**NodeXL for Network analysis**
**Demo/hands-on at NICAR 2012, St Louis, Feb 24**

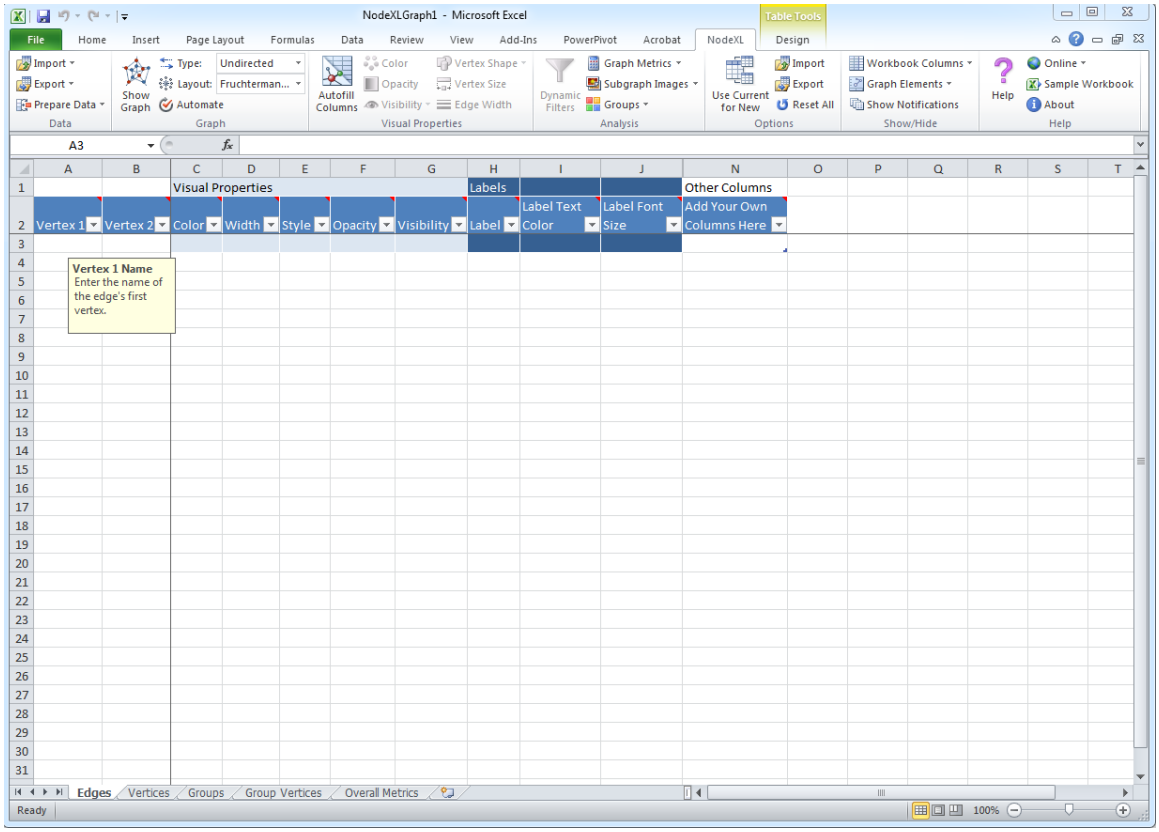
**Peter Aldhous, San Francisco Bureau Chief**

NewScientist

peter@peteraldhous.com

NodeXL is a template for Microsoft Excel 2007 and 2010, which makes network analysis easy and intuitive:

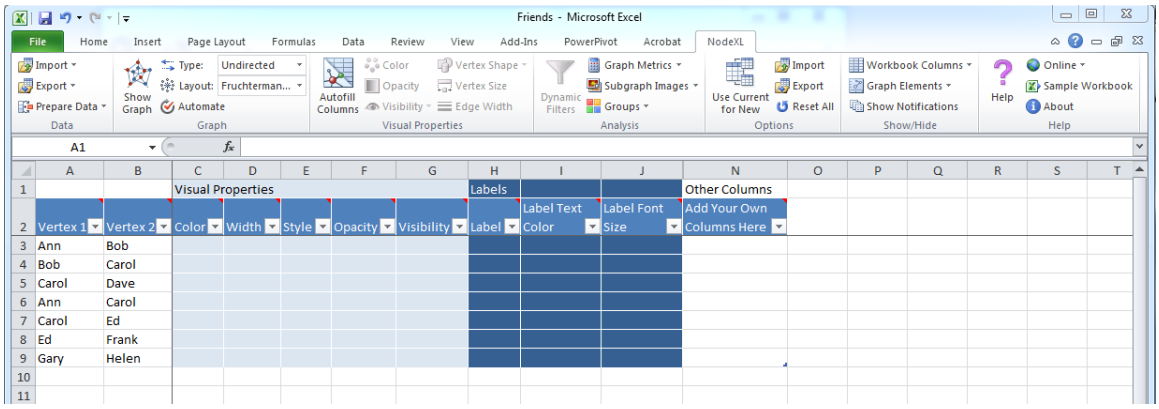
**Download** the template from NodeXL site, then open:




Notice that NodeXL has its own menu ribbon, and that the first worksheet is called **Edges**.

Network graphs show the connections, or **Edges**, between entities such as people or organizations. These entities are known as **Nodes** or **Vertices.**

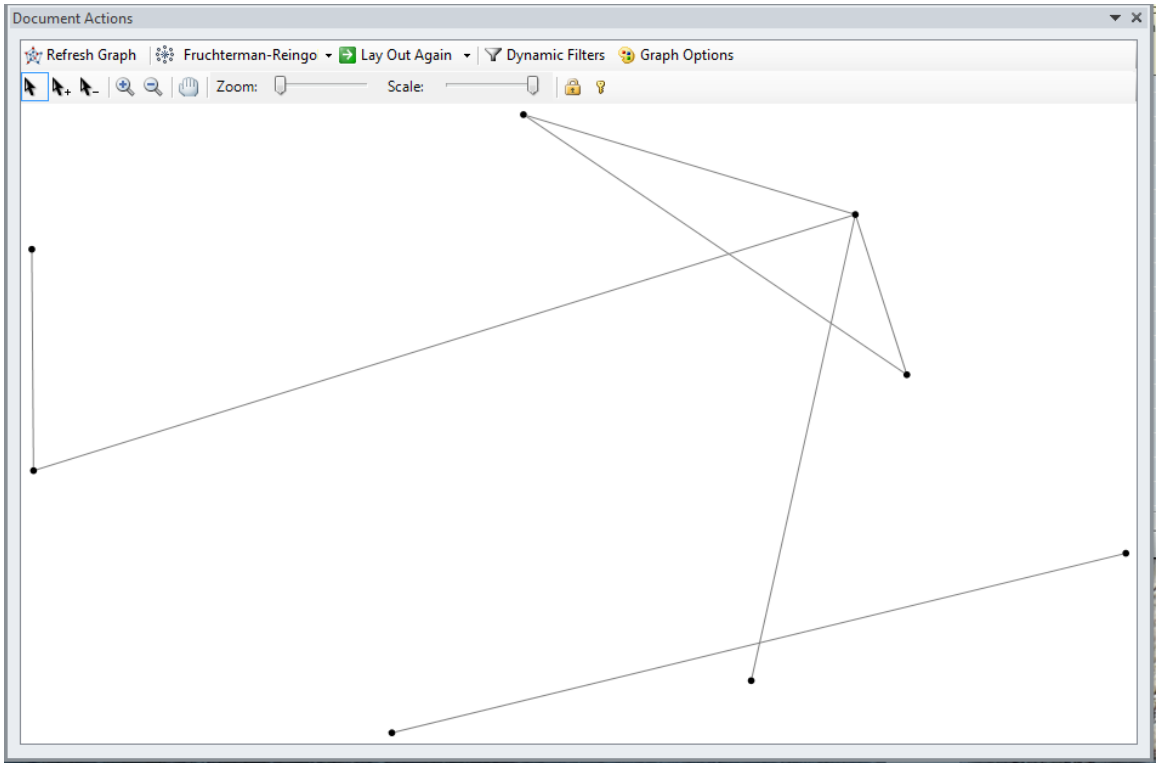
You can enter your own data into NodeXL by typing a list of the edges in the network into this sheet. We'll do that for a series of hypothetical friends. **Save** the template under a new name, and then enter the following data into the **Edges** sheet:



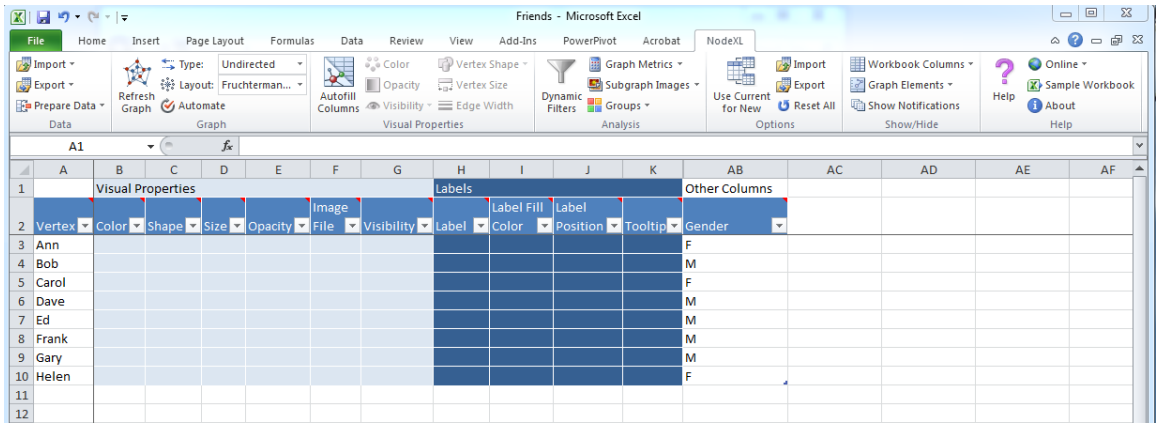

In this case, we're just recording whether the people are friends – a relationship that doesn't have a direction – so we can leave the graph **Type** as **Undirected**.

(If we were recording whether each friend had invited the other to a party, then this should be changed to **Directed**, and we would need a second edge with the names reversed if Ann, for instance, had also invited Carol to a party.)

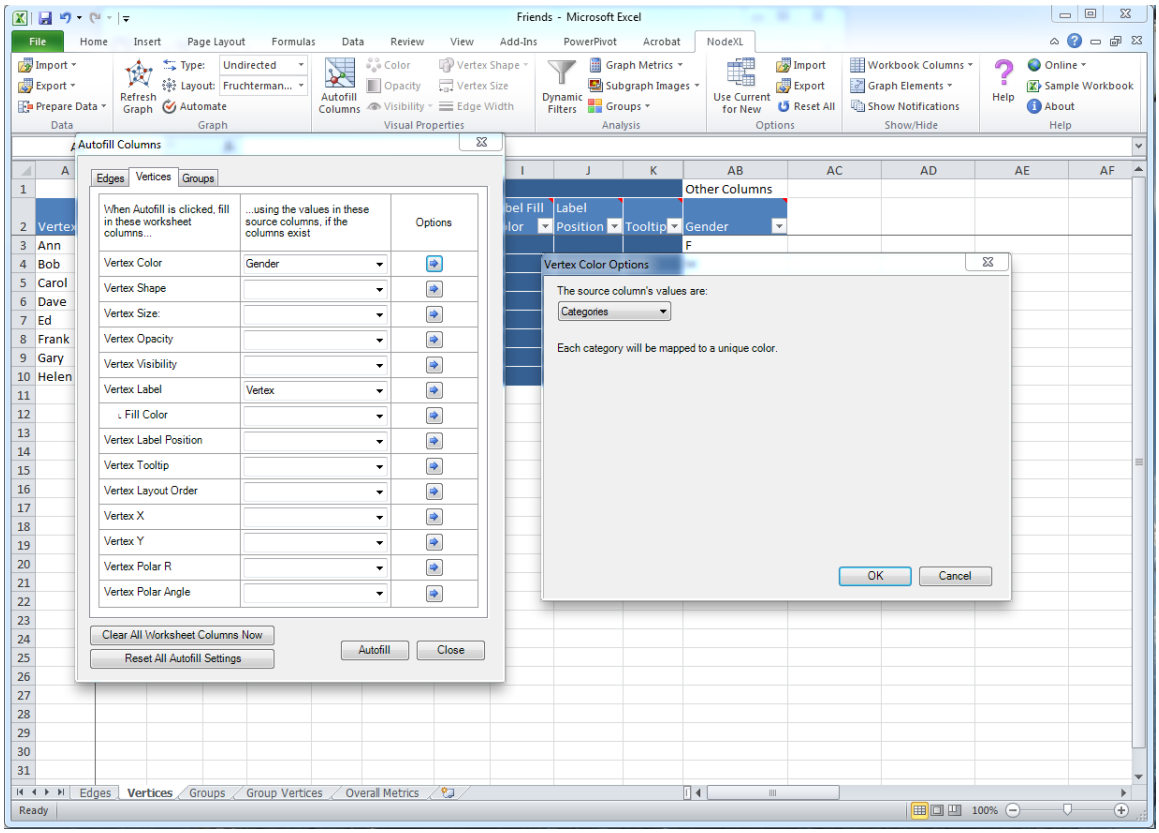
Now click **Show Graph** in the NodeXL menu or in the Window marked Document Action. A simple graph showing the network should then appear in this window:



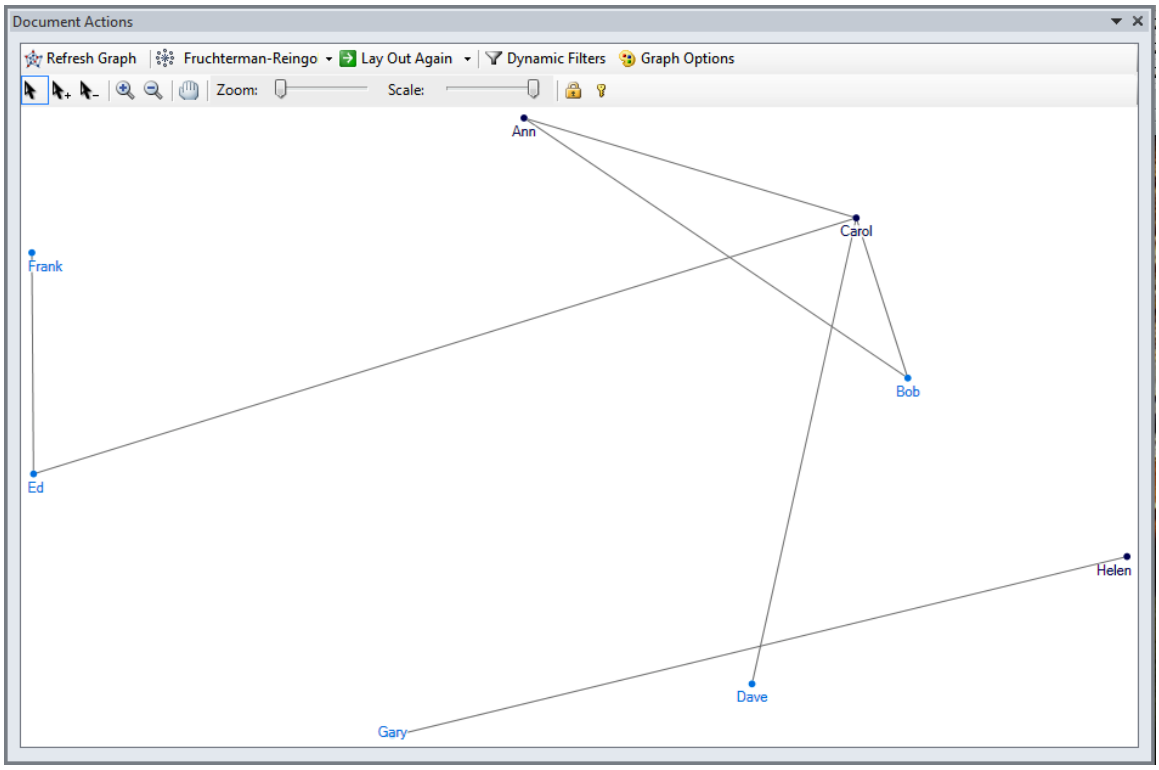
Switch to the **Vertices** sheet, where the name of each friend will have appeared. Add the gender of each under **Other Columns**:




Click **Autofill Columns**, select the **Vertices** tab, and tell NodeXL to label each friend with their name, and color them according to their gender. For the latter, click on the **Options** arrow, and tell NodeXL that the values in the column are **Categories** and click **OK**:



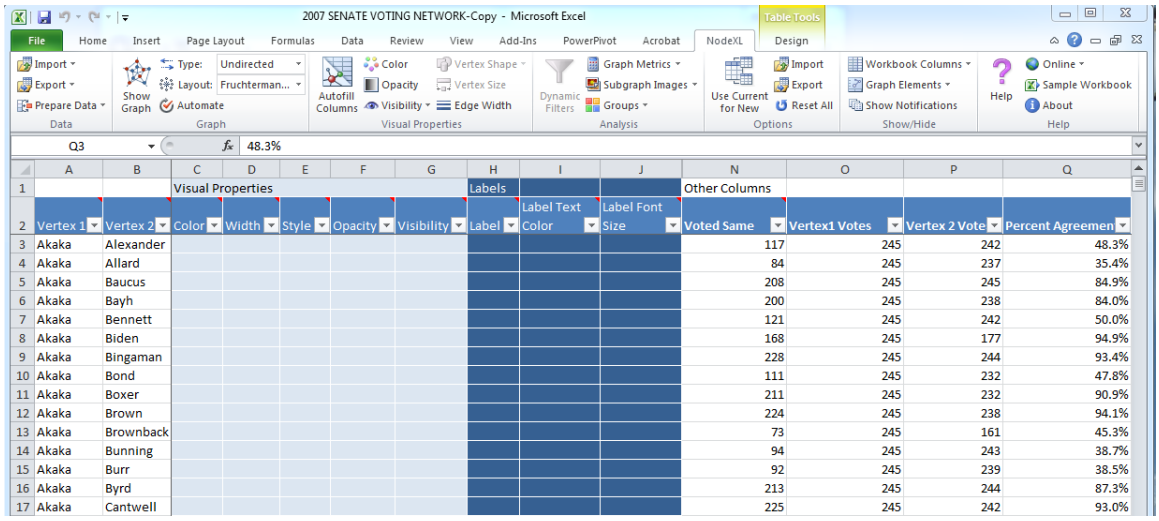
Click **Autofill** then **Close**. The Color column will now have populated with RGB color values, and the Label column will contain the friends' names. The graph should have redrawn, but if necessary, click **Refresh Graph**:




Having learned these basics, we'll explore a more interesting network, based on voting patterns in the US Senate in 2007. The data was compiled by *Slate* and downloaded from the NodeXL teaching site.

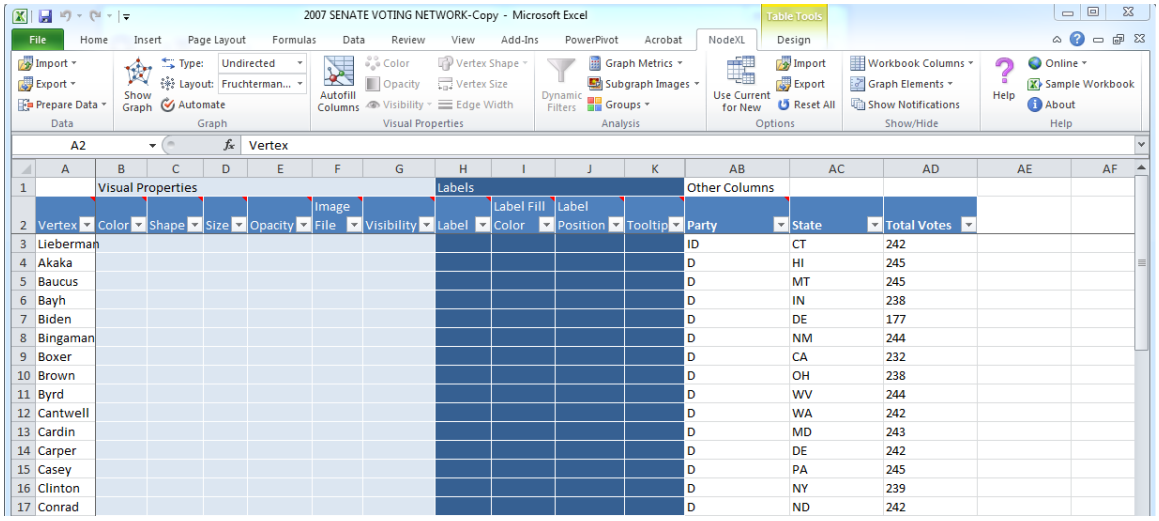
In a blank NodeXL template, select **Import>From NodeXL Workbook Created on Another Computer**, and open the file.

The **Edges** worksheet gives a list of pairs of Senators, with information on how they voted, including a column giving the percentage of times the members of the pair voted the same way:



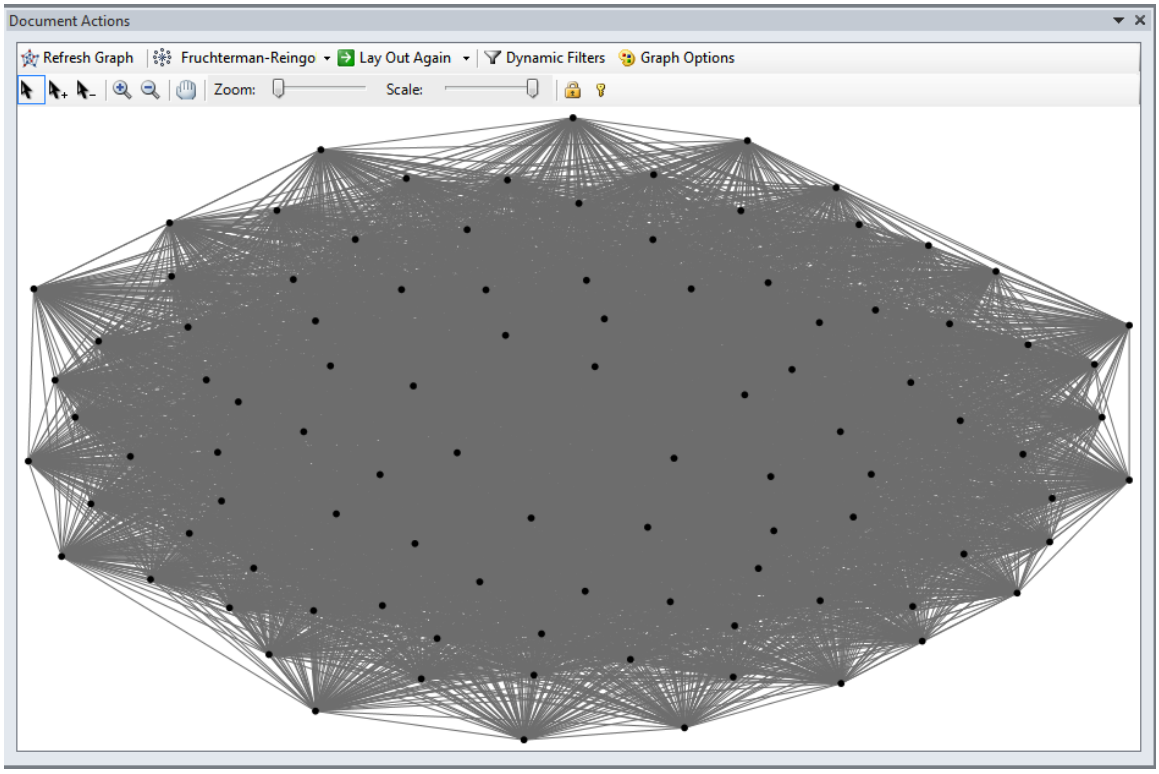

| | A | B | C | D | E | F | G | H | I | J | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | Visual Properties | | | | | Labels | | | Other Columns | | | |
| 2 | Vertex 1 | Vertex 2 | Color | Width | Style | Opacity | Visibility | Label | Label Text Color | Label Font Size | Voted Same | Vertex1 Votes | Vertex 2 Vote | Percent Agreemen |
| 3 | Akaka | Alexander | | | | | | | | | 117 | 245 | 242 | 48.3% |
| 4 | Akaka | Allard | | | | | | | | | 84 | 245 | 237 | 35.4% |
| 5 | Akaka | Baucus | | | | | | | | | 208 | 245 | 245 | 84.9% |
| 6 | Akaka | Bayh | | | | | | | | | 200 | 245 | 238 | 84.0% |
| 7 | Akaka | Bennett | | | | | | | | | 121 | 245 | 242 | 50.0% |
| 8 | Akaka | Biden | | | | | | | | | 168 | 245 | 177 | 94.9% |
| 9 | Akaka | Bingaman | | | | | | | | | 228 | 245 | 244 | 93.4% |
| 10 | Akaka | Bond | | | | | | | | | 111 | 245 | 232 | 47.8% |
| 11 | Akaka | Boxer | | | | | | | | | 211 | 245 | 232 | 90.9% |
| 12 | Akaka | Brown | | | | | | | | | 224 | 245 | 238 | 94.1% |
| 13 | Akaka | Brownback | | | | | | | | | 73 | 245 | 161 | 45.3% |
| 14 | Akaka | Bunning | | | | | | | | | 94 | 245 | 243 | 38.7% |
| 15 | Akaka | Burr | | | | | | | | | 92 | 245 | 239 | 38.5% |
| 16 | Akaka | Byrd | | | | | | | | | 213 | 245 | 244 | 87.3% |
| 17 | Akaka | Cantwell | | | | | | | | | 225 | 245 | 242 | 93.0% |

The **Vertices** worksheet lists each Senator, and gives their Party affiliation, State, and the number of times they voted:




| | A | B | C | D | E | F | G | H | I | J | K | AB | AC | AD | AE | AF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | Visual Properties | | | | | | Labels | | | | Other Columns | | | | |
| 2 | Vertex | Color | Shape | Size | Opacity | Image File | Visibility | Label | Label Fill Color | Label Position | Tooltip | Party | State | Total Votes | | |
| 3 | Lieberman | | | | | | | | | | | ID | CT | 242 | | |
| 4 | Akaka | | | | | | | | | | | D | HI | 245 | | |
| 5 | Baucus | | | | | | | | | | | D | MT | 245 | | |
| 6 | Bayh | | | | | | | | | | | D | IN | 238 | | |
| 7 | Biden | | | | | | | | | | | D | DE | 177 | | |
| 8 | Bingaman | | | | | | | | | | | D | NM | 244 | | |
| 9 | Boxer | | | | | | | | | | | D | CA | 232 | | |
| 10 | Brown | | | | | | | | | | | D | OH | 238 | | |
| 11 | Byrd | | | | | | | | | | | D | WV | 244 | | |
| 12 | Cantwell | | | | | | | | | | | D | WA | 242 | | |
| 13 | Cardin | | | | | | | | | | | D | MD | 243 | | |
| 14 | Carper | | | | | | | | | | | D | DE | 242 | | |
| 15 | Casey | | | | | | | | | | | D | PA | 245 | | |
| 16 | Clinton | | | | | | | | | | | D | NY | 239 | | |
| 17 | Conrad | | | | | | | | | | | D | ND | 242 | | |

Click **Show Graph** to see the following, in which each Senator is connected to all of the others, because every pair voted the same way at least once:



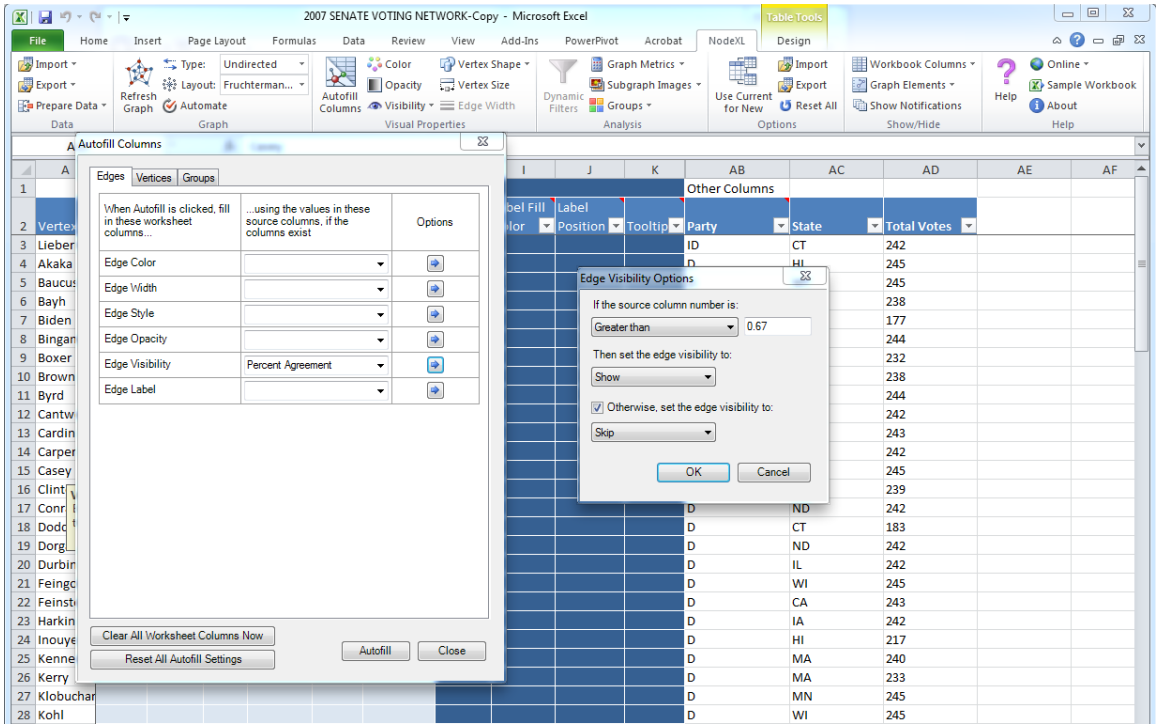

NodeXL's strength is the ease with which you can now filter and customize the network visualization.

The first task with complex networks like this one is often to filter them to reveal their core structure. This can be done in two ways. Clicking **Dynamic Filters** brings up a series of sliders that you can use to adjust the visibility of edges and nodes in the network graph. See how some of the edges disappear as you move the left slider for **Percent Agreement** toward the right.

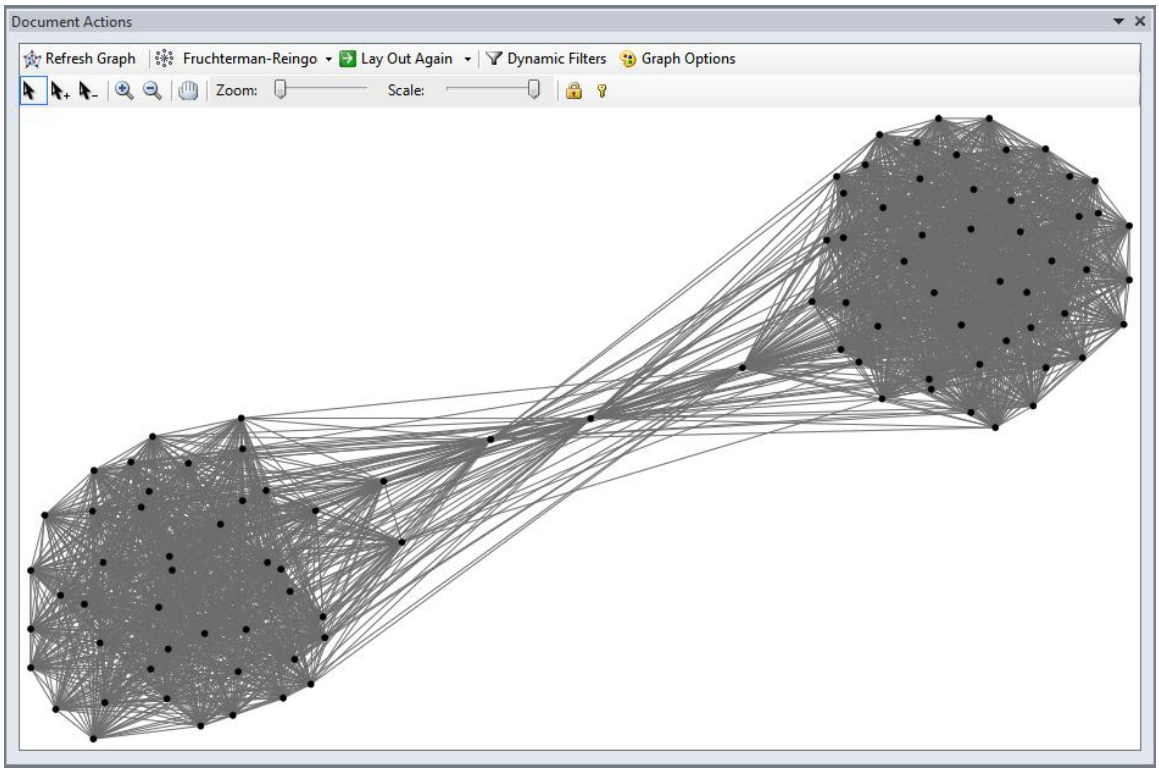
This does not, however, make any changes to the network that is being analyzed. Dynamic filters will not, for instance, cause any change in the results obtained by calculating metrics describing the network.

Instead, we are going to filter the network using **Autofill Columns**, allowing us to run subsequent analyses on this filtered view of the data.

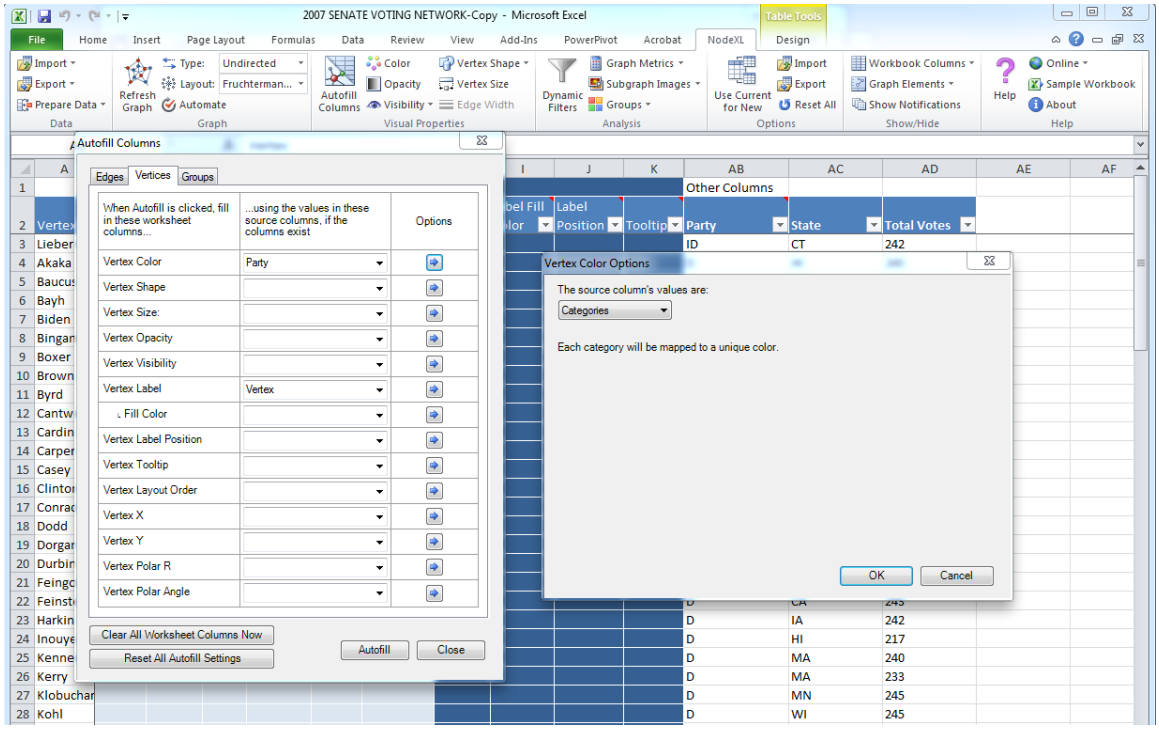
Click **Autofill Columns** and select the **Edges** tab. We will filter so Senators are connected only if they voted the same way at least two-thirds of the time. Select **Edge Visibility = Percent Agreement**, fill in **Options** as follows, and click **OK**:



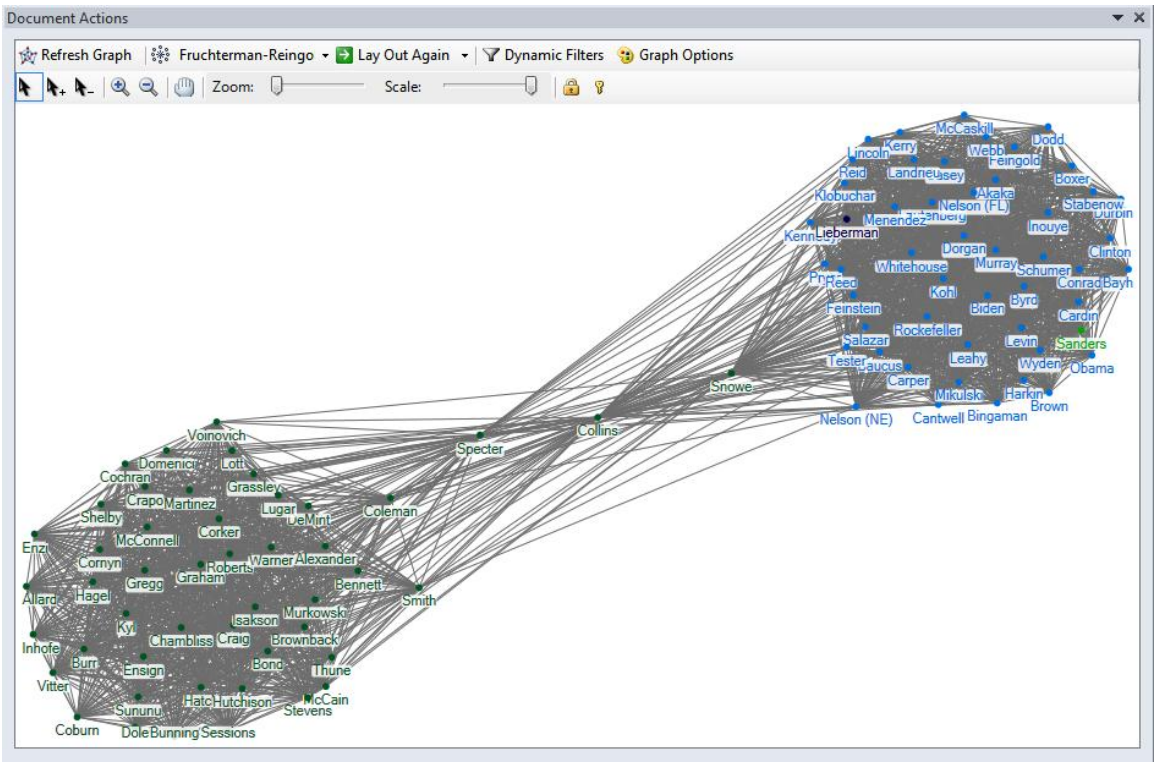

Then click **Autofill**, and the graph should redraw. Make sure the Layout is set to **Fruchterman-Reingold**, which is the automatic layout algorithm that works best with this data, and click **Lay Out Again** until you have two clear clusters:



Presumably these two clusters are Democrats and Republicans, but we can confirm that by using **Autofill Columns**. On the **Vertices** tab, select **Vertex Label = Vertex** to label each Senator with their name, and **Vertex Color = Party** and then **Options = Categories** and **OK**:




Click **Autofill** and **Close**, then **Lay Out Again** until you have something like the following:




The most interesting Senators in the network are the three Republicans who sit between the two main party clusters: Specter, Collins and Snowe. We can calculate some network metrics and customize the graph to illustrate their importance for the overall dynamics of the Senate.

NodeXL can calculate common metrics used to describe networks, including the following:

**Degree** is a simple count of the number of connections for each node. For directed networks, it is divided into **In-degree**, for the number of incoming connections, and **Out-degree,** for outgoing connections.

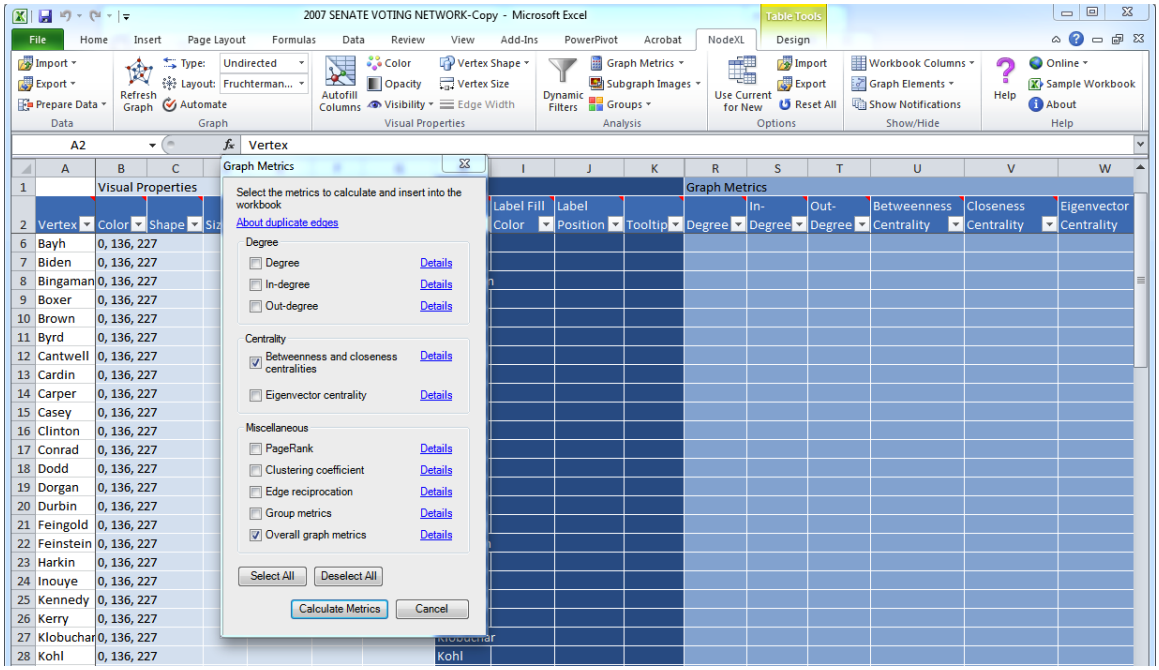
**Eigenvector centrality** accounts not only for the node's own degree, the also the degrees of the nodes to which it connects.

**Betweenness centrality** essentially reveals how important each node is in providing a "bridge" between different parts of the network. It highlights the nodes that, if removed, would cause a network to fall apart.

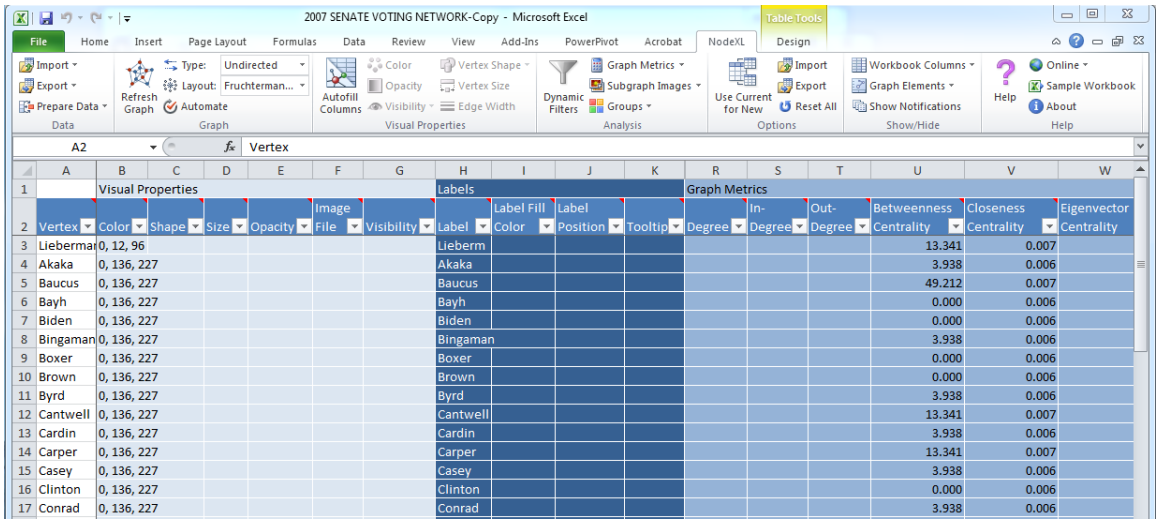
**Closeness centrality** is a measure of how close each node is, on average, to all of the other nodes in a network. It highlights the nodes that connect to the others through a lower number of edges – think Kevin Bacon Game.

For our purposes, betweenness centrality is a good measure, as it should highlight those Senators who provide a bipartisan link between the two core party blocs.

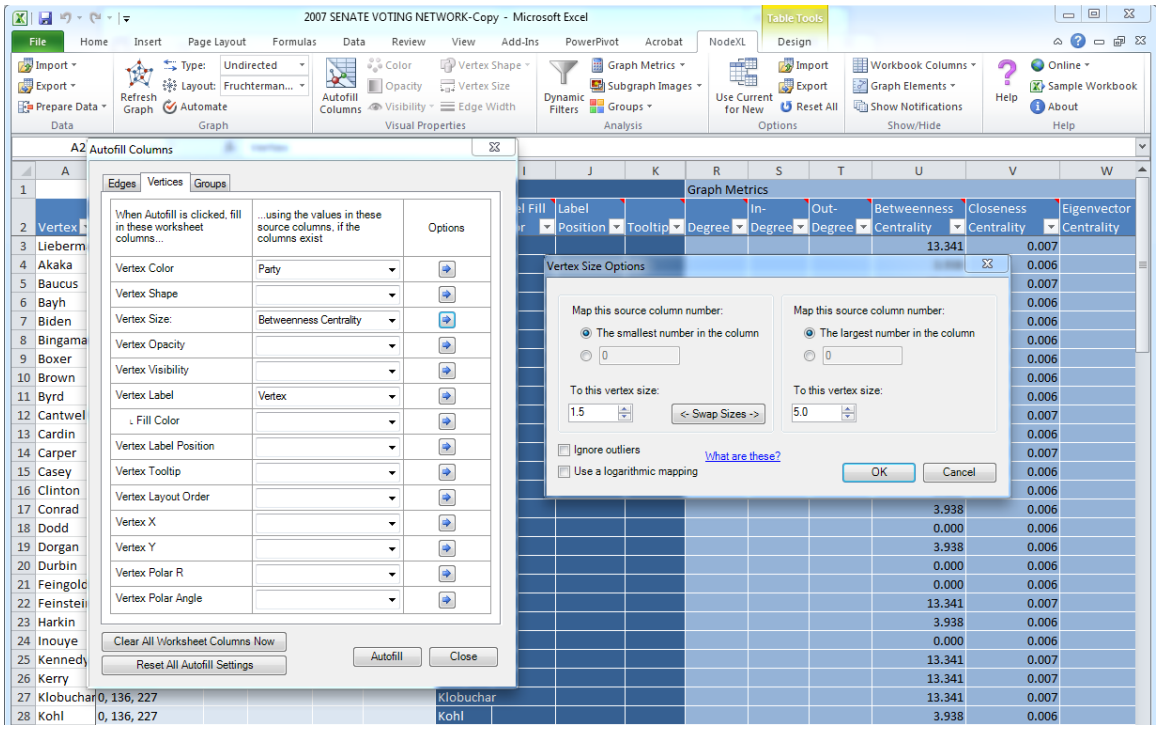
Select **Graph Metrics**, check **Betweenness and closeness centralities** and click **Calculate Metrics:**



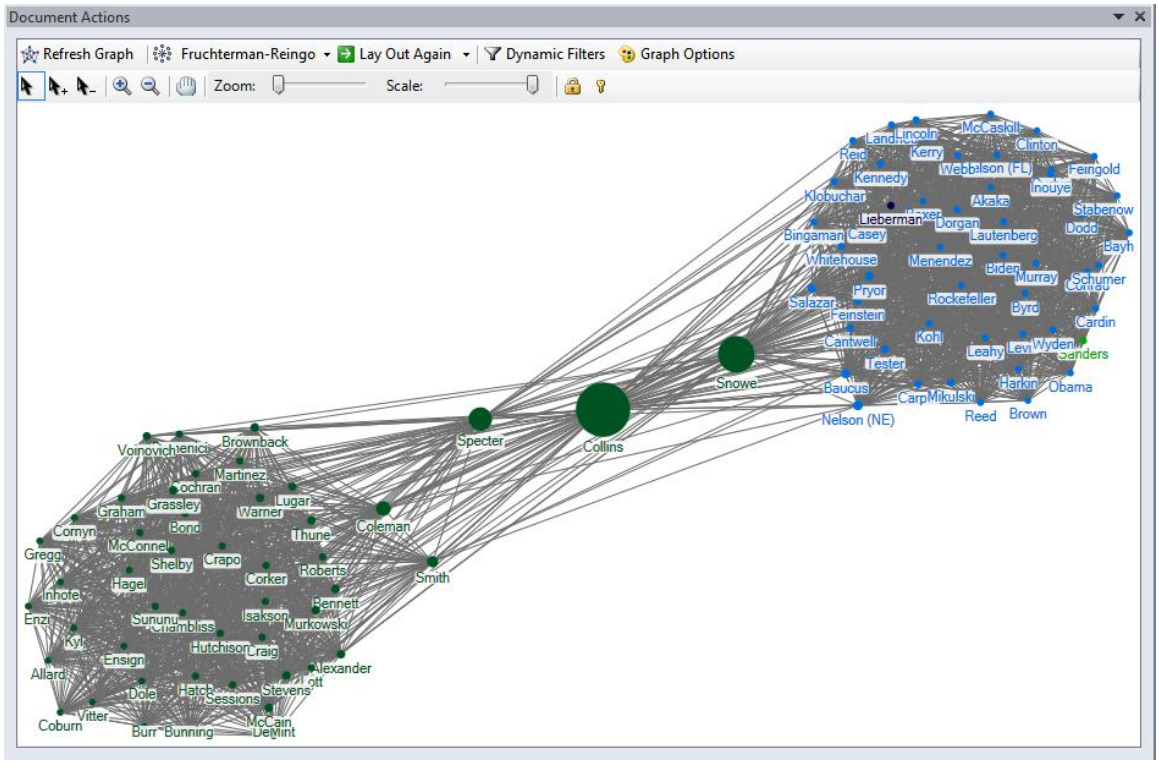

The relevant columns in the Vertices sheet will now have been populated with data:



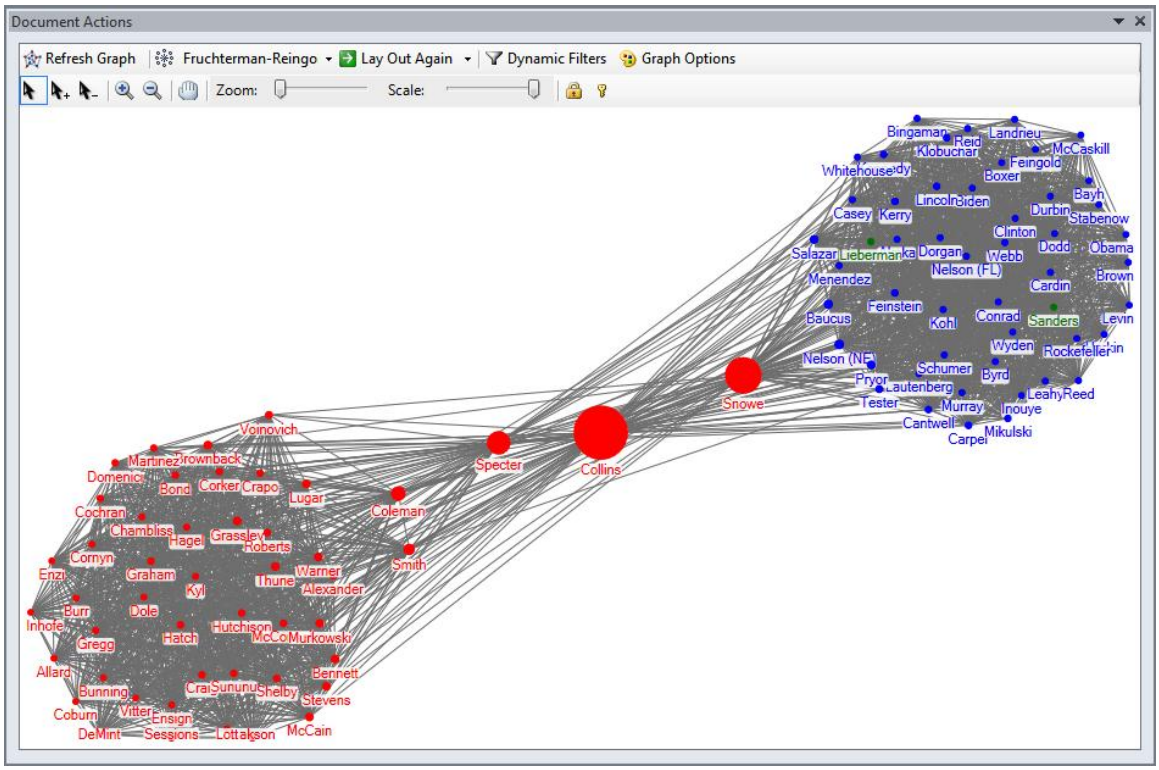

Now we can size the Senators according to their betweenness centrality. Click **Autofill Columns**, select the **Vertices** tab, and set **Vertex Size = Betweenness Centrality**. Using **Options**, set the maximum size to **5**, and click **OK**:



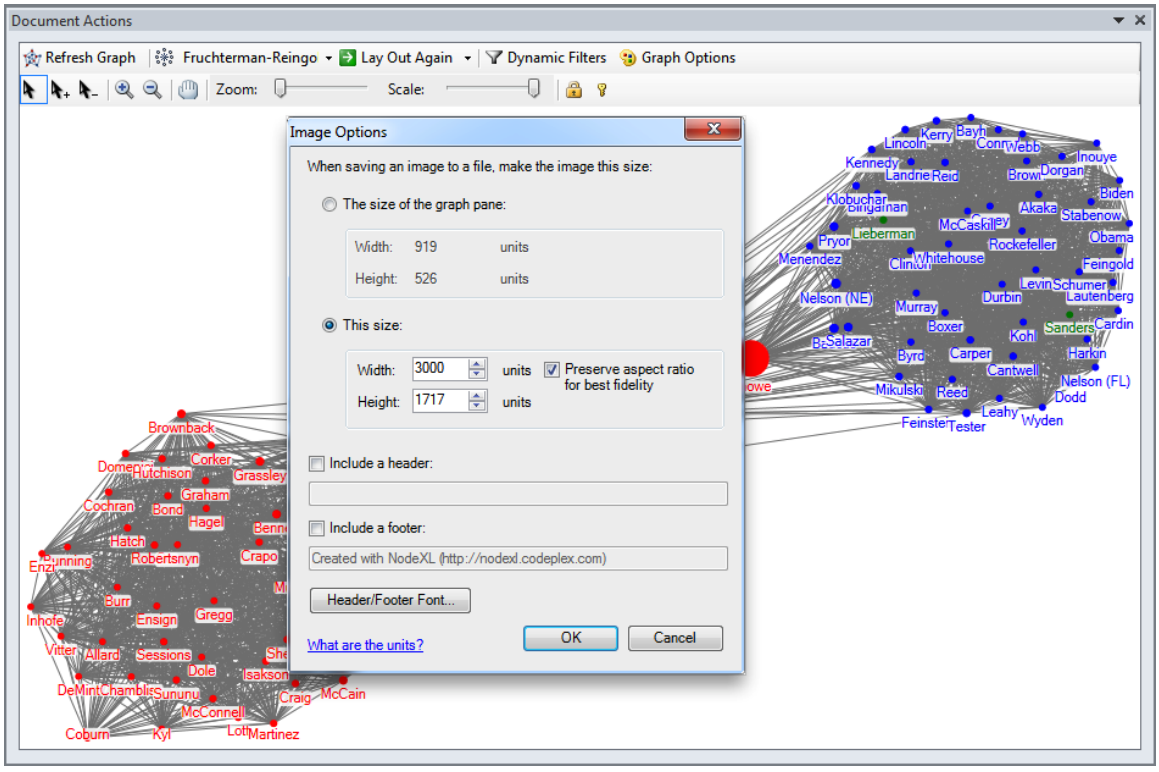

Click **Autofill** and **Close**, then **Lay Out Again** until you have something like the following:



Now change the parties' colors to the familiar red and blue, and make the independents green. In the **Vertices** sheet, replace the RGB values for Lieberman, the Independent Democrat, with Green. Akaka, the first of the Democrats, becomes Blue – copy this value down the column until you get to Wyden, the last Democrat. Then make the Republicans, from Alexander to Warner, Red. Finally, make Sanders, the Independent, Green. Click **Refresh Graph**, then **Lay Out Again** until you have something like the following:



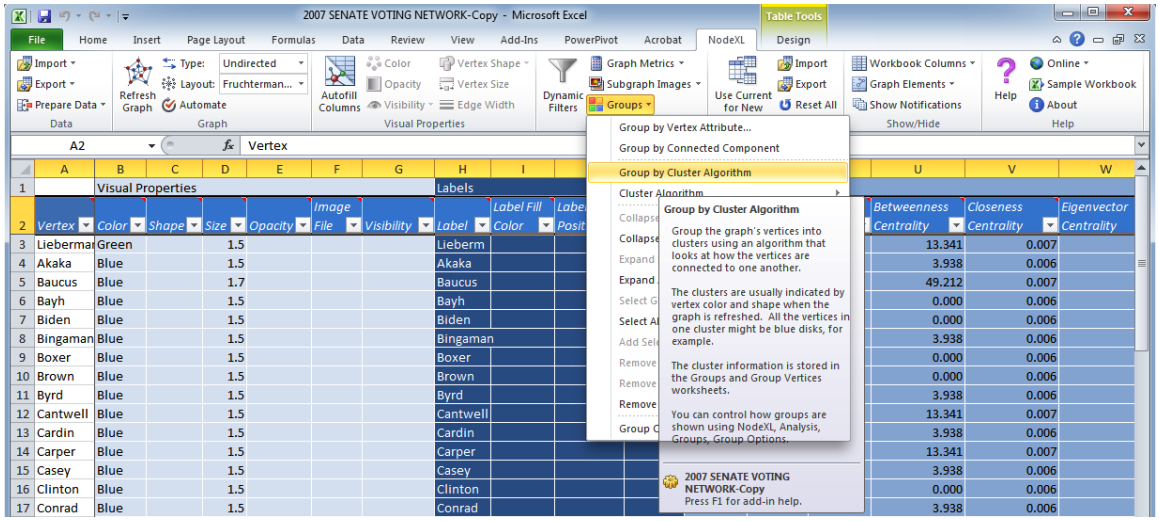

Now is a good time to save the image. **Right click** on the graph, and select **Save Image to File>Image Options** to customize the size. Setting a width of around 3000 pixels is a good idea, to get good resolution:



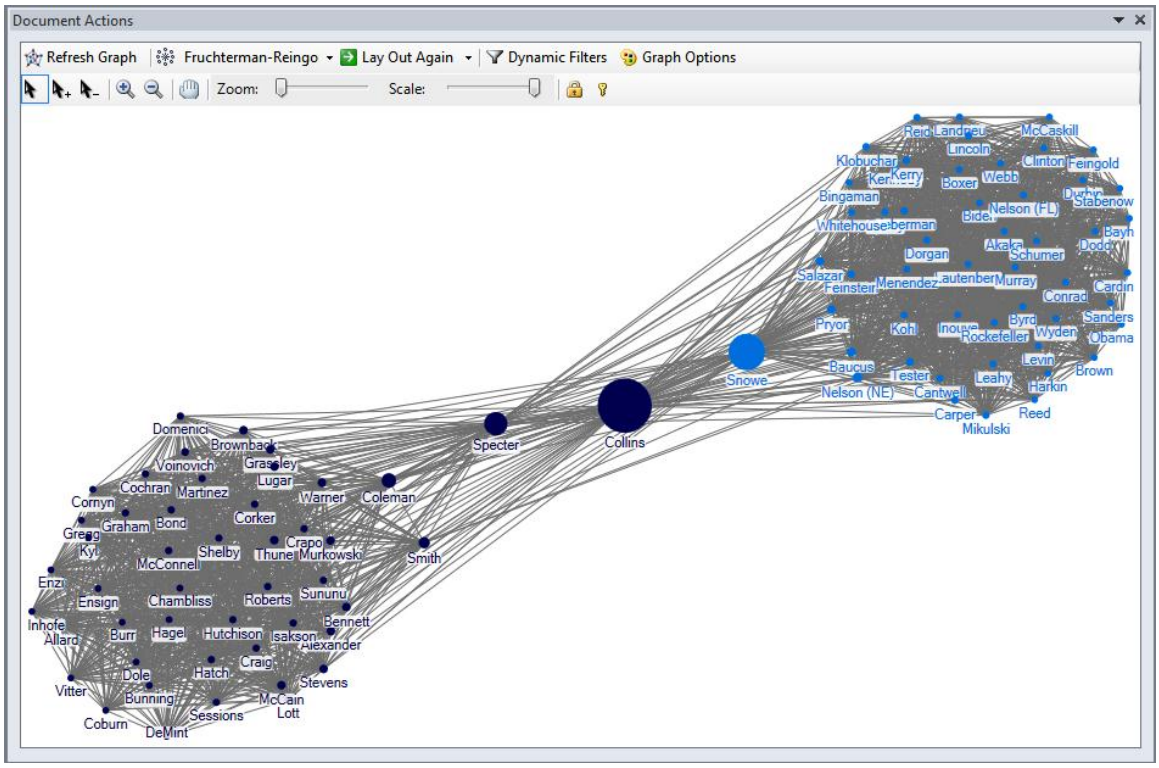

**Right click** again, and then select **Save Image to File>Save Image** to save in formats including PNG and JPEG.

We've colored the Senators by their party affiliation, but NodeXL can also use clustering algorithms to detect clusters of vertices with similar patterns of connections.

Select **Groups>Cluster Algorithm** and check **Clauset-Newman-Moore**. Then select **Groups>Group by Cluster Algorithm**:



Click **Refresh Graph** and then **Lay Out Again** until you have something like the following:




Olympia Snowe is a Republican, but on the basis of her pattern of voting in 2007, the clustering algorithm has decided she actually clusters with the Democrats. See what happens when you repeat the process using the **Wakita-Tsurimi** algorithm.

To learn more about how to use NodeXL, read the tutorial, or refer to the book *Analyzing Social Media Networks With NodeXL*.