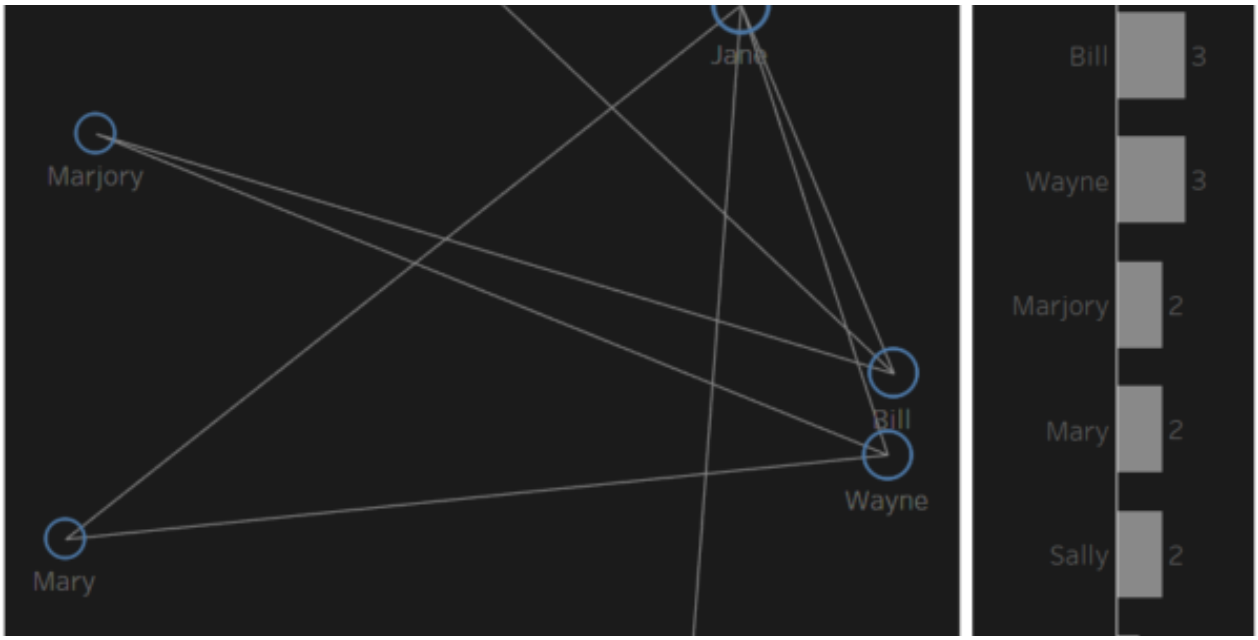


Alteryx Love Tableau: Data Preparation for Network Graph

Published by Data Psycho on July 20, 2017

I have been researching about Network Graph for last one month (approx.). The maximum help I have found form here. At the beginning the blog was very confusing for me but after spending some time I configured everything. So creating a network is not a problem at all but data preparation is pain in the ass (yes, you got it !). There is a lot of conversation and suggestion about it, such as use of Path Shelf, Gephi or Manual Approach using python. But then a lot of things to do to prepare the coordinate for the relation. So I have create my own Alteryx package to create the coordinates which was more easy for me to build.

If you do not want to read the whole bullshit then go to the bottom and download the zip file.You will have everything.

Dig Dipper:

Before you start, You should know:

- Basic Alteryx Knowledge
- Basic SQL Knowledge (inner Join)

- Intermediate knowledge on R language

It is also possible to do the whole operation by using only R or Python (Numpy, Panda).

Some time you want to kill your customers, when they give you a creepy data and want you to show some insights. Yes, lets start with the worse case when You just have two attribute in your data the relationship and a value. I am using the same data set form that blog (That man deserve a lifetime free tableau license :)).

Relationship	Amount
Wayne-->Marjory	$20,000
Mary-->Jane	$13,000
...	...

The first step should be split the relation by '- ->' and clean the Trailing space in each column using Text to Column and Data Cleansing tool in Alteryx . I call the new generated column as *'FromFlow'* and *'ToFlow'*, the new table should looks like as follows:

MessageFlow	FromFlow	ToFlow	Amount
Ken --> Bill	Ken	Bill	10000
Mary-->Jane	Mary	Jane	13000
Wayne-->Marjory	Wayne	Wayne	20000
...	...	...	...

Now I need each row twice; one for the From object and one for the To object. So I take same input and do the same cleaning. But after that pivot the table using *'FromFlow'* and *'ToFlow'* and create the following table to join with the previous table.

MessageFlow	FlowName	NodeName

MessageFlow	FlowName	NodeName
Ken --> Bill	FromFlow	Ken
Ken --> Bill	ToFlow	Bill
Mary-->Jane	FromFlow	Mary
...	...	...

Time to join the second and third table using *'MessageFlow'* as primary key to get desired data frame where one relation has two rows with different Node Name (From and To). Then I do some basic cleaning such as setting null *'Amount'* to the cells where *'FlowName'* is *'ToFlow'* and I get the following table:

MessageFlow	FromFlow	ToFlow	Amount	FlowName	NodeName
Ken --> Bill	Ken	Bill	10000	FromFlow	Ken
Ken --> Bill	Ken	Bill		ToFlow	Bill
Mary-->Jane	Mary	Jane	13000	FromFlow	Mary
...	...	...	...	...	...

Here comes the interesting part, create the x and y coordinate so that each person/object get a fixed coordinate in a two dimensional space. For that, first I create a new sub-flow where I take each person/object once using Unique Tool in Alteryx. Then I send the unique group to R tool and using a small function I assign a unique coordinate to each person/object. The R code is as follows:

```
data <- read.Alteryx("#1", mode="data.frame")
rowNo <- nrow(data)
# assign a redious of a circle
radius <- 1000
# divide the angle according to the number of object
theta <- seq(0, (2*pi-0.17), length=rowNo)
# generating the x and y coordinate of a circle with a
xCoords <- round(radius * cos(theta)*100, 0)
yCoords <- round(radius *sin(theta)*100, 0)
# create the final data frame
dataExt <- data.frame(data, xCoords, yCoords, stringsA
write.Alteryx(dataExt, 1)
```

Let me explain, every person/object get a fixed coordinate point over a circle. As it is a circle first I divide the angle (2π) according to the number of objects. Then I generate X and Y coordinate according to the radius and angle. It is based on polar equation of circle. Actually that's the moment when you realize high school math is not dead and still rocks.

So Finally the golden moment, now I have all the ingredient I need for the damn Network Graph. Look out the following table:

NodeName	xCoords	yCoords
Bill	100000	0
Jane	64229	76646
Ken	-17492	98458
...		

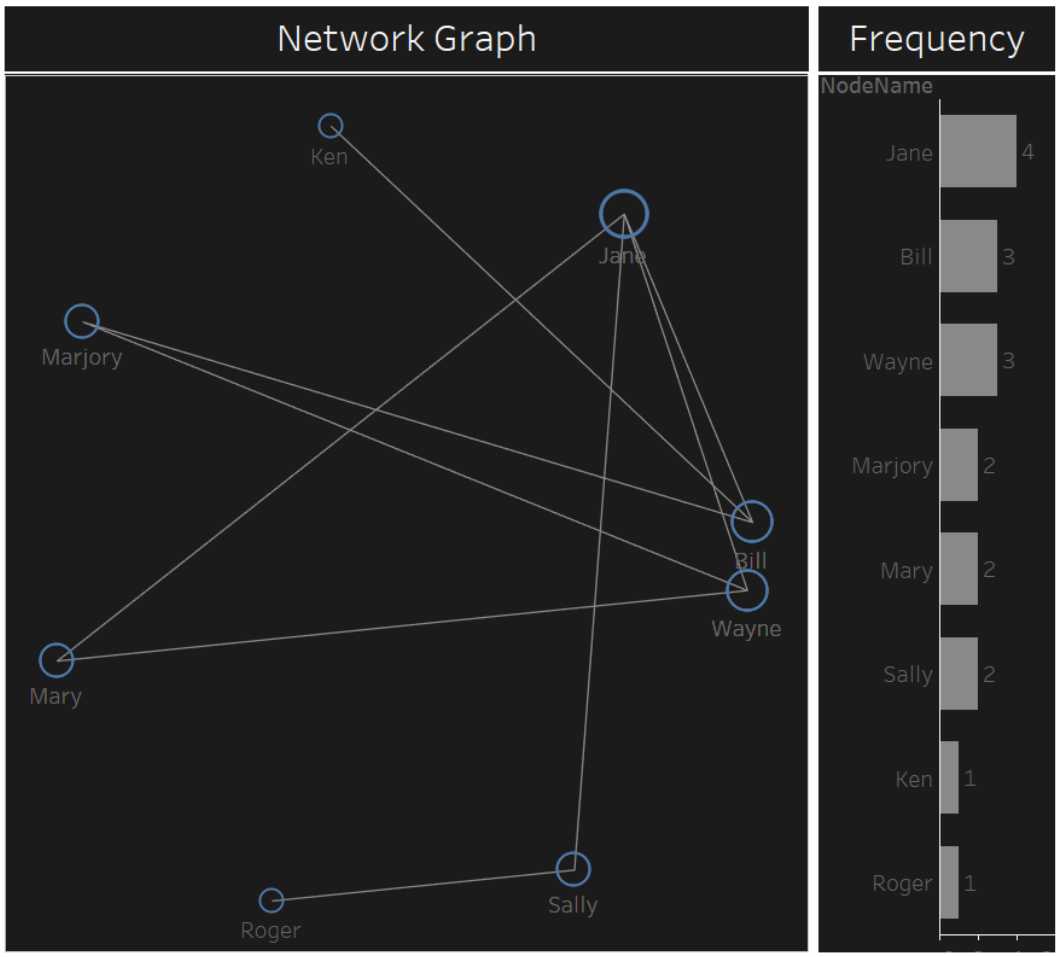
So it is possible to try any kind of function instead of Circle to give the graph different shape. you can try wave function such as mixture of sin, cos, tan or may be parabola, quadratic equation. You just need to make sure that the coordinate point are unique for each object name.

Now if I Join that table with previous table by *'NodeName'*, then I will have the full data. I also have used formula tool to replicate the same *'yCoords'* to *'yCircle'* which gives me the whole data set as follows:

MessageFlow	...	Amount	NodeName	xCoords	yCoords	yCircle
Bill-->Jane	...	10000	Bill	100000	0	0
Ken --> Bill	...		Bill	100000	0	0
Marjory-->Bill	...	13000	Bill	100000	0	0
...	...	...	...	...	...	...

So that's all you need. Wait but where is the workflow I promised. Do not worry, CLICK HERE to download the zip file. I also have added the tableau workbook as bonus.

When you read that it should sound very easy to do. But honestly it took more that 20 hours to configure everything. It was a very good brain storming for me though.



Published in Alteryx R Tableau

Alteryx R Tableau

Previous Post
Alteryx Good, Bad & Ugly: Introduction & Lets Create a Template

No Newer Posts
Return to Blog