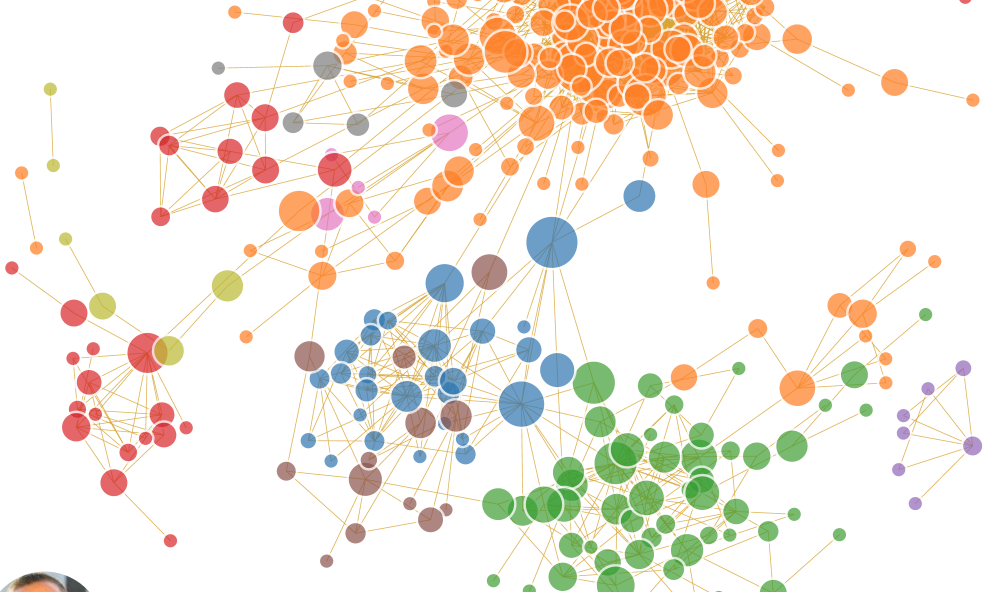


Michael Jules
Data Visualization Developer

How To Visualize Your LinkedIn Social Network

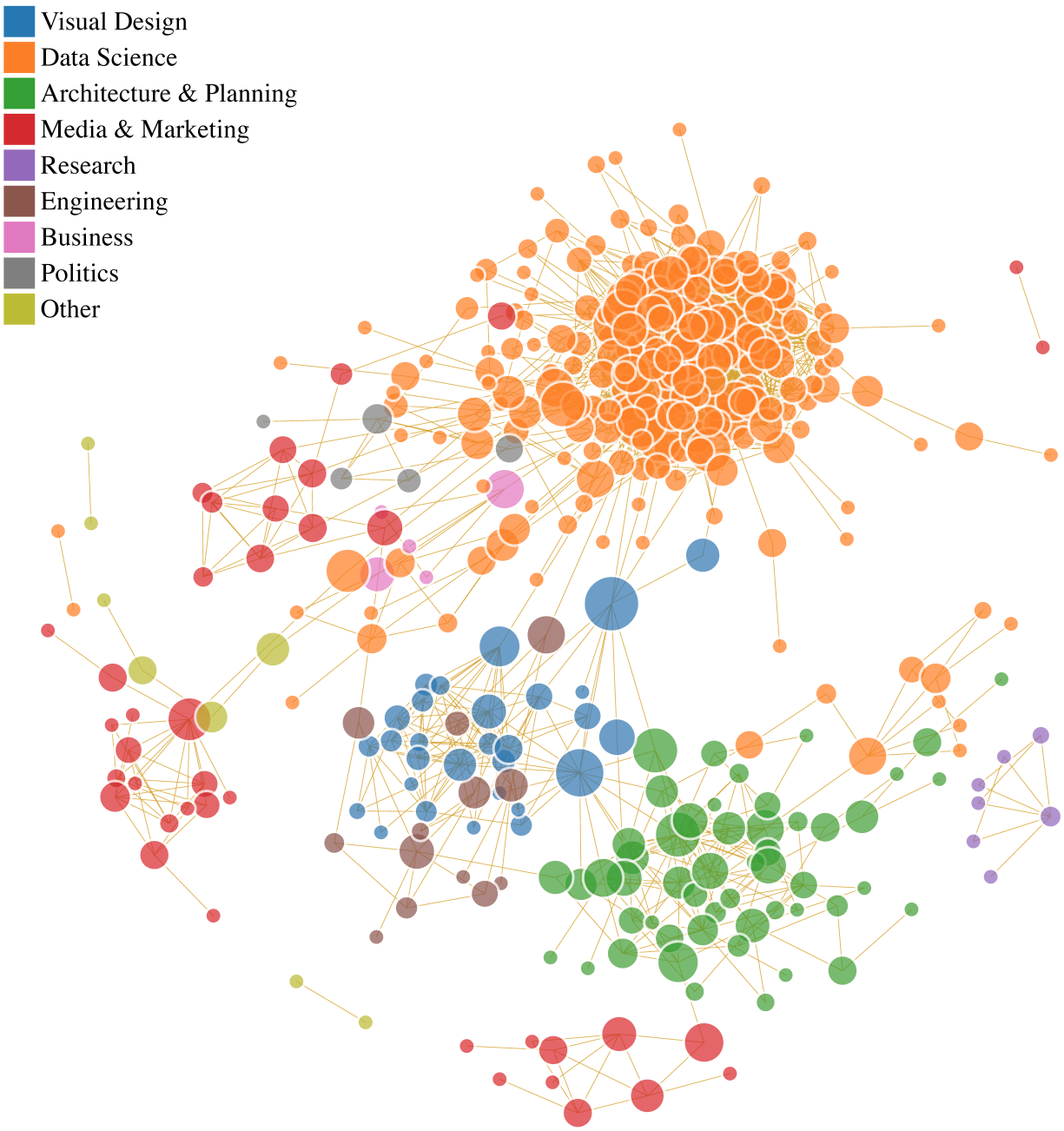
Overview. A social network characterizes the structure of society by describing the relationships and interactions that *connect* people. This post illustrates how to create an interactive data visualization of your LinkedIn professional-social network using R and D3. In this post, I'll focus on the science behind social networks, while corresponding replication materials are appended below and housed on my Github repository.

Motivation. For decades statisticians have worked under the assumption that individuals behave independently. In reality, we know the world is not like this - human behavior is influenced by one's social milieu. Recent advances in computing power now afford data scientists the opportunity to visualize the structure of social behavior as it exists in our connected world. Data visualization can be a powerful tool for exploring complex networks, however, care should be taken when displaying public-facing network graphs. The biggest challenge with network visualizations is conveying complex information simply and clearly so that your audience internalizes the *substantive* takeaway point of the graph. Below are are three practical applications of how we can use data visualization and social network analysis to learn about how we are connected.

I. Identifying Communities in Networks

In statistical analysis of networks, clusters are used to identify communities within large groups of people. Take Figure 1, which displays people I'm directly connected to on LinkedIn with ties representing our mutual connections. Each person is a node (dot) and I use color to represent a person's professional community membership in the network.

Figure 1: **My LinkedIn Social Network Graph**



Note: Graph displays a social network of 387 LinkedIn users who are directly connected to me. Node color represents community membership in network. The size of the node corresponds to a person's network influence. Isolate nodes removed for visualization (n=112). *Data Source:* LinkedIn. Data scraped from http://socilab.com/ Replication materials and code on Github: github.com/michaeljules/social-network-viz . Interactive D3 visualization can found on blocks: http://bl.ocks.org/michaeljules or my website: michaeljules.xyz

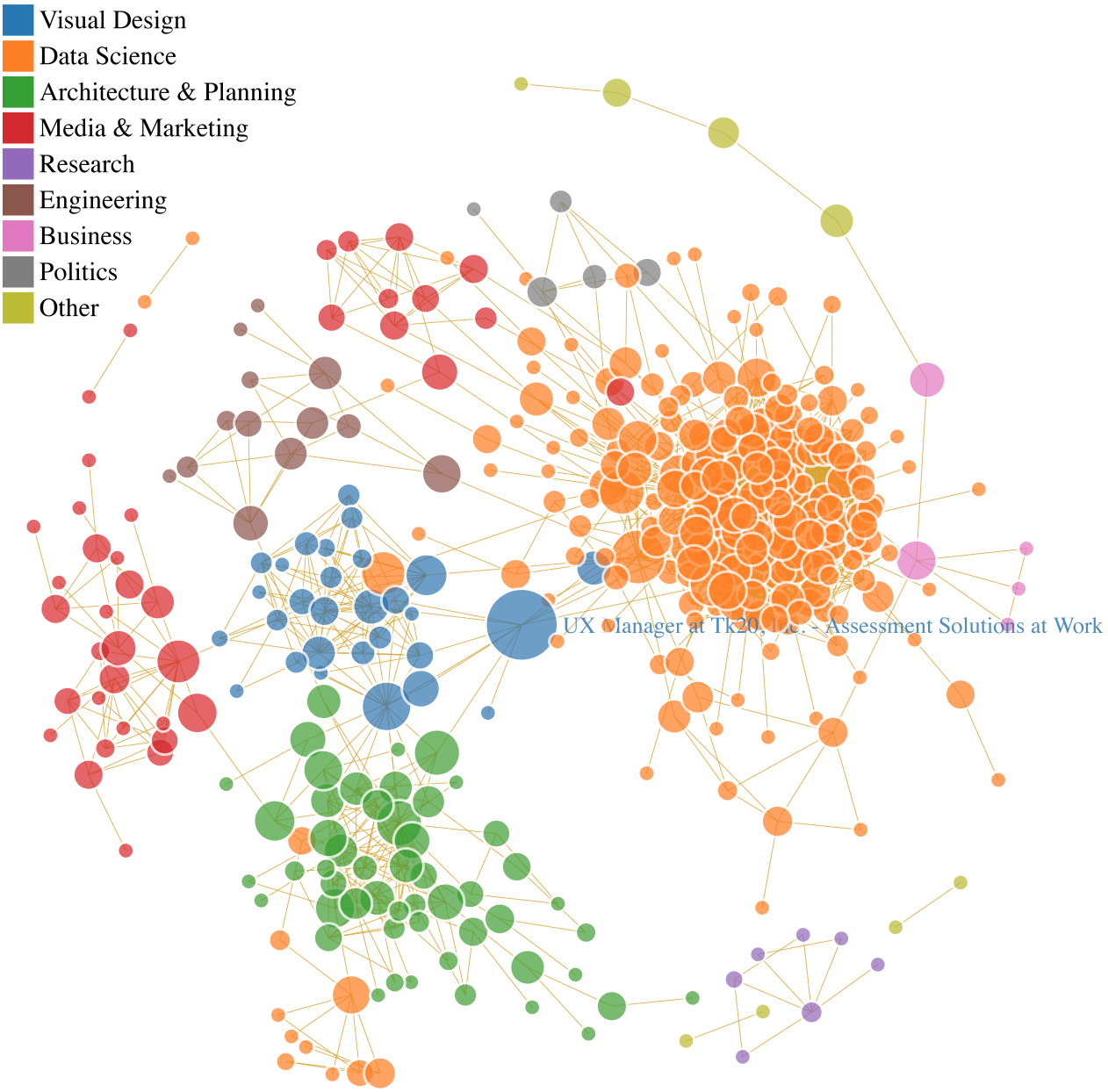
Think of communities as subgroups, tribes, social cliques, or neighborhoods within a city. In this example, communities were identified using the R function in the igraph library *cluster_label_prop*, which is a fast, nearly linear time algorithm for detecting community structure in networks. The function works by labeling the nodes with unique labels and then updating the labels by majority voting in the neighborhood of the nodes.

Visualizing your professional network is important to understand where and how your skills map onto sectors and industries outside your domain. Figure 1 shows how I am primarily connected to the data science community (shown with orange dots), yet I work at the intersection of data science and visual design (shown with blue dots). The next section explains how we can find people who are "bridges" to connect two disparate network communities and in the process make our own networks more diverse.

II. Identifying Influence in Networks

In Figure 2 I use node size to represent a person's influence in a network. To quantify a person's influence in a network we can use metrics of centrality. To calculate the size of each node -- I use the centrality measure *betweenness centrality* -- defined as the number of shortest paths from all nodes to all others that pass through that node. As Figure 2 shows, the large blue highlighted node in the middle of the network is a *bridge* -- a person connecting my data science world with the visual design world.

Figure 2: **Identifying Influence in Social Networks**



Note: Graph displays a social network of 387 LinkedIn users who are directly connected to me. Node color represents community membership in network. The size of the node corresponds to a person's network influence. Isolate nodes removed for visualization (n=112). *Data Source:* LinkedIn. Data scraped from http://socilab.com/ Replication materials and code on Github: github.com/michaeljules/social-network-viz . Interactive D3 visualization can found on blocks: http://bl.ocks.org/michaeljules or my website: michaeljules.xyz

*"Pay special attention to the **bridges** between network clusters, as they are the most likely to be influential (an alumni of your school that introduced you to your current employer)." -- Craig Tutterow*

III. The Future: Tracking Network Change Over Time

The most exciting innovations in network analysis are focused on dynamic networks: how networks change over time. Although, dynamic network analysis is new and still in the development phase, there are exciting advances being made. For example, the R package NDTV renders dynamic network data as movies and interactive animations to bring our social network data to life.

Concluding. I hope you found this exercise useful. You can contact me at michael@beautifuldataviz.com for any questions or suggestions you might have, or to report any bugs in the code. Replication materials, code, and additional resources below. The interactive D3 visualization of my Linkedin network can found on blocks or my website: michaeljules.xyz.

Supplemental Materials

\+

R CODE

This repository is a tutorial demonstrating how to use your LinkedIn data to visualize your professional-social network in R and D3.JS.

1. Install and Load Libraries

```
library("networkD3")
library("rjson")
library("jsonlite")
library("igraph")
```

2. Read in your LinkedIn Data

```
l <- fromJSON("https://raw.githubusercontent.com/michaeljules/social-network-viz/master/data/data.js",flatten=TRUE)
```

3. Translate Data to Adjacency Matrix then Network Object

```
networkData <- l$reducedMatrix
rownames(networkData) <- paste0(l$publicConnections$firstName[1:499]," ",l$publicConnections$lastName[1:499])
colnames(networkData) <- rownames(networkData)

g <- graph.adjacency(networkData,mode="undirected")
```

4. Add Node Attributes to the Network:

```
V(g)$industry <- l$publicConnections$industry[1:499]
V(g)$location <- l$publicConnections$location.name[1:499]
V(g)$names <- paste0(l$publicConnections$firstName[1:499]," ",l$publicConnections$lastName[1:499])
V(g)$click <- l$publicConnections$pictureUrl[1:499]
V(g)$country <- l$publicConnections$location.country.code[1:499]
V(g)$headline <- l$publicConnections$headline[1:499]
```

5. Calculate Network Centrality Scores For Each Node

```
b <- betweenness(g, v=V(g), directed = FALSE, weights = NULL,
                 nobigint = TRUE, normalized = FALSE)
# Take the Square Root of the betweenness measure so nodes are not too large
V(g)$betweenness <- sqrt(b)
```

6. Remove Isolates From Network

```
#identify isolated nodes
bad.vs <- V(g)[degree(g) == 0]

# remove isolated nodes
g <- delete.vertices(g, bad.vs)
```

7. Identify Network Communities

```
c3 <- cluster_label_prop(g)
# Note: cluster_label_prop is a fast, nearly linear time algorithm for detecting community structure in networks. In

# Assign community membership to network nodes
V(g)$c3 <- c3$membership

# Label Group Memberships -- These are self-identified network communites based on the function above.
c3$group <- c3$membership
c3$group[c3$membership==1] <- "Visual Design"
c3$group[c3$membership==2] <- "Data Science"
c3$group[c3$membership==3] <- "Architecture & Planning"
c3$group[c3$membership==4] <- "Media & Marketing"
c3$group[c3$membership==5] <- "Media & Marketing"
c3$group[c3$membership==6] <- "Research"
c3$group[c3$membership==7] <- "Engineering"
c3$group[c3$membership==8] <- "Media & Marketing"
c3$group[c3$membership==9] <- "Business"
c3$group[c3$membership==10] <- "Politics"
c3$group[c3$membership==11] <- "Media & Marketing"
c3$group[c3$membership==12] <- "Data Science"
c3$group[c3$membership==13] <- "Data Science"
c3$group[c3$membership==14] <- "Other"
c3$group[c3$membership==15] <- "Other"
c3$group[c3$membership==16] <- "Other"
c3$group[c3$membership==17] <- "Data Science"
c3$group[c3$membership==18] <- "Other"
c3$group[c3$membership==19] <- "Engineering"
c3$group[c3$membership==20] <- "Politics"
c3$group[c3$membership==21] <- "Media & Marketing"
c3$group[c3$membership==22] <- "Other"
c3$group[c3$membership==23] <- "Other"
```

8. Translate Network Graph to D3

```
library(networkD3)

g_d3 <- igraph_to_networkD3(g)

## Append Node Attributes
g_d3$nodes$location <- V(g)$location
g_d3$nodes$industry <- V(g)$industry
g_d3$nodes$click    <- V(g)$click
g_d3$nodes$betweenness  <- V(g)$betweenness
g_d3$nodes$country <- V(g)$country
g_d3$nodes$position <- V(g)$position
g_d3$nodes$group <- V(g)$neighborhood
g_d3$nodes$headline <- V(g)$headline
g_d3$nodes$cle <- V(g)$cle
g_d3$nodes$c3 <- V(g)$c3

net <- forceNetwork(Links=g_d3$links,
             Nodes=g_d3$nodes,
             width=700,
             height=700,
             NodeID = 'headline',
             Group  = 'c3',
             radiusCalculation = JS("Math.sqrt(d.nodesize)+5"),
             Nodesize = 'betweenness',
             charge = -30,
             linkWidth = .4,
             colourScale = JS("d3.scale.category10()"),
             linkColour = "goldenrod",
             zoom = FALSE,
             legend = TRUE,
             opacity = .85,
             opacityNoHover = 0,
             bounded = FALSE,
             fontFamily = "San Francisco",
             fontSize = 14)
```

Replication Materials:

1. Replication materials and code for this post can be found on Github: github.com/michaeljules/social-network-viz .
2. I scraped my LinkedIn data from Craig Tutterow's excellent open source resource: http://socilab.com/

Additional References:

- Christopher Gandrud, J.J. Allaire and Kenton Russell (2016). networkD3: D3 JavaScript Network Graphs from R. R package version 0.2.11.
- Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006.

- Skye Bender-deMoll (2016). ndtv: Network Dynamic Temporal Visualizations. R package version 0.10.0.