

# Untitled

## Affiliation

### Introduction

Background, what's out there (visualization tools,) why this is useful (because there are not that many detailed examples showing the code, talk about your experience in Sunbelt "what's the format of the data", look for papers talking about computing literacy) and our goal (start to finish network visualization: load the data, process it a little bit, and plot it).

### Network visualization in a nutshell: Things to consider

Talk about the different aspects about network viz the user needs to consider: layout, vertex size, vertex colour, vertex shape, edges, edges width, etc. Talk about the different components and how can we use them (to represent what, for example.) The size of the network, and type of the network (egocentric, small, large, bipartite, etc.)

In terms of the layouts, what are the things we need to consider (we can mention R packages that implement layouts in R).

### Start to finish example

#### Example 1: School data

#### Cleaning Data.

First, the data needs to be pulled in. It can be found at (INSERT LINK HERE). After we pull it in, let's get a glimpse of what the data looks like.

```
# attaching packages
library(igraph)
library(data.table)
library(devtools)
install_github("USCCANA/netplot")
library(netplot)
```

```
# loading and cleaning data
```

```
students <- fread("../data/middle_school/pone.0153690.s001.csv")
interactions <- fread("../data/middle_school/pone.0153690.s003.csv")
```

```
print(students)
```

	id	grade	gender	unique	lunch	initials	Num
1:	2003	7	0	0	1		386
2:	2004	8	1	1	1		402
3:	2006	7	1	1	2		288

4:	2008	8	0	1	1	199
5:	2009	7	1	0	1	147
---						
674:	NA	8	0	0	99	171
675:	NA	8	0	1	99	270
676:	NA	8	0	1	99	327
677:	NA	99	1	0	99	378
678:	NA	7	1	0	99	277

```
print(interactions)
```

	id	contactGender	contactGrade	contactId	ClassPer
1:	2004	1	8	3127	
2:	2004	0	8	2620	
3:	2004	1	8	99	
4:	2004	1	8	99	
5:	2004	1	8	99	
---					
10777:	3448	1	7	99	
10778:	3448	1	7	99	
10779:	3448	1	7	99	
10780:	3448	1	7	3439	
10781:	3448	1	7	99	

In order to use the data, we need to remove all of the 'N/A's and miscoding in the datasets. Also, we see a large number of students who only have interactions with themselves (they do not talk to anyone else through the day), so these "isolates" need to be removed in order for the graph to be more easily read.

```
# filtering out 'N/A's in the 'students' data frame
students <- students[!is.na(id)]
```

```
# filtering down to gender being "0" or "1"
students <- students[gender %in% c("0", "1")]
```

```
# filter out 'N/A's in 'id' and 'contactId'
interactions <- interactions[!is.na(id) & !is.na(contactId)]
```

```
# Which connections are not OK?
```

```
ids <- sort(unique(students$id))
```

```
# narrowed our data from 10781 to 5150
```

```
interactions <- interactions[(id %in% ids) & (contactId %in% ids)]
```

```
source(file = "../misc/color_nodes_function.R")
```

After, the two datasets need to be combined together.

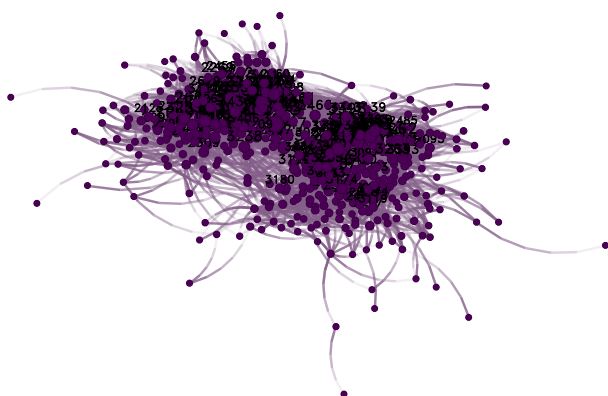
```
## Creating matrix from datasets
net <- graph_from_data_frame(
  d = interactions[, .(id, contactId)],
  directed = FALSE, vertices = as.data.frame(students)
)
```

```
## Getting only connected individuals
```

```
net_with_no_isolates <- induced_subgraph(net, which(degree(net) > 0))
```

Finally, we plot it, effectively showing this network graph.

```
## Plot with no isolates
nplot(
  net_with_no_isolates
)
```



```
vertex.nsizes = ifelse(V(net_with_no_isolates)$grade
vertex.size.range = c(0.015, 0.015),
edge.color = ~ego(alpha = 1, col = "lightgray") + alt
vertex.label = NULL,
edge.curvature = pi/6,
edge.line.breaks = 10
)
```

```
# add radial gradient fill
```

```
grades <- set_vertex_gpar(grades,
  element = "core",
  fill = lapply(get_vertex_gpar,
    radialGradient(c("white", i
  )))
```

```
# add legend to graph
```

```
grades_general <- nplot_legend(
  grades,
  labels = c("7th", "8th"),
  pch = c(21, 21),
  gp = gpar(
    fill = c("gray40", "red3")),
  packgrob.args = list(side = "bottom"),
  ncol = 2
)
```

```
grades_general
```

```
grid.text("Split According to Grade", x = .2, y = .87,
```

### Split according to Grade.

Here, we are taking the data set and coloring the nodes according to grades. The 7th graders are in gray and the 8th graders are in red.

```
## adjust 'grade' to factor
```

```
V(net_with_no_isolates)$grade <- as.factor(V(net_with_no_isolates)$grade)
```

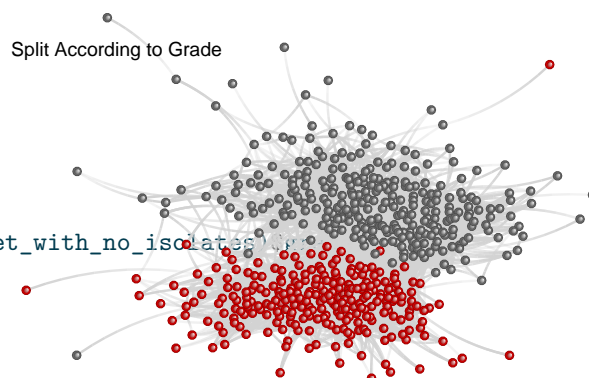
```
# plotting connections among grades ####
set.seed(77)
```

```
a_colors <- color_nodes(net_with_no_isolates, "grade", c("gray40", "red3"),
attr(a_colors, "map")
```

```
7 8
```

```
"#666666" "#CD0000"
```

```
grades <- nplot(
  net_with_no_isolates,
  vertex.color = color_nodes(net_with_no_isolates, "grade", c("gray40", "red3")),
```



### Split according to Gender.

Next, the data will be split according to gender, with male being yellow and female being green. Also, the male points are circles, while the female points are diamonds.

```

# let's get a graph for the gender data
V(net_with_no_isolates)$gender <- as.factor(V(net_with_no_isolates)$gender)

a_colors <- color_nodes(net_with_no_isolates,"gender",c("lightgoldenrod2","forestgreen"))
attr(a_colors, "map")

0      1
"#EEDC82" "#228B22"

## plot
set.seed(77)
gender <- nplot(
  net_with_no_isolates,
  vertex.color = color_nodes(net_with_no_isolates, "gender", c("lightgoldenrod2","forestgreen")),
  vertex.size.range = ifelse(V(net_with_no_isolates)$gender == 0, 10, 4),
  vertex.size = 10,
  edge.color = ~ego(alpha = 0.33, col = "gray") + alter(alpha = 0.33, col = "gray"),
  vertex.label = NULL,
  edge.line.breaks = 10
)

# add legend to graph
nplot_legend(
  gender,
  labels = c("Male", "Female"),
  pch = c(21,23),
  gp = gpar(
    fill = c("lightgoldenrod2","forestgreen")),
  packgrob.args = list(side = "bottom"),
  ncol = 2
)

grid.text("Split According to Gender", x = .2, y = .87, just = "bottom")

# now let's do the same with lunch period
V(net_with_no_isolates)$lunch <- as.factor(V(net_with_no_isolates)$lunch)

b_colors <- color_nodes(net_with_no_isolates,"lunch",c("purple","palegreen","steelblue"))
attr(b_colors, "map")

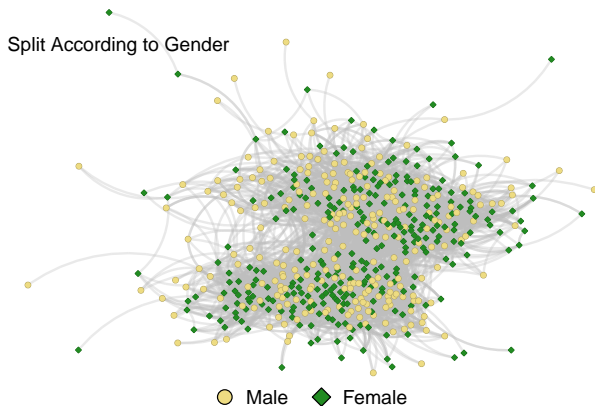
1      2      99
"#A020F0" "#98FB98" "#4682B4"

## plot
set.seed(77)
lunch <- nplot(
  net_with_no_isolates,
  vertex.color = color_nodes(net_with_no_isolates, "lunch", c("purple","palegreen","steelblue")),
  vertex.size.range = ifelse(V(net_with_no_isolates)$lunch == 1, 10, 4),
  vertex.size = 10,
  edge.color = ~ego(alpha = 0.33, col = "gray") + alter(alpha = 0.33, col = "gray"),
  vertex.label = NULL,
  edge.line.breaks = 10
)

# add legend to graph
nplot_legend(
  lunch,
  labels = c("First", "Second", "Other"),
  pch = c(23,24,21),
  gp = gpar(
    fill = c("purple","palegreen","steelblue")),
  packgrob.args = list(side = "bottom"),
  ncol = 3
)

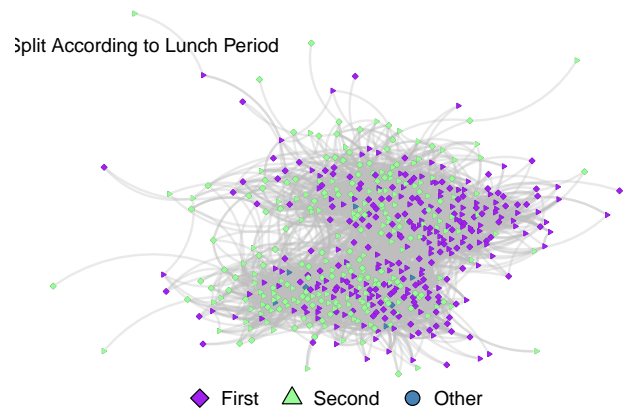
grid.text("Split According to Lunch Period", x = .2, y = .87, just = "bottom")

```



### Split according to Lunch Period.

Here is the data split according to the different lunch periods students might be in.



### Changing Lines to Dashes.

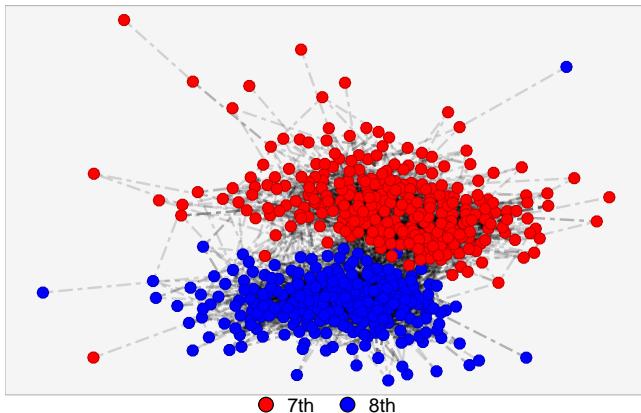
One of the perks of `netplot` is the ability to be fully customizable, right out of the box. First, here is an example of the same dataset, but the edges are dashed instead of full and straight instead of curved.

```
set.seed(77)

grades <- nplot(
  net_with_no_isolates,
  bg.col = "#F5F5F5",
  vertex.color = color_nodes(net_with_no_isolates, "grade", c("red", "blue")),
  vertex.size.range = c(0.02, 0.02),
  edge.color = ~ego(alpha = .15, col = "black") + alter(alpha = .15, col = "black"),
  vertex.label = NULL,
  edge.width.range = c(2,2),
  edge.line.lty = 6,
  edge.line.breaks = 1
)

# add legend to graph
grades_dashed <- nplot_legend(
  grades,
  labels = c("7th", "8th"),
  pch = c(21,21),
  gp = gpar(
    fill = c("red", "blue")),
  packgrob.args = list(side = "bottom"),
  ncol = 2
)

grades_dashed
```



### Colored Edges and Skipped Vertices

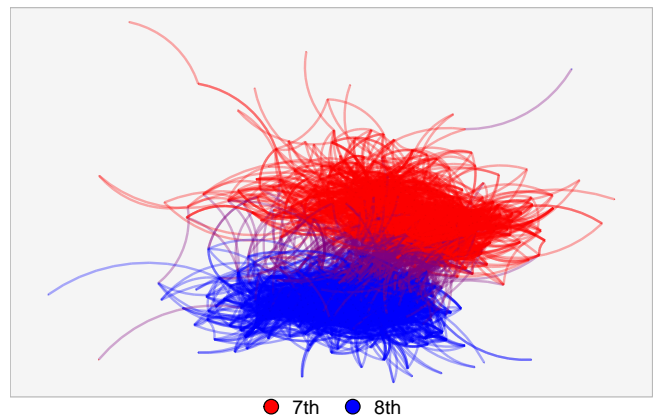
This selection shows how to skip vertices and add colors to the edges.

```
set.seed(77)

grades <- nplot(
  net_with_no_isolates,
  bg.col = "#F5F5F5",
  vertex.color = color_nodes(net_with_no_isolates, "grade", c("red", "blue")),
  vertex.size.range = c(0.0001, 0.0001),
  edge.color = ~ego(alpha = 0.33) + alter(alpha = 0.33, col = c("red", "blue")),
  vertex.label = NULL,
  edge.width.range = c(2,2),
  edge.line.breaks = 10
)

# add legend to graph
grades_edge_colored <- nplot_legend(
  grades,
  labels = c("7th", "8th"),
  pch = c(21,21),
  gp = gpar(
    fill = c("red", "blue")),
  packgrob.args = list(side = "bottom"),
  ncol = 2
)

grades_edge_colored
```



### Changing Background Color

We can also add a background to the plot, including a gradient.

```
set.seed(77)

grades <- nplot(
  net_with_no_isolates,
  bg.col = linearGradient(c("lightpink", "lightskyblue"))
```

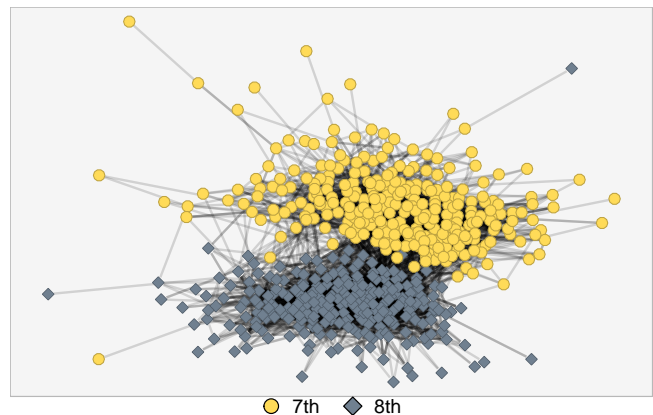
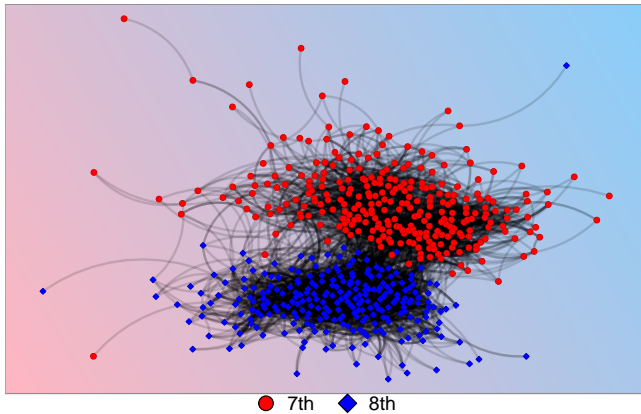
```

vertex.color = color_nodes(net_with_no_isolates, "grade", c("red", "blue")),
vertex.nsizes = ifelse(V(net_with_no_isolates)$grade == 7, 10, 4),
vertex.size.range = c(0.01, 0.01),
edge.color = ~ego(alpha = 0.15, col = "black") # alter(alpha = 0.15, col = "black"),
vertex.label = NULL,
edge.line.breaks = 10
)

# add legend to graph
grades_background <- nplot_legend(
  grades,
  labels = c("7th", "8th"),
  pch = c(21, 23),
  gp = gpar(
    fill = c("red", "blue")),
  packgrob.args = list(side = "bottom"),
  ncol = 2
)

grades_different_color

```



### Different Colors

We can also change vertex colors and edges to straight lines.

```

set.seed(77)

grades <- nplot(
  net_with_no_isolates,
  bg.col = "#F5F5F5",
  vertex.color = color_nodes(net_with_no_isolates, "grade", c("#FFDB58", "#708090")),
  vertex.nsizes = ifelse(V(net_with_no_isolates)$grade == 7, 10, 4),
  vertex.size.range = c(0.02, 0.02),
  edge.color = ~ego(alpha = .15, col = "black") + alter(alpha = .15, col = "black"),
  vertex.label = NULL,
  edge.width.range = c(2, 2),

```

*Example 2: LTCF data*