visNetwork R Package Review

What is visNetwork?

- visNetwork is an R package, which interfaces with the vis.js JavaScript library to enable interactive network visualization

What skill level / background is needed?

- Outside of a foundational understanding of R, no additional background is needed to create a visualization of a basic network graph. Once you work through a couple of examples to get a feel for the package, you will be up and running fairly quickly, as you can create a graph in one line of code. Please find helpful links below in the *Resources* section of this review document to get started. As seen in these examples, you will need to create two dataframes prior to visualization, one for the nodes and one for the edges of your network graph. Thus, depending on your data set, be sure to allocate the appropriate time for creating these dataframes.

What does visNetwork do well?

- Presentation
  - The out-of-the-box network graphs you can create using this package are visually appealing and can be used in presentations, especially if you are working with a small graph. Otherwise you may run into some performance issues (see below). You can also drag any node to re-position the graph on the page, as well as create a hierarchical layout of the network if you so choose. The package also allows you to integrate with Shiny and igraph to control the deployment or look of your network graph.
- Customize / Flexible
  - You can customize almost anything. There are countless features you can change, from the ability to control, as groups or individually, the color, shape, size, and labels of both the nodes and edges, to more advanced features such as using custom images to display the nodes.
- Interactive
  - There is a nice set of built-in functionality that allows for exploratory data analysis. For example, you can set up descriptive hovers over the nodes and edges. There are functions that allow you to highlight a node's neighbors by clicking on that node or by creating a drop-down menu to scroll to a particular node of interest. You can also create a similar drop-down menu to scroll to a specific group of the network you would like to analyze. Additional functionality includes the ability to create buttons to zoom in or zoom out of the graph and the ability to add and delete nodes after the visualization is created. These are just a few of the many interesting functions you can play around with.

What are the drawbacks of visNetwork?

- Performance
  - From my experience, if you are working with any data set that has more than a few hundred edges, the runtime will be slow. This made it difficult to see the graph and utilize any of the interactive features of the package. I thus recommend this package as a best option for the presentation of small networks.
- Computations
  - This package is focused on visualizing networks rather than assist with network computations, which we saw in-class was possible using the Gephi software. As a result, I wouldn't recommend this package for any detailed computational work you may want to do.

Resources:

- Tutorial (with code): http://datastorm-open.github.io/visNetwork/
- Examples (without code): http://visjs.org/network_examples.html
- Documentation: http://visjs.org/docs/network/