

Dot Graph How-To

I. Introduction

Starting with TAMS Analyzer 2.39, TA can produce dot files for use with the GraphViz program (available as a quartz/aqua program free at <http://www.pixelglow.com/graphviz/download/>). It is also available as an X11 program through the fink package manager—fink install graphviz). Dot Graphs are a type of diagram similar to concept maps. They show the relationship between different “nodes”. Here is an example, straight from TAMS Analyzer that shows the relationship between codes that deal with gender and the different documents I’m analyzing as part of a study of the bureaucracy of human subjects.

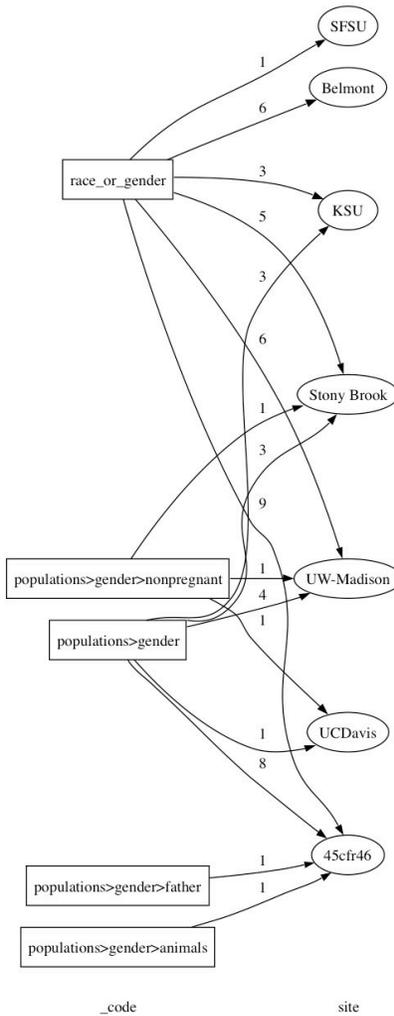


Figure 1: A sample GraphViz Graph/Map

I was able to tell TAMS Analyzer to produce a map connecting a selection between `_code` and `site`, a universal code that indicates where each document came from. I was able to choose that each line be labeled with the number of occurrences (lines are called “edges” in GraphViz terminology). I was able to indicate that `_code` should be marked

with rectangles and site “nodes” indicated by ellipses. I could also have made the edges dotted or dashed. Additionally I could have made the arrows two directional or single directed. TA then kicks out a text file with an extension “.dot” Once the file is generated, I just dropped the file on the GraphViz and voila: beautiful mapping between these two codes.

2. How to generate dot graph diagrams

GraphViz takes files that end in “.dot” These are text files with a series of simple commands that indicate which node (square or ellipse) connects to which other node. In TA, these are generated from result windows. TA treats each column of a result window as a variable and the rows as indicating what the values of those variables are. TA, in generating graph files, allows you to map (graph) one of these variable-columns onto one or more others. However, only universal code columns, the repeat code columns, and the `_code` column are eligible to be mapped.

There are two ways that TA can graph (map) these columns. It can take the left most eligible column and connect those values to the values in each of the other eligible columns (1st to many). Alternatively, it can connect all of the columns to each other (Many to many).

There are three steps that you should execute before picking Reports->dotGraph Output. First, you should move the column that you want to be the one to map onto the others so that it is left-most. In the graph above it was `_code`. So I had to drag the `_code` column to be the first column (or at least to the left of any repeat and universal code, i.e., eligible, columns). Second, you should select the data you are interested in graphing. Graphs are often more effective with less rather than more data. In my example I selected “gender” from the `_code` column. Finally, if you want, you should set a “code level” (Results->Code level). Code level 0 means that no code level comparisons are carried out. Code level 1 means $a > b$ is equal to $a > c$ (and will just appear as “a” in the graph). Code level 2 means $a > b > c$ is equal to $a > b > d$ and will appear as $a > b$, but will be distinct from $a > c$ or a .

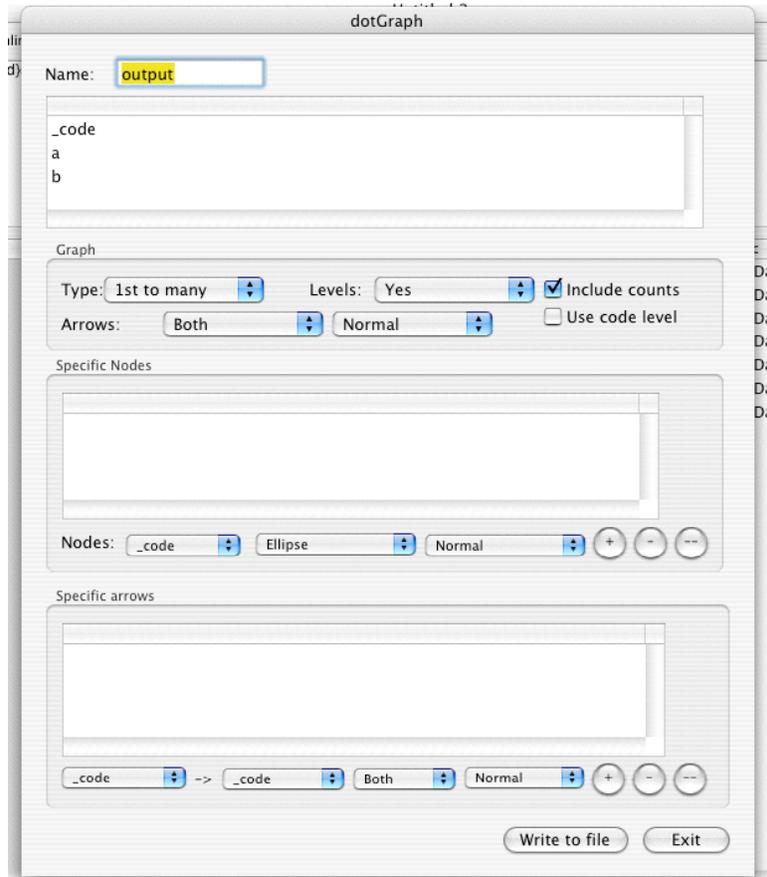


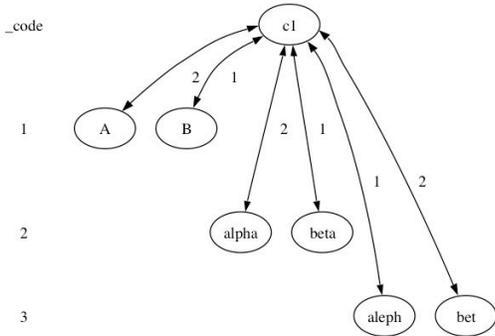
Figure 2: the dotGraph Output dialogue

Having rearranged your columns, selected your data, and set the code level you should now pick Reports->dotGraph Output. What you'll see is figure 2.

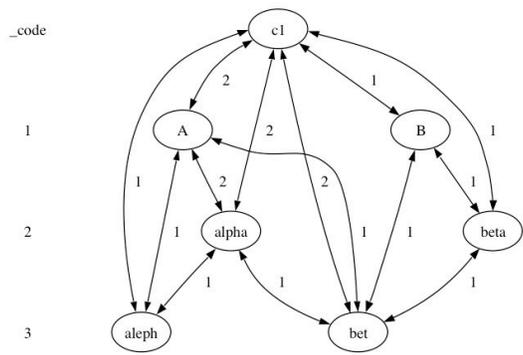
You need to work this dialogue from top to bottom. First fill in a **one-word** name for this graph. You could just leave it as is if you like. Then select the fields that you want mapped. You can apple-click and shift click to make sure that all of the variables/columns/codes you want are selected.

Then set the overall graph characteristics. Is it a 1st (first in the above table) to many or many to many or tree mapping?

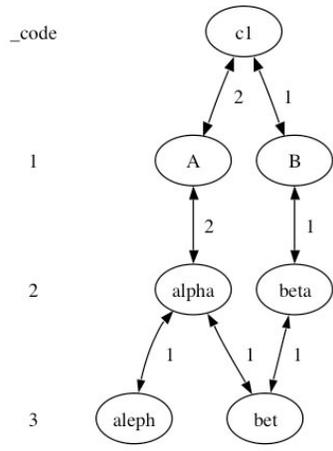
To see the difference between these here are examples of each generated from the exact same data (repeat codes called 1, 2, and 3; data code called c1):



1st To Many: The left most column is mapped to each subsequent column but they are not mapped to each other.



Many To Many: Each level is mapped to all the others



Tree: Each level is mapped only to the next most level (column).

The choice of map really is indicated by the meaning of your repeat and universal variables.

That is selected with the “Type” menu. The “Levels” menu indicates whether you want a scale like the one I have in figure one (indicating which column is `_code` and sites). You can also set some general characteristics for edges: whether arrows appear at both ends and whether the edges should be solid or dashed or dotted.

Now you have the opportunity to set the characteristics of specific variables (columns from the result window). You could give the values of a specific column a square or ellipse shaped box, for instance. I used the specific nodes box to give the site values “boxes” rather than ellipses, which is the default. You could also make the border around the node dotted or shade the box. When you’ve picked the values from the menus under the specific nodes area you need to use the + button to add it to the criteria for this graph.

Then you can pick characteristics for edges. You could make all the lines from variable a to variable b dotted or double arrowed, for instance. When picking which edge you need to specify a **from** and a **to**, separated by the “->” symbol. Always pick the field higher up in the menu to be your from and the one further down to be your to. Again, nothing is registered til you click the “+” button, which adds that criteria to your map.

Now you’re ready to write the dot file. Click the “write to file” button, give it a name at the standard save dialog box that appears and then use Finder to drag and drop your newly created “.dot” file to the GraphViz application. Make sure you play with the settings for your graph/map in Graphviz as well. Your map will initially be from top to bottom, for instance. To make the “gender” map left to right, I changed the value of “Rank Direction” under the Graph tab in the settings drawer.

I think the only way to realize what this can do is by playing with options and trying different ways of graphing your data. The dotGraph dialogue wont close until you click the exit button so you can modify the values compare the graphs they produce.

Enjoy and happy data mining!