Appendix

Danai Avgerinou, Holly Capell, Shannon McNish, Taylor Pellerin, Kaya Tollas

4/22/2018

Marvel Social Network dataset

```
el <- read_csv("./data/hero-network.csv")
```

```
## Parsed with column specification:
## cols(
##   hero1 = col_character(),
##   hero2 = col_character()
## )
```

```
el = as.matrix(el)
marvel <- graph.edgelist(el,directed=FALSE)

wel <- read_csv("./data/weighted_edges.csv")
```

```
## Parsed with column specification:
## cols(
##   hero1 = col_character(),
##   hero2 = col_character(),
##   weight = col_integer()
## )
```

(a) What do the vertices and edges represent in this example?

Here, the nodes represent superheroes and edges represent them coexisting in a comic book together. In the raw df, there may be duplicate edges. In the weighted df, there are no duplicates.

(b) How many vertices and edges are there in the data set?

```
#Count the total number of nodes
vcount(marvel)
```

```
## [1] 6421
```

```
#Count the total number of edges
ecount(marvel)
```

```
## [1] 574467
```

```
dim(wel)
```

```
## [1] 167100      3
```

There are 574,467 total edges, including duplicates, 167,100 weighted deduplicated edges, and 6,421 nodes in this data set

(c) Is the network weighted? If so, what do the weights represent?

Yes, the node weights represent the number of comic books that each character appeared in. The edge weights represent the number of times that each character appeared in a comic book together.

(d) Is the network directed or undirected?

This network is undirected, since sharing space in a comic book is mutual.

(e1) How much storage will it require to store the network using sparse representation?

Using a sparse representation, we only need to store the edges and their weights, and which heroes are involved. Each edge in this edge list takes 2 heros and one weight. There are 167,100 weighted edges and so we only need to store 501,300 observations.

(e2) How about as an adjacency matrix?

In a weighted adjacency matrix, we would have to store (1/2)(n^2) - n weights where n = 6,206, the number of heros. This is because we have an undirected graph, so we only need the upper triangular quadrant of the matrix. We can also ignore the main diagonal, since we do not care if a character has self references. This comes out to 20,640,312 data points. If we also store the node weights here, we can put these on the "main diagonal"" of the adjacency matrix, leaving us with a total of (1/2)(n^2), or 20,646,738 entries to keep track of.

(f) What kinds of questions would you like to investigate for this application?

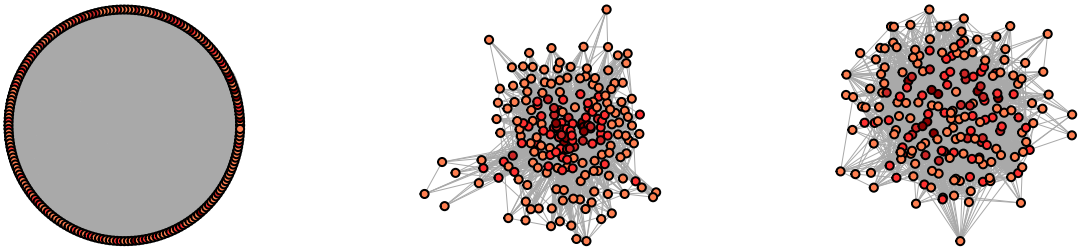
(g) Extract 3 induced subgraphs, each with 200 random nodes, from the dataset that you've looked at in (2). Visualize each of these networks using circular, Kamad-Kawai and Fructerman Reingold layouts (total of 9 plots). Consider what attributes in your chosen network could be used to recolor or resize the vertices. Are there any interesting network properties that become apparent when coloring or resizing the vertices?

```
wel <- read.csv("./data/weighted_edge_top_1000.csv")
wel <- subset(wel, select = c("hero1", "hero2", "weight"))
g <- graph.data.frame(wel, directed = F)
```

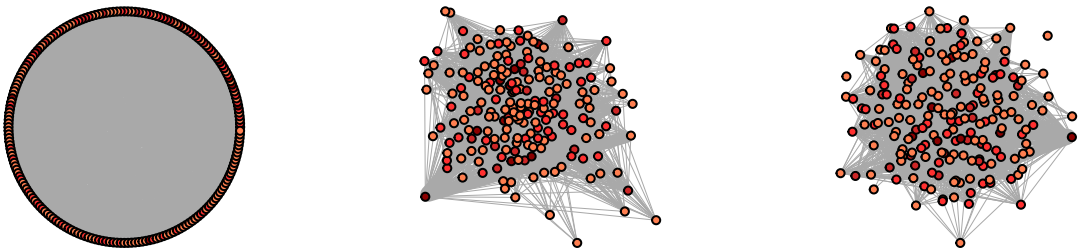
```
#import the sample_attributes
a <- read.csv("./data/weighted_nodes_top_1000.csv")

a$logged_weight <- log(a$weight)
labels <-  c(1,2,3,4)
a$weight_grp = cut(a$logged_weight, 4, include.lowest=TRUE, labels = labels)


V(g)$node_value <- as.numeric(a$weight_grp[match(V(g)$name,a$hero)])

V(g)$color=V(g)$node_value

V(g)$color=gsub("1","coral",V(g)$color)
V(g)$color=gsub("2","firebrick1",V(g)$color)
V(g)$color=gsub("3","firebrick3",V(g)$color)
V(g)$color=gsub("4","darkred",V(g)$color)
```

```
g1 <- induced.subgraph(g, sample(V(g), 200))
g2 <- induced.subgraph(g, sample(V(g), 200))
g3 <- induced.subgraph(g, sample(V(g), 200))
```

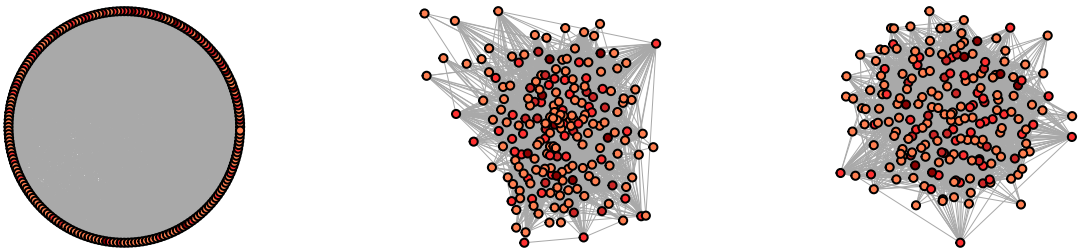
```
par(mfrow=c(1,3))
plot(g1, layout=layout.circle, edge.width=E(g1)$weight/50, vertex.label=NA, vertex.size=7)
plot(g1, layout=layout_with_fr(g1), edge.width=E(g1)$weight/50, vertex.label=NA, vertex.size=7)
plot(g1, layout=layout_with_kk(g1), edge.width=E(g1)$weight/50, vertex.label=NA, vertex.size=7)
```



```
par(mfrow=c(1,3))
plot(g2, layout=layout.circle, edge.width=E(g1)$weight/50, vertex.label=NA, vertex.size=7)
plot(g2, layout=layout_with_fr(g1), edge.width=E(g1)$weight/50, vertex.label=NA, vertex.size=7)
plot(g2, layout=layout_with_kk(g1), edge.width=E(g1)$weight/50, vertex.label=NA, vertex.size=7)
```



```
par(mfrow=c(1,3))
plot(g3, layout=layout.circle, edge.width=E(g1)$weight/50, vertex.label=NA, vertex.size=7)
plot(g3, layout=layout_with_fr(g1), edge.width=E(g1)$weight/50, vertex.label=NA, vertex.size=7)
plot(g3, layout=layout_with_kk(g1), edge.width=E(g1)$weight/50, vertex.label=NA, vertex.size=7)
```



(h) Run an exploratory analysis of the graph following along the visualizations, summary statistics, and community detection work shown in the Network Descriptions.Rmd file on Canvas. Use these techniques to answer the questions you posed in problem (1).

```
# Checking whether or not the network is connected
igraph::is.connected(marvel)
```

```
## [1] FALSE
```

The network is not connected.

```
# decompose the graph into its connected components
comps <- decompose.graph(marvel) #calculates the connected components

#count the number of vertices in each connected component
table(sapply(comps, vcount))
```

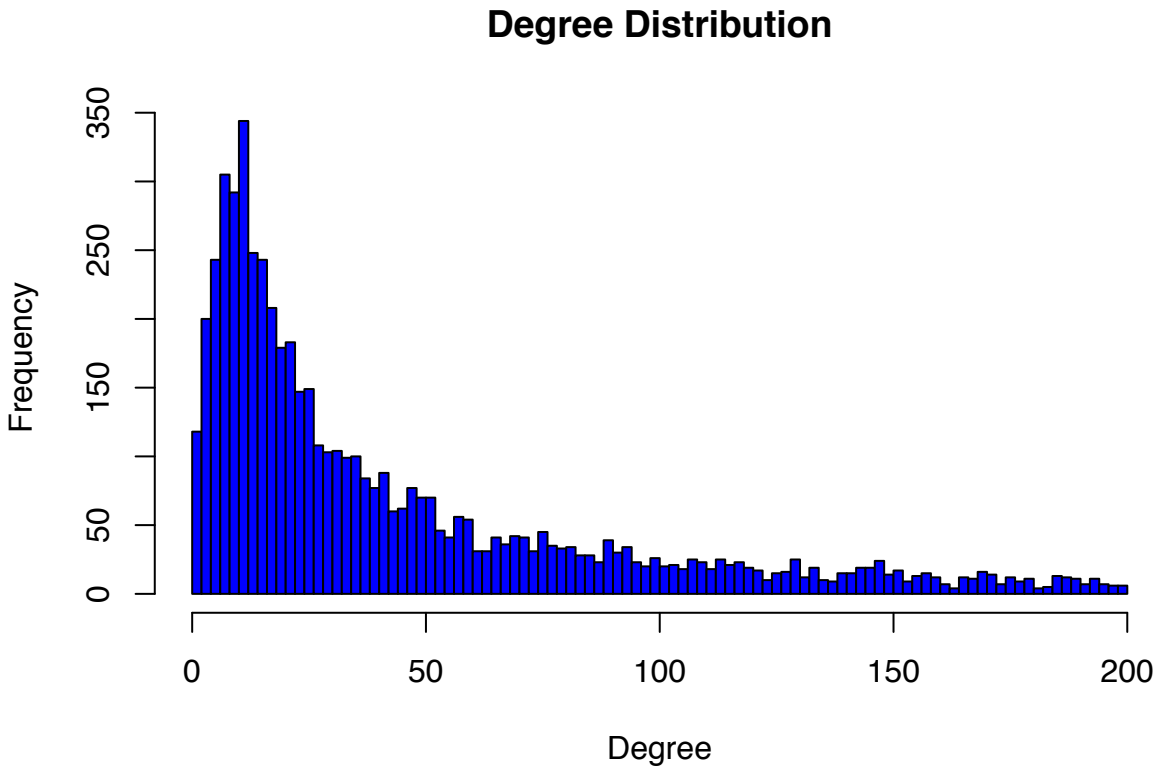
```
##
##    2    7    9 6403
##    1    1    1    1
```

The network has one large connected component and 3 smaller ones.

```r
d.marvel <- igraph::degree(marvel)  # this is the degree sequence

summary(d.marvel)  # get the distribution of the degrees
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    13.0    33.0   178.9   102.0 16499.0
```

```r
d.marvel.small <- d.marvel[d.marvel<200]  # clip outliers

#plot
hist(d.marvel.small,col="blue", breaks = 75,
     xlab="Degree", ylab="Frequency",
     main="Degree Distribution")
```



There are some very connected heros, with the node with Captain America having the max degree of 16,499 connections. The distribution with outliers clipped shows the degrees peak around degree 10.

```r
# get network diameter - max of shortest path distances excluding infinite paths

diameter(marvel)
```

```
## [1] 5
```

This shows us that the maximum Marvel Number (for our main connected component) is 5, meaning that someone knows someone who knows someone . . . who knows Ant Man.

Convert Network Edge Data

```r
# create igraph from edgelist
el <- read_csv("./data/hero-network.csv")
```

```
## Parsed with column specification:
## cols(
##   hero1 = col_character(),
##   hero2 = col_character()
## )
```

```
marvel.edgelist = as.matrix(el)

marvel.igraph <- graph.edgelist(marvel.edgelist, directed=FALSE)
# adjacency matrix from igraph
marvel.adjacency <- as_adj(marvel.igraph)
```

```
# create statnet object from edgelist
marvel.network <- network(marvel.edgelist)

# plot(marvel.network, main = paste(""), usearrows = TRUE, edge.col = "grey50")#, vertex.col = pblog.la
```

Network Centrality Measures and Visualization

Calculate In-Degree, Out-Degree, Eigenvector, Betweenness, and Closeness Centralities

```
#in degree centrality
in.degree.centrality <- colSums(as.matrix(marvel.adjacency))

#out degree centrality
out.degree.centrality <- rowSums(as.matrix(marvel.adjacency))

#eigenvector centrality
eigenvector.centrality <- eigen_centrality(marvel.igraph, directed = FALSE)$vector

#normalized betweenness centrality
betweenness.centrality <- estimate_betweenness(marvel.igraph, directed = TRUE, cutoff = 10)

#closeness centrality
closeness.centrality <- estimate_closeness(marvel.igraph, mode = "total", normalized = FALSE, cutoff =
```

```
## Warning in estimate_closeness(marvel.igraph, mode = "total", normalized =
## FALSE, : At centrality.c:2784 :closeness centrality is not well-defined for
## disconnected graphs
```

```
# These are all matrices...

# in.degree.centrality
# out.degree.centrality
# eigenvector.centrality
# betweenness.centrality
# closeness.centrality
```

Plot Network According to Centrality Measures.

```
# #plot in-degree
# plot(marvel.network, main = paste("In-Degree Centrality"), usearrows = TRUE, vertex.cex = in.degree.c
#
```

```
# #plot out-degree
# plot(marvel.network, main = paste("Out-Degree Centrality"), usearrows = TRUE, vertex.cex = out.degree
#
# #plot eigenvector
# plot(marvel.network, main = paste("Eigenvector Centrality"), usearrows = TRUE, vertex.cex = eigenvect
#
# #plot betweenness
# plot(marvel.network, main = paste("Betweenness Centrality"), usearrows = TRUE, vertex.cex = betweenne
#
# #plot closeness
# plot(marvel.network, main = paste("Closeness Centrality"), usearrows = TRUE, vertex.cex = closeness.c
```

Check out Ego network

```
##################################

#An ego network is nothing but the neighborhood of a given "ego" / node

#Ego network of the main instructor (node 1)
#ego <- induced.subgraph(marvel, igraph::neighborhood(marvel, 1, 'BLACK WIDOW/NATASHA')[[1]])

#get degrees of nodes
deg <- igraph::degree(marvel)
dSorted <- sort.int(deg,decreasing = T, index.return = F)

topDegreeNames <- dSorted[1:20]
```

```
degreetop20bet <- subset(betweenness.centrality, names(betweenness.centrality) %in% topDegreeNames)
degreetop20bet
```

```
## named numeric(0)
```

```
degreetop20Eigen <- subset(eigenvector.centrality, names(eigenvector.centrality) %in% topDegreeNames)
degreetop20Eigen
```

```
## named numeric(0)
```

Which heros are central to the network and how does that compare to the number of connections

```
#get degrees of nodes
deg <- igraph::degree(marvel.igraph)
dSorted <- sort.int(deg,decreasing = T, index.return = F)

#get top 20 nodes with most degrees
topDegree <- dSorted[1:20]
topDegreeNames <- names(topDegree)

t1 <- data.frame(Hero=names(topDegree), Degrees=topDegree, row.names=NULL)
kable(t1, caption= "Top 20 Heros by Number of Degrees")
```

Table 1: Top 20 Heros by Number of Degrees

Hero	Degrees
CAPTAIN AMERICA	16499
SPIDER-MAN/PETER PAR	13717
IRON MAN/TONY STARK	11817
THOR/DR. DONALD BLAK	11427
THING/BENJAMIN J. GR	10681
WOLVERINE/LOGAN	10353
HUMAN TORCH/JOHNNY S	10237
SCARLET WITCH/WANDA	9911
MR. FANTASTIC/REED R	9775
VISION	9696
INVISIBLE WOMAN/SUE	9326
BEAST/HENRY &HANK& P	9287
CYCLOPS/SCOTT SUMMER	9099
STORM/ORORO MUNROE S	8795
HAWK	8483
WASP/JANET VAN DYNE	8426
COLOSSUS II/PETER RA	7863
PROFESSOR X/CHARLES	7840
HULK/DR. ROBERT BRUC	7515
ANT-MAN/DR. HENRY J.	7343

Calculate eigenvector and betweenness centralities:

```
#eigenvector centrality
eigenvector.centrality <- eigen_centrality(marvel.igraph, directed = FALSE)$vector
eSorted <- sort.int(eigenvector.centrality,decreasing = T, index.return = F)[1:20]

#normalized betweenness centrality
betweenness.centrality <- estimate_betweenness(marvel.igraph, directed = TRUE, cutoff = 10)
bSorted <- sort.int(betweenness.centrality,decreasing = T, index.return = F)[1:20]
```

Look at betweenness and eigenvector centrailty of those in top 20 by degree

```
degreetop20bet <- subset(betweenness.centrality, names(betweenness.centrality) %in% topDegreeNames)
degreetop20bet
```

```
##  IRON MAN/TONY STARK HULK/DR. ROBERT BRUC SPIDER-MAN/PETER PAR
##            1067304.5             743858.0            2676603.3
##  WASP/JANET VAN DYNE                 HAWK  SCARLET WITCH/WANDA
##             205191.7             623125.4             415569.1
##      CAPTAIN AMERICA      WOLVERINE/LOGAN               VISION
##            1935707.5            1062680.3             281312.2
## STORM/ORORO MUNROE S MR. FANTASTIC/REED R THING/BENJAMIN J. GR
##             330653.0             565053.9             681995.6
##  INVISIBLE WOMAN/SUE BEAST/HENRY &HANK& P CYCLOPS/SCOTT SUMMER
##             366192.0             660047.4             324658.7
## THOR/DR. DONALD BLAK  PROFESSOR X/CHARLES HUMAN TORCH/JOHNNY S
##             770925.3             315234.1             539222.0
## COLOSSUS II/PETER RA ANT-MAN/DR. HENRY J.
##             230577.9             208261.8
```

```
degreetop20Eigen <- subset(eigenvector.centrality, names(eigenvector.centrality) %in% topDegreeNames)
degreetop20Eigen
```

```
##  IRON MAN/TONY STARK HULK/DR. ROBERT BRUC SPIDER-MAN/PETER PAR
##            0.7951503            0.3543299            0.4576505
##  WASP/JANET VAN DYNE                 HAWK  SCARLET WITCH/WANDA
##            0.6964646            0.6255454            0.7519413
##      CAPTAIN AMERICA      WOLVERINE/LOGAN               VISION
##            1.0000000            0.4673382            0.7407603
## STORM/ORORO MUNROE S MR. FANTASTIC/REED R THING/BENJAMIN J. GR
##            0.4360988            0.7980751            0.8266349
##  INVISIBLE WOMAN/SUE BEAST/HENRY &HANK& P CYCLOPS/SCOTT SUMMER
##            0.7681152            0.5575718            0.4824879
## THOR/DR. DONALD BLAK  PROFESSOR X/CHARLES HUMAN TORCH/JOHNNY S
##            0.7051696            0.4142344            0.8121594
## COLOSSUS II/PETER RA ANT-MAN/DR. HENRY J.
##            0.3787070            0.6047572
```

```
t2 <- data.frame(Hero=names(bSorted), Betweeness=bSorted, row.names=NULL)
kable(t2, caption= "Top 20 Heros by Betweeness Centrality")
```

Table 2: Top 20 Heros by Betweeness Centrality

Hero	Betweeness
SPIDER-MAN/PETER PAR	2676603.3
CAPTAIN AMERICA	1935707.5
IRON MAN/TONY STARK	1067304.5
WOLVERINE/LOGAN	1062680.3
DR. STRANGE/STEPHEN	846621.3
HAVOK/ALEX SUMMERS	806363.7
THOR/DR. DONALD BLAK	770925.3
HULK/DR. ROBERT BRUC	743858.0
THING/BENJAMIN J. GR	681995.6
DAREDEVIL/MATT MURDO	677435.4
BEAST/HENRY &HANK& P	660047.4
HAWK	623125.4
MR. FANTASTIC/REED R	565053.9
HUMAN TORCH/JOHNNY S	539222.0
FURY, COL. NICHOLAS	492337.6
SILVER SURFER/NORRIN	439868.2
SCARLET WITCH/WANDA	415569.1
SUB-MARINER/NAMOR MA	396431.1
PUNISHER II/FRANK CA	395655.4
SHE-HULK/JENNIFER WA	388712.0

```
t3 <- data.frame(Hero=names(eSorted), Eigenvector=eSorted, row.names=NULL)
kable(t3, caption= "Top 20 Heros by Eigenvector Centrality")
```

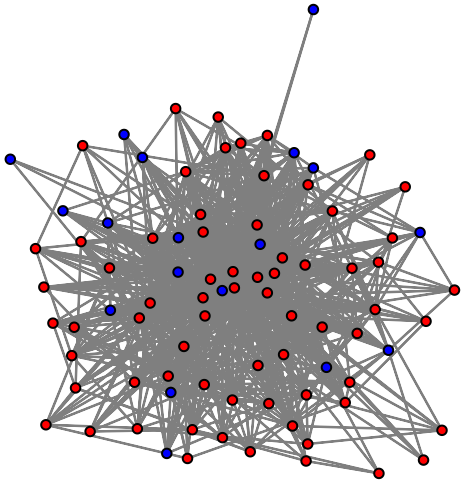
Table 3: Top 20 Heros by Eigenvector Centrality

Hero	Eigenvector
CAPTAIN AMERICA	1.0000000
THING/BENJAMIN J. GR	0.8266349

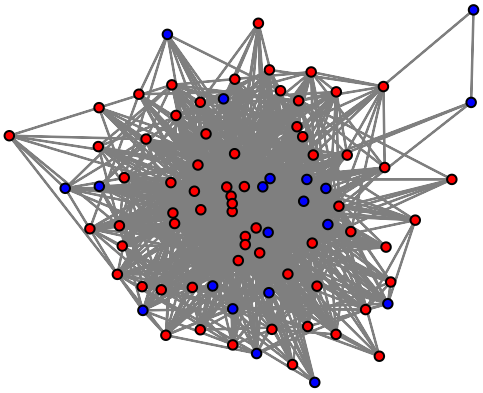
Hero	Eigenvector
HUMAN TORCH/JOHNNY S	0.8121594
MR. FANTASTIC/REED R	0.7980751
IRON MAN/TONY STARK	0.7951503
INVISIBLE WOMAN/SUE	0.7681152
SCARLET WITCH/WANDA	0.7519413
VISION	0.7407603
THOR/DR. DONALD BLAK	0.7051696
WASP/JANET VAN DYNE	0.6964646
HAWK	0.6255454
ANT-MAN/DR. HENRY J.	0.6047572
BEAST/HENRY &HANK& P	0.5575718
WONDER MAN/SIMON WIL	0.5073018
CYCLOPS/SCOTT SUMMER	0.4824879
WOLVERINE/LOGAN	0.4673382
SPIDER-MAN/PETER PAR	0.4576505
STORM/ORORO MUNROE S	0.4360988
PROFESSOR X/CHARLES	0.4142344
SHE-HULK/JENNIFER WA	0.4059961

PLOTS

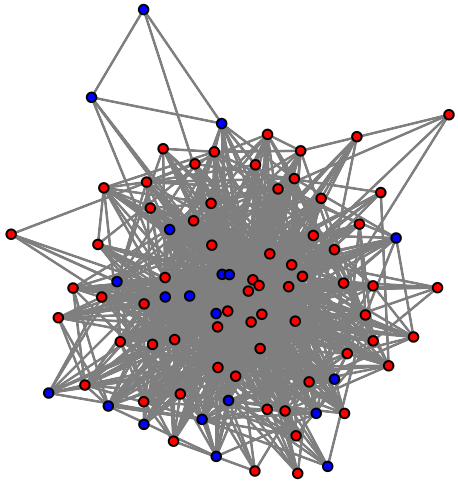
```
# Betweenness Centrality

bSorted <- sort.int(betweenness.centrality,decreasing = T, index.return = F)

topBetweenNames <- bSorted[1:20]
top20 <- names(topBetweenNames)

# Create list of top 100 heros
top100 <- bSorted[1:100]
top100 <- names(top100)

# Create attribute of whether or not hero is in our top 20
a2 <- as.matrix(a$hero)
a2 <- replace(a2, a2 %in% top20, 4)
a2 <- replace(a2, !(a2 %in% top20) & a2 != 4, 2)

df <- data.frame(a$hero, a2)

# Create subgraph of top 100 heroes

top_100 <- a$hero %in% top100
g_small <- induced.subgraph(g, V(g)[top_100])

g.edgelist <- as_edgelist(g_small)
small.network <- network(g.edgelist, directed = F)
plot(small.network, vertex.col = a2, edge.col = "grey50")
```



```r
# Degree

dSorted <- sort.int(deg,decreasing = T, index.return = F)

topDegreeNames <- dSorted[1:20]
top20 <- names(topDegreeNames)

# Create list of top 100 heros
top100 <- dSorted[1:100]
top100 <- names(top100)

# Create attribute of whether or not hero is in our top 20
a2 <- as.matrix(a$hero)
a2 <- replace(a2, a2 %in% top20, 4)
a2 <- replace(a2, !(a2 %in% top20) & a2 != 4, 2)

df <- data.frame(a$hero, a2)

# Create subgraph of top 100 heroes

top_100 <- a$hero %in% top100
g_small <- induced.subgraph(g, V(g)[top_100])

g.edgelist <- as_edgelist(g_small)
small.network <- network(g.edgelist, directed = F)
plot(small.network, vertex.col = a2, edge.col = "grey50")
```



```r
# Eigenvector Centrality

eSorted <- sort.int(eigenvector.centrality,decreasing = T, index.return = F)

topEigNames <- eSorted[1:20]
top20 <- names(topEigNames)

# Create list of top 100 heros
top100 <- eSorted[1:100]
top100 <- names(top100)

# Create attribute of whether or not hero is in our top 20
a2 <- as.matrix(a$hero)
a2 <- replace(a2, a2 %in% top20, 4)
a2 <- replace(a2, !(a2 %in% top20) & a2 != 4, 2)

df <- data.frame(a$hero, a2)

# Create subgraph of top 100 heroes

top_100 <- a$hero %in% top100
g_small <- induced.subgraph(g, V(g)[top_100])

g.edgelist <- as_edgelist(g_small)
small.network <- network(g.edgelist, directed = F)
plot(small.network, vertex.col = a2, edge.col = "grey50")
```



HW2 Q2

Danai Avgerinou, Holly Capell, Shannon McNish, Taylor Pellerin, Kaya Tollas

4/22/2018

```
el <- read_csv("~/CompStats/HW/hw_2/hero-network.csv")
```

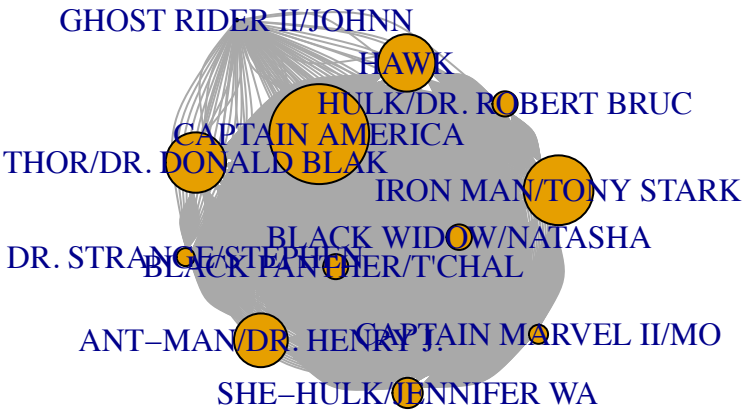
```
## Parsed with column specification:
## cols(
##   hero1 = col_character(),
##   hero2 = col_character()
## )
```

```
el = as.matrix(el)

marvel <- graph.edgelist(el,directed=FALSE)
```

Avenger analysis

Below we investigate the universe of the Avengers as compared to Captain America's neighborhood. This will give us insight into the relative importance of the Avengers

```
marvel_nodes <-  V(marvel)
captain_america <- which(marvel_nodes$name == 'CAPTAIN AMERICA')
thor <- which(marvel_nodes$name == 'THOR/DR. DONALD BLAK')
iron_man <- which(marvel_nodes$name == 'IRON MAN/TONY STARK')
black_panther <- which(marvel_nodes$name == 'BLACK PANTHER/T\'CHAL')
captain_marvel <- which(marvel_nodes$name == 'CAPTAIN MARVEL II/MO')
ghost_rider <- which(marvel_nodes$name == 'GHOST RIDER II/JOHNN')
she_hulk <- which(marvel_nodes$name == 'SHE-HULK/JENNIFER WA')
hulk <- which(marvel_nodes$name == 'HULK/DR. ROBERT BRUC')
black_widow <- which(marvel_nodes$name == 'BLACK WIDOW/NATASHA')
dr_strange <- which(marvel_nodes$name == 'DR. STRANGE/STEPHEN')
ant_man <- which(marvel_nodes$name == 'ANT-MAN/DR. HENRY J.')
hawk <- which(marvel_nodes$name == 'HAWK')

avenger_ids <- c(captain_america, thor, iron_man, black_panther, black_widow, captain_marvel, ghost_rid
avengers <-  induced.subgraph(marvel, avenger_ids)
plot(avengers, layout = layout.kamada.kawai, vertex.size = degree(avengers)/50)
```




```
betweenness(avengers) %>% sort(decreasing=T)
```

```
##      CAPTAIN AMERICA  BLACK WIDOW/NATASHA THOR/DR. DONALD BLAK
##           0.59587143           0.35689566           0.26439218
##  IRON MAN/TONY STARK  DR. STRANGE/STEPHEN ANT-MAN/DR. HENRY J.
##           0.26217077           0.13692511           0.13689140
##                 HAWK HULK/DR. ROBERT BRUC BLACK PANTHER/T'CHAL
##           0.12020409           0.10887434           0.01777503
## SHE-HULK/JENNIFER WA GHOST RIDER II/JOHNN CAPTAIN MARVEL II/MO
##           0.00000000           0.00000000           0.00000000
```

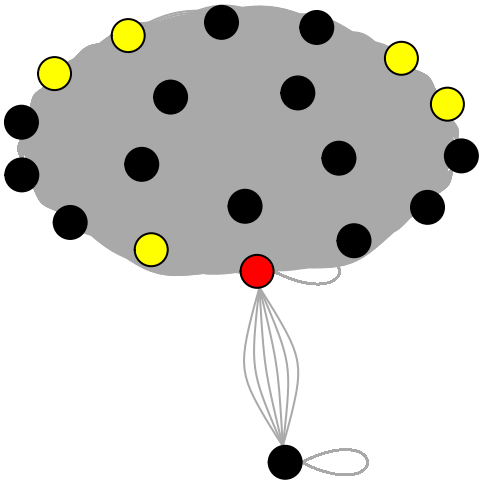
```
eigen_centrality(avengers)$vector %>% sort(decreasing=T)
```

```
##      CAPTAIN AMERICA  IRON MAN/TONY STARK THOR/DR. DONALD BLAK
##            1.0000000            0.8130392            0.7133454
##                 HAWK ANT-MAN/DR. HENRY J. SHE-HULK/JENNIFER WA
##            0.6736223            0.6413664            0.3387975
## BLACK PANTHER/T'CHAL  BLACK WIDOW/NATASHA HULK/DR. ROBERT BRUC
##            0.3069200            0.2990746            0.2497170
## CAPTAIN MARVEL II/MO  DR. STRANGE/STEPHEN GHOST RIDER II/JOHNN
##            0.2230927            0.1673285            0.0232608
```

Now we will visualize Captain America's ego network. This is a subgraph of popular heroes connected to Captain America who are also highly-connected in the network.

```
ego.captain <- induced.subgraph(marvel, igraph::neighborhood(marvel, 1, 'CAPTAIN AMERICA')[[1]])
vertices.captain <-  V(ego.captain)
friends <- degree(ego.captain) %>% sort(decreasing=T)
irrelevant <- degree(ego.captain) < friends[20] #only keep his most connected friends
ego.cap <- delete_vertices(ego.captain, vertices.captain[irrelevant])
vertices.captain <-  V(ego.cap)

captain_america <- which(vertices.captain$name == 'CAPTAIN AMERICA')
thor <- which(vertices.captain$name == 'THOR/DR. DONALD BLAK')
iron_man <- which(vertices.captain$name == 'IRON MAN/TONY STARK')
black_widow <- which(vertices.captain$name == 'BLACK WIDOW/NATASHA')
black_panther <- which(vertices.captain$name == 'BLACK PANTHER/T\'CHAL')
captain_marvel <- which(vertices.captain$name == 'CAPTAIN MARVEL II/MO')
ghost_rider <- which(vertices.captain$name == 'GHOST RIDER II/JOHNN')
she_hulk <- which(vertices.captain$name == 'SHE-HULK/JENNIFER WA')
hulk <- which(vertices.captain$name == 'HULK/DR. ROBERT BRUC')
black_widow <- which(vertices.captain$name == 'BLACK WIDOW/NATASHA')
```

```
dr_strange <- which(vertices.captain$name == 'DR. STRANGE/STEPHEN')
ant_man <- which(vertices.captain$name == 'ANT-MAN/DR. HENRY J.')
hawk <- which(vertices.captain$name == 'HAWK')

avenger_ids <- c(captain_america, thor, iron_man, black_panther, hulk, black_widow, captain_marvel, ghos

isAvenger <- rep(1,vcount(ego.cap))
isAvenger[avenger_ids] <- 2
isAvenger[captain_america] <- 3

# specify a vector of 2 colors for Avengers or other
colorvec <- c('black','yellow', 'red')
# assign colors to nodes
V(ego.cap)$color <- colorvec[isAvenger]
plot(ego.cap, layout = layout.kamada.kawai, vertex.label = NA)
```



Now we check closeness centrality for Captain America's co-hero network.

```
closeness(ego.cap, mode = c("out", "in", "all", "total"),
  weights = NULL, normalized = FALSE)
```

```
##  IRON MAN/TONY STARK HULK/DR. ROBERT BRUC SPIDER-MAN/PETER PAR
##           0.05000000           0.05000000           0.05000000
##  WASP/JANET VAN DYNE                 HAWK  SCARLET WITCH/WANDA
##           0.05000000           0.05000000           0.05000000
##      CAPTAIN AMERICA      WOLVERINE/LOGAN               VISION
##           0.05263158           0.05000000           0.05000000
## STORM/ORORO MUNROE S MR. FANTASTIC/REED R THING/BENJAMIN J. GR
##           0.05000000           0.05000000           0.05000000
##  INVISIBLE WOMAN/SUE BEAST/HENRY &HANK& P CYCLOPS/SCOTT SUMMER
##           0.05000000           0.05000000           0.05000000
## THOR/DR. DONALD BLAK  PROFESSOR X/CHARLES HUMAN TORCH/JOHNNY S
##           0.05000000           0.05000000           0.05000000
## ANT-MAN/DR. HENRY J.    PATRIOT/JEFF MACE
##           0.05000000           0.02702703
```

It appears everyone is approximately equidistant on this network.

Now we look at eigenvector centrality. It is interesting to note that after Captain America, the most central nodes are all part of the Fantastic Four. A few X-Men show up here too, which confirms that those leagues

are the most popular and influential in the MCU.

```
eigenvector.centrality <- eigen_centrality(ego.cap, directed = FALSE)$vector
eigenvector.centrality %>% sort(decreasing=T)
```

```
##      CAPTAIN AMERICA THING/BENJAMIN J. GR HUMAN TORCH/JOHNNY S
##           1.00000000           0.97710945           0.96919580
## MR. FANTASTIC/REED R  INVISIBLE WOMAN/SUE  IRON MAN/TONY STARK
##           0.96672860           0.92793912           0.81591066
##  SCARLET WITCH/WANDA               VISION  WASP/JANET VAN DYNE
##           0.74739953           0.73365024           0.71218653
## THOR/DR. DONALD BLAK                 HAWK ANT-MAN/DR. HENRY J.
##           0.70212065           0.63019048           0.62260462
## BEAST/HENRY &HANK& P SPIDER-MAN/PETER PAR HULK/DR. ROBERT BRUC
##           0.41757826           0.37056642           0.30975186
## CYCLOPS/SCOTT SUMMER      WOLVERINE/LOGAN  PROFESSOR X/CHARLES
##           0.29683752           0.28458036           0.26377313
## STORM/ORORO MUNROE S    PATRIOT/JEFF MACE
##           0.26174395           0.01520648
```

Community_Detection

April 24, 2018

In [1]:
```
import networkx as nx
import collections
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import csv
from plotly.offline import download_plotlyjs, init_notebook_mode,  iplot, plot
import community
from operator import itemgetter
init_notebook_mode(connected=True)
```

helpful link: http://programminghistorian.github.io/ph-submissions/lessons/published/exploring-and-analyzing-network-data-with-python

In [2]:
```
with open('/Users/danaiavg/Desktop/marvel-comic-social-network/data/weighted_edge_top_
    nodereader = csv.reader(nodecsv) # Read the csv
    # Retrieve the data (using Python list comprhension and list slicing to remove the
    nodes = [n for n in nodereader]
```

In [3]:
```
nodes = nodes[1:]
```

In [4]:
```
for row in nodes:
    del row[0]
```

In [5]:
```
node1 = [n[0] for n in nodes]
node2 = [n[1] for n in nodes]
node1.extend(node2)
my_nodes = list(set(node1))
my_nodes.sort()
```

In [6]:
```
G=nx.Graph()
```

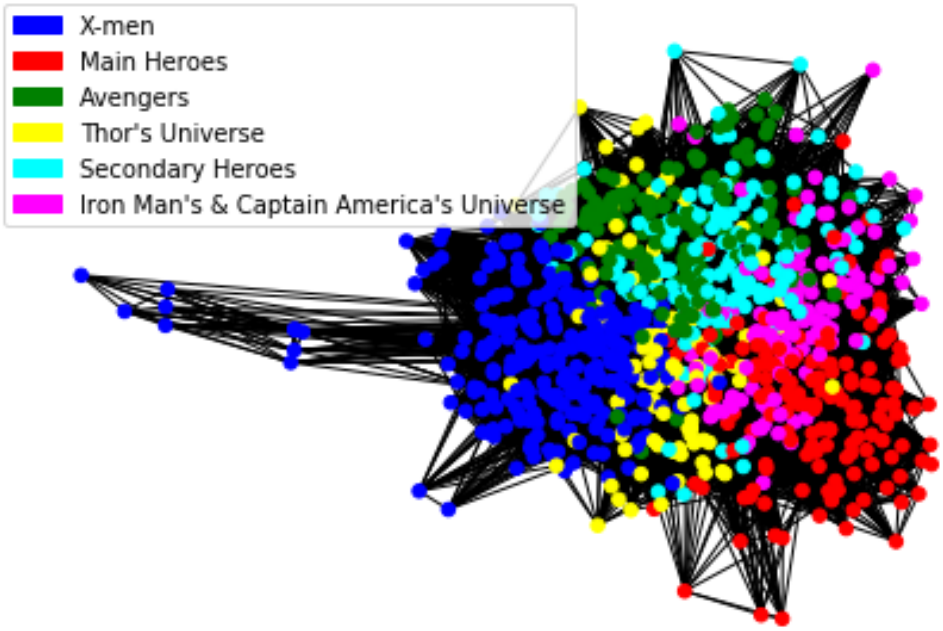
In [7]:
```
my_edges=nodes
for row in my_edges:
    del row[2]
```

In [8]:
```
G.add_nodes_from(my_nodes)
G.add_edges_from(my_edges)
```

```
In [9]: print(nx.is_connected(G))
```

```
True
```

```
In [10]: from community import community_louvain
         communities = community_louvain.best_partition(G)

In [11]: nx.set_node_attributes(G, 'modularity', communities)

In [12]: values = [communities.get(node) for node in G.nodes()]

         values1 = []
         for i in values:
             if i == 0:
                 values1.append('blue')
             elif i ==1:
                 values1.append('red')
             elif i ==2:
                 values1.append('green')
             elif i ==3:
                 values1.append('yellow')
             elif i ==4:
                 values1.append('cyan')
             elif i ==5:
                 values1.append('magenta')

In [16]: import matplotlib.patches as mpatches

         nx.draw_spring(G, node_color = values1, node_size=30, with_labels=False)
         plt.legend(handles=[mpatches.Patch(color='blue', label='X-men'),
                             mpatches.Patch(color='red', label='Main Heroes'),
                             mpatches.Patch(color='green', label='Avengers'),
                             mpatches.Patch(color='yellow', label="Thor's Universe"),
                             mpatches.Patch(color='cyan', label='Secondary Heroes'),
                             mpatches.Patch(color='magenta', label="Iron Man's & Captain Ameri
                            ])

         plt.savefig("/Users/danaiavg/Downloads/shit.png", format="PNG")
```




```
In [14]: # Iron Man/ Captain America Universe
         print(communities['IRON MAN/TONY STARK'])
         print(communities['CAPTAIN AMERICA'])
         print(communities['PATRIOT/JEFF MACE'])
```

```
5
5
5
```

```
In [15]: # Avenger
         print(communities['SCARLET WITCH/WANDA'])
         print(communities['BLACK WIDOW/NATASHA'])
         print(communities['HULK/DR. ROBERT BRUC'])
         print(communities['SHE-HULK/JENNIFER WA'])
         print(communities['ANT-MAN/DR. HENRY J.'])
         print(communities['VISION'])
         print(communities['THING/BENJAMIN J. GR'])
         print(communities['INVISIBLE WOMAN/SUE'])
```

```
2
2
2
```

```
2
2
2
2
2
```

In [17]:
```
# Xmen
print(communities['WOLVERINE/LOGAN'])
print(communities['STORM/ORORO MUNROE S'])
print(communities['CYCLOPS/SCOTT SUMMER'])
print(communities['PROFESSOR X/CHARLES'])
print(communities['DEADPOOL/JACK/WADE W'])
print(communities['ROGUE /'])
print(communities['BEAST/HENRY &HANK& P'])
```

```
0
0
0
0
0
0
0
```

In [18]:
```
# Main Heroes
print(communities['DAREDEVIL/MATT MURDO'])
print(communities['SPIDER-MAN/PETER PAR'])
print(communities['MR. FANTASTIC/REED R'])
print(communities['HUMAN TORCH/JOHNNY S'])
```

```
1
1
1
1
```

In [ ]:
```
# [k for k,v in communities.items() if v ==5]
```