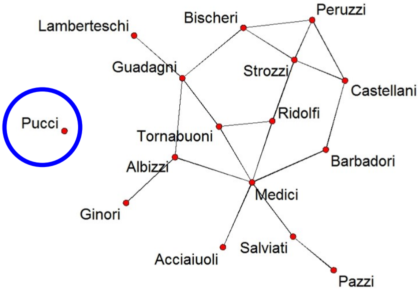
We also have easy access to nodes, edges, and their attributes with:

```
E(net)       # The edges of the "net" object
V(net)       # The vertices of the "net" object
E(net)$type  # Edge attribute "type"
V(net)$media # Vertex attribute "media"
```

Descriptives

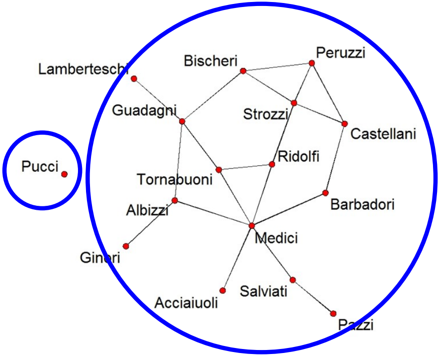


Descriptives



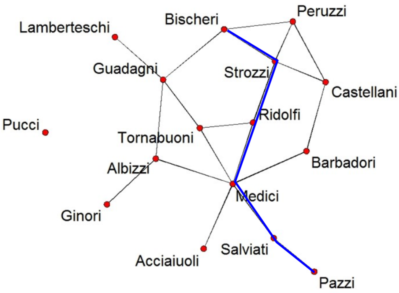
- One isolate

Descriptives



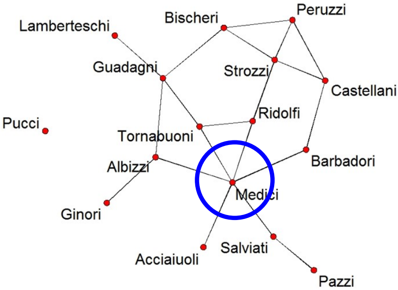
- One isolate
- Two components

Descriptives



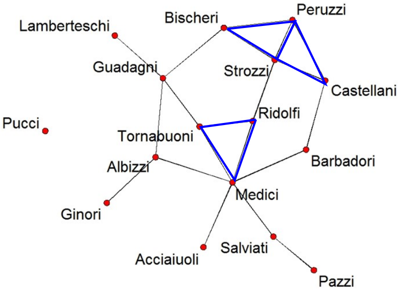
- One isolate
- Two components
- Diameter is 5

Descriptives



- One isolate
- Two components
- Diameter is 5
- Medici is most popular

Descriptives



- One isolate
- Two components
- Diameter is 5
- Medici is most popular
- Three triads

Degree

For each node, its degree is

Degree

For each node, its degree is

- the number of nodes adjacent to it

Degree

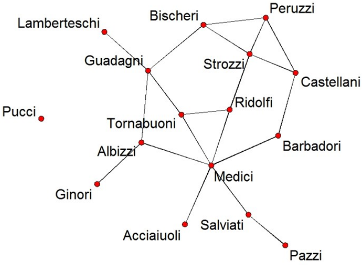
For each node, its degree is

- the number of nodes adjacent to it
- or, the number of lines incident with it

Degree

For each node, its degree is

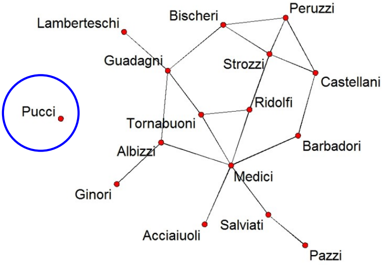
- the number of nodes adjacent to it
- or, the number of lines incident with it



Degree

For each node, its degree is

- the number of nodes adjacent to it
- or, the number of lines incident with it

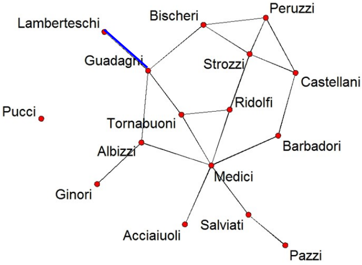


- Pucci has degree 0

Degree

For each node, its degree is

- the number of nodes adjacent to it
- or, the number of lines incident with it

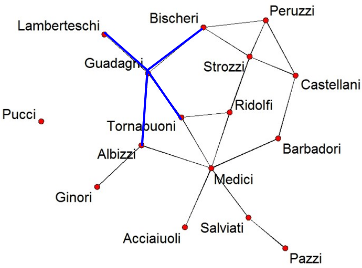


- Pucci has degree 0
- Lamberteschi has degree 1

Degree

For each node, its degree is

- the number of nodes adjacent to it
- or, the number of lines incident with it

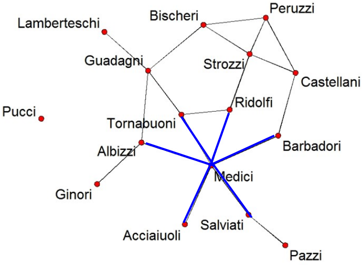


- Pucci has degree 0
- Lamberteschi has degree 1
- Guadagni has degree 4

Degree

For each node, its degree is

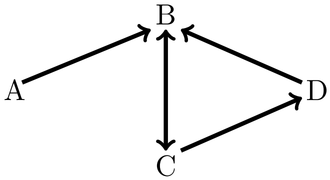
- the number of nodes adjacent to it
- or, the number of lines incident with it



- Pucci has degree 0
- Lamberteschi has degree 1
- Guadagni has degree 4
- Medici has degree 6

Directed Degree

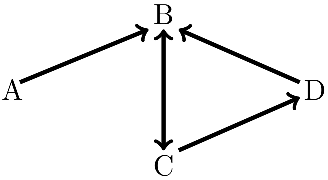
In directed graphs,



Directed Degree

In directed graphs,

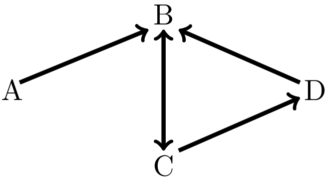
- *Indegree* indicates the number of received ties



Directed Degree

In directed graphs,

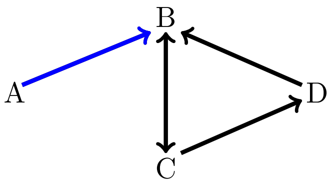
- *Indegree* indicates the number of received ties
- *Outdegree* indicates the number of sent ties



Directed Degree

In directed graphs,

- *Indegree* indicates the number of received ties
- *Outdegree* indicates the number of sent ties

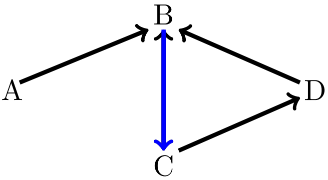


- A has 1 outdegree

Directed Degree

In directed graphs,

- *Indegree* indicates the number of received ties
- *Outdegree* indicates the number of sent ties

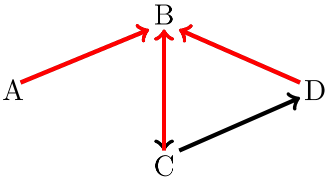


- A has 1 outdegree
- B has 1 outdegree

Directed Degree

In directed graphs,

- *Indegree* indicates the number of received ties
- *Outdegree* indicates the number of sent ties

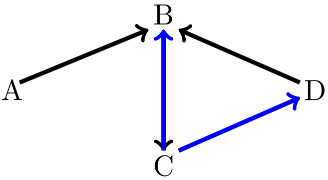


- A has 1 outdegree
- B has 1 outdegree and 3 indegree

Directed Degree

In directed graphs,

- *Indegree* indicates the number of received ties
- *Outdegree* indicates the number of sent ties

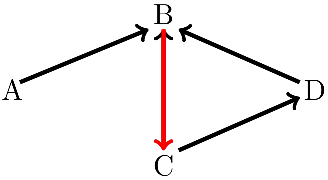


- A has 1 outdegree
- B has 1 outdegree and 3 indegree
- C has 2 outdegree

Directed Degree

In directed graphs,

- *Indegree* indicates the number of received ties
- *Outdegree* indicates the number of sent ties

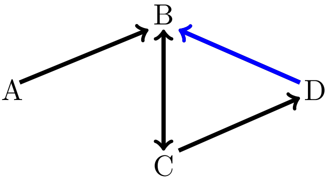


- A has 1 outdegree
- B has 1 outdegree and 3 indegree
- C has 2 outdegree and 1 indegree

Directed Degree

In directed graphs,

- *Indegree* indicates the number of received ties
- *Outdegree* indicates the number of sent ties

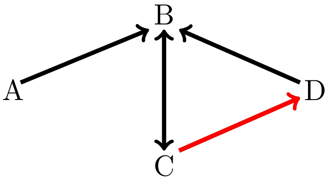


- A has 1 outdegree
- B has 1 outdegree and 3 indegree
- C has 2 outdegree and 1 indegree
- D has 1 outdegree

Directed Degree

In directed graphs,

- *Indegree* indicates the number of received ties
- *Outdegree* indicates the number of sent ties



- A has 1 outdegree
- B has 1 outdegree and 3 indegree
- C has 2 outdegree and 1 indegree
- D has 1 outdegree and 1 indegree

Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

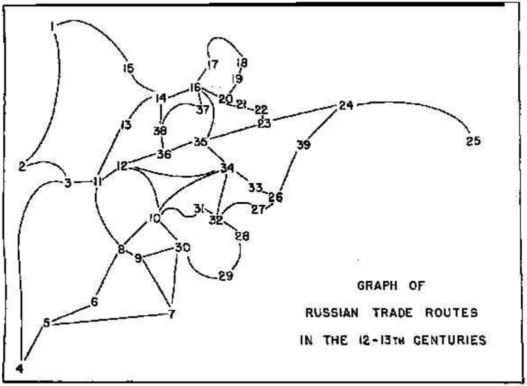
Betweenness Centrality

Proportion of shortest paths that the given node lies on

Betweenness Centrality

Proportion of shortest paths that the given node lies on

Figure 2. *Graph of Russian trade routes in the 12th - 13th centuries.*

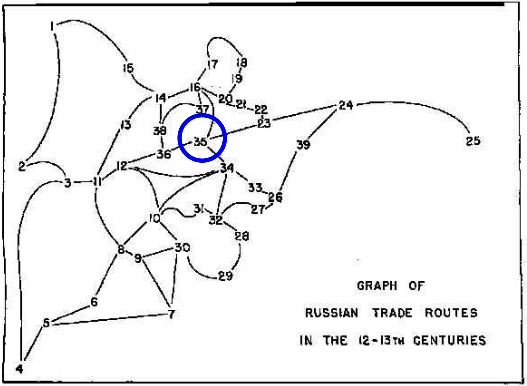


Forrest Pitts 1978 "The River Trade Ntework of Russia, Revisited"

Betweenness Centrality

Proportion of shortest paths that the given node lies on

Figure 2. *Graph of Russian trade routes in the 12th - 13th centuries.*



Forrest Pitts 1978 "The River Trade Ntework of Russia, Revisited"

Additional Centrality Measures

Additional Centrality Measures

- Closeness: the sum of the distances to all other nodes in the network

Additional Centrality Measures

- Closeness: the sum of the distances to all other nodes in the network
- Eigenvector: an iterated measure where nodes have higher eigenvector centrality if they are tied to nodes with high eigenvector centrality

Additional Centrality Measures

- Closeness: the sum of the distances to all other nodes in the network
- Eigenvector: an iterated measure where nodes have higher eigenvector centrality if they are tied to nodes with high eigenvector centrality
- Many, many others

Graph Level Indices

Density

Density

- Number of ties, expressed as a percentage of the number of possible ties

Density

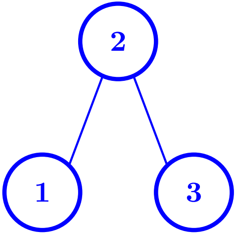
- Number of ties, expressed as a percentage of the number of possible ties
- For directed graphs: $\frac{E}{N(N-1)}$

Density

- Number of ties, expressed as a percentage of the number of possible ties
- For directed graphs: $\frac{E}{N(N-1)}$
- For undirected graphs: $\frac{\frac{E}{N(N-1)}}{2}$

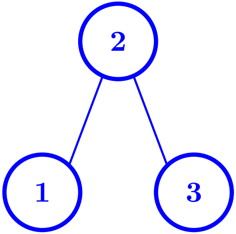
Density

- Number of ties, expressed as a percentage of the number of possible ties
- For directed graphs: $\frac{E}{N(N-1)}$
- For undirected graphs: $\frac{\frac{E}{N(N-1)}}{2}$



Density

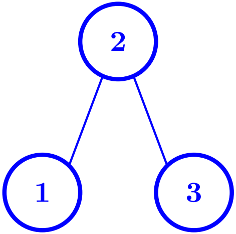
- Number of ties, expressed as a percentage of the number of possible ties
- For directed graphs: $\frac{E}{N(N-1)}$
- For undirected graphs: $\frac{\frac{E}{N(N-1)}}{2}$



$$= \frac{\frac{2}{3(3-1)}}{2} = \frac{4}{6}$$

Mean Degree

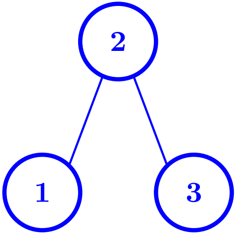
The mean degree, $\bar{d}$, of all nodes in the graph is



Mean Degree

The mean degree, $\bar{d}$, of all nodes in the graph is

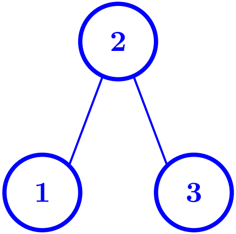
$$\bar{d}(n) = \frac{\sum_{i=1}^{N} d(n_i)}{N} = \frac{2E}{N}$$



Mean Degree

The mean degree, $\bar{d}$, of all nodes in the graph is

$$\bar{d}(n) = \frac{\sum_{i=1}^{N} d(n_i)}{N} = \frac{2E}{N}$$
$$= \frac{1+2+1}{3} = \frac{4}{3}$$

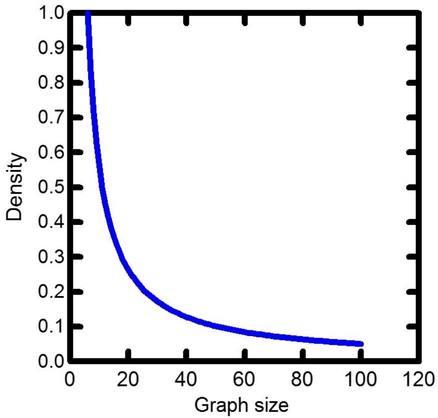


Size, Density, and Mean Degree

If we hold the mean degree constant, but vary size, what happens to density?

Size, Density, and Mean Degree

If we hold the mean degree constant, but vary size, what happens to density?



Centralization

Centralization

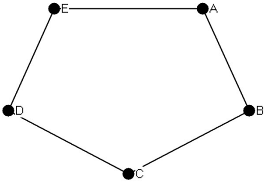
- Extent to which centrality is concentrated on a single vertex

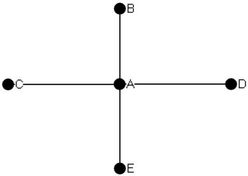
Centralization

- Extent to which centrality is concentrated on a single vertex
- Calculated as the sum of the differences between each node's centrality score and the maximum score

Centralization

- Extent to which centrality is concentrated on a single vertex
- Calculated as the sum of the differences between each node's centrality score and the maximum score
- Most centralized structure is usually a star network





Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Dyad Census

Dyad Census

Mutual (M)

Intro

R Intro

Basic SNA Measures

Graph Level Indices

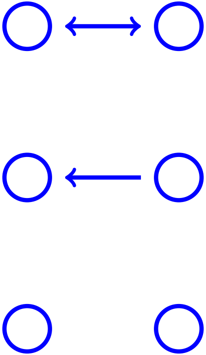
Simulation

Dyad Census

Mutual (M)

Assymetric (A)

Dyad Census



Mutual (M)

Assymetric (A)

Null (N)

Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Reciprocity

Reciprocity

- Dyadic: the proportion of dyads that are symmetric
  $\frac{M+N}{M+A+N}$

Reciprocity

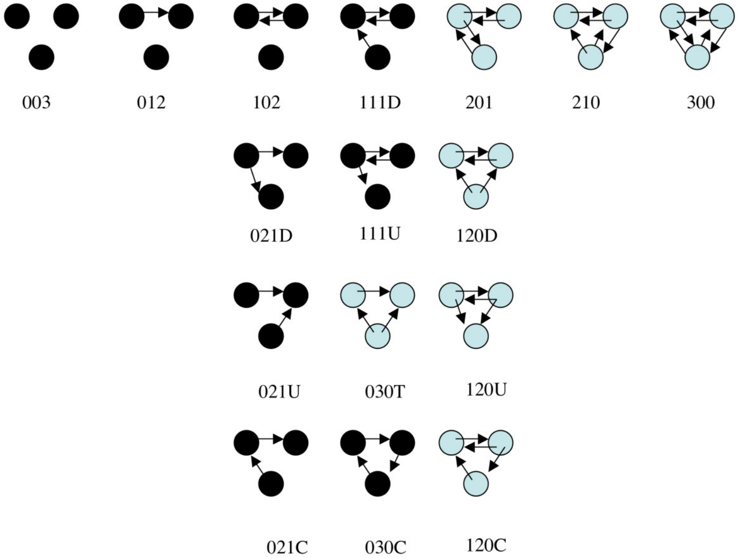
- Dyadic: the proportion of dyads that are symmetric $\frac{M+N}{M+A+N}$
- Dyadic non-null: the proportion of non-null dyads that are reciprocal $\frac{M}{M+A}$

Reciprocity

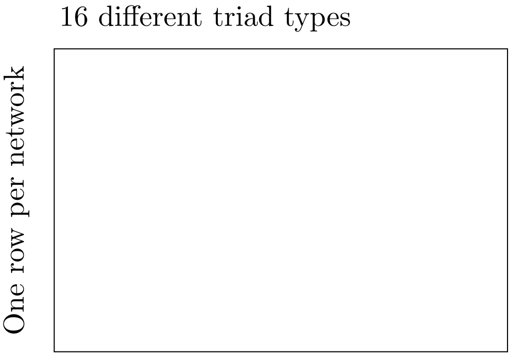
- Dyadic: the proportion of dyads that are symmetric $\frac{M+N}{M+A+N}$
- Dyadic non-null: the proportion of non-null dyads that are reciprocal $\frac{M}{M+A}$
- Edgewise: $\frac{2*M}{2*M+A}$

Triad Census

Triad Census



Triad Census



Triad Census

16 different triad types

One row per network

i,j cell is the number
of triad type j
in network i

Triad Census

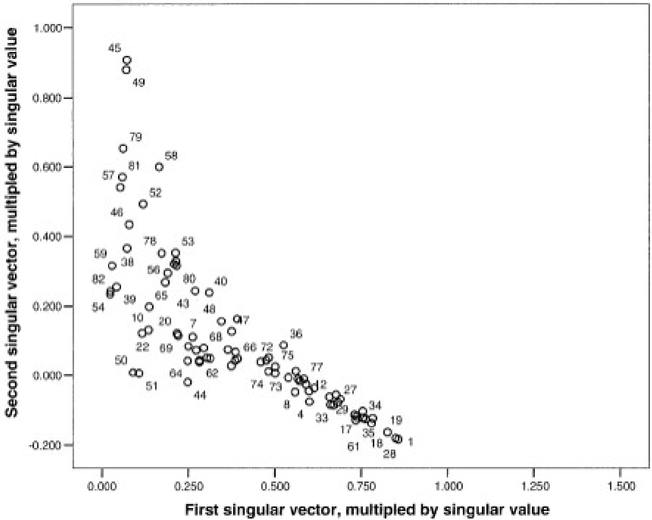


FIGURE 2. Singular value decomposition of triad census array, first two left singular vectors, multiplied by singular values, $N = 82$ networks.

Triad Census

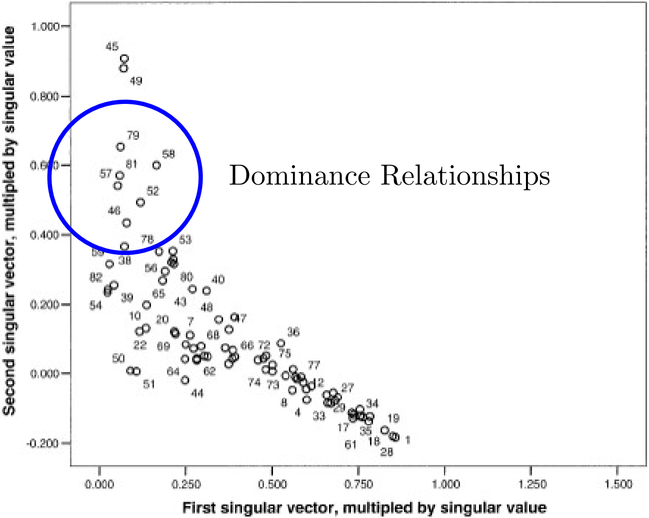


FIGURE 2. Singular value decomposition of triad census array, first two left singular vectors, multiplied by singular values, $N = 82$ networks.

Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Triad Census

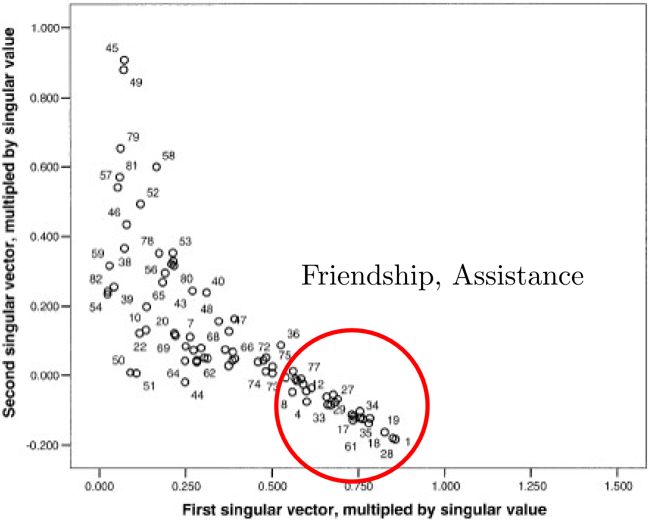
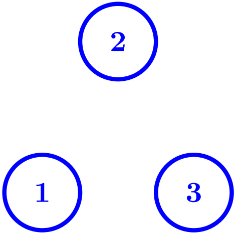


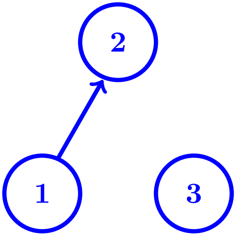
FIGURE 2. Singular value decomposition of triad census array, first two left singular vectors, multiplied by singular values, $N = 82$ networks.

Transitivity

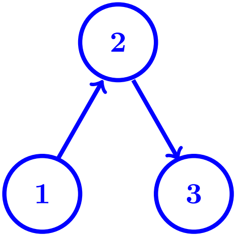
Transitivity



Transitivity



Transitivity



Intro

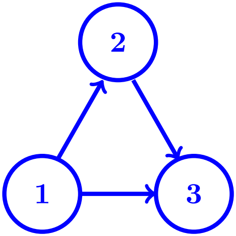
R Intro

Basic SNA Measures

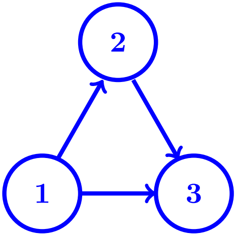
Graph Level Indices

Simulation

Transitivity

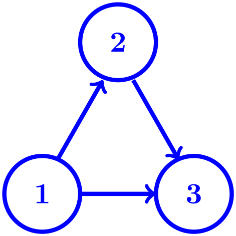


Transitivity



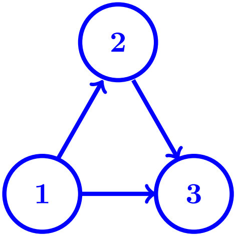
- Usually calculated as the fraction of completed two-paths

Transitivity



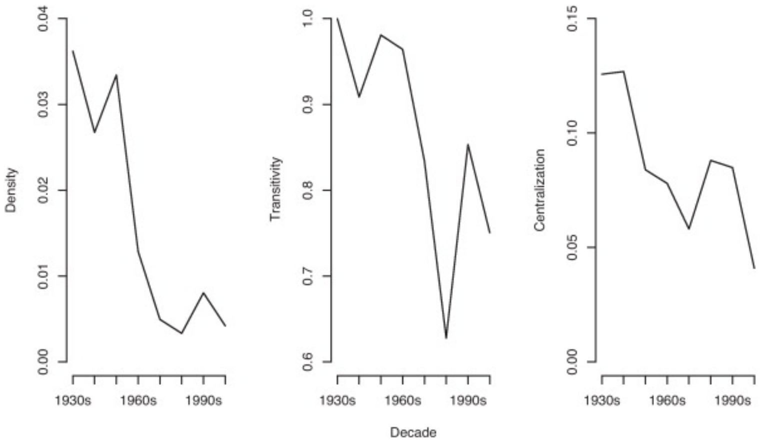
- Usually calculated as the fraction of completed two-paths
- Related to Grannovetter's 'forbidden triad'

Transitivity



- Usually calculated as the fraction of completed two-paths
- Related to Grannovetter's 'forbidden triad'
- Can be directed or undirected

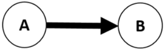
Example



Janet Box-Steffensmeier and Dino Christenson
"The evolution and formation of amicus curiae networks"
Social Networks 2012

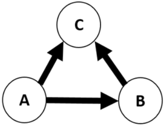
Example

a)



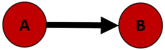
A sends information to B.

c)



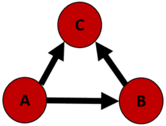
Transitive triad such that A sends information to B and C, and B also sends information to C. The smallest example of a ‘chamber.’

b)



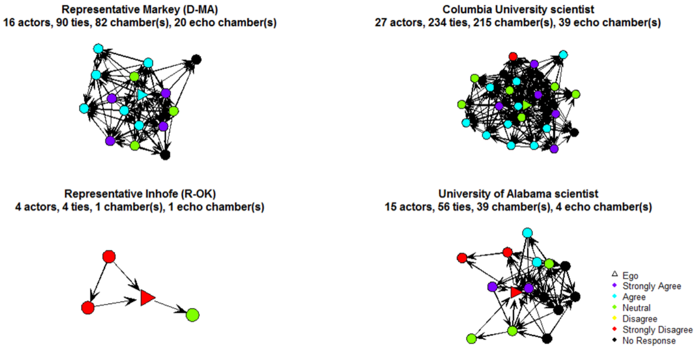
A and B agree so A’s information echoes B’s understanding.

d)



A transitive triad where each actor already holds the same position – an echo chamber.

Example


Representative Markey (D-MA)
16 actors, 90 ties, 82 chamber(s), 20 echo chamber(s)
Columbia University scientist
27 actors, 234 ties, 215 chamber(s), 39 echo chamber(s)
Representative Inhofe (R-OK)
4 actors, 4 ties, 1 chamber(s), 1 echo chamber(s)
University of Alabama scientist
15 actors, 56 ties, 39 chamber(s), 4 echo chamber(s)
Ego
Strongly Agree
Agree
Neutral
Disagree
Strongly Disagree
No Response

Example


3:42 pm ET
U.S. SENATE
SEN. JAMES INHOFE
R-Oklahoma
C-SPAN2
c-span.org

Code Time

- Section 3-4.7

Forbidden Triad or Structural Hole?

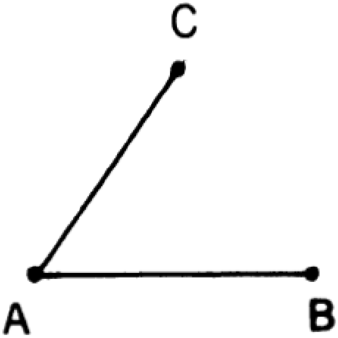


FIG. 1.—Forbidden triad

Forbidden Triad or Structural Hole?

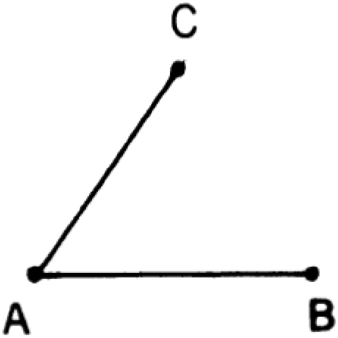
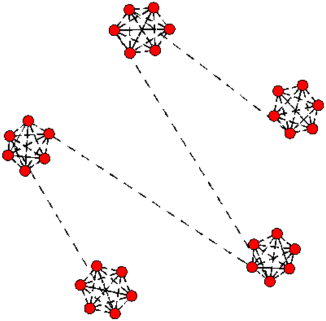


FIG. 1.—Forbidden triad

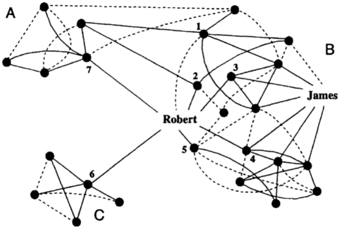
- Granovetter, Mark S. 1973. "The Strength of Weak Ties"

Forbidden Triad or Structural Hole?



- Granovetter, Mark S. 1973. "The Strength of Weak Ties"

Forbidden Triad or Structural Hole?



- Burt, Ronald S. 2004. "Structural Holes: The Social Structure of Competition"

Extensions

Attributes!

Extensions

Attributes!

- Properties of nodes, edges, or even networks

Extensions

Attributes!

- Properties of nodes, edges, or even networks
- Pretty much anything you can measure could be an attribute

Extensions

Attributes!

- Properties of nodes, edges, or even networks
- Pretty much anything you can measure could be an attribute
- Extension based on node attributes: Brokerage

Extensions

Attributes!

- Properties of nodes, edges, or even networks
- Pretty much anything you can measure could be an attribute
- Extension based on node attributes: Brokerage
- Extension based on edge attributes: Structural Balance

Brokerage

- Brokerage is a process "by which intermediary actors facilitate transactions between other actors lacking access to or trust in one another" (Marsden 1982)

Brokerage

- Brokerage is a process “by which intermediary actors facilitate transactions between other actors lacking access to or trust in one another” (Marsden 1982)
- Brokers play a crucial role in knitting together diverse groups of people, organizations, parties

Brokerage

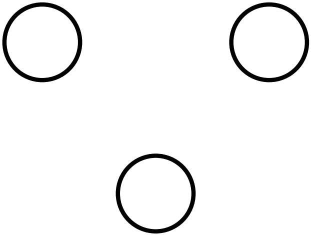
- Brokerage is a process “by which intermediary actors facilitate transactions between other actors lacking access to or trust in one another” (Marsden 1982)
- Brokers play a crucial role in knitting together diverse groups of people, organizations, parties
- Brokers can gain a lot – early access to information, prestige

Brokerage

- Brokerage is a process “by which intermediary actors facilitate transactions between other actors lacking access to or trust in one another” (Marsden 1982)
- Brokers play a crucial role in knitting together diverse groups of people, organizations, parties
- Brokers can gain a lot – early access to information, prestige
- But can also be distrusted by everyone

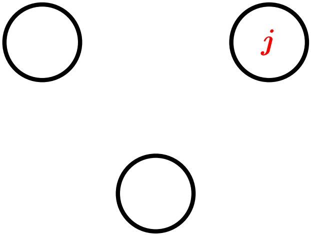
Brokerage: Formal Concept

In a network N with edges E,



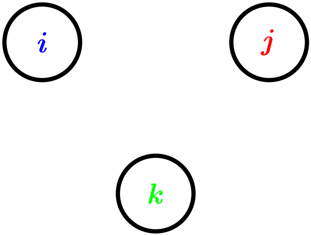
Brokerage: Formal Concept

In a network N with edges E,
node j brokers



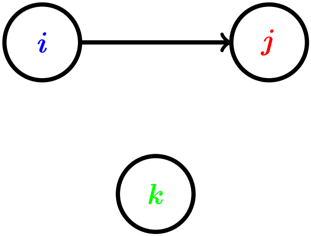
Brokerage: Formal Concept

In a network N with edges E, node j brokers nodes i and k



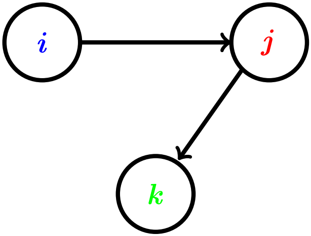
Brokerage: Formal Concept

In a network N with
edges E,
node j brokers
nodes i and k
if $e_{ij} \in E$



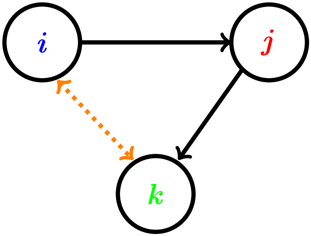
Brokerage: Formal Concept

In a network N with edges E,
node j brokers nodes i and k
if $e_{ij} \in E$
and $e_{jk} \in E$



Brokerage: Formal Concept

In a network N with edges E,
node j brokers nodes i and k
if $e_{ij} \in E$
and $e_{jk} \in E$
but $e_{ik} \notin E$



Brokerage: Formal Concept

Gould and Fernandez (1989, 1994)

Brokerage: Formal Concept

Gould and Fernandez (1989, 1994)

- formalized the concept

Brokerage: Formal Concept

Gould and Fernandez (1989, 1994)

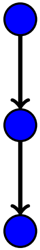
- formalized the concept
- added a vertex attribute component

Brokerage: Formal Concept

Gould and Fernandez (1989, 1994)

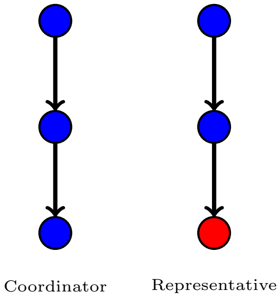
- formalized the concept
- added a vertex attribute component
- compared empirical brokerage counts to counts from random graphs conditioned on the number of edges

Brokerage

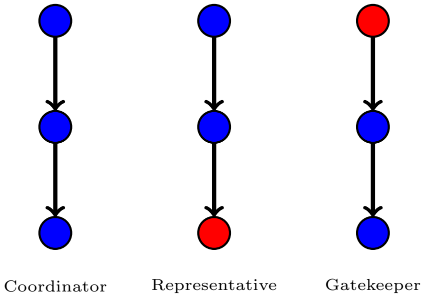


Coordinator

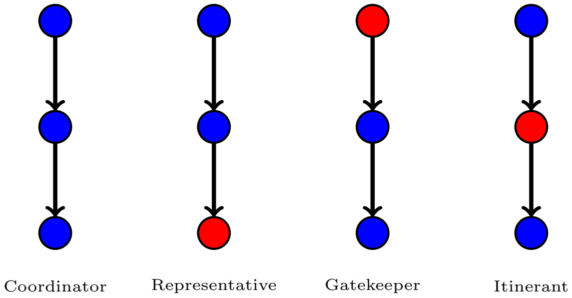
Brokerage



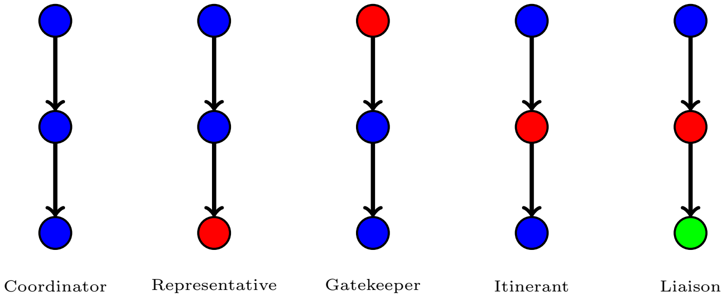
Brokerage



Brokerage



Brokerage


Coordinator
Representative
Gatekeeper
Itinerant
Liaison

Gould and Fernandez' Findings

Gould and Fernandez' Findings

- the benefits of brokerage are mediated both by the type of organization (the node sets) and the type of brokerage chain

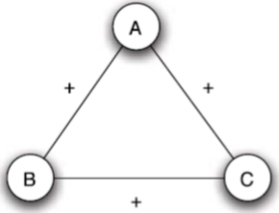
Gould and Fernandez' Findings

- the benefits of brokerage are mediated both by the type of organization (the node sets) and the type of brokerage chain
- non-governmental organizations were found to have more influence when they held any type of brokerage position

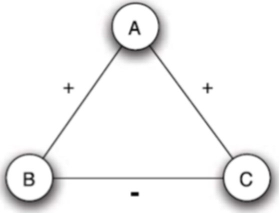
Gould and Fernandez' Findings

- the benefits of brokerage are mediated both by the type of organization (the node sets) and the type of brokerage chain
- non-governmental organizations were found to have more influence when they held any type of brokerage position
- governmental organizations gained influence only when they held "outsider" brokerage roles in itinerant and liaison chains

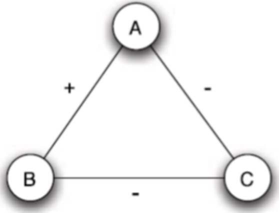
Structural Balance



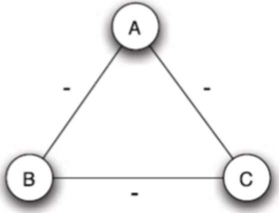
(a) *A, B, and C are mutual friends: balanced.*



(b) *A is friends with B and C, but they don't get along with each other: not balanced.*

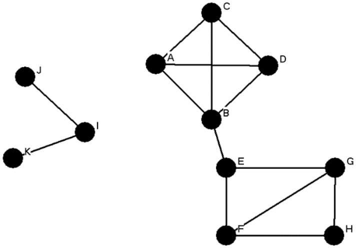


(c) *A and B are friends with C as a mutual enemy: balanced.*

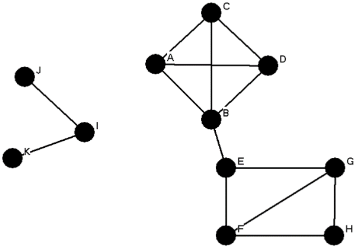


(d) *A, B, and C are mutual enemies: not balanced.*

Subgroups

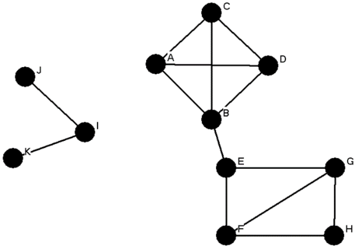


Subgroups



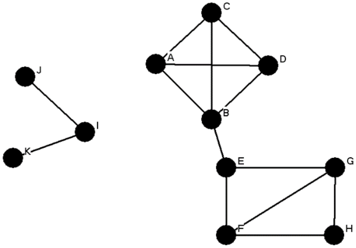
- Component: A maximal connected subgraph

Subgroups

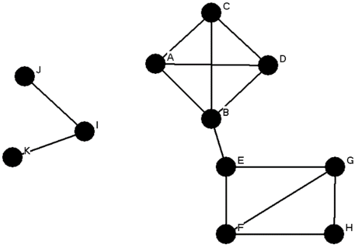


- Component: A maximal connected subgraph
- A subgraph is *maximal* with respect to some property if it has the property, but loses it with the addition of more nodes or edges

Cliques

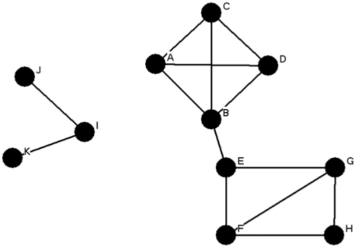


Cliques



- A maximally complete subgraph of 3 or more

Cliques



- A maximally complete subgraph of 3 or more
- All nodes are adjacent to all others in the subgraph

Intro

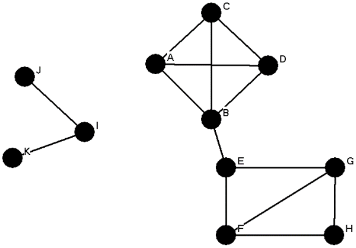
R Intro

Basic SNA Measures

Graph Level Indices

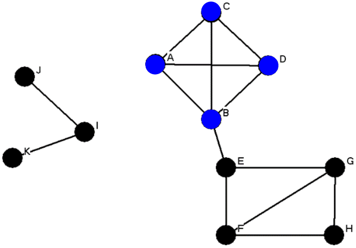
Simulation

Cliques



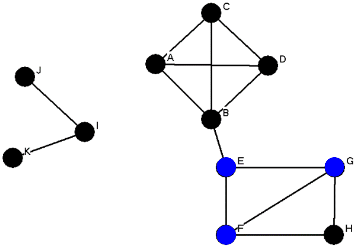
- A maximally complete subgraph of 3 or more
- All nodes are adjacent to all others in the subgraph
- No nodes can be added that have that property

Cliques



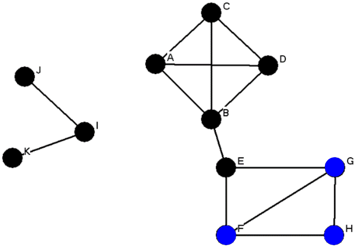
- A maximally complete subgraph of 3 or more
- All nodes are adjacent to all others in the subgraph
- No nodes can be added that have that property

Cliques



- A maximally complete subgraph of 3 or more
- All nodes are adjacent to all others in the subgraph
- No nodes can be added that have that property

Cliques



- A maximally complete subgraph of 3 or more
- All nodes are adjacent to all others in the subgraph
- No nodes can be added that have that property

Limitations of Cliques

- A very strict definition
- Often networks contain many small overlapping cliques
- Cliques have no internal structure

Relaxations of Cliques

Subgroups based on reachability or diameter

Relaxations of Cliques

Subgroups based on reachability or diameter

- n-clique: a maximal subgraph in which the **largest geodesic distance** between any two nodes is no greater than n

Relaxations of Cliques

Subgroups based on reachability or diameter

- n-clique: a maximal subgraph in which the **largest geodesic distance** between any two nodes is no greater than n
- n-clan: an n-clique in which the geodesic distance between all nodes in the subgraph is no greater than n **for paths within the subgraph**

Relaxations of Cliques

Subgroups based on reachability or diameter

- n-clique: a maximal subgraph in which the **largest geodesic distance** between any two nodes is no greater than n
- n-clan: an n-clique in which the geodesic distance between all nodes in the subgraph is no greater than n **for paths within the subgraph**
- n-club: a maximal subgraph of diameter n

Relaxations of Cliques

Subgroups based on nodal degree

Relaxations of Cliques

Subgroups based on nodal degree

- k-plex: a subgrah of g nodes in whcich each node is adjacent to no fewer than $g - k$ ndoes in the subgraph

Relaxations of Cliques

Subgroups based on nodal degree

- k-plex: a subgrah of g nodes in whcich each node is adjacent to no fewer than $g - k$ ndoes in the subgraph
- k-core: a subgraph in which each node is adjacent to at least k other nodes in the subgroup

What to do with these subgroup definitions?

What to do with these subgroup definitions?

- characterize the overall structure of the network

What to do with these subgroup definitions?

- characterize the overall structure of the network
- analyze the pattern of co-membership in cliques, etc.

K-Core Example

RESEARCH ARTICLE

The Critical Periphery in the Growth of Social Protests

Pablo Barberá[1]*, Ning Wang[2], Richard Bonneau[3,4], John T. Jost[1,5,6], Jonathan Nagler[6], Joshua Tucker[6], Sandra González-Bailón[7]*

1 Center for Data Science, New York University, New York, New York, 10003, United States of America, **2** Mathematical Institute and Oxford Internet Institute, University of Oxford, Oxford, OX26GG, United Kingdom, **3** Center for Genomics and System Biology, New York University, New York, New York, 10003, United States of America, **4** Simons Center for Data Analysis, Simons Foundation, New York, New York, 10010, United States of America, **5** Department of Psychology, New York University, New York, New York, 10003, United States of America, **6** Department of Politics, New York University, New York, New York, 10012, United States of America, **7** Annenberg School for Communication, University of Pennsylvania, Philadelphia, Pennsylvania, 19104, United States of America

* pablo.barbera@nyu.edu (PB); sgonzalezbailon@asc.upenn.edu (SGB)

K-Core Example

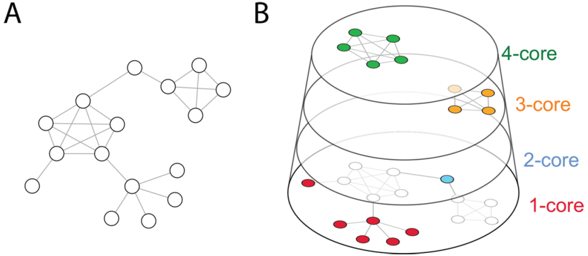


Fig 2. Schematic representation of the *k*-core decomposition for a random network with N = 16 vertices and E = 24 edges. This technique recursively prunes the network to remove nodes with the lowest degree. The coreness of a vertex is *k* if it belongs to the *k*-core but not to the (*k*+1)-core.

K-Core Example

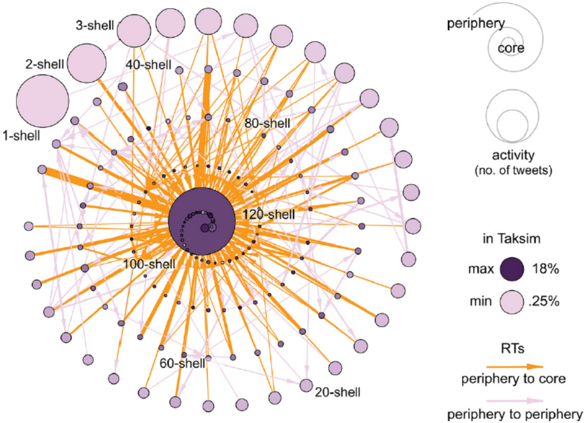
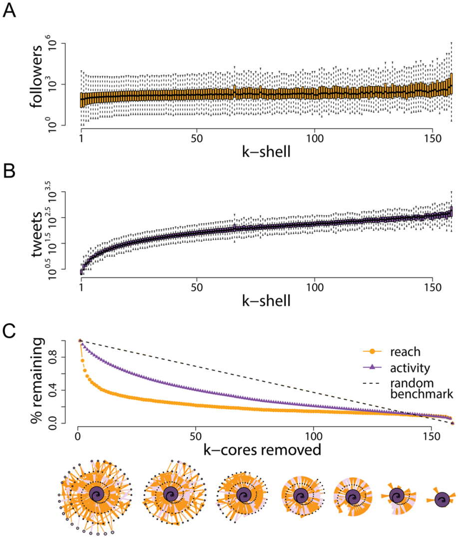


Fig 3. *K*-core decomposition of the network of retweets that emerged during the 2013 Taksim Gezi Park protests in Turkey (see S1 Text). Participants have been grouped in their corresponding *k*-shells, here represented by nodes. Lower *k*-shells contain participants at the periphery of the network; higher *k*-shells contain core participants. Node size is proportional to aggregated activity, measured as total number of protest messages (not just retweets). Arcs indicate retweeting activity, and their width is proportional to normalized strength (arcs with lower strength have been filtered to improve the visualization of the network). The darkness of nodes is proportional to the percentage of participants who reported being in the Taksim Gezi Park (the geographical epicenter of the protests), as indicated by the geographic information attached to their tweets. Most of these participants are at the core of the network where most RTs are also sourced from, thus allowing information to flow from the core to the periphery.

Intro
R Intro
Basic SNA Measures
Graph Level Indices
Simulation

K-Core Example



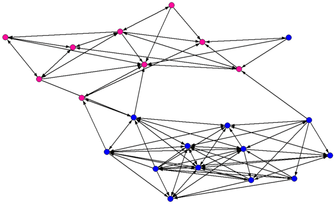
Block Models

Block Models

- Attribute Based

Block Models

- Attribute Based



Block Models

- Attribute Based

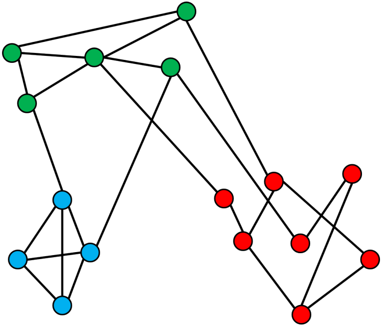
```
                        Boys                 Girls
         1 2 4 5 5 7 8 9 9 0 1 2 6   3 4 3 6 7 8 0 1 2
         1 2 1 1 5 1 1 1 9 1 1 1 1   1 4 3 6 7 8 2 2 2
       ------------------------------------------------
       1  1 |     1 1 1   1           1 |     1   1         |
       2  2 |       1 1 1     1   1   1 |                   |
      14 14 | 1 1   1 1 1 1   1   1 1 1 |                   |
      15 15 | 1   1   1   1   1   1   1 |                   |
Boys   5  5 | 1 1 1       1   1       1 |     1             |
      17 17 | 1 1   1 1   1 1 1   1   1 |                   |
      18 18 | 1     1 1           1     |                   |
      19 19 |     1 1         1       1 |                   |
       9  9 |     1 1                 1 |                   |
      10 10 |                           |       1 1       1 |
      11 11 |   1   1     1             |                   |
      12 12 |   1 1       1 1     1   1 |                 1 |
      16 16 |     1 1         1         |                   |
       ------------------------------------------------
      13 13 |                           |   1         1 1 1 |
       4  4 | 1                         | 1       1   1 1   |
       3  3 |                           |       1 1         |
Girls  6  6 |                   1       |                   |
       7  7 |                           |     1             |
       8  8 |                           |       1 1   1     |
      20 20 |                           | 1 1   1 1 1   1 1 |
      21 21 |                           | 1 1     1         |
      22 22 |                           | 1     1 1   1     |
       ------------------------------------------------
```

Block Models

- Attribute Based
- Relation Based

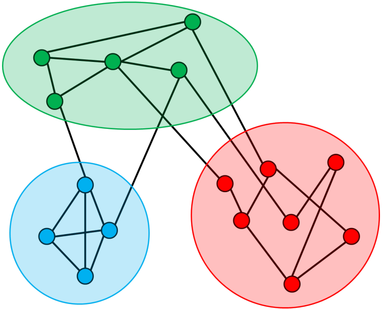
Block Models

- Attribute Based
- Relation Based



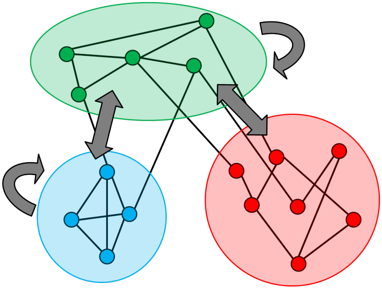
Block Models

- Attribute Based
- Relation Based



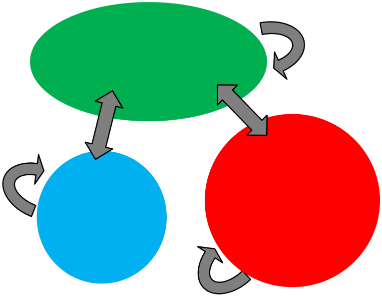
Block Models

- Attribute Based
- Relation Based



Block Models

- Attribute Based
- Relation Based



Intro

R Intro

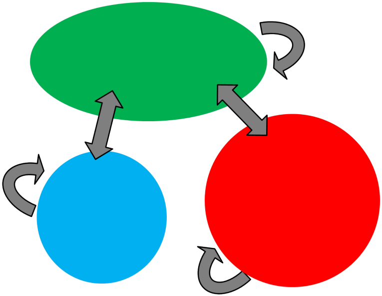
Basic SNA Measures

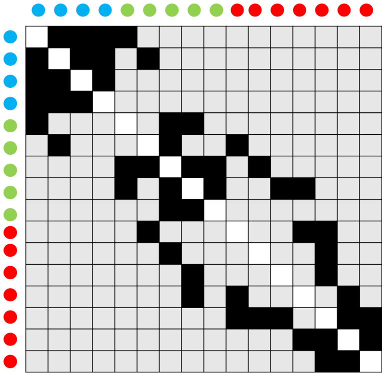
Graph Level Indices

Simulation

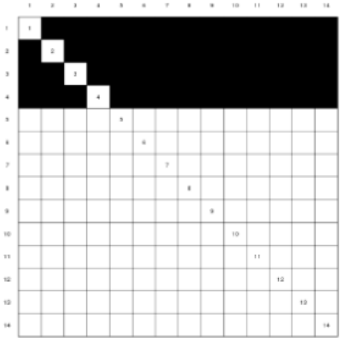
Block Models

- Attribute Based
- Relation Based





Core-Periphery Models



Intro

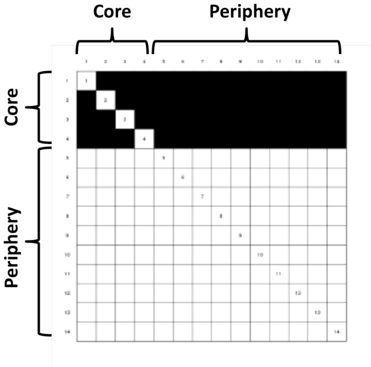
R Intro

Basic SNA Measures

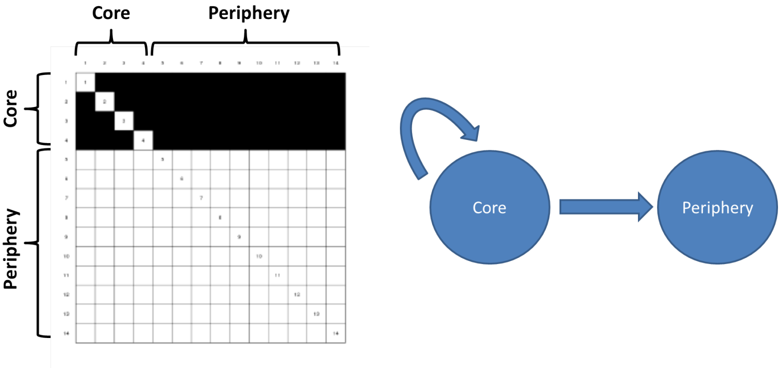
Graph Level Indices

Simulation

Core-Periphery Models



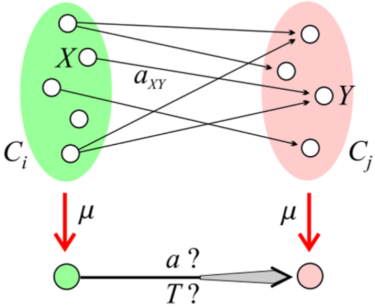
Core-Periphery Models



Even more blockmodeling

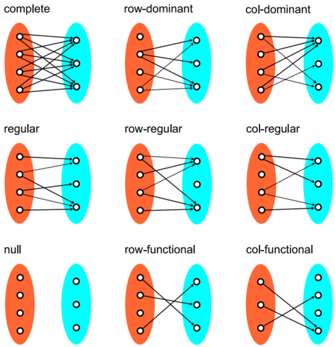
Generalized Blockmodeling

A *blockmodel* consists of structures obtained by identifying all units from the same cluster of the clustering **C**. For an exact definition of a blockmodel we have to be precise also about which blocks produce an arc in the *reduced graph* and which do not, and of what *type*. Some types of connections are presented in the figure on the next slide. The reduced graph can be represented by relational matrix, called also *image matrix*.



Doreian, Batagelj, and Ferligoj *Generalized Blockmodeling*, 2010

Even more blockmodeling



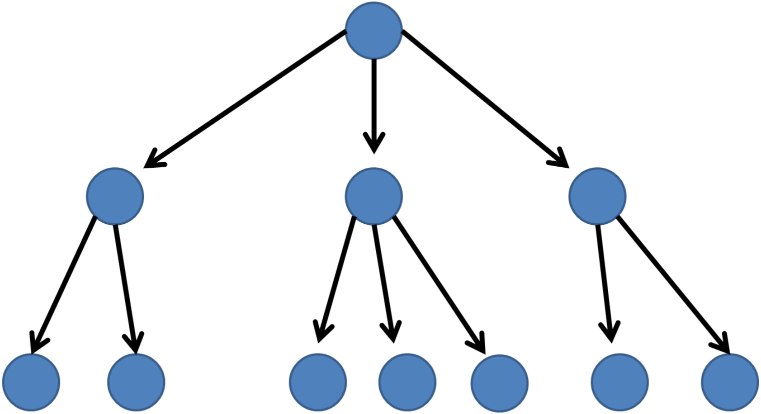
null	nul	all 0 *
complete	com	all 1 *
regular	reg	1-covered rows and columns
row-regular	rre	each row is 1-covered
col-regular	cre	each column is 1 -covered
row-dominant	rdo	∃ all 1 row *
col-dominant	cdo	∃ all 1 column *
row-functional	rfn	∃! one 1 in each row
col-functional	cfn	∃! one 1 in each column
non-null	one	∃ at least one 1

* except this may be diagonal

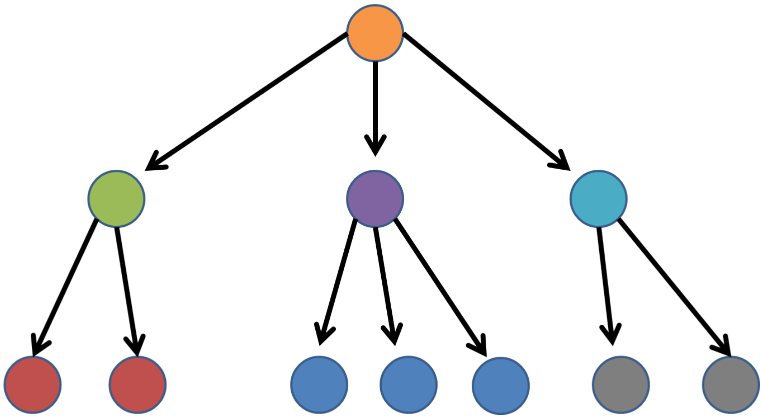
Doreian, Batagelj, and Ferligoj *Generalized Blockmodeling*, 2010

Structural Equivalence

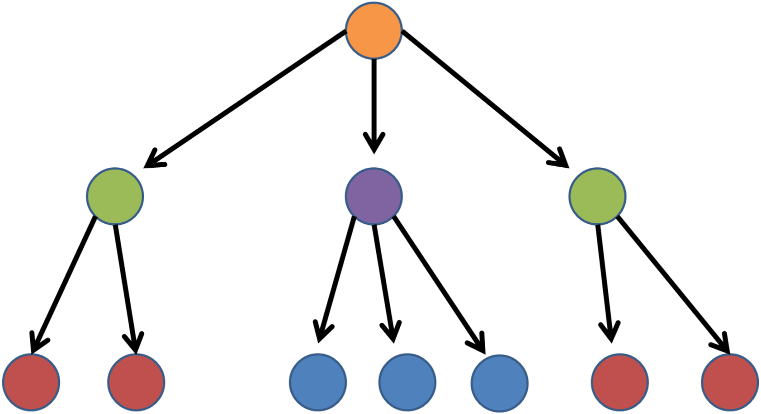
Structural Equivalence



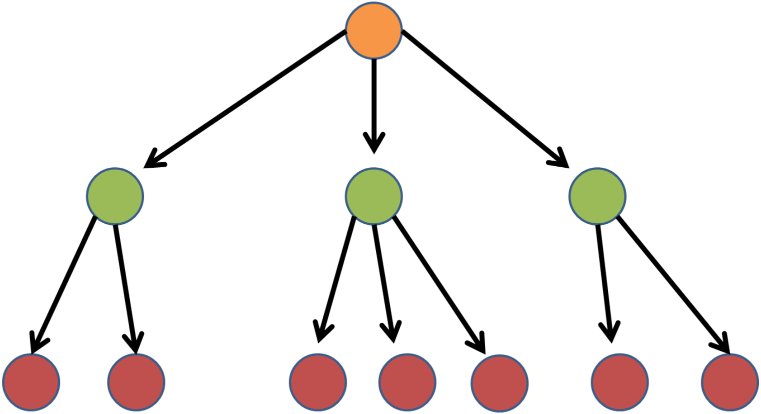
Structural Equivalence



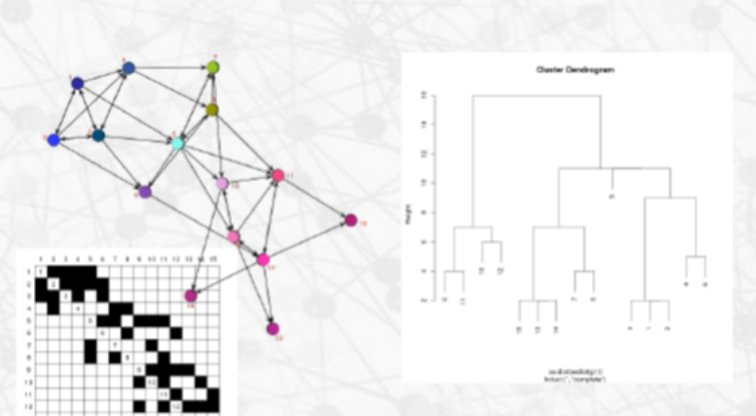
Automorphic Equivalence



Regular Equivalence

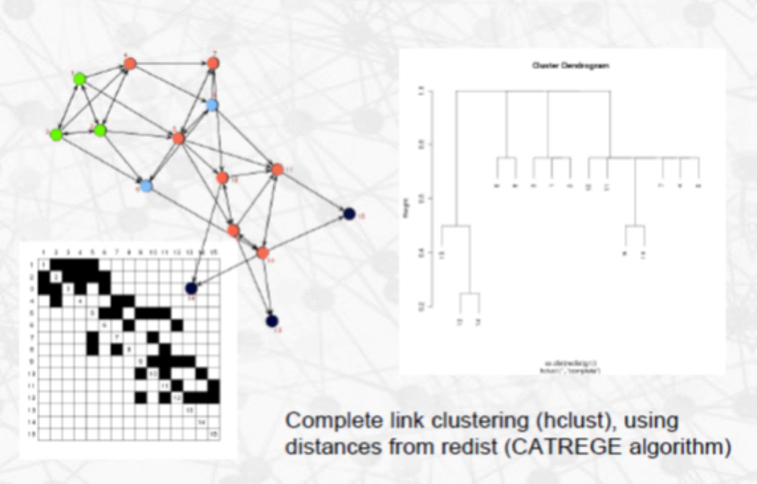


Structural Equivalence Example



Complete link clustering (hclust), using distances from sedist (Hamming distances)

Regular Equivalence Example



Community Structure

Community Structure

- Modularity index measures the 'goodness' of an assignment of nodes to subgroups by comparing the number of ties within groups as compared to the number of expected ties

Community Structure

- Modularity index measures the ‘goodness’ of an assignment of nodes to subgroups by comparing the number of ties within groups as compared to the number of expected ties

$$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j$$

Intro

R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Community Structure

- Modularity index measures the 'goodness' of an assignment of nodes to subgroups by comparing the number of ties within groups as compared to the number of expected ties

$$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j$$

- where m is the number of nodes, A is the adjacency matrix, k is the node's degree, and s is an indicator of which group the node is assigned

Community Structure

- Modularity index measures the 'goodness' of an assignment of nodes to subgroups by comparing the number of ties within groups as compared to the number of expected ties

$$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j$$

- where m is the number of nodes, A is the adjacency matrix, k is the node's degree, and s is an indicator of which group the node is assigned
- find the assigments of s that maximizes Q

Community Structure

- Modularity index measures the 'goodness' of an assignment of nodes to subgroups by comparing the number of ties within groups as compared to the number of expected ties

$$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j$$

- where m is the number of nodes, A is the adjacency matrix, k is the node's degree, and s is an indicator of which group the node is assigned
- find the assigments of s that maximizes Q
- M.E.J. Newman "Modularity and community structure in networks" *Proc Natl Acad Sci* 2006

Bernoulli ‘Random’ Graphs

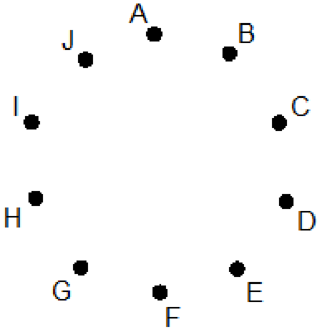
Bernoulli 'Random' Graphs

- A probability distribution for a success/failure

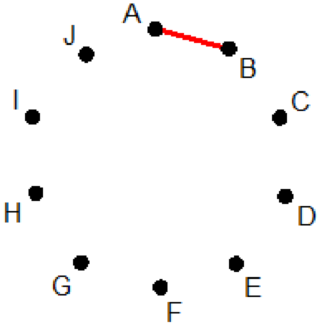
Bernoulli ‘Random’ Graphs

- A probability distribution for a success/failure
- Best known example is the coin flip

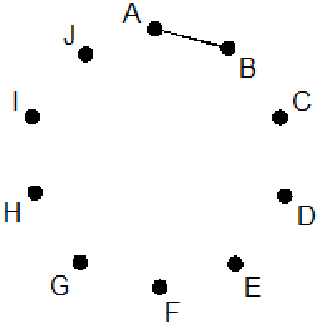
Bernoulli 'Random' Graphs



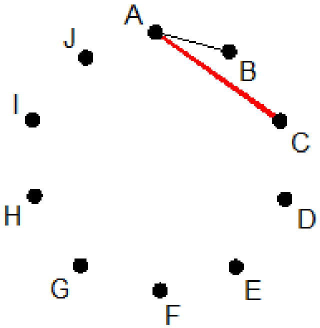
Bernoulli ‘Random’ Graphs



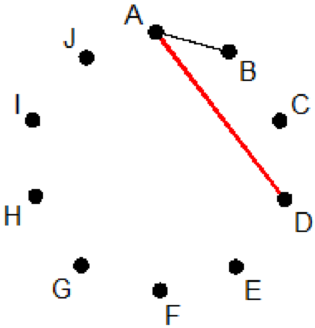
Bernoulli 'Random' Graphs



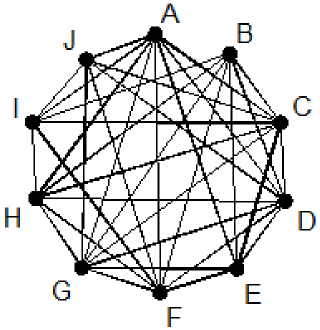
Bernoulli ‘Random’ Graphs



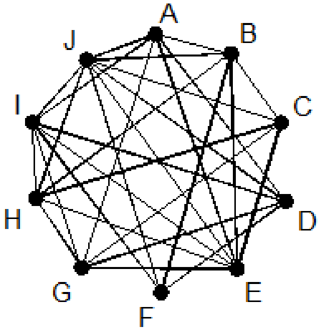
Bernoulli ‘Random’ Graphs



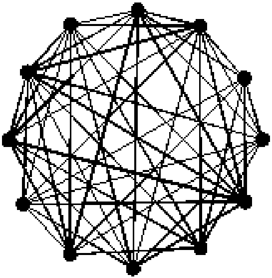
Bernoulli 'Random' Graphs



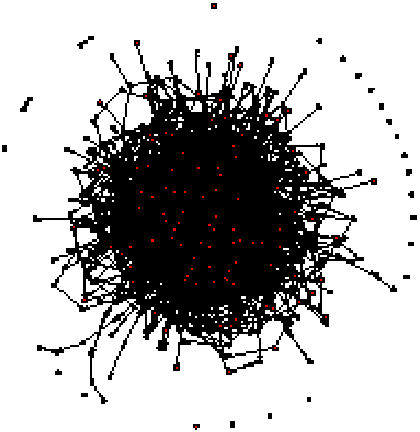
Bernoulli 'Random' Graphs



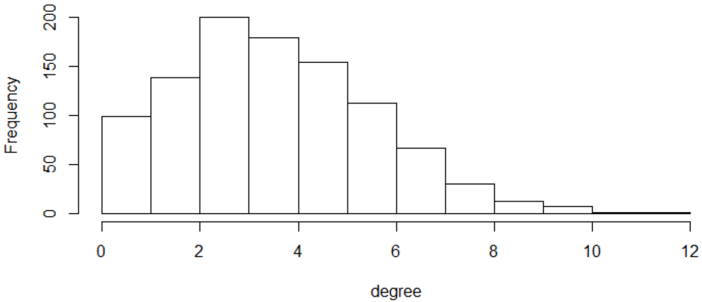
Bernoulli 'Random' Graphs



Bernoulli 'Random' Graphs



Bernoulli 'Random' Graphs



Small Worlds

What were the characteristics of Milgram's small world?

Watts-Strogatz Model

Watts-Strogatz Model

- high clustering coefficient

Watts-Strogatz Model

- high clustering coefficient
- low diameter

Clustering Coefficient and Diameter

Clustering Coefficient and Diameter

- The ***diameter*** of a graph is the longest shortest path between any pair nodes

Clustering Coefficient and Diameter

- The ***diameter*** of a graph is the longest shortest path between any pair nodes
- The ***clustering coefficient*** of a graph is the number of triangles divided by the number of two-paths

Intro

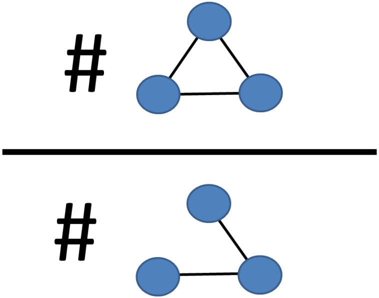
R Intro

Basic SNA Measures

Graph Level Indices

Simulation

Clustering Coefficient

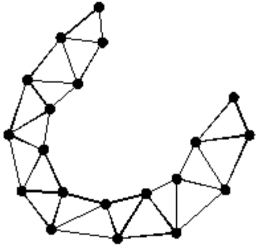


Watts-Strogatz Model

- high clustering coefficient
- low diameter

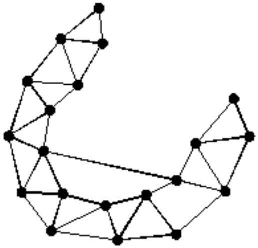
Watts-Strogatz Model

- high clustering coefficient
- low diameter
- starts with a lattice structure

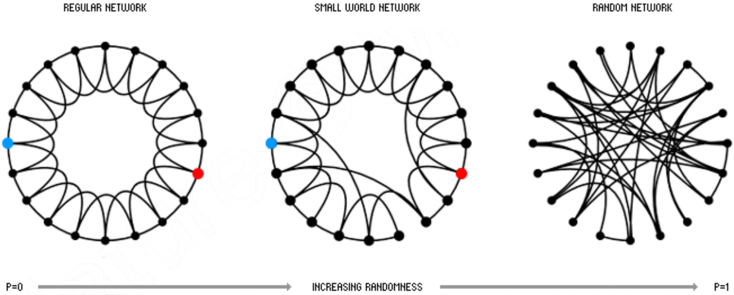


Watts-Strogatz Model

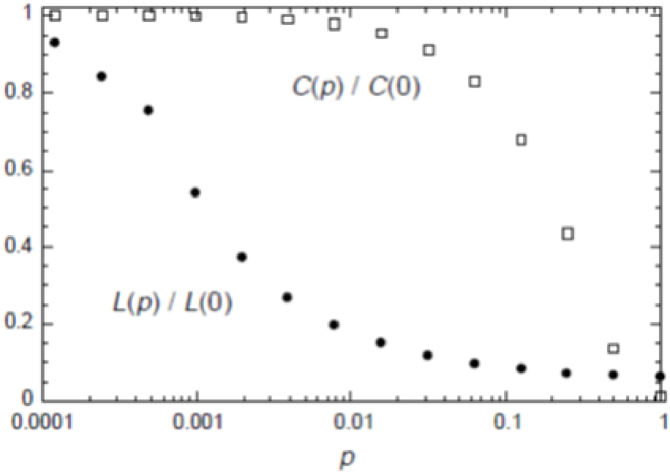
- high clustering coefficient
- low diameter
- starts with a lattice structure
- randomly re-wires ties



Watts-Strogatz Model



Watts-Strogatz Model



Watts-Strogatz Model

	L_{actual}	L_1	C_{actual}	C_1
Film Actors	3.65	2.99	0.79	0.00027
Power Grid	18.7	12.4	0.08	0.005
C. Elegans	2.65	2.25	0.28	0.05

Code Time

- the rest! whew!