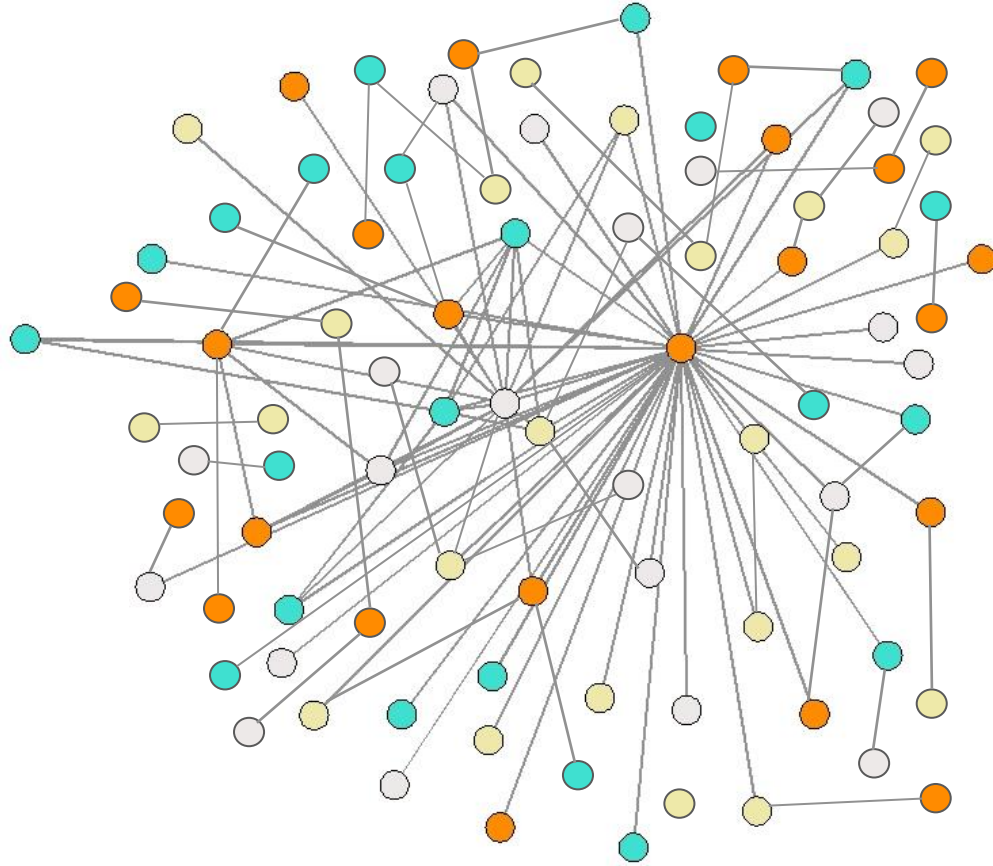


2018



A Novice's Guide:

Social Network Analysis Using R Statnet

A Novice's Guide: Social Network Analysis Using R Statnet

Florida Department of Health in Orange County

Danielle Rankin | Danielle.Rankin@flhealth.gov

Contents

1. Overview of the Social Network Analyses Toolkit.....	1
1.1 R Software and Statnet Package.....	1
1.2 Download and Install R Package Statnet.....	2
1.3 Getting Started in R Statnet.....	3
1.3.1 Change Work Directory—	3
1.3.2 Setting Packages for Analysis—	4
1.4 R Coding Rules and Structure	4
1.5 General R Statnet Commands.....	6
2. General Social Network Terminology	7
3. Social Network Analysis Tutorial	9
3.1 Creating a Weighted Adjacency Matrix	9
3.2 Creating a Weighted Edge List	10
3.3 Uploading/Importing Weighted Adjacency Matrix Dataset(s) Into R	11
3.4 Conducting a Social Network Analysis With a Weighted Adjacency Matrix	12
3.5 Uploading/Importing Weighted Edge List Dataset(s) Into R	13
3.6 Conducting a Social Network Analysis With a Weighted Edge List	14
4. Social Network Centrality Measure Terminology	15
5. Sociogram Analysis.....	19
5.1 Network Level Indices.....	19
5.2 Vertex/Node Level Indices.....	21
6. Helpful Resources.....	23
6.1 YouTube Resources.....	23
6.2 Text Resources	23
7. Toolkit Evaluation Survey.....	24
8. Acknowledgements.....	24
9. References	24

1. Overview of the Social Network Analyses Toolkit

In the realm of public health, there has been an increasing trend in exploration of social network analyses (SNAs). SNAs are methodological and theoretical tools that describe the connections of people, partnerships, disease transmission, the interorganizational structure of health systems, the role of social support, and social capital.¹ The Florida Department of Health in Orange County (DOH-Orange) created this toolkit to provide a baseline understanding on how to conduct SNAs using R Statnet.

The toolkit will introduce general terminology and illustrations for the user to become familiar with SNAs. Next, the toolkit will demonstrate dataset creation (i.e., adjacency matrices and edgelist) and then will introduce SNA calculation measures (i.e., centrality measures) that are generally used in analyses. The end of the toolkit will go through a tutorial of conducting a SNA with step-by-step coding using R Statnet.

Objectives of this toolkit:

1. Recognize SNA terminology and depictions
2. Practice importing/uploading SNA data into R Statnet
3. Conduct SNA graph and node-level calculations
4. Interpret SNA results and output

To conduct this tutorial, you will need:

1. Any version of Microsoft Excel
2. R Software and RStudio
3. R Statnet Package

There are additional calculations that can be conducted with SNAs that are not included in this toolkit. [Section 6](#) will provide several resources and concepts to perform additional SNA functions.

1.1 R Software and Statnet Package

R Software

R is a language and environment for statistical computing and graphics.² The software system is available as free software and runs on Windows and macOS. To download the R software, go to www.r-project.org/.

RStudio

RStudio is an integrated development environment for R.³ The software is available in open source and runs on Windows, Mac, and Linux. To download RStudio, go to www.rstudio.com/products/RStudio/#Desktop.

Statnet Package

Statnet is a suite of software packages for statistical network analysis that implement the recent advances in the statistical modeling of networks.⁴ The statnet package runs and is installed in R. [Section 1.2](#) provides a demonstration for installing the statnet package.

1.2 Download and Install R Package Statnet

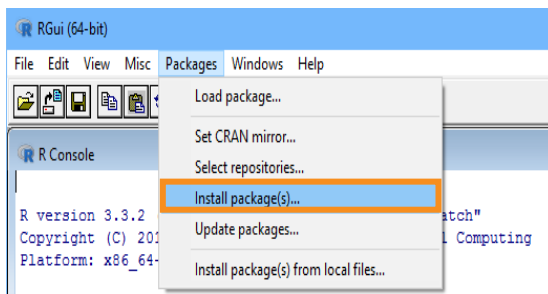
RSoftware

Step 1. Open R

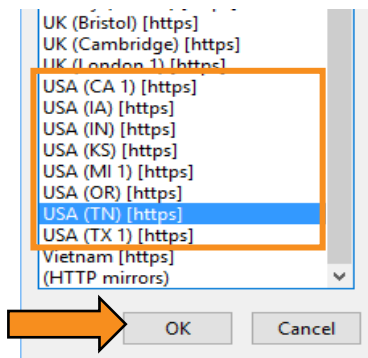
Step 2. Select “Packages” from the taskbar taskbar



Step 3. Select “Install package(s)...” from the drop-down menu



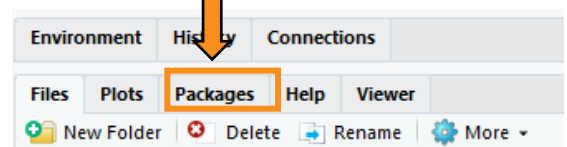
Step 4. Once selected, an HTTPS CRAN mirror selection box will appear. Scroll down and select any of the “USA” selections. Then select “OK.”



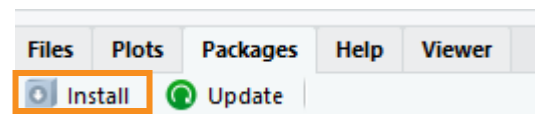
RStudio

Step 1. Open RStudio

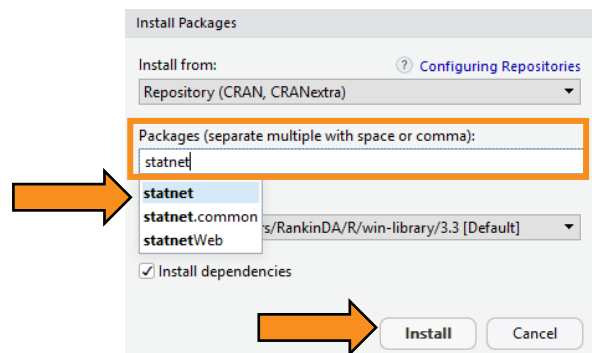
Step 2. Select “Packages” from right side of screen



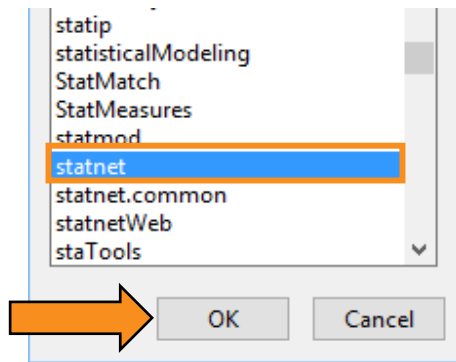
Step 3. Select Install



Step 4. Once selected, an Install Packages box will appear. Under “Packages,” type statnet. Select Statnet, and then select “Install.”



Step 5. A “Packages” selection box will appear. Scroll down and select “statnet.” Then select “OK.”



Step 5. Your R console will then indicate the statnet package is installed.

```
> install.packages("statnet")
Installing package into '\\chd48sfp01/users/Rankinda/R/win-
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.
Content type 'application/zip' length 28492 bytes (27 KB)
downloaded 27 KB
package 'statnet' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
C:\Users\rankinda\AppData\Local\Temp\RtmpGGQDHQ\downl
```

Step 6. Your R console will then indicate the statnet package is installed

```
trying URL 'https://mirrors.nics.utk.edu/cran/bin/windows/contrib/3.3/statnet_20.
Content type 'application/zip' length 28482 bytes (27 KB)
downloaded 27 KB
```

```
package 'statnet' successfully unpacked and MD5 sums checked
```

```
The downloaded binary packages are in
C:\Users\rankinda\AppData\Local\Temp\RtmpYNB4rI\downloaded_packages
```

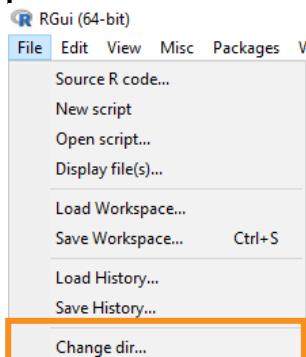
Note: Once the statnet package is installed, it is stored in your R package unless you uninstall it. Therefore, there is no need to install the statnet package each session. Occasionally, R will prompt you to update the package. You can do this through the “Packages” selection on the taskbar. **Alternatively, you can install packages through the install.packages command (i.e., install.packages(“statnet”)).**

1.3 Getting Started in R Statnet

1.3.1 Change Work Directory— For ease of access to your files, change your work directory to the location your files are saved.

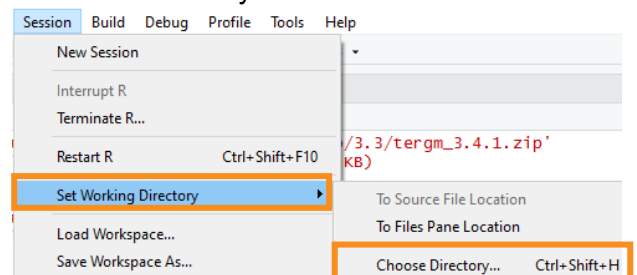
R Software

Step 1. Go to File and Select “Change dir...”

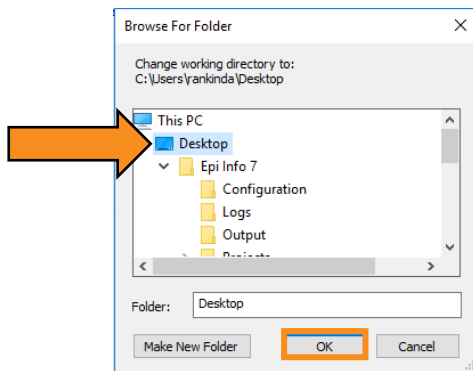


RStudio

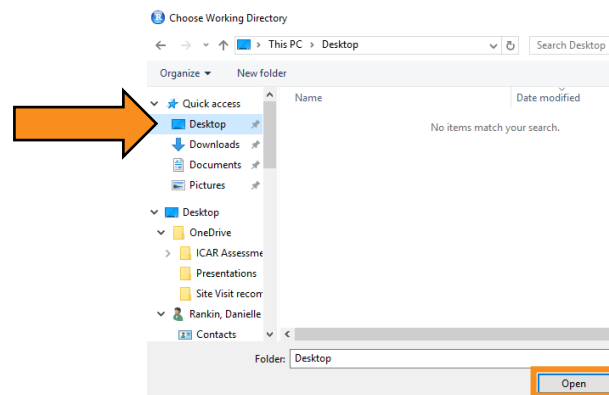
Step 1. Go to Session and Select “Set Working Directory”; then “Choose Directory...”



Step 2. A “Browse For Folder” box will appear; scroll through the browser box to locate the folder to which your files are saved. For this tutorial, we will save our SNA demo files on our “Desktop.” Find and select the “Desktop” folder in the browser box; once you have located your “Desktop” folder select “OK.”



Step 2. A “Choose Working Directory” box will appear. For this tutorial, we will save our SNA demo files on our “Desktop.” Find and select the “Desktop” folder in the box; once you have located your “Desktop” folder select “Open.”



1.3.2 Setting Packages for Analysis— In order for R to conduct the desired analyses, one must prompt R with the necessary package(s). Note the remaining coding in the subsequent sections are the same codes and steps for R Software and RStudio.

Step 1. To use the downloaded statnet package, click into the R console and type the following code and press “ENTER:”

```
> library(statnet)
```

R will present a “>” with the cursor blinking, once the package has completely loaded

1.4 R Coding Rules and Structure

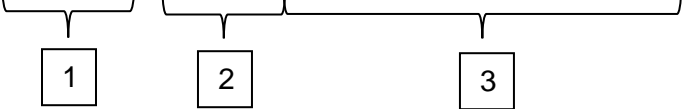
Before beginning this step-by-step tutorial, it is important to understand some general rules and structure of commands/coding in R.

Rules

1. R runs through command lines (provides R with the command to run) or batch files (a series of commands).²⁻⁵
2. R is case sensitive.²⁻⁵
3. R will run the code/command when “ENTER” is pressed on the keyboard.
4. R will return a “>” when ready for the next command.
5. R will return a “+” if the command or code is not complete. For instance, if you have a command with an argument and the end parenthesis is forgotten, R will return a “+.”
6. R uses “,” to separate arguments/parameters.
7. R assignments are indicated by using “<-” **NOT** “=.” For instance, to assign x to equal 3 you would type x<- 3.
8. R comments you want to assign to your command/code begin and end with “#” (e.g., `#comment example#`).

Structure

Code: `dataname<-command(x, arguments/parameters)`



1. The first portion (before "<-") of the code will generally be a name you create and will refer to your dataset. For instance, the example shows the data are referred as `dataname`. Typically, the name should be kept short and concise for ease and to avoid data entry error.
2. The second portion of the code generally contains the command. A command communicates with R to perform a task.
3. The third portion of the code contains the following:
 - a. "x"—the data name you created and is being analyzed.
 - b. Commas are used within the parentheses to separate each reference/argument/parameter
 - c. Arguments/parameters differ for each command. If no arguments/parameters are included in the code, R will refer to the default.

Command: `command(x, arguments/parameters)`

Notice the format of commands follow the second and third portion of the code described above, except for commands such as `library(x)` where "x" will represent the package desired.




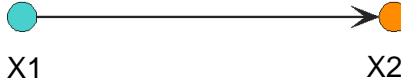
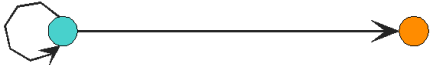
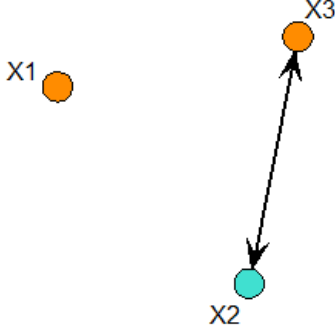
Note: For additional information on the structure of coding and commands in R, refer to the built-in user guide (i.e., `help(x)` or `?(x)`).

1.5 General R Statnet Commands

Type	Function/Term	Command [†]
General	Help/Assistance	help(x) ? Type “?” before any command (e.g. ?library)
	Loading Package	library(x)
	Importing csv Files	read.csv(x)
	View	view(x)
	Install Package	install.packages(“package_name”)
	Update Package	update.packages(“package_name”)
Network Measures	Network Summary	summary(x)
	Size	network.size(x)
	Density	gden(x)
	Reciprocity	grecip(x)
	Edgewise Reciprocity	grecip(x, measure= “edgewise”)
	Degree	centralization(x, degree)
	Outdegree	centralization(x, degree, cmode= “outdegree”)
	Indegree	centralization(x, degree, cmode= “indegree”)
	Betweenness	centralization(x, betweenness)
	Closeness	centralization(x, closeness)
	Eigenvector	centralization(x, evcent)
	Dyads	network.dyadcount(x)
Vertex/Node Level	Degree	degree(x)
	Outdegree	degree(x, cmode= “outdegree”)
	Indegree	degree (x, cmode= “indegree”)
	Betweenness	betweenness(x)
	Closeness	closeness(x)
	Eigenvector	evcent(x)
Sociogram/Networks	Plotting sociogram	gplot(x)

Note: (x) after each command denotes the file name you create when importing the dataset into R. [†]Commands were created from references 2-4, 14.

2. General Social Network Terminology

Term	Definition ¹⁷	Depiction
Vertex/Vertices	The individual actors within a network (e.g., individual persons, hospital facilities, companies, etc.) also are called nodes, points, or actors.	
Edge	The connections between vertices or nodes; other common names are lines, ties, or (if directed) arcs.	
Undirected graph	A graph that does not indicate a directionality of connection between vertices.	
Directed graph	A graph, also known as a digraph, that indicates a direction of connection between vertices. A digraph consists of endpoints that contain two elements. 1. Tail/sender 2. Head/receiver (e.g., the blue vertex [X1] is the tail/sender in the relationship; the orange vertex [X2] is the head/receiver in the relationship).	
Loop	An edge whose endpoint is itself (e.g., a hospital transferred a patient to a different unit within the hospital).	
Isolate	Vertex that does not have edges within the network (e.g., "X1" is an isolate within the network).	

Term	Definition [™]	Depiction																																																																								
Adjacency matrix	<p>A square matrix with as many rows and columns as there are vertices in the dataset (e.g., “1” is equivalent to “X1” and “2” is equivalent to “X2”). An adjacency matrix is used to import the network dataset into R, usually used when analyzing small datasets (e.g., datasets ≤100 by 100). In an unweighted adjacency matrix, “0” or “1” is used to represent the relationship between edge(s). For instance, 1 represents a connection between two vertices (e.g., vertex 1/X1 has a connection [“1”] with vertex 2/X2, but does not have a connection [“0”] with itself; vertex 2/X2 has a connection with vertex 1/X1, but does is not connected [“0”] to itself. A weighted adjacency matrix contains a continuous range of values; “0” indicates no connection. The continuous range of values illustrates edge qualities or strengths, which is used to identify the strength of relationship, tie, connection, etc.</p>	<p>Unweighted Adjacency Matrix</p> <table><tr><th>▲</th><th>X1</th><th>▼</th><th>X2</th><th>▼</th></tr><tr><td>1</td><td>0</td><td></td><td>1</td><td></td></tr><tr><td>2</td><td>1</td><td></td><td>0</td><td></td></tr></table> <p>Weighted Adjacency Matrix</p> <table><tr><th>▲</th><th>X1</th><th>▼</th><th>X2</th><th>▼</th></tr><tr><td>1</td><td>0</td><td></td><td>5</td><td></td></tr><tr><td>2</td><td>17</td><td></td><td>0</td><td></td></tr></table>	▲	X1	▼	X2	▼	1	0		1		2	1		0		▲	X1	▼	X2	▼	1	0		5		2	17		0																																											
▲	X1	▼	X2	▼																																																																						
1	0		1																																																																							
2	1		0																																																																							
▲	X1	▼	X2	▼																																																																						
1	0		5																																																																							
2	17		0																																																																							
Edge list	<p>A two-column list of two vertices that are connected in a network. An edge list is another method to import a network dataset into R, usually used when analyzing larger datasets. To include edge weights, one must create another column that represents the “weight.” The weighted connections or relationships among vertex pairs must be calculated prior to creating the edge list. One should avoid duplicate entries for a vertex pair (e.g., the connection between “X1” and “X3” appeared in the original dataset 100 times, therefore was summed together and represented in the “Weight” column).</p> <p>NOTE: If the network/graph is undirected, then the vertex order that is displayed in the first column (“V1”) versus the second column (“V2”) is irrelevant. If the network/graph is directed, then R reads the first column (“V1”) in the edge list as the tail/sender and the second column (“V2”) as the head/receiver.</p>	<p>Edge List</p> <table><tr><th>▲</th><th>└.V1</th><th>▼</th><th>V2</th><th>▼</th></tr><tr><td>1</td><td>X1</td><td></td><td>X2</td><td></td></tr><tr><td>2</td><td>X1</td><td></td><td>X3</td><td></td></tr><tr><td>3</td><td>X2</td><td></td><td>X3</td><td></td></tr><tr><td>4</td><td>X3</td><td></td><td>X1</td><td></td></tr><tr><td>5</td><td>X3</td><td></td><td>X2</td><td></td></tr></table> <p>Edge List With Edge Weights</p> <table><tr><th>▲</th><th>└.V1</th><th>▼</th><th>V2</th><th>▼</th><th>Weight</th><th>▼</th></tr><tr><td>1</td><td>X1</td><td></td><td>X2</td><td></td><td>5</td><td></td></tr><tr><td>2</td><td>X1</td><td></td><td>X3</td><td></td><td>100</td><td></td></tr><tr><td>3</td><td>X2</td><td></td><td>X3</td><td></td><td>3</td><td></td></tr><tr><td>4</td><td>X3</td><td></td><td>X1</td><td></td><td>1</td><td></td></tr><tr><td>5</td><td>X3</td><td></td><td>X2</td><td></td><td>30</td><td></td></tr></table>	▲	└.V1	▼	V2	▼	1	X1		X2		2	X1		X3		3	X2		X3		4	X3		X1		5	X3		X2		▲	└.V1	▼	V2	▼	Weight	▼	1	X1		X2		5		2	X1		X3		100		3	X2		X3		3		4	X3		X1		1		5	X3		X2		30	
▲	└.V1	▼	V2	▼																																																																						
1	X1		X2																																																																							
2	X1		X3																																																																							
3	X2		X3																																																																							
4	X3		X1																																																																							
5	X3		X2																																																																							
▲	└.V1	▼	V2	▼	Weight	▼																																																																				
1	X1		X2		5																																																																					
2	X1		X3		100																																																																					
3	X2		X3		3																																																																					
4	X3		X1		1																																																																					
5	X3		X2		30																																																																					

Note: Formats other than adjacency matrices and edge lists can be used to import a network dataset for SNAs. [™]Definitions were created from references 1-4, 5-6.

3. Social Network Analysis Tutorial

In this tutorial, we are creating and assessing a demonstration dataset to analyze the relationship among five county health departments (CHDs) in Florida. This is a mock dataset and was created for the educational purpose of this toolkit.

Note: The R package needs to be installed on your PC before starting this tutorial.

3.1 Creating a Weighted Adjacency Matrix

Step 1. To start, open any version of Microsoft Excel.

Step 2. The five counties for which we will be creating a demonstration weighted adjacency matrix are Duval, Okaloosa, Orange, Palm Beach, and Pinellas. In cells “A2” through “A6,” type the name of each of the five counties. Next, type the five counties in cells “B1” through “F1.” Once completed, the Microsoft Excel spreadsheet should look like the following:

	A	B	C	D	E	F
1		Duval	Okaloosa	Orange	Palm Beach	Pinellas
2	Duval					
3	Okaloosa					
4	Orange					
5	Palm Beach					
6	Pinellas					

Step 3. Now we will input the weighted connections into the Excel spreadsheet:

	A	B	C	D	E	F
1		Duval	Okaloosa	Orange	Palm Beach	Pinellas
2	Duval	0	0	3	0	0
3	Okaloosa	0	0	1	5	2
4	Orange	0	3	1	2	5
5	Palm Beach	0	5	0	0	7
6	Pinellas	0	2	3	5	0

Step 4. You have created your weighted adjacency matrix; now go to File -> Save As; save your file as sna_demo in a CSV (comma delimited) format on your desktop.

Note: If the project you are trying to complete consists of an unweighted adjacency matrix, the spreadsheet will be similarly arranged, but will **ONLY** contain values of “0” or “1” in the rows and columns. See “Unweighted Adjacency Matrix” in [Section 2](#).

3.2 Creating a Weighted Edge List

Step 1. To start, open any version of Microsoft Excel.

Step 2. We will be creating the weighted edge list with the same data used in [Section 3.1](#). First, we will label the first column with V1, the second column with V2, and the third column with WEIGHT. The Microsoft Excel spreadsheet should look like the following:

	A	B	C
1	V1	V2	WEIGHT
2			
3			

Step 3. Next, we will add the connections sent in column “V1” and the connections received in column “V2.” Start with the connections Duval County initiated/sent into the Microsoft Excel spread sheet:

	A	B	C
1	V1	V2	WEIGHT
2	Duval	Orange	

Step 4. Now, we will continue adding the remaining CHDs into the Microsoft Excel spreadsheet:

	A	B	C
1	V1	V2	WEIGHT
2	Duval	Orange	
3	Okaloosa	Orange	
4	Okaloosa	Palm Beach	
5	Okaloosa	Pinellas	
6	Orange	Okaloosa	
7	Orange	Orange	
8	Orange	Palm Beach	
9	Orange	Pinellas	
10	Palm Beach	Okaloosa	
11	Palm Beach	Pinellas	
12	Pinellas	Okaloosa	
13	Pinellas	Orange	
14	Pinellas	Palm Beach	

Step 5. Next, we need to input the weights associated with each connection into the Microsoft Excel spreadsheet:

	A	B	C
1	V1	V2	WEIGHT
2	Duval	Orange	3
3	Okaloosa	Orange	1
4	Okaloosa	Palm Beach	5
5	Okaloosa	Pinellas	2
6	Orange	Okaloosa	3
7	Orange	Orange	1
8	Orange	Palm Beach	2
9	Orange	Pinellas	5
10	Palm Beach	Okaloosa	5
11	Palm Beach	Pinellas	7
12	Pinellas	Okaloosa	2
13	Pinellas	Orange	3
14	Pinellas	Palm Beach	5

Step 6. You have created your weighted edge list, now go to File -> Save As; save your file as sna_edgelistdemo in a CSV (comma delimited) format on your desktop.

Note: Recall that edge lists are typically used to analyze large datasets. If your project is undirected and not weighted, then the vertex order and the “WEIGHT” column is unnecessary.

3.3 Uploading/Importing Weighted Adjacency Matrix Dataset(s) Into R

Step 1. To upload the csv adjacency matrix file “sna_demo” we created in [Section 3.1](#), click into the R console and type the following code and press “ENTER:”
`> snademo<-read.csv('sna_demo.csv', header=TRUE, row.names=1)`

As in [1.3.2](#), R will present a “>” with the cursor blinking once the dataset has been uploaded.

Note: R is case sensitive; if you receive an error message after you press “ENTER,” double check the file name.

Step 2. View the dataset you uploaded; click into the R console, type the file name and press “ENTER:” `> snademo`

R will populate the following:

	Duval	Okaloosa	Orange	Palm.Beach	Pinellas
Duval	0	0	3	0	0
Okaloosa	0	0	1	5	2
Orange	0	3	1	2	5
Palm Beach	0	5	0	0	7
Pinellas	0	2	3	5	0

3.4 Conducting a Social Network Analysis With a Weighted Adjacency Matrix

Step 1. Although we input the data into an adjacency matrix form, we need to tell R that we want the data to be read as a matrix. To do so, click into the R console, type the following code and press “ENTER:”

```
> snademo.m<-as.matrix(snademo)
```

Step 2. Next, we need to create a network to plot and command R how we want our data portrayed. Click into the R console, type the following code and press “ENTER:”

```
> snademo.n<-network(snademo.m, matrix.type="adjacency", directed=TRUE, loops=TRUE, ignore.eval=FALSE, names.eval="weight")
```

Note: The command `ignore.eval=FALSE` instructs R to not ignore the edge values; `names.eval= “weight”` commands R to store the edge values as an edge attribute called “weight.” For additional arguments and parameters within the network command, refer to the help/assistance command in R (i.e., `help(x)`).

Step 3. View the network. Click into the R console, type the following code and press “ENTER:”

```
> snademo.n
```

R will populate the network attributes:

```
Network attributes:
vertices = 5
directed = TRUE
hyper = FALSE
loops = TRUE
multiple = FALSE
bipartite = FALSE
total edges= 13
missing edges= 0
non-missing edges= 13

Vertex attribute names:
vertex.names

Edge attribute names:
weight
```

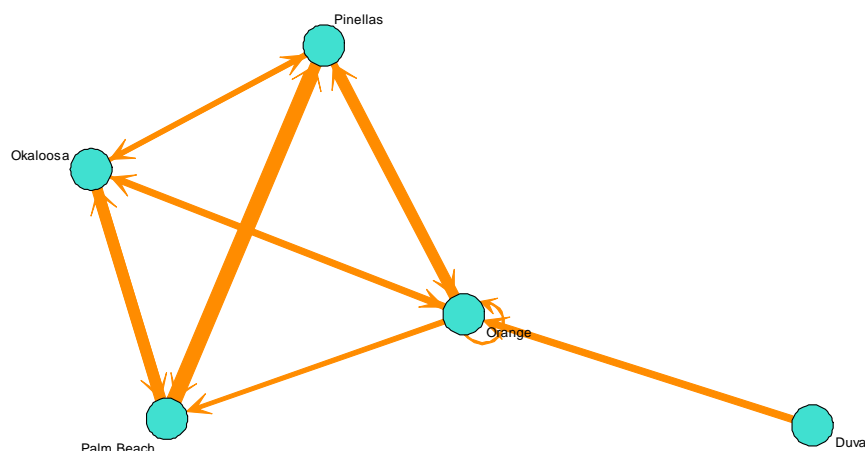
Interpretation: The network attributes indicate the network is directed and contains loops, 5 vertices, and 13 edges. Additionally, the network attributes display the edge attributes (weights) and are stored with the name “weight.” This indicates our previous command, `names.eval= “weight”`, was successful.

Step 4. Plot the network. Click in the R console, type the following code, and press “ENTER:”

```
> gplot(snademo.n, displaylabels=TRUE, label.cex=0.75, label.col="black", vertex.col="turquoise", edge.col="darkorange", edge.lwd = snademo.n%e%'weight', arrowhead.cex = 0.6, diag=TRUE)
```

Note: `edge.lwd = snademo.n%e%'weight'` commands R to reflect the edge values (strength of relationship) from the adjacency matrix in the arcs/edges. R will display the edge values through line thickness. `%e%` is an operator initiating the edge values to be displayed in the plot. `diag=TRUE` commands R to display loops in the plot. If you do not prompt R to display loops, R will default to FALSE (no loops displayed). To understand additional parameters and arguments within the `gplot` command, use the help/assistance command in R or refer to [A guide to the `gplot` function of the `sna` library for R](#).¹⁴ The network visualizations demonstrated in this toolkit are displayed with R Statnet's default layout "Fruchterman Reingold (1991)." For additional layout options, refer to [Social network analysis with `sna`](#).⁶

R will populate your social network analysis:



Interpretation: Before running calculations, the sociogram provides a visual depiction of the relationships among the five CHDs in Florida. The network appears to be moderately connected (centralization by degree). Orange, Pinellas, and Okaloosa CHDs have the majority of connections (degree=6), but Orange CHD contains a loop. Palm Beach and Pinellas counties appear to have the strongest relationship shared in the network (edge thickness). Additionally, Orange CHD acts as a bridge to connect Duval CHD to the other departments within the network (betweenness centrality). If you were to remove Orange CHD from the network, Duval would not be connected, thus making Orange CHD a pertinent actor in the network.

3.5 Uploading/Importing Weighted Edge List Dataset(s) Into R

Step 1. To upload the csv edge list file “`sna_edgelistdemo`” we created in [Section 3.2](#), click into the R console and type the following code and press “ENTER:”

```
> snaedgedemo<-read.csv('sna_edgelistdemo.csv', header=TRUE)
```

3.6 Conducting a Social Network Analysis With a Weighted Edge List

Step 1. Next, we need to create a network to plot and command R how we want our data portrayed. Click into the R console, type the following code and press “ENTER:”

```
> snaedgedemo.n<-network(snaedgedemo[,1:3], matrix.type="edgelist",  
directed=TRUE, loops=TRUE, ignore.eval=FALSE)
```

Note: [,1:3] tells R that we want to include all three columns from our dataset in the edge list. Additionally, since column 3 is already named “WEIGHT,” we do not need to use the command names.eval.

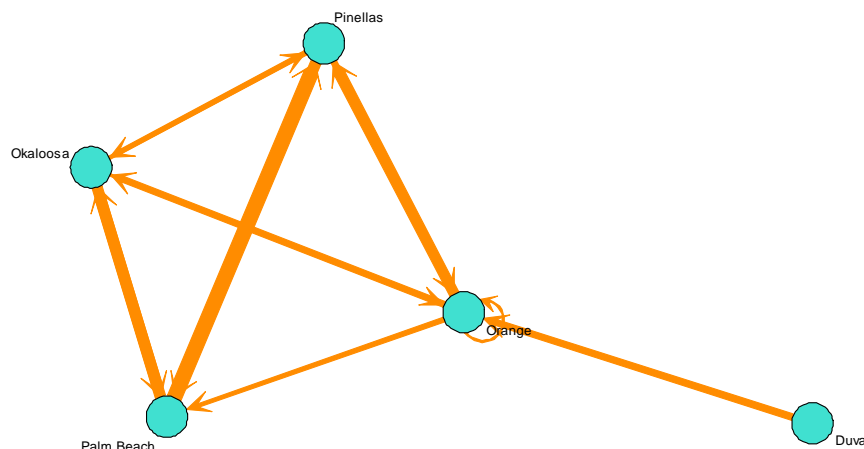
Step 2. View the network. Click into the R console, type the following code and press “ENTER:”

```
> snaedgedemo.n  
Network attributes:  
vertices = 5  
directed = TRUE  
hyper = FALSE  
loops = TRUE  
multiple = FALSE  
bipartite = FALSE  
total edges= 13  
missing edges= 0  
non-missing edges= 13  
  
Vertex attribute names:  
vertex.names  
  
Edge attribute names:  
WEIGHT
```

Note: Refer to the interpretation from [Section 3.4](#). Since we used the same data to create the weighted adjacency matrix and weighted edge list, the calculations and outputs are the same.

Step 3. Plot the network. Click in the R console, type the following code, and press “ENTER:”

```
> gplot(snaedgedemo.n, displaylabels=TRUE,label.cex=0.75,  
label.col="black", vertex.col="turquoise", edge.col="darkorange",  
edge.lwd = snaedgedemo.n%e%'WEIGHT', arrowhead.cex = 0.6, diag=TRUE)
```



Note: Refer to the interpretation in Step 4 of [Section 3.4](#).

4. Social Network Centrality Measure Terminology

Level	Term	Definition ^π	Interpretation
Network	Density	<p>The ratio of observable edges to the potential edges in a network; determines the network's cohesion. To calculate, one must first calculate the denominator of the Network Density fraction ("Potential Edges"). In the potential edges equation, "e" is the number of vertices in the network. "Observable edges" are the number of edges that are present in the network.</p> <p>1. Potential Edges:</p> $\frac{e * (e - 1)}{2}$ <p>2. Network Density:</p> $\frac{\text{Observable Edges}}{\text{Potential Edges}}$	<p>The network density is valued between 0 and 1. A value closer to 0 indicates a sparse network; a value closer to 1 indicates a tightly connected network.</p> <p>R Output [1] 0.4166667</p> <p>The R output depicted above is interpreted as a moderately sparse network.</p>
	Reciprocity	<p>The proportion of vertex pairs (dyad[s]) that are mutually or symmetrically linked.</p> <p>Reciprocity:</p> $\frac{\# \text{ of Mutual Dyads}}{\# \text{ of Adjacent Dyads}}$ <p>Alternatively, in a directed graph one can also calculate "edgewise" reciprocity, which is the fraction of reciprocated arcs/edges.</p> <p>Edgewise Reciprocity:</p> $\frac{\# \text{ of Reciprocated Arcs}}{\text{Total Number of Arcs}}$	<p>Reciprocity and edgewise reciprocity are interpreted as percentages.</p> <p>Reciprocity R Output Mut 0.8333333</p> <p>The R output indicates there is an 83.3% tendency a vertex will return a tie to another vertex.</p> <p>Edgewise Reciprocity R Output Mut 0.8</p> <p>The R output indicates 80% of the arcs/edges within the network are reciprocated.</p>

Level	Term	Definition ^π	Interpretation
Network	Diameter	The longest geodesic (shortest path between vertices) within the network. Diameter counts the number of instances it takes to get from one vertex to the other while using the shortest path. The greatest instance (e.g., longest geodesic) in the network represents the diameter.	<p>The diameter output will be valued as “Inf” where geodesics do not exist.</p> <p>R Output</p> <pre>\$gdist [,1] [,2] [,3] [,4] [1,] 0 1 2 Inf [2,] 1 0 1 Inf [3,] 2 1 0 Inf [4,] 1 2 3 0</pre> <p>The R output indicates the network diameter between vertices is 3.</p>
	Centralization	<p>The magnitude a network is influenced by a central vertex. The network can be influenced by one or several vertices within the network. Centralization can be calculated in several ways (e.g., degree, betweenness, and closeness), but depends on what you are trying to interpret from your network.</p> <ol style="list-style-type: none"> Degree: measures the extent to which one or a few vertices influence the network due to having a significantly greater amount of connections in the network. Betweenness: measures the extent to which one or a few vertices influence the network by creating geodesics between vertices within the network. Closeness: measures the extent to which one or a few vertices influence the network by the reciprocal of how far all vertices are from other vertices in the network. 	<p>Centralization measures (i.e., degree, betweenness, and closeness) output ranges from 0 to 1. A value approaching 0 is interpreted as a decentralized network (no vertex exceeds others within the network); a value approaching 1 indicates a strongly-centralized network (a vertex exceeds others within the network).</p> <p>1. Centralization by Degree R Output</p> <pre>[1] 0.25</pre> <p>The centralization measured by degree indicates the network is decentralized and not entirely dominated by a single vertex.</p> <p>2. Centralization by Betweenness R Output</p> <pre>[1] 0.15</pre> <p>Similarly, the centralization measured by betweenness indicates the network is decentralized.</p> <p>3. Centralization by Closeness R Output</p> <pre>[1] 0.6666667</pre> <p>Notice the centralization by closeness output is closer to 1. The network centralization by closeness is slightly centralized by one or a few vertices that are closer (e.g., can transfer information faster) than others in the network.</p>

Level	Term	Definition ^π	Level
Subgraph £	Dyad £	A pair of vertices and the possible edge(s)/connection(s) between them.	R Output [1] 16 The R output indicates there are 16 possible edges between the dyads in the network.
	Triad £	Three vertices/nodes and the edges/connections between them.	
	Components £	Subgroups of vertices that are connected to each other, but are disconnected from the remaining part of the network.	
Vertex	Centrality	Measures the central importance of each given vertex. Centrality can be measured in different ways (i.e., degree centrality, betweenness centrality, closeness centrality, and eigenvector centrality) and each measure provides a different interpretation. Note: Centrality is different than centralization. Centralization describes the extent a network is influenced by one or several vertices.	
	Degree Centrality	The number of connections a vertex contains with other vertices within the network. In directed graphs, two additional degree measurements are calculated to understand the direction of degree (refer to Section 2 Term Directed Graphs-endpoints). <ol style="list-style-type: none"> Outdegree: the number of connections a vertex exports/sends to other vertices within the network. Indegree: the number of connections a vertex imports/receives from other vertices within the network. Both outdegree and indegree measures sum to equal the degree for each vertex.	The degree, outdegree, and indegree centrality outputs range from 0 to infinity. Degree Centrality R Output [1] 3 4 2 1 Degree centrality is commonly presented with the range, mean or median, and mode. For example, the degree ranges from 1 to 4 (mean=2.5; median=2.5). In this scenario, there is no mode. Additionally, you can identify which vertex contains the most amount of connections within the network. In this scenario, Vertex 2 has the highest degree (4). 1. Outdegree R Output [1] 1 2 1 1 Outdegree is presented with the range, mean or median, and mode. The outdegree R output ranges from 1 to 2 (mean=1.25; median=1; mode=1). Vertex 2 exports/sends the most connections to other vertices within the network. 2. Indegree R Output [1] 2 2 1 0

			<p>Indegree is presented with the range, mean or median, and mode. The indegree R output ranges from 0 to 2 (mean=1.25; median=1.5; mode=2). Vertex 1 and Vertex 2 receive more connections than other vertices within the network.</p>
	Betweenness Centrality	<p>The number of instances a vertex serves as a connector between dyad(s) (e.g., a vertex that connects two vertices; the shortest path [geodesic] between the dyad[s]).</p>	<p>Betweenness centrality output ranges from 0 to infinity.</p> <p>Betweenness Centrality R Output</p> <p>[1] 2 3 0 0</p> <p>The R output indicates Vertex 1 acts as a connector to 2 dyads within the network; Vertex 2 is a connector to 3 dyads within the network. Vertex 3 and Vertex 4 are not connectors in the network.</p>
	Closeness Centrality	<p>The distance from one vertex to another vertex. Closeness is calculated in terms of shortest distance (e.g., the speed information can be sent or received from vertex to vertex). Closeness is calculated when graphs are not strongly connected, but will equal 0 for any vertex that lacks a path to another vertex. Due to this caveat, closeness in networks with isolates are often not presented.⁴</p>	<p>Closeness centrality ranges from 0 to 1.</p> <p>Closeness Centrality R Output</p> <p>[1] 0.0 0.0 0.0 0.5</p> <p>The closeness centrality output indicates Vertex 1, Vertex 2, and Vertex 3 lack a path to all vertices in the network. Vertex 4 has a closeness of 0.5, indicating its average distance is smaller than all other vertices in the network.</p>
	Eigenvector Centrality	<p>Measures how closely connected one vertex is to another vertex with a strong degree (e.g., vertex with the majority of connections in the network).</p>	<p>Eigenvector centrality ranges from 0 to 1.</p> <p>Eigenvector Centrality R Output</p> <p>[1] 0.3904 0.4792</p> <p>[3] 0.7809 0.0887</p> <p>Vertex 3 is the strongest connected vertex in the network to the vertex with the most degree.</p>

Note: [1] denotes the calculation output for the first vertex. When there are multiple lines of data being output, the second line will have [xx] with the number referring to the first vertex of the second line of output. This will be discussed with more detail in Section 2. £ denotes there are several calculations that can be conducted at the subgraph level, but are more on the intermediate level of SNAs. For the purpose of this toolkit, the subgraph section provides a basic understanding, but does not go in-depth in analysis and interpretations. ⁴Definitions were created from references 2-4, 6-13.

5. Sociogram Analysis

In this section, we will assess the network using terms and commands covered in [Sections 2 and 4](#). The coding and commands demonstrated will be shown with both the weighted adjacency matrix (sna_demo) and weighted edge list (sna_edgelistdemo) created in [Sections 3.1 and 3.2](#). The “sna_demo” dataset code will appear first and will be followed by the “sna_edgelistdemo” code.

5.1 Network Level Indices

Step 1. Calculate the network’s density. Click in the R console, type the following code, and press “ENTER:”

```
> gden(sna_demo.n, ignore.eval=FALSE, diag=TRUE)
> gden(sna_edgelistdemo.n, ignore.eval=FALSE, diag=TRUE)
```

R will output:

```
[1] 0.52
```

Interpretation: The CHDs represented in our network are moderately connected. Most CHDs in the network share a connection/relationship. Recall that density measures the network’s cohesion and is valued between 0 and 1.

Note: The command ignore.eval=FALSE instructs R to not ignore the edge values; diag=TRUE commands R to incorporate loops in the calculation. If you do not prompt R to incorporate edge values or loops, R will default no edge values and no loops. If you are working with an undirected network, you do not need to prompt R with “ignore.eval.” Additionally, if you are not working with loops in your network, you do not need to prompt R with “diag=TRUE.”

Step 2. Next, we will calculate the network’s reciprocity to demonstrate how to calculate if our dataset was undirected. Our data are directed; therefore, edgewise reciprocity (Step 3) is the appropriate calculation in our scenario. Click in the R console, type the following code, and press “ENTER:”

```
> grecip(sna_demo.n)
> grecip(sna_edgelistdemo.n)
```

R will output:

```
Mut
0.8
```

Interpretation: There is an 80% tendency a CHD will return a relationship to another CHD.

Step 3. Calculate the edgewise reciprocity. Click in the R console, type the following code, and press “ENTER:”

```
> grecip(sna_demo.n, measure="edgewise")
> grecip(sna_edgelistdemo.n, measure="edgewise")
```

R will output:

```
Mut
0.8333333
```

Interpretation: There is an 83.3% tendency a CHD will reciprocate a relationship with another CHD.

Step 4. Now we will calculate the network's diameter. Click in the R console, type the following code and press "ENTER:" > `geodist(snademo.n)`
> `geodist(snaedgedemo.n)`

R will output:

```

$counts
      [,1] [,2] [,3] [,4] [,5]
Duval  —> [1,]   1   1   1   1   1
Okaloosa —> [2,]   0   1   1   1   1
Orange  —> [3,]   0   1   1   1   1
Palm Beach —> [4,]   0   1   2   1   1
Pinellas —> [5,]   0   1   1   1   1

$gdist
      [,1] [,2] [,3] [,4] [,5]
Duval  —> [1,]   0   2   1   2   2
Okaloosa —> [2,] Inf   0   1   1   1
Orange  —> [3,] Inf   1   0   1   1
Palm Beach —> [4,] Inf   1   2   0   1
Pinellas —> [5,] Inf   1   1   1   0

```

Interpretation: The diameter (longest geodesic) in the network is 2.

Note: R outputs \$counts and \$gdist. The actual diameter calculation is displayed under the \$gdist. \$counts refers to the raw number it takes to go from one vertex to the next. Recall that the diameter output will be valued "Inf" where geodesics do not exist. Additionally, [1,]-[5,] represent the vertices in the network. When there are multiple lines of output, R will start the line with [x,], where "x" represents the vertex that starts the line of output. The format of the output should remind you of the adjacency matrix we created in Section 2.1.

Step 5. Calculate the network's centralization (i.e., degree, betweenness, and closeness). Click in the R console and type the following codes; press "ENTER" after each code:

Centralization by Degree code:

```

> centralization(snademo.n, degree, ignore.eval=FALSE, diag=TRUE)
> centralization(snaedgedemo.n, degree, ignore.eval=FALSE, diag=TRUE)

```

R's degree output:

```
[1] 0.36
```

Centralization by Betweenness code:

```

> centralization(snademo.n, betweenness)
> centralization(snaedgedemo.n, betweenness)

```

R's betweenness output:

```
[1] 0.2291667
```

Centralization by Closeness code:

```

> centralization(snademo.n, closeness)
> centralization(snaedgedemo.n, closeness)

```

R's closeness output:

```
[1] 0.7142857
```

Network Centralization Interpretation: Centralization by degree and betweenness indicates the network is sparse (degree=0.36; betweenness=0.23). Thus, the network is not influenced by one CHD in the network. Notice the centralization by closeness output is closer to 1. The network centralization by closeness is slightly centralized by one or a few CHDs that are closer (e.g., can transfer information faster) than others in the network.

5.2 Vertex/Node Level Indices

Step 1. Calculate degree for each vertex within the network. Click in the R console, type the following code, and press “ENTER:”

```
> degree(snademo.n, ignore.eval=FALSE, diag=TRUE)
> degree(snaedgedemo.n, ignore.eval=FALSE, diag=TRUE)
```

R will output:

```
[1] 1 6 7 5 6
    ↓ ↓ ↓ ↓ ↓
    Duval Orange Pinellas
      ↓   ↓
    Okaloosa Palm
           Beach
```

Explanation: Notice the R output for vertex/node level indices correspond to the row and column order displayed in the adjacency matrix/imported dataset. Recall that the order of the five counties was Duval, Okaloosa, Orange, Palm Beach, and Pinellas. Therefore, the degree output for Duval County (vertex 1) indicates it has one total connection. Okaloosa County (vertex 2) and Pinellas County (vertex 5) have six total connections. Orange County (vertex 3) has 7 connections and Palm Beach County (vertex 4) has five total connections.

Interpretation: The CHDs total degree ranges from 1 to 7 (median=6, mode=6).

Step 2. Next, since we have a directed network, we will calculate outdegree (sent connections). Click in the R console, type the following code, and press “ENTER:”

```
> degree(snademo.n, cmode="outdegree", ignore.eval=FALSE, diag=TRUE)
> degree(snaedgedemo.n, cmode="outdegree", ignore.eval=FALSE,
diag=TRUE)
```

R will output:

```
[1] 1 3 4 2 3
    ↓ ↓ ↓ ↓ ↓
    Duval Orange Pinellas
      ↓   ↓
    Okaloosa Palm
           Beach
```

Interpretation: The CHDs outdegree ranges from 1 to 4 (median=3, mode=3). Orange CHD has the greatest outdegree, but Okaloosa and Pinellas CHDs possess nearly the same amount of outdegrees in the network.

Step 3. Now we will calculate indegree (received connections). Click in the R console, type the following code, and press “ENTER:”

```
> degree(snademo.n, cmode="indegree", ignore.eval=FALSE, diag=TRUE)
> degree(snaedgedemo.n, cmode="indegree", ignore.eval=FALSE,
diag=TRUE)
```

R will output:

```
[1] 0 3 4 3 3
     ↓ ↓ ↓ ↓ ↓
   Duval Orange Pinellas
       ↓ ↓
   Okaloosa Palm
       Beach
```

Interpretation: CHD indegree ranges from 0 to 4 (median=3, mode=3). Orange CHD has the greatest indegree within the network.

Step 4. Calculate betweenness. Click in the R console, type the following code, and press “ENTER:” `> betweenness(snademo.n)` `> betweenness(snaedgedemo.n)`

R will output:

```
[1] 0.0 0.5 3.0 0.0 0.5
     ↓ ↓ ↓ ↓ ↓
   Duval Okaloosa Orange Palm Pinellas
       Beach
```

Interpretation: CHD betweenness ranges from 0 to 3 (median=1.7, mode=0). Orange County acts as the primary connector in the network.

Step 5. Calculate closeness. Click in the R console, type the following code, and press “ENTER:” `> closeness(snademo.n)` `> closeness(snaedgedemo.n)`

R will output:

```
[1] 0.5714286 0.0000000 0.0000000 0.0000000 0.0000000
     ↓ ↓ ↓ ↓ ↓
   Duval Okaloosa Orange Palm Beach Pinellas
```

Interpretation: Duval CHD is the only vertex that can reach every vertex. Refer to the network sociogram to further understand the direction of connections. Because we have a directed graph, Duval CHD sends the relationship to Orange CHD, therefore can reach all other vertices in the network. Since Orange CHD only receives the relationship, the other CHDs, including Orange CHD, do not have a pathway to connect to Duval CHD. The lack of connection to Duval CHD is represented by “0” in the output.

Step 6. Calculate eigenvector. Click in the R console, type the following code, and press “ENTER:” `> evcent(snademo.n, ignore.eval=FALSE, diag=TRUE)`
`> evcent(snaedgedemo.n, ignore.eval=FALSE, diag=TRUE)`

R will output:

```
[1] 0.2114066 0.4756648 0.6342197 0.3171099 0.4756648
     ↓ ↓ ↓ ↓ ↓
   Duval Okaloosa Orange Palm Beach Pinellas
```

Interpretation: From Step 1 in [Section 5.2](#), we know Orange CHD contains the highest degree in the network. Therefore, it is no surprise that the eigenvector centrality depicts

Orange CHD as the strongest connected in the network. Although Orange CHD has the greatest eigenvector centrality, Okaloosa and Pinellas CHDs also contribute substantially to the network. Note in larger networks, the vertex with the highest degree centrality does not always indicate the node will have the highest eigenvector centrality.

6. Helpful Resources

6.1 YouTube Resources

- Cook J. Importing social network data into r through csv files [video]. YouTube. www.youtube.com/watch?v=3wxEC0svnJc&t=872s. Published October 10, 2016. Accessed February 2, 2018.
- Cook J. Introduction to exploring social network structure with visualization in r [video]. YouTube. www.youtube.com/watch?v=N_mMXvNPE8Y&t=1227s. Published October 10, 2016. Accessed February 2, 2018.
- Cook J. Introduction: r and igraph for edge lists and social network graphs [video]. YouTube. www.youtube.com/watch?v=isBm5RTslow&t=5s. Published September 25, 2016. Accessed February 2, 2018.
- McCulloch J. Betweenness centrality [video]. YouTube. www.youtube.com/watch?v=0CCrq62TF7U&t=638s. Published September 10, 2017. Accessed February 2, 2018.
- McCulloch J. Centralization [video]. YouTube. www.youtube.com/watch?v=-ANEqyrJOac. Published October 25, 2017. Accessed February 2, 2018.
- McCulloch J. Degree centrality [video]. YouTube. www.youtube.com/watch?v=iiVeQklELyc&t=419s. Published September 10, 2017. Accessed February 2, 2018.
- McCulloch J. Graph measures [video]. YouTube. www.youtube.com/watch?v=N3wbYrsJrbk&t=230s. Published October 25, 2017. Accessed February 2, 2018.
- McCulloch J. Mod02D closeness [video]. YouTube. www.youtube.com/watch?v=2ELP6yd21tw. Published September 10, 2017. Accessed February 2, 2018.
- McCulloch J. Mod02A terminology [video]. YouTube. www.youtube.com/watch?v=5Hw1OmWOLA8. Published September 10, 2017. Accessed February 2, 2018.

6.2 Text Resources

- Fraizer M. R color cheatsheet. www.nceas.ucsb.edu/~frazier/RSpatialGuides/colorPaletteCheatsheet.pdf. Accessed February 2, 2018.
- McMillan C. Big Data Social Science Blog [Internet]. Penn State University. 2016 Dec- [cited 2018 February 2]. Available from: <https://sites.psu.edu/bdssblog/2016/12/05/introduction-to-network-analysis-in-r/>.
- Ognyanova K. Network visualization with r. <http://kateto.net/network-visualization>. Accessed February 2, 2018.
- Acton R., Butts C., Goodreau S.M. An introduction to network analysis with r and statnet. Workshop presented at: Sunbelt XXXII Workshop Series; March 13, 2012. Available from: <https://statnet.org/trac/raw->

[attachment/wiki/Resources/introToSNAinR_sunbelt_2012_tutorial.pdf](#). Accessed February 2, 2018.

- Shizuka D. Shizuka Lab School of Biological Science University of Nebraska-Lincoln. Plotting networks: weighted edges. www.shizukalab.com/toolkits. Published February 9, 2013. Accessed February 2, 2018.

7. Toolkit Evaluation Survey

We created a survey using Survey Monkey® to evaluate the usefulness of this toolkit: https://www.surveymonkey.com/r/SNA_Toolkit_Survey. The survey does not collect any personal information and will close on September 28, 2018 at 5:00 p.m. EST. We would really appreciate your input and feedback to assist us in developing future editions.

8. Acknowledgements

This toolkit was supported by grant 2017-121401 from the National Association of County and City Health Officials (NACCHO). We would like to thank the following individuals and divisions for their input and guidance in creating this toolkit: Taylor Campion, Alvina Chu, Karen Elliott, Sarah Matthews, Dr. Rachel Slayton, and the team from the Centers for Disease Control and Prevention Division of Healthcare Quality Promotion.

9. References

1. Luke D, Harris J. Network Analysis in Public Health: History, Methods, and Applications. ANNUAL REVIEW OF PUBLIC HEALTH [serial online]. 2007:69. Available from: British Library Document Supply Centre Inside Serials & Conference Proceedings, Ipswich, MA. Accessed July 21, 2017.
2. R Core Team [computer program]. Version 3.3.2. Vienna, Austria: R Foundation for Statistical Computing. www.R-project.org/. Published 2016. Accessed February 2, 2018.
3. RStudio Team [computer program]. Version 1.1.383. Boston, MA: RStudio: Integrated Development for R. www.rstudio.com/. Published 2016. Accessed February 2, 2018.
4. Handcock M, Hunter D, Butts C, Goodreau S, Krivitsky P, Bender-deMoll S, Morris M. Statnet: software tools for the statistical analysis of network data. The Statnet Project. www.statnet.org. Published 2016. Accessed February 2, 2018.
5. Acton R., Butts C., Goodreau S.M. An introduction to network analysis with r and statnet. Workshop presented at: Sunbelt XXXII Workshop Series; March 13, 2012. Available from: https://statnet.org/trac/raw-attachment/wiki/Resources/introToSNAinR_sunbelt_2012_tutorial.pdf. Accessed February 2, 2018.
6. Butts C, Social network analysis with sna. Journal of Statistical Software. 2008; 24(6). Accessed February 2, 2018.
7. Petrescu-Prahova M, Belza B, Leith K, et al. Using social network analysis to assess mentorship and collaboration in a public health network. Preventing Chronic Disease. 2015:12.
8. McCulloch J. Betweenness centrality [video]. YouTube. www.youtube.com/watch?v=0CCrq62TF7U&t=638s. Published September 10, 2017. Accessed February 2, 2018.

9. McCulloch J. Centralization [video]. YouTube. www.youtube.com/watch?v=-ANEqyrJOac. Published October 25, 2017. Accessed February 2, 2018.
10. McCulloch J. Degree centrality [video]. YouTube. www.youtube.com/watch?v=iiVeQkIELyc&t=419s. Published September 10, 2017. Accessed February 2, 2018.
11. McCulloch J. Graph measures [video]. YouTube. www.youtube.com/watch?v=N3wbYrsJrbk&t=230s. Published October 25, 2017. Accessed February 2, 2018.
12. McCulloch J. Mod02D closeness [video]. YouTube. www.youtube.com/watch?v=2ELP6yd21tw. Published September 10, 2017. Accessed February 2, 2018.
13. McCulloch J. Mod02A terminology [video]. YouTube. www.youtube.com/watch?v=5Hw1OmWOLA8. Published September 10, 2017. Accessed February 2, 2018.
14. Clarkson M. A guide to the gplot function of the sna library for r. www.melissaclarkson.com/resources/R_guides/. Accessed February 2, 2018.