Data:
The data is from the survey taken in 1994-95 on 90,118 junior or high school students representing 84 communities. The original data are constructed from the in-school questionnaire, asking students 68 questions relating to their sex, race, grade, family information and friendship (for example, list five male and female friends). Since we’re only interested in analyzing the networks, which are the friendships between students in each community and also each student’s personal information, so the data I chose representing 1338 students from one school is just part of the whole dataset and consists of each student’s sex, race, grade, and his list of friends.

Import data:
The file I have are two text files, one of which records each student’s ID, sex, race and grade (ID is from 1 to 1338, Sex is coded 1=male, 2=female, 0=unreported. Race is coded 1=white, 2=black, 3=hispanic, 4=asian, 5=mixed/other, 0=unreported. Grade is recorded as a number between 7 and 12 with 0=unreported.), the other one records lists of pairs of numbers indicating the friendships between two students.

Import the data from text into the NodeXL template, making the “edge” sheet have the list of friendships and “vertex” sheet have the list of students, after which, create columns for “sex”, “race”, “grade” and populate the data.

Headlines:
1. Students make friends with the same race
   In order to differentiate students with different races, I use different colors indicating race (blue for white, lime for black, fuchsia for Hispanic, yellow for Asian, orange for mixed/other), and select Harel-Koren Fast Multiscale layout, then generate figure 1). And then group by cluster (Clauset-Newman-Moore), generate figure 2).

   From figure 1) below, the vertices are roughly clustered into three groups, each of which is dominated by one color, revealing that friends have the same race. To have an even more clear idea, take a look at figure 2) and we can conclude that, students tend to have friends who are of the same race as they are. Because clusters are generated based on the edges, vertices that are connected are grouped in one cluster. For the larger five clusters, each cluster has a dominant color indicating one race, no matter it’s white, black or Hispanic, drawing the conclusion that students tend to make friends with the same race.
2. **White and black students tend to make friends of the same grade while Hispanic students don’t**

Based on Figure 2) and in order to differentiate students in different grades, I use different shapes indicating grades (triangle for students in grade 9, circle for students in grade 10, solid triangle for students in grade 11, disk for students in grade 12). And group by cluster (Clauset-Newman-Moore), then generate Figure 3).

From figure 3) below, we can see that, there is one large cluster for Hispanic students (the one with more fuchsia disks) where the students are in grade ranging from 9-12 in a distributive manner. While there is one large cluster for black students (the one with more lime disks) that has four kinds of shapes but mainly each shape is clustered together, and also three large clusters for white students (the ones with more blue disks), two of which mainly have students in grade 9 and grade 10 respectively (denoted as triangle and circle) and the other one mainly has students with higher grades (denoted as solid triangle and disk) but also has the same shapes clustered together. So compared to Hispanic students, white and black students tend to make students with those who are of the same grade (or say, same age) while Hispanic students don't seem to have that preference.
3. Compared to other students, Asian students with higher grades have fewer friends

In order to differentiate students that are of different races and have different number of friends, I calculated the matrix of in-degree for each vertex and give the vertex different sizes (smaller size for vertex that has smaller in-degree and bigger size for vertex that has larger in-degree), and use different colors indicating race (blue for white, lime for black, fuchsia for Hispanic, yellow for Asian, orange for mixed/other), disk shape for all the vertices. Group the data by grades, and filter for Asian students, white students, black students, Hispanic students and mixed/other students separately.

From figure 4) below, we can see that for Asian students, which are denoted as yellow color, the disks in the higher grade group have smaller size on average, meaning that they have fewer friends because size indicates the number of friends they have. While by looking at 5), 6), 7), 8) each for white, black, Hispanic and mixed/other students respectively, there aren’t such big difference for higher grade and lower grade students. Each cluster has large size of disks as well as small size of disks.

![Figure 4) Diagram showing different grades and races with disks of varying sizes and colors.](http://nodexl.codeplex.com)
Critique of the NodeXL experience

Overall speaking, NodeXL is a powerful tool in providing valuable network visualization. Embedding in Excel, it provides additional useful tools and service for network visualization while still keeps data import and manipulation as significantly easy and simple as it is when we use Excel. In addition, the auto function it provides, including Autofill columns and Autofill vertices once we have edges, makes the filling task much more efficient than ever before, especially when we have a large data set.

Apart from all the fabulous advantages, however, when I used it to accomplish my homework, I found some issues with NodeXL that, if can be solved when possible, would
help not only me, but also all the others who use this for network visualization to have a more powerful and efficient tool. Here're the issues:

1. It’s known to almost everybody that NodeXL only works in Windows system, which is a pity, because nowadays large portions of people use other systems like MacOS or Unix, and these potential users can be lost if installing a Windows system is a headache for them. So a solution for this can be to develop NodeXL to make it work with other systems.

2. NodeXL doesn’t well support undo, so each time I mistakenly autofilled the columns, I wasn’t able to go back to where I was before autofilling, and the only option for me is to redo the whole thing again from where I began, and that really wasted me a lot of time when autofill is supposed to save time for users. So a solution for this can be to support undo whenever the user makes an operation.

3. Speaking of Autofill, the color is automatically set for each category without users being able to select their own preferences for colors. And the size is linearly or logarithmic mapped to the source column values without providing the ability of one-to-one mapping to the source column values. So a solution for this can be to provide more customized features for the user.

4. When clustering, it’s hard to know what each of these clusters is. For example, when I use Clauset-Newman-Moore to cluster, I’m only provided with the clusters’ name without detailed information about what kind of properties the vertices in each cluster have. So a solution for this would be to have an additional column explaining the purpose of each cluster.

5. Background pictures of the visualization can’t be resized when user zooms in/out. So a solution of this would be to resize it when zooming in/out.

6. Even though it’s easy to filter vertices that have some specific properties, it’s hard to filter edges that have some common property, for example, whether this edge links two people that have different sex, and I’m interested in that because I would like to know if boys have more female friends, girls have more male friends or vice versa. But right now I haven’t been able to figure it out unless I manually delete some edges from the sheet, which is a huge amount of work for thousands of edges. So a solution of this would be to provide the functionality of filtering edges, the vertices of which they link have some specific properties.

Reference: