# Social Network Analysis (Team One)

# Social Network Analysis: Amazon Purchases

```
#By Team One: Pam Askins, Mason Hansen, Abhinav Kashyap, Shiyi Li, Xinfang Zhan
g


#install.packages("igraph")
#install.packages("dplyr")
#install.packages("corrplot")
#install.packages("RColorBrewer")
library(igraph)
library(dplyr)
library(corrplot)
library(RColorBrewer)
```

# Step 1: Books Only

```
#Data Prep
setwd("/Users/mason/Desktop/MSBA/Fall18/BANA277_f18/")
products <- read.csv("products.csv", header=T)
co_purchase <- read.csv("copurchase.csv", header=T)

# slicing the products table to get the subset that are just Books
bookproducts <- products[products$group=="Book",]

# reducing the subset to include only those with salesranks within 0-150000, in
clusive
bookproducts <- bookproducts[bookproducts$salesrank<=150000 & bookproducts$sale
srank>=0,]

# slicing the copurchase table so that it only includes copurchases that involv
ed books
copurchase_books <- co_purchase[co_purchase$Source %in% bookproducts$id & co_pu
rchase$Target %in% bookproducts$id,]

# converting all ids to characters
bookproducts$id <- as.character(bookproducts$id)
copurchase_books$Source <- as.character(copurchase_books$Source)
copurchase_books$Target <- as.character(copurchase_books$Target)

# examining new structures and dataframes
str(bookproducts)
```

```
## 'data.frame':    35250 obs. of  7 variables:
##  $ id        : chr  "12" "33" "39" "45" ...
##  $ title     : Factor w/ 246817 levels ", said the shotgun to the head.",..:
68416 57846 135896 24322 168265 236805 197233 226298 153931 110825 ...
##  $ group     : Factor w/ 9 levels "Baby Product",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ salesrank : num  24741 97166 57186 48408 27507 ...
##  $ review_cnt: int  12 4 22 4 2 11 3 12 4 2 ...
##  $ downloads : int  12 4 22 4 2 11 3 12 4 2 ...
##  $ rating    : num  4.5 5 3.5 4 4 4.5 4.5 4.5 5 2.5 ...
```

```
str(copurchase_books)
```

```
## 'data.frame':    22460 obs. of  2 variables:
##  $ Source: chr  "12" "74" "77" "79" ...
##  $ Target: chr  "261" "282" "422" "82" ...
```

```
nrow(bookproducts)
```

```
## [1] 35250
```

```
# looking to see if there are duplicate copurchases which there are not
nrow(copurchase_books); nrow(unique(copurchase_books[,c("Source", "Target")]))
```

```
## [1] 22460
```

```
## [1] 22460
```

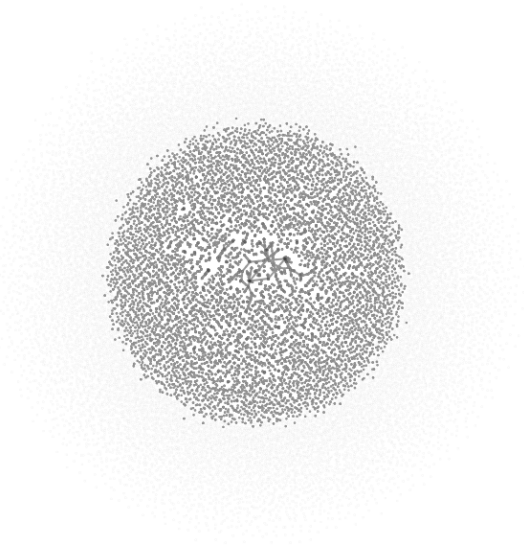
```
# organizing the copurchase dataframe by Source, then Target
copurchase_books <- copurchase_books[order(copurchase_books$Source, copurchase_
books$Target),]

# creating the network using the bookproducts as nodes and copurchase relations
hips as edges
net <- graph_from_data_frame(d=copurchase_books, vertices=bookproducts, directe
d=T)
#net <- simplify(net, remove.multiple = F, remove.loops = T) # not necessary be
cause no dups
# examining the nodes and edges
V(net)
```

```
## + 35250/35250 vertices, named, from 125a013:
##     [1] 12     33     39     45     74     77     78     79     82
##    [10] 105    110    117    120    121    127    130    131    136
##    [19] 137    147    148    170    173    185    187    193    196
##    [28] 199    203    215    224    244    249    255    256    257
##    [37] 261    265    268    269    274    275    278    282    296
##    [46] 302    307    321    322    326    333    335    338    343
##    [55] 344    347    393    396    399    413    422    426    435
##    [64] 439    448    450    472    494    495    498    510    527
##    [73] 543    556    566    569    577    578    582    583    596
##    [82] 603    611    626    633    640    649    650    652    654
## + ... omitted several vertices
```

```
E(net)
```

```
## + 22460/22460 edges from 125a013 (vertex names):
##  [1] 10001 ->18016  100013->94857  10003 ->10004  100030->100032
##  [5] 100030->103483 100032->100030 100032->103483 100038->70463
##  [9] 10004 ->10003  100048->136574 100049->101647 100087->78223
## [13] 100092->42211  100099->77013  100113->119171 100122->101136
## [17] 100130->110237 100153->93866  100179->73678  100182->76870
## [21] 100187->83521  100192->142537 100192->95038  100197->142832
## [25] 100197->154843 100210->136340 100267->104366 100274->114409
## [29] 100292->90863  100333->124238 100333->143508 100333->92080
## [33] 100375->95869  100377->94558  100398->117804 100402->104773
## [37] 100402->111049 100416->129350 100442->77441  100457->127072
## + ... omitted several edges
```

```
#plotting the network which looks like a giant sphere with concentration around
 a donut ring
plot(net, edge.arrow.size=.01, edge.curved=0, edge.arrow.width=.02, vertex.size
=.01,
     vertex.label=NA,
     vertex.color="orange", vertex.frame.color="#555555")
```



# Step 2 through 4: Degrees

```
# Examinging information about the nodes in order to determine a subcomponent w
ith most links
in_degree = degree(net, mode = 'in')
out_degree = degree(net, mode = 'out')
tot_degree = degree(net, mode = 'all')
max(in_degree)
```

```
## [1] 53
```

```
max(out_degree)
```

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

```
max(tot_degree)
```

```
## [1] 53
```

```
# finding the node with highest total degrees
maxdegreenode <- V(net)$name[degree(net)==max(tot_degree)]

# creating a subcomponent of the first node with highest degrees
sub <- subcomponent(net, maxdegreenode[1], mode = "all")

# generating a subnetwork based on subcomponent
subg <- induced_subgraph(net, sub)

#saving the node names and putting them into a vector
#subid<- V(subg)$name
V(subg)$label <- V(subg)$name
subid <- V(subg)$label

#examining the new nodes and edges
V(subg)
```

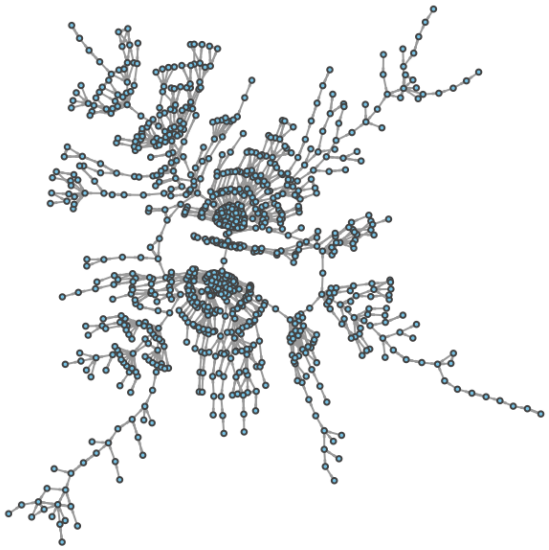
```
## + 904/904 vertices, named, from 16daff5:
##   [1] 33     77     78     130    148    187    193    224    302    321
##  [11] 322    422    448    556    577    626    724    976    1051   1201
##  [21] 1322   1644   1673   1817   1822   1851   1971   1985   2057   2071
##  [31] 2145   2161   2210   2279   2285   2326   2330   2332   2343   2345
##  [41] 2423   2470   2501   2505   2558   2563   2572   2603   2657   2658
##  [51] 2806   2807   2895   2959   3032   3119   3155   3191   3217   3306
##  [61] 3427   3588   3670   3737   3861   3909   4002   4014   4038   4068
##  [71] 4099   4140   4174   4184   4185   4197   4205   4222   4223   4319
##  [81] 4345   4429   4977   4993   4994   5018   5059   5137   5163   5164
##  [91] 5293   5355   5388   5410   5623   5638   5639   5655   5670   5821
## + ... omitted several vertices
```

```
E(subg)
```

```
## + 1173/1173 edges from 16daff5 (vertex names):
##  [1] 77  ->422  130 ->78   148 ->302  187 ->78   187 ->321  187 ->322
##  [7] 193 ->224  224 ->33   224 ->193  321 ->78   321 ->187  321 ->322
## [13] 322 ->78   322 ->187  322 ->321  422 ->77   422 ->1644 556 ->78
## [19] 577 ->33   626 ->33   724 ->302  1051->302  1644->422  1644->5293
## [25] 1817->976  1822->193  1822->724  1851->78   1971->193  2071->3155
## [31] 2210->2279 2210->2285 2279->2210 2279->2326 2285->2330 2326->193
## [37] 2326->2210 2330->2343 2330->2345 2332->4140 2343->2285 2343->2330
## [43] 2423->5410 2470->556  2501->3588 2505->2501 2558->33   2572->4184
## [49] 2572->4185 2657->2658 2658->77   2806->2807 2807->302  2959->1673
## [55] 3032->2558 3119->976  3191->2279 3217->4319 3306->2071 3306->4345
## + ... omitted several edges
```

# Step 5: Graph + Diameter

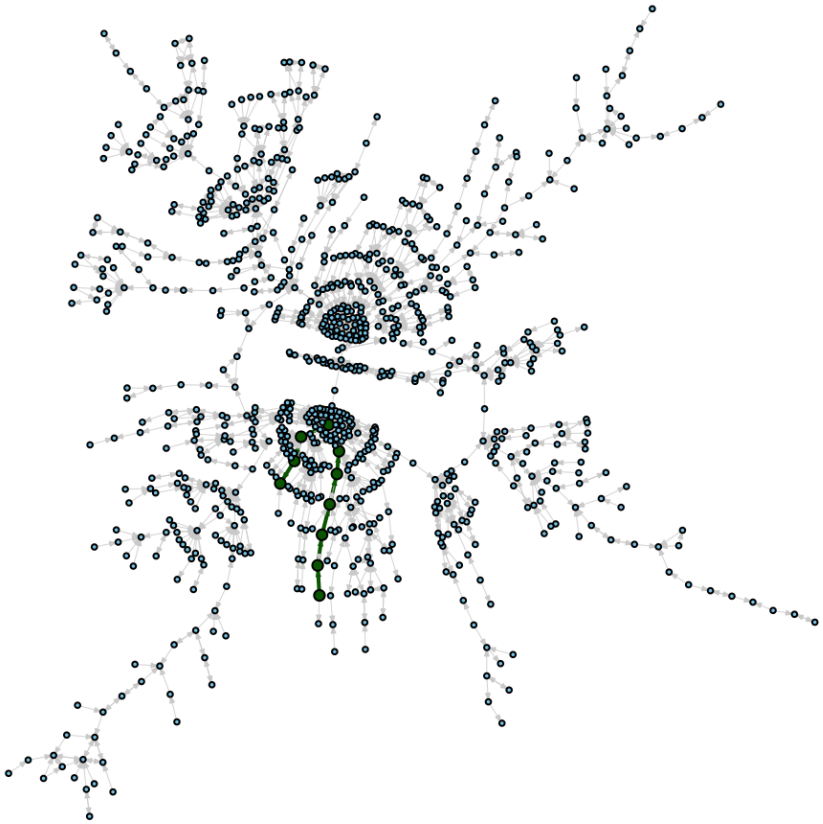
```
# graphing the new subcomponent
set.seed(123)
plot(subg, edge.arrow.size=.1, edge.curved=0, edge.arrow.width=.2, vertex.size=
2,
     vertex.label=NA,
     vertex.color="skyblue", vertex.frame.color="#555555",
     layout = layout.kamada.kawai)
```



```
# identifying the diameter, finding and coloring its path so that it will be vi
sible in graph
diameter(subg, directed=T, weights=NA)
```

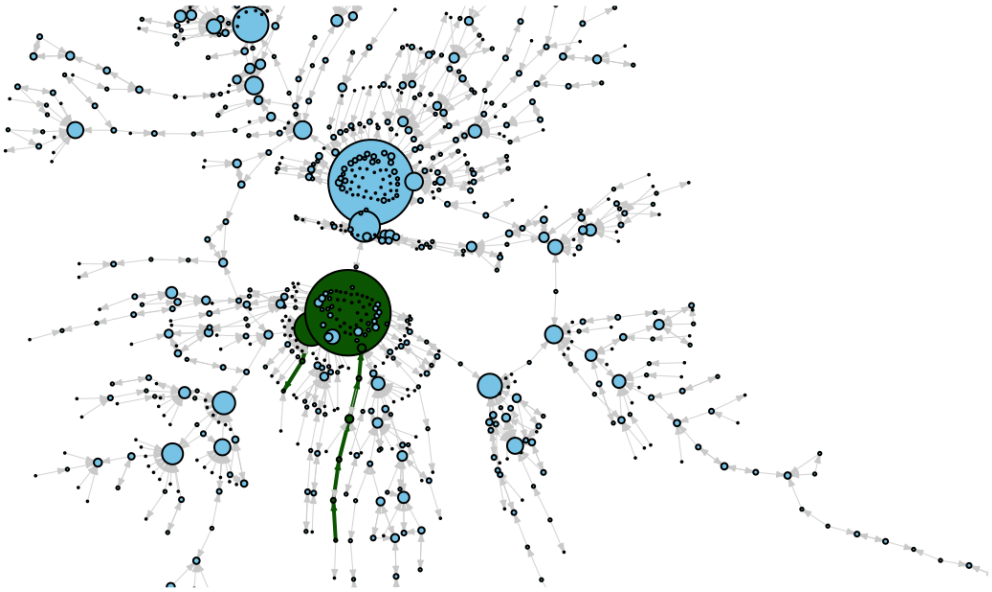
```
## [1] 9
```

```
nodes.diameter <- get.diameter(subg)

# setting colors for default nodes & edges of graph and then the diameter size
 & color as different
V(subg)$color<-"skyblue"
V(subg)$size <- 2
V(subg)[nodes.diameter]$color<-"darkgreen"
V(subg)[nodes.diameter]$size<-4
E(subg)$color<-"lightgrey"
E(subg)$width <- .4
E(subg,path=nodes.diameter)$color<-"darkgreen"
E(subg,path=nodes.diameter)$width<-2

# investigating different standard layouts to determine which is best
# by assiging the layout to a variable, stabilizes the graph for comparison
l <- layout_with_kk(subg)
#l <- layout_with_lgl(subg)
#l<- layout_with_fr(subg)

# setting norms to spread out graph
l <- norm_coords(l, ymin=-1, ymax=1, xmin=-1, xmax = 1)

# graph showing the diameter
plot(subg, edge.arrow.size=.15,
     rescale=F, layout=l*1.5,
     vertex.label=NA,
     layout=l)
```

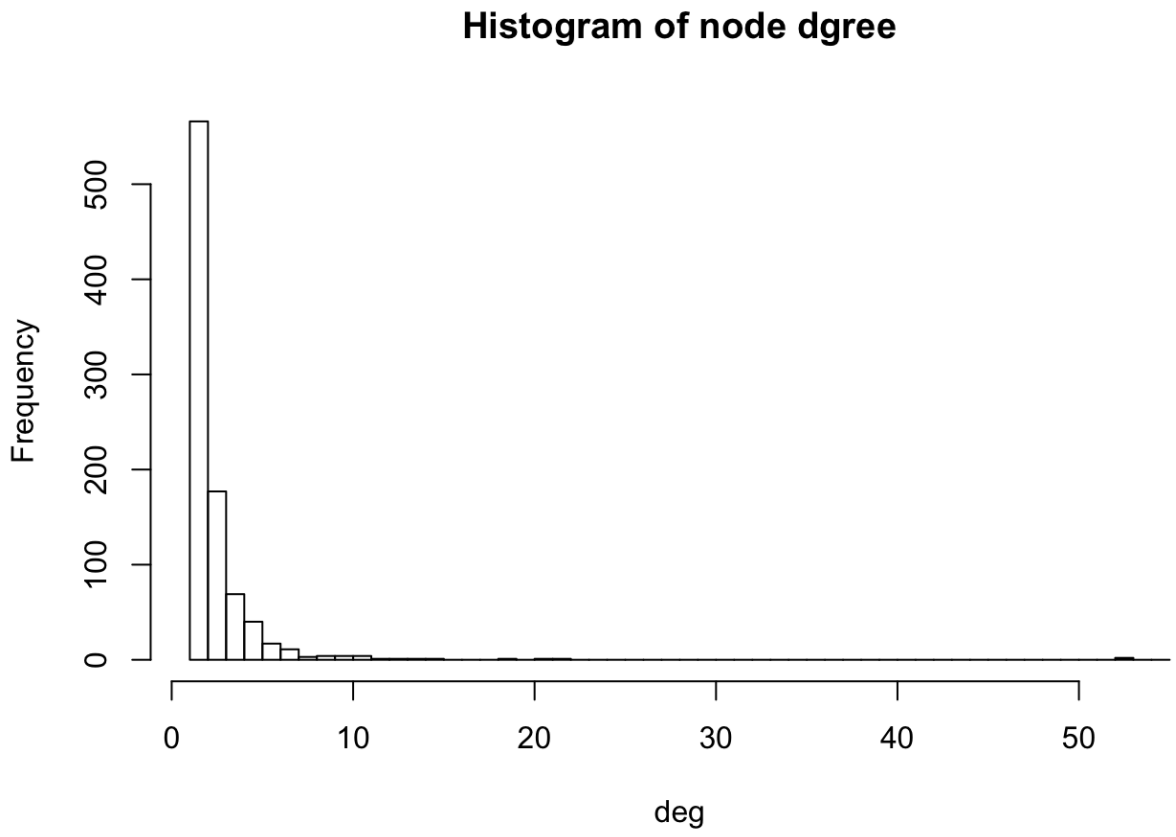


# Step 6 and 7: Subcomponent Analysis

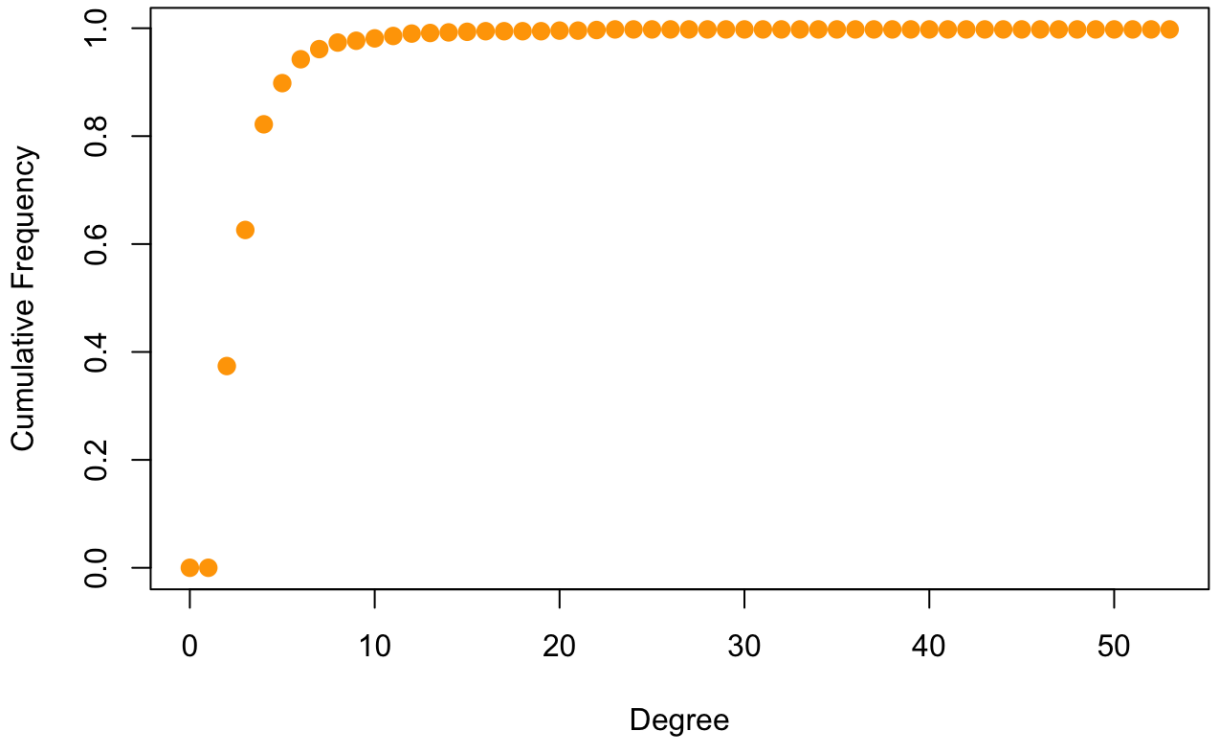
```
# looking at the degrees of the subcomponent
indeg <- degree(subg, mode ='in')
outdeg <- degree(subg, mode = 'out')
deg <- degree(subg, mode = "all")

# below are duplicate methods of generating the degree for subcomponent
#centr_degree(subg, mode='in', normalized = T)
#centr_degree(subg, mode = 'all', normalized = T)
# generating a dataframe for the subcompoent of the books with the ids and degr
ees for comparison
subbookproducts <- data.frame(subid, indeg, outdeg, deg)

# generating a subcomponent copurchase dataframe in case it is needed
subbokcopurchase <- as_edgelist(subg, names = T)

# graphing with size as a multiple of the total degrees
plot(subg,
     vertex.size=deg*.6,
     vertex.label=NA,
     edge.arrow.size=.2,
     rescale=F, layout=l*2,
     layout=l
     )
```



```
# examining the histogram and degree distribution of the degrees of the subcomp
onent
hist(deg, breaks = 1:55, main="Histogram of node dgree")
```




```
deg.dist <- degree_distribution(subg, cumulative = T, mode="all")
plot(x=0:max(deg), y=1-deg.dist, pch=19, cex=1.2, col='orange',
     xlab="Degree", ylab="Cumulative Frequency")
```




```
# determining the edge density of the subcomponent in two different manners
edge_density(subg, loops = F)
ecount(subg)/(vcount(subg)*(vcount(subg)-1))

# examining the reciprocity of the subcomponent
reciprocity(subg)
```

```
## [1] 0.001436951
## [1] 0.001436951
## [1] 0.286445
```

```
# examining the centrality and placing values in subcomponent table
subbookproducts$clos <- closeness(subg, mode = 'all', weights = NA)
centr_clo(subg, mode = 'all', normalized = T)

#subbookproducts$eigen <- eigen(subg, directed=T, normalized=T)
centr_eigen(subg, directed=T, normalized = T)

subbookproducts$btwn <- betweenness(subg, directed=T, weights=NA)
#edge_betweenness(subg, directed = T, weights = NA) #could add to subcopurchase
books table
centr_betw(subg, directed = T, normalized = T)
```

```
cent_c <- centr_clo(subg, mode = 'all', normalized = T)
  head(cent_c$res, n = 50)
  cent_c$centralization
  cent_c$theoretical_max
```

```
##  [1] 0.14559819 0.08168250 0.10761530 0.09733750 0.09119370 0.09723269
##  [7] 0.12770471 0.13509874 0.10028876 0.09721176 0.09723269 0.07555853
## [13] 0.08127081 0.10604815 0.12725479 0.14315155 0.10753841 0.07459727
## [19] 0.09121212 0.10259032 0.07034354 0.07027784 0.07784483 0.06942416
## [25] 0.11660640 0.09716991 0.11325724 0.06169297 0.08717058 0.07304643
## [31] 0.08413305 0.10801435 0.11171595 0.10693984 0.11012195 0.11704472
## [37] 0.10988075 0.06806875 0.10403226 0.09910009 0.09540412 0.09590059
## [43] 0.12877924 0.12056075 0.13354037 0.10033333 0.10464712 0.06806875
## [49] 0.07023411 0.07553325
## [1] 0.1074443
## [1] 451.2499
```

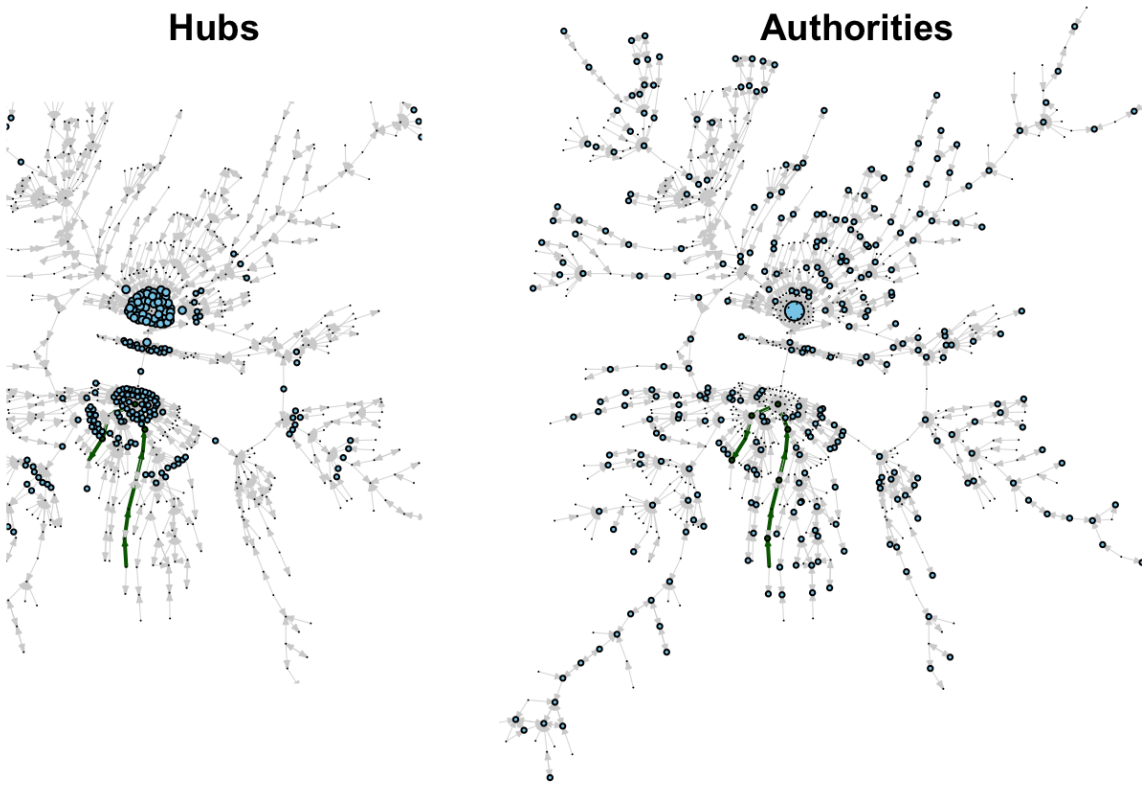
```
cent_e <- centr_eigen(subg, directed=T, normalized = T)
  head(cent_e$vector, n= 50)
  cent_e$value
  cent_e$centralization
 cent_e$theoretical_max
```

```
##  [1] 0.000000e+00 0.000000e+00 2.136383e-16 4.442920e-18 0.000000e+00
##  [6] 1.098040e-16 0.000000e+00 0.000000e+00 5.000000e-01 8.822909e-17
## [11] 1.074547e-16 0.000000e+00 1.598883e-17 1.317858e-17 0.000000e+00
## [16] 0.000000e+00 5.245155e-18 1.089702e-17 6.116620e-19 2.548901e-18
## [21] 0.000000e+00 0.000000e+00 1.852212e-17 0.000000e+00 0.000000e+00
## [26] 0.000000e+00 0.000000e+00 6.685681e-19 3.865707e-18 4.253519e-18
## [31] 0.000000e+00 2.522561e-17 0.000000e+00 0.000000e+00 0.000000e+00
## [36] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [41] 2.784464e-17 5.445626e-18 0.000000e+00 0.000000e+00 7.202638e-18
## [46] 3.991073e-17 0.000000e+00 5.821956e-18 0.000000e+00 3.760185e-18
## [1] 2.732051
## [1] 0.9964208
## [1] 903
```

```
cent_b <-centr_betw(subg, directed = T, normalized = T)
  head(cent_b$res, n=50)
  cent_b$centralization
  cent_b$theoretical_max
```

```
##  [1]   0  12   0   1   2   2  40  31   0   0   2  15   0   4   4  31   1
## [18]   0   2   0   0  14   0   0   0   0   0   0   0   2   0   0  17  22
## [35]  15  12  37   0   9   0  19   0 298  45  22   0   0   0   0   5
## [1] 0.0003616307
## [1] 735498918
```

```
#Below only works for undirected graphs -- not in our situation
#csubg <- cluster_edge_betweenness(subg)
#plot(csubg,
#     net,
#     vertex.size = 10,
#     vertex.label.cex = 0.8)
hs <- hub.score(subg, weights=NA)$vector
as <- authority_score(subg, weights = NA)$vector
subbookproducts$hs <- hs
subbookproducts$as <- as
par(mfrow=c(1,2))
plot(subg, vertex.size=hs*4, vertex.label=NA, edge.arrow.size=0.2, rescale=F, l
ayout=l*2,layout=l, main="Hubs")
plot(subg, vertex.size=as*10, vertex.label=NA, edge.arrow.size=0.2, rescale=F,
 layout=l*2,layout=l, main='Authorities')
```



```
par(mfrow=c(1,1))

# building on to the subbookproducts table by adding in the mean sales rank, re
view count, and
# rating of each of the subcomponent nodes and in degree neighbors by creating
 a vector of the
# incoming neighbor nodes and then using this to build a small slice of the ori
ginal book products
# dataframe from which the means can be generated for the requested columns
for (i in 1:904) {
  subbookproducts$salesrank[V(subg)[i]] <- bookproducts$salesrank[V(subg)[i]]
  subbookproducts$review_cnt[V(subg)[i]] <- bookproducts$review_cnt[V(subg)[i]]
  subbookproducts$downloads[V(subg)[i]] <- bookproducts$downloads[V(subg)[i]]
  subbookproducts$rating[V(subg)[i]] <- bookproducts$rating[V(subg)[i]]
  ngh_nodes <- neighbors(subg, V(subg)[i], mode='in' )
  tempnodes <- as_ids(ngh_nodes)
  tempdf <- filter(bookproducts,  bookproducts$id %in% tempnodes)
  subbookproducts$ngh_mn_salesrank[V(subg)[i]]<- mean(tempdf$salesrank)
  subbookproducts$ngh_mn_review_cnt[V(subg)[i]]<- mean(tempdf$review_cnt)
  subbookproducts$ngh_mn_rating[V(subg)[i]]<- mean(tempdf$rating)
}
# Now we have a table of information about all the nodes within the subcomponen
t
summary(subbookproducts)
```

```
##      subid          indeg             outdeg           deg
##  101035 :  1   Min.   : 0.000   Min.   :0.000   Min.   : 1.000
##  10121  :  1   1st Qu.: 0.000   1st Qu.:1.000   1st Qu.: 1.000
##  101336 :  1   Median : 1.000   Median :1.000   Median : 2.000
##  101725 :  1   Mean   : 1.298   Mean   :1.298   Mean   : 2.595
##  101765 :  1   3rd Qu.: 1.250   3rd Qu.:2.000   3rd Qu.: 3.000
##  101832 :  1   Max.   :53.000   Max.   :4.000   Max.   :53.000
##  (Other):898
##       clos                btwn               hs
##  Min.   :4.253e-05   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:8.528e-05   1st Qu.:  0.000   1st Qu.:0.00000
##  Median :1.010e-04   Median :  0.000   Median :0.00000
##  Mean   :1.018e-04   Mean   :  3.775   Mean   :0.05717
##  3rd Qu.:1.219e-04   3rd Qu.:  2.000   3rd Qu.:0.00000
##  Max.   :1.612e-04   Max.   :298.000   Max.   :1.00000
##
##        as             salesrank        review_cnt        downloads
##  Min.   :0.000000   Min.   :    19   Min.   :   0.00   Min.   :   0.00
##  1st Qu.:0.000000   1st Qu.: 31208   1st Qu.:   2.00   1st Qu.:   2.00
##  Median :0.000000   Median : 65346   Median :   7.00   Median :   7.00
##  Mean   :0.001479   Mean   : 69787   Mean   :  36.29   Mean   :  35.64
##  3rd Qu.:0.000000   3rd Qu.:106011   3rd Qu.:  19.00   3rd Qu.:  19.00
##  Max.   :1.000000   Max.   :149890   Max.   :5539.00   Max.   :4995.00
##
##      rating      ngh_mn_salesrank ngh_mn_review_cnt ngh_mn_rating
##  Min.   :0.000   Min.   :  1596   Min.   :   0.00   Min.   :0.000
##  1st Qu.:4.000   1st Qu.: 44144   1st Qu.:   3.00   1st Qu.:3.500
##  Median :4.500   Median : 73840   Median :   8.00   Median :4.333
##  Mean   :3.971   Mean   : 73422   Mean   :  25.58   Mean   :3.875
##  3rd Qu.:5.000   3rd Qu.:101274   3rd Qu.:  19.19   3rd Qu.:4.667
##  Max.   :5.000   Max.   :149844   Max.   :1015.00   Max.   :5.000
##                  NA's   :386      NA's   :386       NA's   :386
```

```
str(subbookproducts)
```

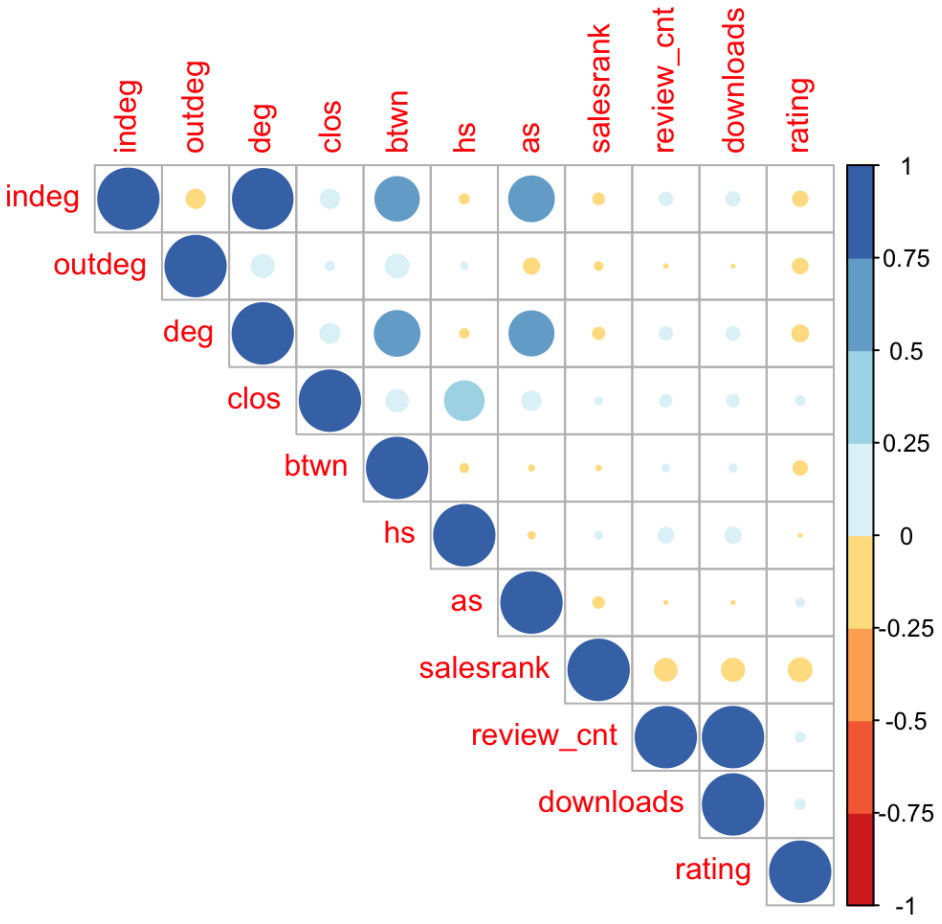
```
## 'data.frame':    904 obs. of  15 variables:
##  $ subid            : Factor w/ 904 levels "101035","10121",..: 541 824 831
86 143 260 277 362 511 534 ...
##  $ indeg            : num  53 3 11 1 1 3 10 2 22 2 ...
##  $ outdeg           : num  0 1 0 1 1 3 1 2 0 3 ...
##  $ deg              : num  53 4 11 2 2 6 11 4 22 5 ...
##  $ clos             : num  1.61e-04 9.05e-05 1.19e-04 1.08e-04 1.01e-04 ...
##  $ btwn             : num  0 12 0 1 2 2 40 31 0 0 ...
##  $ hs               : num  4.75e-15 5.14e-16 1.01e-15 5.53e-04 3.59e-05 ...
##  $ as               : num  1 0 0.000575 0 0 ...
##  $ salesrank        : num  24741 97166 57186 48408 27507 ...
##  $ review_cnt       : int  12 4 22 4 2 11 3 12 4 2 ...
##  $ downloads        : int  12 4 22 4 2 11 3 12 4 2 ...
##  $ rating           : num  4.5 5 3.5 4 4 4.5 4.5 4.5 5 2.5 ...
##  $ ngh_mn_salesrank : num  82153 41744 73179 19415 46701 ...
##  $ ngh_mn_review_cnt: num  21.1 4 157.8 6 0 ...
##  $ ngh_mn_rating    : num  4.1 4.67 4.5 4.5 0 ...
```

# Step 8: Logistic Regression Analysis

```
# Running Poisson logistic regression to determine which variables have biggest
 impact on sales rank
fit.salesrank <- glm(salesrank ~ review_cnt + downloads + rating + ngh_mn_sales
rank + ngh_mn_review_cnt + ngh_mn_rating
                     + indeg + outdeg + deg + clos + btwn + hs + as, data = sub
bookproducts, family = poisson(link = log))
summary(fit.salesrank)
```

```
##
## Call:
## glm(formula = salesrank ~ review_cnt + downloads + rating + ngh_mn_salesrank
+
##     ngh_mn_review_cnt + ngh_mn_rating + indeg + outdeg + deg +
##     clos + btwn + hs + as, family = poisson(link = log), data = subbookprodu
cts)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -370.20  -160.14   -25.52   125.52   426.56
##
## Coefficients: (1 not defined because of singularities)
##                      Estimate Std. Error  z value Pr(>|z|)
## (Intercept)         1.134e+01  1.101e-03 10298.39   <2e-16 ***
## review_cnt          8.483e-03  1.009e-04    84.11   <2e-16 ***
## downloads          -1.049e-02  1.020e-04  -102.85   <2e-16 ***
## rating             -4.743e-02  1.101e-04  -430.63   <2e-16 ***
## ngh_mn_salesrank    4.321e-07  4.555e-09    94.86   <2e-16 ***
## ngh_mn_review_cnt -5.368e-04  2.896e-06  -185.36   <2e-16 ***
## ngh_mn_rating     -2.498e-03  1.246e-04   -20.04   <2e-16 ***
## indeg             -6.067e-03  7.028e-05   -86.33   <2e-16 ***
## outdeg            -8.930e-03  2.092e-04   -42.69   <2e-16 ***
## deg                       NA         NA       NA       NA
## clos               6.268e+02  7.906e+00    79.29   <2e-16 ***
## btwn               2.188e-04  1.079e-05    20.27   <2e-16 ***
## hs                 1.811e-02  9.695e-04    18.68   <2e-16 ***
## as                -6.025e-01  6.814e-03   -88.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 16900894  on 517  degrees of freedom
## Residual deviance: 15840167  on 505  degrees of freedom
##   (386 observations deleted due to missingness)
## AIC: 15846727
##
## Number of Fisher Scoring iterations: 6
```

```
# Running the correlation to identify collinearity
# For visual representation; those with NA correlations have been removed.
correlations<- cor(subbookproducts[,-1])
thresh <- 0.0000001
correlations0 <- correlations
diag(correlations0) <- 0
new<- sort(unique(c(which(abs(correlations0)>= thresh,arr = TRUE))))
correlations <- correlations[new,new]
corrplot(correlations[new,new], method = "circle", type = "upper", col = brewe
r.pal(n=8, name = "RdYlBu"))
```




```
# Removing total degrees (deg) because it is highly correlated to indegree  It
 didn't compute in the first regression because it is
# 100% defined by the two other variables in the regression (indeg + outdeg)
fit.salesrank_final <- glm(salesrank ~ review_cnt + downloads + rating + ngh_mn
_salesrank + ngh_mn_review_cnt + ngh_mn_rating
                     + indeg + outdeg + clos + btwn + hs + as, data = subbookpr
oducts, family = poisson())
summary(fit.salesrank_final)
```

```
##
## Call:
## glm(formula = salesrank ~ review_cnt + downloads + rating + ngh_mn_salesrank
+
##     ngh_mn_review_cnt + ngh_mn_rating + indeg + outdeg + clos +
##     btwn + hs + as, family = poisson(), data = subbookproducts)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -370.20  -160.14   -25.52   125.52   426.56
##
## Coefficients:
##                      Estimate Std. Error  z value Pr(>|z|)
## (Intercept)         1.134e+01  1.101e-03 10298.39   <2e-16 ***
## review_cnt          8.483e-03  1.009e-04    84.11   <2e-16 ***
## downloads          -1.049e-02  1.020e-04  -102.85   <2e-16 ***
## rating             -4.743e-02  1.101e-04  -430.63   <2e-16 ***
## ngh_mn_salesrank    4.321e-07  4.555e-09    94.86   <2e-16 ***
## ngh_mn_review_cnt -5.368e-04  2.896e-06  -185.36   <2e-16 ***
## ngh_mn_rating     -2.498e-03  1.246e-04   -20.04   <2e-16 ***
## indeg             -6.067e-03  7.028e-05   -86.33   <2e-16 ***
## outdeg            -8.930e-03  2.092e-04   -42.69   <2e-16 ***
## clos               6.268e+02  7.906e+00    79.29   <2e-16 ***
## btwn               2.188e-04  1.079e-05    20.27   <2e-16 ***
## hs                 1.811e-02  9.695e-04    18.68   <2e-16 ***
## as                -6.025e-01  6.814e-03   -88.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 16900894  on 517  degrees of freedom
## Residual deviance: 15840167  on 505  degrees of freedom
##   (386 observations deleted due to missingness)
## AIC: 15846727
##
## Number of Fisher Scoring iterations: 6
```

```
# By taking the exp of each coefficient, we can say that salesrank will increas
e by that amount, exp(coefficient), when the variable increases by 1 unit
# This means that the actual sales will decrease based on the increased sales r
ank
Increase <- data.frame(exp(fit.salesrank_final$coefficients))
Increase
```

```
##                   exp.fit.salesrank_final.coefficients.
## (Intercept)                                  8.387912e+04
## review_cnt                                   1.008519e+00
## downloads                                    9.895626e-01
## rating                                       9.536802e-01
## ngh_mn_salesrank                             1.000000e+00
## ngh_mn_review_cnt                            9.994633e-01
## ngh_mn_rating                                9.975054e-01
## indeg                                        9.939510e-01
## outdeg                                       9.911101e-01
## clos                                        1.677189e+272
## btwn                                         1.000219e+00
## hs                                           1.018272e+00
## as                                           5.474584e-01
```