Module 3: Visualization RStudio

Goals for Module 3: Visualization RStudio

~Participants will be introduced to various plotting options in igraph and learn how to customize network images.

Lab 3 Files

The `Lab 3 Files` folder contains all of the files needed to complete the Module 3: Visualization RStudio. There are four files in this folder: three csv data files that end and one script file. Participants are familiar with all three csv files, which are the same files we used in the Module 2: Data RStudio lab.

The four files included in the `Lab 3 Files` folder are:

`edgelist.csv`

`attributes.csv`

`event.csv`

`visualization.R`

If you need a refresher of what these files look like outside of RStudio, you can view the csv files in either a spreadsheet software program or a simple text-editing program, and you can view the visualization script file in a text editing program.

Visualization.R in RStudio

A. Open RStudio on your computer.

B. Set the working directory to the **`Lab 3 Files`** folder.

1. Select the Session menu located toward the middle of the menu bar at the very top of the RStudio program
2. Select Set Working Directory
3. Select Choose Directory
4. Locate the **`Lab 3 Files`** folder on your computer
5. Click open

C. Open the **`visualization.R`** script.

1. In RStudio click on the **`Open`** icon shaped like a folder located in the upper-left pane of the RStudio window
2. Check that the folder is **`Lab 3 Files`**
3. Select the **`visualization.R`** script
4. The **`visualization.R`** script should open in the top-left pane. It will look like a text file with the file name at the top

D. Complete Module 3: Visualization RStudio part of the lab by following the `visualization.R` script in RStudio. Once you have completed the script return to this document for the final review section of this Lab.
Tasks in the RStudio portion of Module 3: Visualization include:

Setting up RStudio Session
Importing Edgelist and Attribute Data
Setting Up the Social Network
Basic Plot Command
Plot Layout
Adjusting Vertex (Node) Properties
Adjusting Edge Properties
Fixing Node Position
Adjusting the Plotting Region
Saving Work
Two-Mode Data & Two-Mode Network
Plotting Two-Mode Networks
One-Mode Projections
Review

Review of Module 3: Visualization RStudio

Participants reviewed loading edgelist and attribute data and converting the data into a network. Participants were introduced to iterative additions to the basic `plot()` function in igraph. This included customizing the plot layout, customizing the vertex (node) properties, customizing the edge properties, fixing the nodes to certain positions, and adjusting the plotting region. Participants also learned several ways to save plots and scripts. Participants loaded two-mode network event data, generated a two-mode network, created variables for two-mode networks in RStudio, and plotted the two-mode network using the variables created. Participants created one-mode projections from the two-mode network and plotted each of those.

Below is a briefly annotated list of RStudio commands used in the Module 3: Visualization RStudio Lab:

```
dir()                                    # Check the contents of the working directory
library(igraph)                          # Load the igraph library
e<-read.csv("edgelist.csv",header=TRUE)  # Load the edgelist csv files
e                                        # Inspect object e
a<-read.csv("attributes.csv",header=TRUE) # Load the attribute files
a                                        # Inspect object a
g<-graph.data.frame(e,directed=FALSE,vertices=a) # Make a network
g                                        # Inspect network g
V(g)$name                                # Inspect the node attribute "name" in network g
V(g)$gender                              # Inspect the node attribute "gender" in network g
V(g)$gpa                                 # Inspect the node attribute "gpa" in network g
V(g)$age                                 # Inspect the node attribute "age" in network g
V(g)$color<-"blue"                       # Generate a new node attribute called "color" in network g
                                         and assign every node the color blue
V(g)$color[V(g)$gender=="F"]<-"red"      # Replace the blue color with red for all of the nodes with
                                         gender equal to female
E(g)$weight                              # Inspect the edge attribute "weight" in network g
plot(g)                                  # Produce the basic plot of network g
?igraph.plotting                         # Access RStudio help on plots in igraph

pdf("best_network_ever.pdf")             # Open a new graph area and save its contents to a pdf
par(mar=(c(5, 0, 3, 0)),bg="black")      # Set the margins and the background color
set.seed(12345)                          # Set a seed to reuse in order to fix nodes into one position
plot(g,                                  # Start the plot command for network g
layout=layout.kamada.kawai,              # Select a layout option
vertex.size=40,                          # Customize the size of the nodes
vertex.label=V(g)$gpa,                   # Assign the gpa variable as labels for the nodes
vertex.shape="square",                   # Set the shape of the nodes
vertex.color=V(g)$color,                 # Assign the color variable to vary the color of the nodes
vertex.label.color="white",              # Pick a color for the text of the labels
edge.color=E(g)$weight+1,                # Set the color of the lines to vary by the edge weights variable
edge.width=E(g)$weight*3,                # Set the width of the lines to vary by the edge weights variable
edge.lty=E(g)$weight)                    # Set the type of the lines to vary by the edge weights variable
title(main="The Best Network Ever",      # Add a title to the top of the graph area
sub="By A.V. Papachristos",              # Add a subtitle to the bottom of the graph area
col.main="white",                        # Pick a color for the text of the main title
col.sub="lightgrey")                     # Pick a color for the text of the subtitle
legend("topleft",                        # Add a legend and locate it in the top left of the graph area
legend=c("male", "female"),              # Use this text in the legend
col=c("blue", "red"),                    # Assign these two colors to the legend to match the nodes
```

```
pch=15,                                  # Select a plotting symbol to use in the legend
text.col="white")                        # Select a color for text in the legend
dev.off()                                # Save and close this graph area


t <- read.csv("event.csv", header=TRUE, colClasses="character")
                                         # Load the event.csv file containing the two-mode network
t                                        # Inspect t
person <- unique(t$person)               # Make a unique list of all the persons in t
event <- unique(t$event)                 # Make a unique list of the events in t
tf.list <- data.frame(name=c(event, person), type=c(rep(FALSE, length(event)),
rep(TRUE, length(person))), stringsAsFactors=FALSE)
                                 # Stack the two unique lists of persons and events and assign them true and false
two.network <- graph.data.frame(t, directed=FALSE, vertices=tf.list)
                                         # Convert t into a two-mode network
two.network                              # Inspect the network
par(mar=c(c(5, 4, 4, 2) + 0.1), bg="white") # Return the plotting background to default settings
plot(two.network)                        # Run the basic default plot of the two-mode network

V(two.network)$new.label <- c("arrest 1", "arrest 2", "arrest 3", "arrest 4",
"arrest 5", "Willard", "Anna", "Rema", "Rex", "Benny", "Chris", "Jen",
"Ken")                                   # Create a new variable within the two-mode network for labels
V(two.network)$shape <- c("square", "square", "square", "square", "square",
"circle", "circle", "circle", "circle", "circle", "circle", "circle",
"circle")                                # Create a variable within the two-mode network for shapes
V(two.network)$color <- c("cornflowerblue", "cornflowerblue",
"cornflowerblue", "cornflowerblue", "cornflowerblue", "olivedrab3",
"olivedrab3", "olivedrab3", "olivedrab3", "olivedrab3", "olivedrab3",
"olivedrab3", "olivedrab3")              # Create a new variable within the two-mode network for colors
V(two.network)$size <- c(30,30,30,30,30,20,20,20,20,20,20,20,20)
                                         # Create a new variable within the two-mode network for size

plot(two.network,                        # Plot the two-mode network
vertex.label=V(two.network)$new.label,   # Set each node name to the label variable created above
vertex.label.color="black",              # Set all label colors to black
vertex.label.cex=.75,                    # Set all label sizes to 0.75 times the default
vertex.shape=V(two.network)$shape,       # Set each node shape to the shape variable created above
vertex.color=V(two.network)$color,       # Set each node color to the color variable created above
vertex.size=V(two.network)$size,         # Set each node size to the size variable created above
edge.width=4,                            # Set the edge width to 4 times the default
edge.color="black")                      # Set the edge colors to black

p <- bipartite.projection(two.network) # Create the two one-mode projections
p                                        # Inspect the new object p
event.network <- p[[1]]                  # Create a network for just the one-mode events
person.network <- p[[2]]                 # Create a network for just the one-mode persons
event.network                            # Inspect the event network
person.network                           # Inspect the person network
V(event.network)$new.label               # Inspect the labels in the event network
V(event.network)$shape                   # Inspect the shapes in the event network
V(event.network)$color                   # Inspect the colors in the event network
V(event.network)$size                    # Inspect the size in the event network
V(person.network)$new.label              # Inspect the labels in the person network
V(person.network)$shape                  # Inspect the shapes in the person network
V(person.network)$color                  # Inspect the colors in the person network
V(person.network)$size                   # Inspect the size in the person network
```

```
plot(event.network, vertex.label=V(event.network)$new.label,
vertex.label.color="black", vertex.label.cex=.75,
vertex.shape=V(event.network)$shape, vertex.color=V(event.network)$color,
vertex.size=V(event.network)$size, edge.width=4, edge.color="black")
                                        # Plot the event network

plot(person.network, vertex.label=V(person.network)$new.label,
vertex.label.color="black", vertex.label.cex=.75,
vertex.shape=V(person.network)$shape, vertex.color=V(person.network)$color,
vertex.size=V(person.network)$size, edge.width=4, edge.color="black")
                                        # Plot the person network
```

Module 4: Analytics RStudio introduces participants to some of the basic descriptive statistics possible with igraph in RStudio.