SNA Closeness 2_RELACIONAMENTO FORMAL (var0)

Leonardo Martins

17 de julho de 2016

Contents

SNA Descritive Analysis from “Projeto Redes de Atenção às pessoas que consomem álcool e outras Drogas em Juiz de Fora-MG Brazil” - SNArRDJF

Here you can find a basic script to analysis data from SNArRDJF - this script was elaborated considering its use for orther matrix adjacency data from SNArRDJF - Here we are going to analyse:

1 2_RELACIONAMENTO FORMAL (var0)

#########################`#` Basic Preparation ##### ‘#########################

2 Loading objects generated with previous script

```
rm(list = ls()) # removing previous objects to be sure that we don't have objects conflicts name
load("~/SNArRDJF/Robject/2_degree_var0.RData")
```

2.1 Reload packages

```
suppressMessages(library(RColorBrewer))
suppressMessages(library(car))
suppressMessages(library(xtable))
suppressMessages(library(igraph))
suppressMessages(library(miniCRAN))
suppressMessages(library(magrittr))
suppressMessages(library(keyplayer))
suppressMessages(library(dplyr))
suppressMessages(library(feather))
suppressMessages(library(visNetwork))
suppressMessages(library(knitr))
suppressMessages(library(DT))
```

2.2 Adding phantom tools

```
#In order to get dinamic javascript object install those ones. If you get problems installing go to Sta
#devtools::install_github("wch/webshot")
#webshot::install_phantomjs()
```

2.3 Setting a random seed - this is a good strategy to keep the same graph pattern layout in a new report generation

```
set.seed(123)
```

2.4 Simplify Graph - removing loops and duble edges

```
var0<-simplify(var0) #Simplify
```

3 Closeness - centrality based on distance to others in the graph

How close an actor to all the other actors in network?

High closeness centrality - short communication path to others, minimal number of steps to reach others.

Answers the "Kevin Bacon" question:

How many steps are required to access every other vertex from a given vertex?

One practical implication of this metric: it helps you gauge how information might spread within your network, and who might be the best people to leverage if you need to make sure information gets around. Link here: http://www.tc.umn.edu/~alink/R-social-network-analysis.html

Closeness centrality can be defined as a measure of how far other nodes are from the node in question. Nodes with high closeness centrality are likely to be relatively efficient in receiving or transmitting information to/from distant parts of the social network.

Scores may be interpreted as arising from a reciprocal process in which the centrality of each actor is proportional to the sum of the centralities of those actors to whom he or she is connected.

In general, vertices with high eigenvector centralities are those which are connected to many other vertices which are, in turn, connected to many others (and so on). (The perceptive may realize that this implies that the largest values will be obtained by individuals in large cliques (or high-density substructures)

3.1 Closeness Non-normalized

3.1.1 Saving to Igraph object

```
V(var0)$incloseness <- closeness(var0, mode = "in", weights = E(var0)$var0) %>% round(6)
V(var0)$outcloseness <- closeness(var0, mode = "out", weights = E(var0)$var0) %>% round(6)
V(var0)$totalcloseness <- closeness(var0, mode = "total", weights = E(var0)$var0) %>% round(4)
```

3.1.2 Saving to Environment

```
var0_incloseness<- closeness(var0, mode = "in", weights = E(var0)$var0) %>% round(6)
var0_outcloseness<- closeness(var0, mode = "out", weights = E(var0)$var0) %>% round(6)
var0_totalcloseness<- closeness(var0, mode = "total", weights = E(var0)$var0) %>% round(6)
```

3.1.3 Closeness Non-normalized - in

```
summary(var0_incloseness)
```

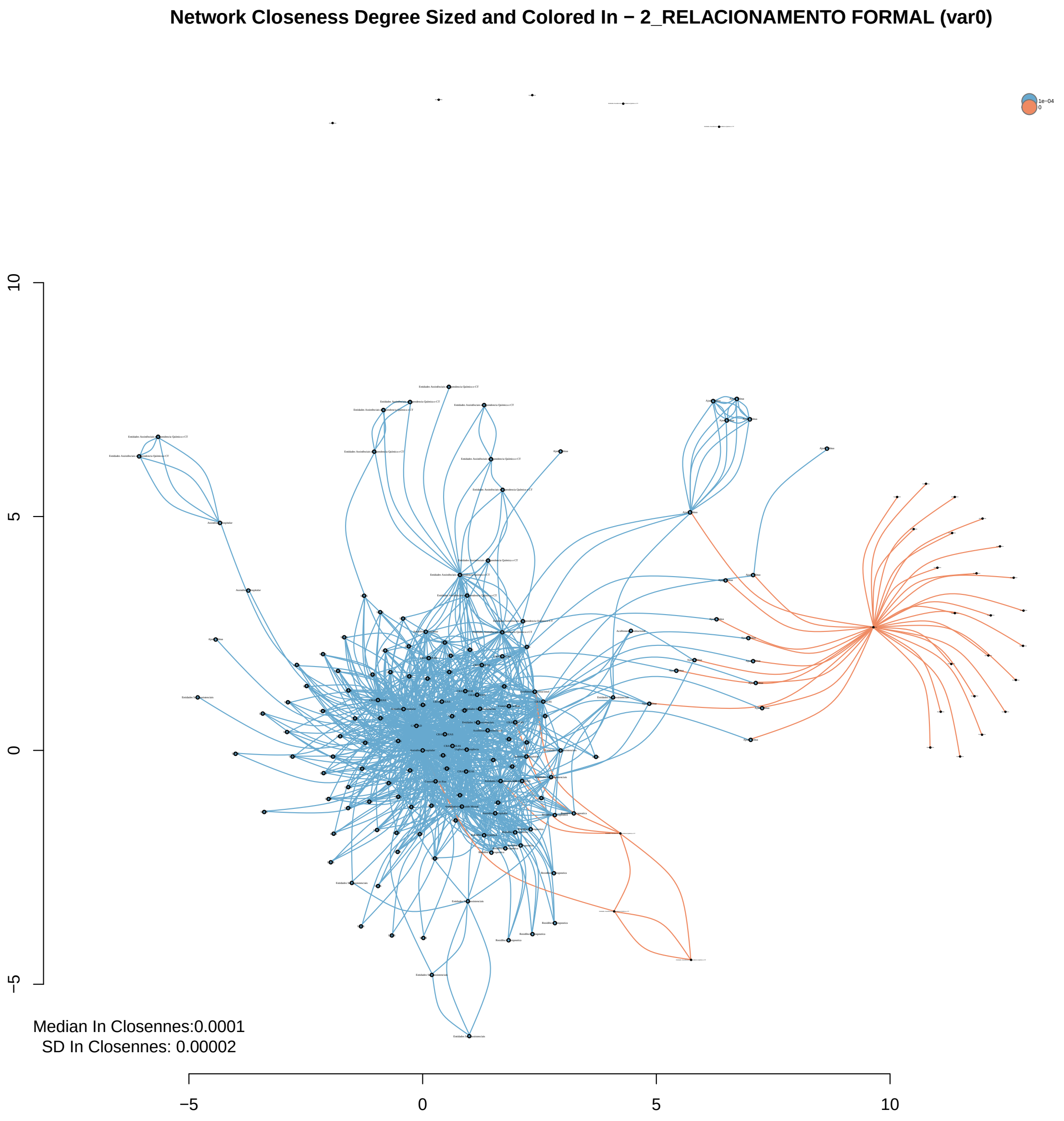
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 2.900e-05 7.200e-05 7.200e-05 6.492e-05 7.200e-05 7.600e-05
```

```
sd(var0_incloseness)
```

```
## [1] 1.638413e-05
```

3.2 Network Plotting Based On Non-normalized Closeness - IN

```
V(var0)$incloseness<-closeness(var0, weights = E(var0)$var0, mode="in")

#Get Variable
V(var0)$var0_color_degree<-round(V(var0)$incloseness,4)

#Creating brewer pallette
vertex_var0_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(var0)$var0_color_degree)), "RdBu"))(
            length(unique(V(var0)$var0_color_degree)))

#Saving as Vertex properties
V(var0)$vertex_var0_color_degree<-
  vertex_var0_color_degree[as.numeric(
  cut(V(var0)$var0_color_degree,
      breaks=length(unique(V(var0)$var0_color_degree))))]

set.seed(123)
#Plotting based only on degree measures
edge.start <- ends(var0, es=E(var0), names=F)[,1]

# Fixing ego
minC <- rep(-Inf, vcount(var0))
maxC <- rep(Inf, vcount(var0))
minC[1] <- maxC[1] <- 0
co <- layout_with_fr(var0, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(var0)$weig


#PLotting
plot(var0,
     layout=co,
     edge.color=V(var0)$vertex_var0_color_degree[edge.start],
     edge.arrow.size=closeness(var0, weights = E(var0)$var0, mode="in"),
     edge.width=E(var0)$weight/mean(E(var0)$weight),
     edge.curved = TRUE,
     vertex.color=V(var0)$vertex_var0_color_degree,
     vertex.size=closeness(var0, weights = E(var0)$var0, mode="in")*10^5,
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(var0,"LABEL_COR"),
     vertex.label.cex=(closeness(var0, weights = E(var0)$var0, mode="in")+10^-5)*2000,
     vertex.label.dist=0,
     rescale=F,
     xlim=range(co[,1]),
     ylim=range(co[,2])
     )
axis(1)
axis(2)


#Solving Problems with legend rendering
a<-V(var0)$var0_color_degree
b<-V(var0)$vertex_var0_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#Adding Legend
legend(x=range(co[,1])[2],
       y=range(co[,2])[2],
       legend=as.character(f),
       pch=21,
       col = "#777777",
       pt.bg=as.character(g),
       pt.cex=2,
       bty="n",
       ncol=1,
       lty=1,
       cex = .3)

#Adding Title
  title("Network Closeness Degree Sized and Colored In - 2_RELACIONAMENTO FORMAL (var0)", sub = "Source
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median In Closennes:%.4f\nSD In Closennes: %.5f",
             median(closeness(var0, mode="in", weights = E(var0)$var0)),
             sd(closeness(var0, mode="in", weights = E(var0)$var0))
             )
       )
```



Source: from authors

3.2.1 Closeness Non-normalized - OUT

```
summary(var0_outcloseness)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 0.0000290 0.0000290 0.0001570 0.0001091 0.0001580 0.0004730
```

```
sd(var0_outcloseness)
```

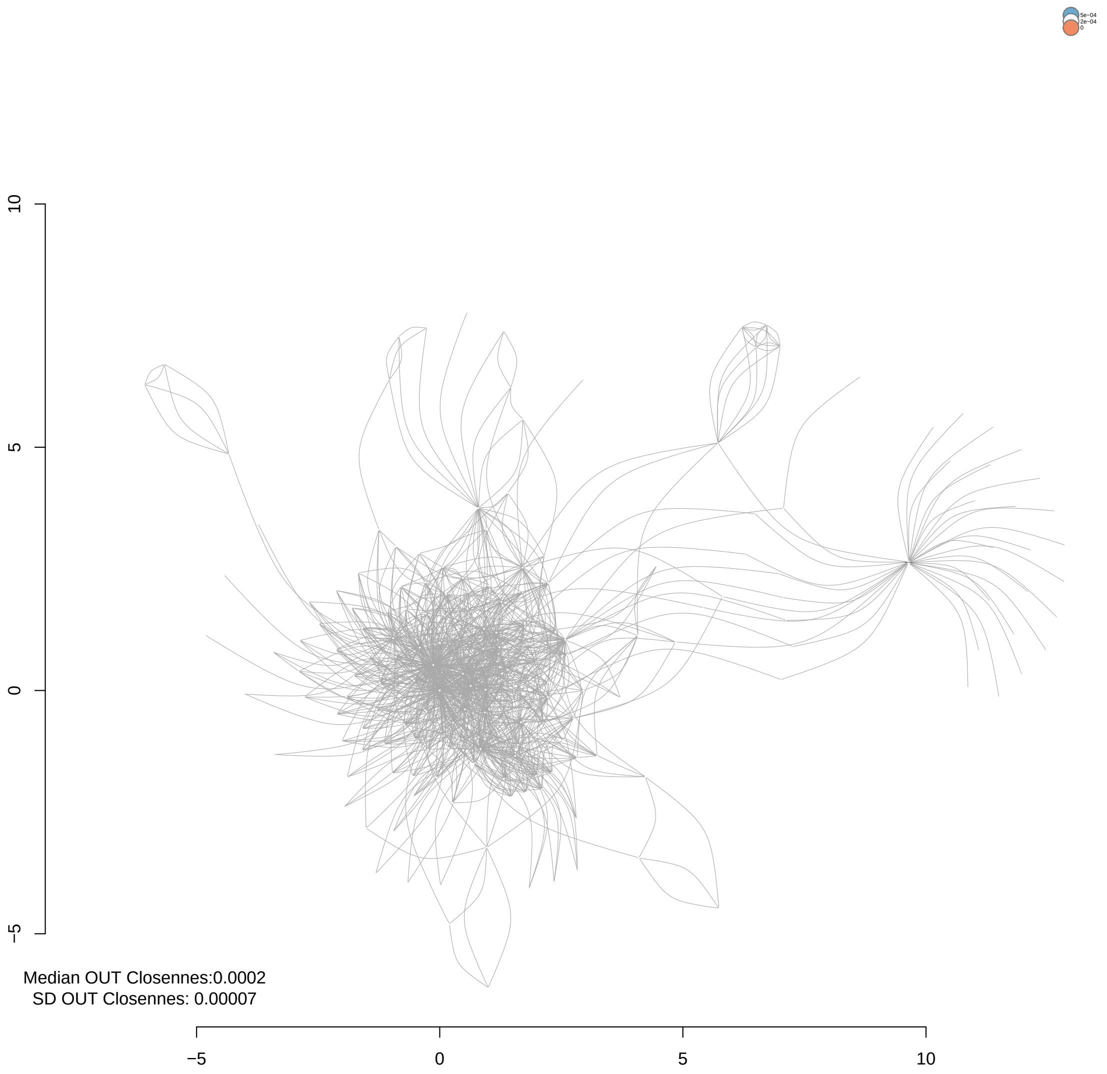
```
## [1] 6.830992e-05
```

3.3 Network Plotting Based On Non-normalized Closeness - OUT

```
V(var0)$outcloseness<-closeness(var0, weights = E(var0)$var0, mode="out")

#Get Variable
V(var0)$var0_color_degree<-round(V(var0)$outcloseness,4)

#Creating brewer pallette
vertex_var0_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(var0)$var0_color_degree)), "RdBu"))(
            length(unique(V(var0)$var0_color_degree)))

#Saving as Vertex properties
V(var0)$vertex_var0_color_degree<-
  vertex_var0_color_degree[as.numeric(
  cut(V(var0)$var0_color_degree,
      breaks=length(unique(V(var0)$var0_color_degree))))]

set.seed(123)
#Plotting based only on degree measures
edge.start <- ends(var0, es=E(var0), names=F)[,1]

# Fixing ego
minC <- rep(-Inf, vcount(var0))
maxC <- rep(Inf, vcount(var0))
minC[1] <- maxC[1] <- 0
co <- layout_with_fr(var0, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(var0)$weig


#PLotting
plot(var0,
     layout=co,
     #edge.color=V(var0)$vertex_var0_color_degree[edge.start],
     edge.arrow.size=closeness(var0, weights = E(var0)$var0, mode="out"),
     edge.width=E(var0)$weight/2*mean(E(var0)$weight),
     edge.curved = TRUE,
     vertex.color=V(var0)$vertex_var0_color_degree,
     vertex.size=closeness(var0, weights = E(var0)$var0, mode="out")*10^4,
     vertex.frame.color="white",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(var0,"LABEL_COR"),
     vertex.label.cex=closeness(var0, weights = E(var0)$var0, mode="out")*200,
     vertex.label.dist=0,
     rescale=F,
     xlim=range(co[,1]),
     ylim=range(co[,2])
     )
axis(1)
axis(2)


#Solving Problems with legend rendering
```

```
a<-V(var0)$var0_color_degree
b<-V(var0)$vertex_var0_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#Adding Legend
legend(x=range(co[,1])[2],
       y=range(co[,2])[2],
       legend=as.character(f),
       pch=21,
       col = "#777777",
       pt.bg=as.character(g),
       pt.cex=2,
       bty="n",
       ncol=1,
       lty=1,
       cex = .3)

#Adding Title
  title("Network Closeness Degree Sized and Colored OUT - 2_RELACIONAMENTO FORMAL (var0)", sub = "Sourc
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median OUT Closennes:%.4f\nSD OUT Closennes: %.5f",
             median(closeness(var0, mode="out", weights = E(var0)$var0)),
             sd(closeness(var0, mode="out", weights = E(var0)$var0))
             )
       )
```

Network Closeness Degree Sized and Colored OUT − 2_RELACIONAMENTO FORMAL (var0)



Source: from authors

3.3.1 Closeness Non-normalized - ALL

```
summary(var0_totalcloseness)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000290 0.0006740 0.0007260 0.0006898 0.0007320 0.0008220
```

```
sd(var0_totalcloseness)
```

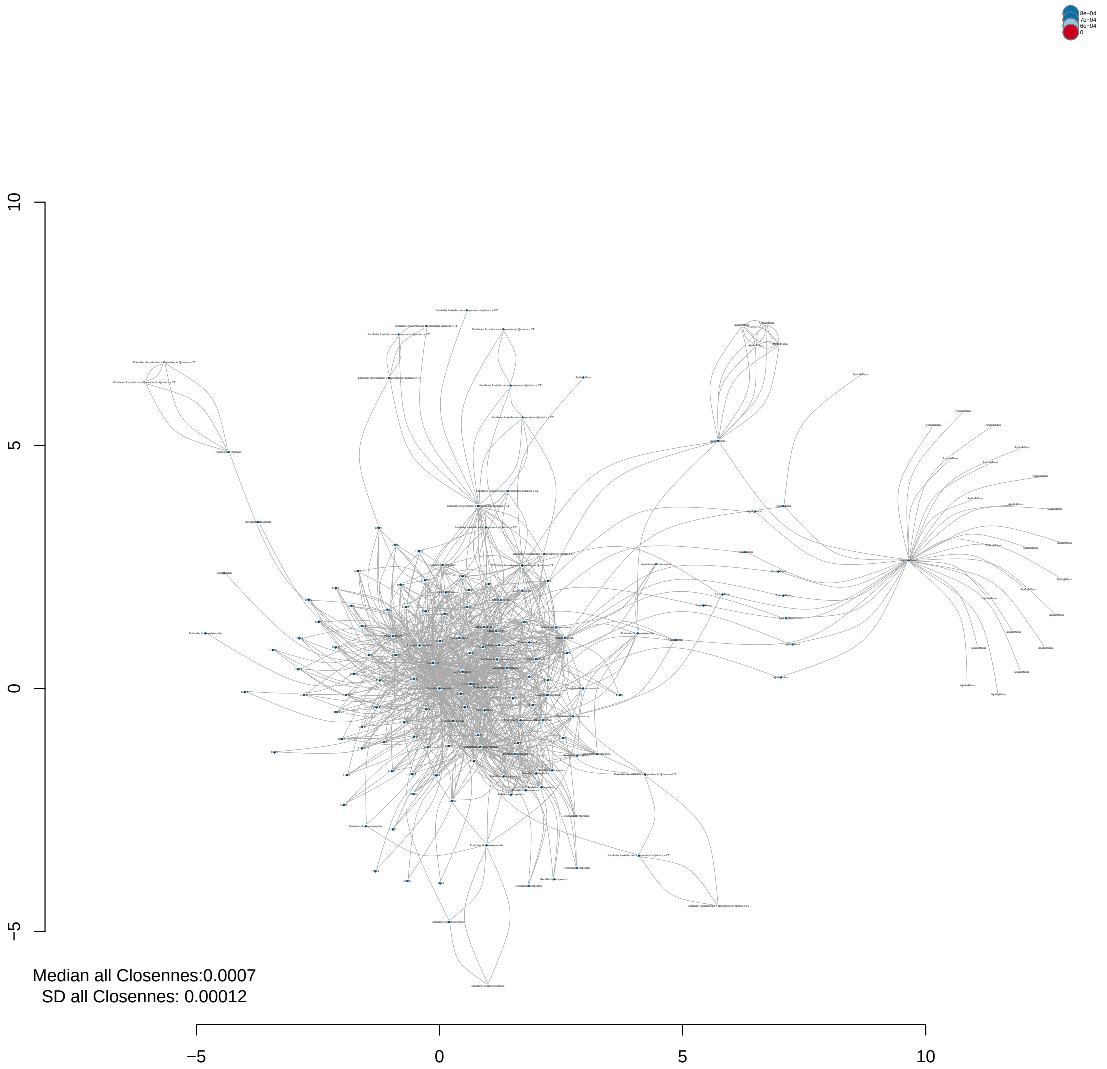
```
## [1] 0.0001176304
```

3.4 Network Plotting Based On Non-normalized Closeness - ALL

```
V(var0)$allcloseness<-closeness(var0, weights = E(var0)$var0, mode="all")

#Get Variable
V(var0)$var0_color_degree<-round(V(var0)$allcloseness,4)

#Creating brewer pallette
vertex_var0_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(var0)$var0_color_degree)), "RdBu"))(
            length(unique(V(var0)$var0_color_degree)))

#Saving as Vertex properties
V(var0)$vertex_var0_color_degree<-
  vertex_var0_color_degree[as.numeric(
  cut(V(var0)$var0_color_degree,
      breaks=length(unique(V(var0)$var0_color_degree))))]

set.seed(123)
#Plotting based only on degree measures
edge.start <- ends(var0, es=E(var0), names=F)[,1]

# Fixing ego
minC <- rep(-Inf, vcount(var0))
maxC <- rep(Inf, vcount(var0))
minC[1] <- maxC[1] <- 0
co <- layout_with_fr(var0, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(var0)$weig


#PLotting
plot(var0,
     layout=co,
     #edge.color=V(var0)$vertex_var0_color_degree[edge.start],
     edge.arrow.size=closeness(var0, weights = E(var0)$var0, mode="all"),
     edge.width=E(var0)$weight/2*mean(E(var0)$weight),
     edge.curved = TRUE,
     vertex.color=V(var0)$vertex_var0_color_degree,
     vertex.size=closeness(var0, weights = E(var0)$var0, mode="all")*10^4,
     vertex.frame.color="white",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(var0,"LABEL_COR"),
     vertex.label.cex=closeness(var0, weights = E(var0)$var0, mode="all")*200,
     vertex.label.dist=0,
     rescale=F,
     xlim=range(co[,1]),
     ylim=range(co[,2])
     )
axis(1)
axis(2)


#Solving Problems with legend rendering
```

```r
a<-V(var0)$var0_color_degree
b<-V(var0)$vertex_var0_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#Adding Legend
legend(x=range(co[,1])[2],
       y=range(co[,2])[2],
       legend=as.character(f),
       pch=21,
       col = "#777777",
       pt.bg=as.character(g),
       pt.cex=2,
       bty="n",
       ncol=1,
       lty=1,
       cex = .3)

#Adding Title
  title("Network Closeness Degree Sized and Colored all - 2_RELACIONAMENTO FORMAL (var0)", sub = "Sourc
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median all Closennes:%.4f\nSD all Closennes: %.5f",
             median(closeness(var0, mode="all", weights = E(var0)$var0)),
             sd(closeness(var0, mode="all", weights = E(var0)$var0))
             )
       )
```

Network Closeness Degree Sized and Colored all − 2_RELACIONAMENTO FORMAL (var0)



Source: from authors

3.5 Closeness Normalized

3.5.1 Saving to Igraph object

```
V(var0)$incloseness_n <- closeness(var0, mode = "in",, weights = E(var0)$var0, normalized = T) %>% roun
V(var0)$outcloseness_n <- closeness(var0, mode = "out", normalized = T, weights = E(var0)$var0) %>% rou
V(var0)$totalcloseness_n <- closeness(var0, mode = "total", normalized = T, weights = E(var0)$var0) %>%
```

3.5.2 Saving to Environment

```
var0_incloseness_n<- closeness(var0, mode = "in", normalized = T, weights = E(var0)$var0) %>% round(6)
var0_outcloseness_n<- closeness(var0, mode = "out", normalized = T, weights = E(var0)$var0) %>% round(6
var0_totalcloseness_n<- closeness(var0, mode = "total", normalized = T, weights = E(var0)$var0) %>% rou
```

3.5.3 Closeness Normalized - IN

```
summary(var0_incloseness_n)
```

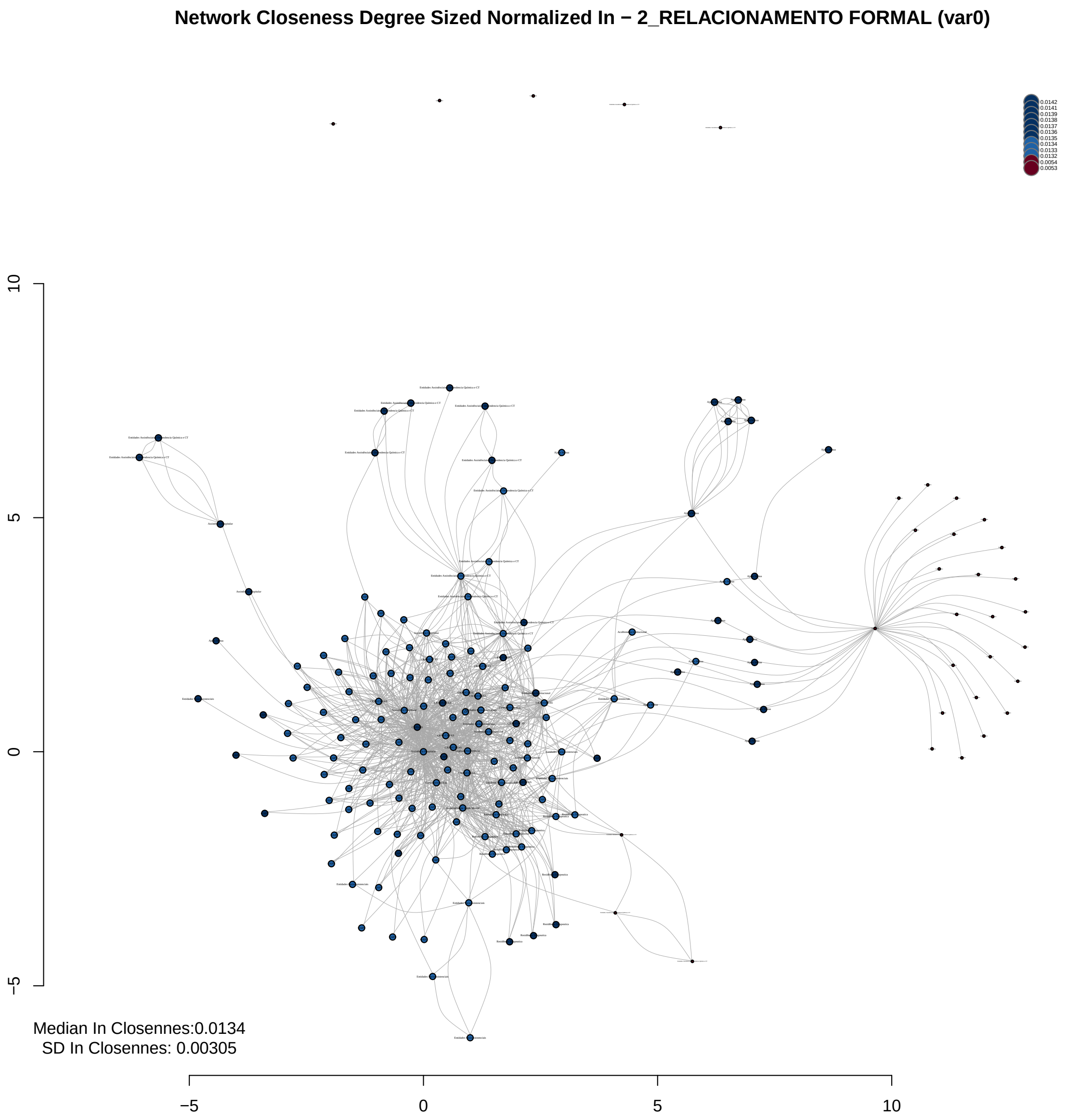
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005348 0.013340 0.013390 0.012070 0.013420 0.014210
```

```
sd(var0_incloseness_n)
```

```
## [1] 0.003055024
```

3.6 Network Plotting Based On Normalized Closeness - IN

```
V(var0)$incloseness_n<-closeness(var0, weights = E(var0)$var0, mode="in", normalized = T)

#Get Variable
V(var0)$var0_color_degree<-round(V(var0)$incloseness_n,4)

#Creating brewer pallette
vertex_var0_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(var0)$var0_color_degree)), "RdBu"))(
            length(unique(V(var0)$var0_color_degree)))

#Saving as Vertex properties
V(var0)$vertex_var0_color_degree<-
  vertex_var0_color_degree[as.numeric(
  cut(V(var0)$var0_color_degree,
      breaks=length(unique(V(var0)$var0_color_degree))))]

set.seed(123)
#Plotting based only on degree measures
edge.start <- ends(var0, es=E(var0), names=F)[,1]

# Fixing ego
minC <- rep(-Inf, vcount(var0))
maxC <- rep(Inf, vcount(var0))
minC[1] <- maxC[1] <- 0
co <- layout_with_fr(var0, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(var0)$wei
#PLotting
plot(var0,
     layout=co,
     #edge.color=V(var0)$vertex_var0_color_degree[edge.start],
     edge.arrow.size=closeness(var0, weights = E(var0)$var0, mode="in",normalized = T),
     edge.width=E(var0)$weight/10*mean(E(var0)$weight),
     edge.curved = TRUE,
     vertex.color=V(var0)$vertex_var0_color_degree,
     vertex.size=(closeness(var0, weights = E(var0)$var0, mode="in",normalized = T))*1000,
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(var0,"LABEL_COR"),
     vertex.label.cex=closeness(var0, weights = E(var0)$var0, mode="in",normalized = T)*10,
     vertex.label.dist=0,
     rescale=F,
     xlim=range(co[,1]),
     ylim=range(co[,2])
     )
axis(1)
axis(2)


#Solving Problems with legend rendering
a<-V(var0)$var0_color_degree
b<-V(var0)$vertex_var0_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#Adding Legend
legend(x=range(co[,1])[2],
       y=range(co[,2])[2],
       legend=as.character(f),
       pch=21,
       col = "#777777",
       pt.bg=as.character(g),
       pt.cex=2,
       bty="n",
       ncol=1,
       lty=1,
       cex = .3)

#Adding Title
  title("Network Closeness Degree Sized Normalized In - 2_RELACIONAMENTO FORMAL (var0)", sub = "Source:
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median In Closennes:%.4f\nSD In Closennes: %.5f",
             median(closeness(var0, mode="in", weights = E(var0)$var0, normalized = T)),
             sd(closeness(var0, mode="in", weights = E(var0)$var0, normalized = T))
             )
       )
```



Source: from authors

###Closeness Normalized - OUT

```
summary(var0_outcloseness_n)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005348 0.005348 0.029260 0.020280 0.029340 0.087900
```

```
sd(var0_outcloseness_n)
```

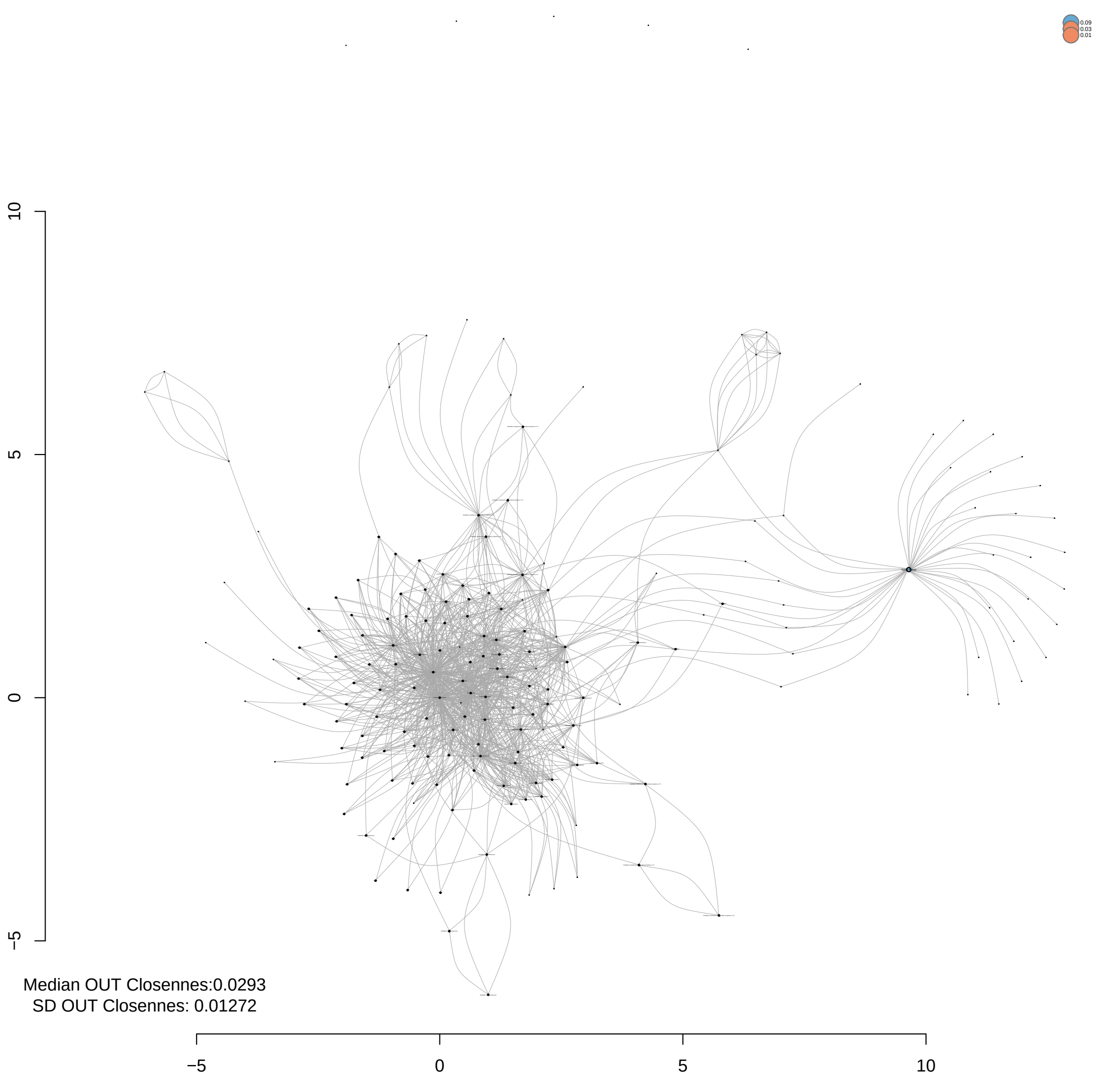
```
## [1] 0.01271714
```

3.7 Network Plotting Based On Normalized Closeness - OUT

```
V(var0)$outcloseness_n<-closeness(var0, weights = E(var0)$var0, mode="out", normalized = T)

#Get Variable
V(var0)$var0_color_degree<-round(V(var0)$outcloseness_n,2)

#Creating brewer pallette
vertex_var0_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(var0)$var0_color_degree)), "RdBu"))(
            length(unique(V(var0)$var0_color_degree)))

#Saving as Vertex properties
V(var0)$vertex_var0_color_degree<-
  vertex_var0_color_degree[as.numeric(
  cut(V(var0)$var0_color_degree,
      breaks=length(unique(V(var0)$var0_color_degree))))]

set.seed(123)
#Plotting based only on degree measures
edge.start <- ends(var0, es=E(var0), names=F)[,1]

# Fixing ego
minC <- rep(-Inf, vcount(var0))
maxC <- rep(Inf, vcount(var0))
minC[1] <- maxC[1] <- 0
co <- layout_with_fr(var0, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(var0)$weig


#PLotting
plot(var0,
     layout=co,
     #edge.color=V(var0)$vertex_var0_color_degree[edge.start],
     edge.arrow.size=closeness(var0, weights = E(var0)$var0, mode="out",normalized = T),
     edge.width=E(var0)$weight/10*mean(E(var0)$weight),
     edge.curved = TRUE,
     vertex.color=V(var0)$vertex_var0_color_degree,
     vertex.size=(closeness(var0, weights = E(var0)$var0, mode="out",normalized = T))*100,
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(var0,"LABEL_COR"),
     vertex.label.cex=closeness(var0, weights = E(var0)$var0, mode="out",normalized = T)*1.5,
     vertex.label.dist=0,
     rescale=F,
     xlim=range(co[,1]),
     ylim=range(co[,2])
     )
axis(1)
axis(2)


#Solving Problems with legend rendering
```

```
a<-V(var0)$var0_color_degree
b<-V(var0)$vertex_var0_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#Adding Legend
legend(x=range(co[,1])[2],
       y=range(co[,2])[2],
       legend=as.character(f),
       pch=21,
       col = "#777777",
       pt.bg=as.character(g),
       pt.cex=2,
       bty="n",
       ncol=1,
       lty=1,
       cex = .3)

#Adding Title
  title("Network Closeness Degree Sized Normalized OUT - 2_RELACIONAMENTO FORMAL (var0)", sub = "Source
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median OUT Closennes:%.4f\nSD OUT Closennes: %.5f",
             median(closeness(var0, mode="out", weights = E(var0)$var0, normalized = T)),
             sd(closeness(var0, mode="out", weights = E(var0)$var0, normalized = T))
             )
       )
```

Network Closeness Degree Sized Normalized OUT − 2_RELACIONAMENTO FORMAL (var0)



Source: from authors

3.7.1 Closeness Normalized - ALL

```
summary(var0_totalcloseness_n)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005348 0.125400 0.135100 0.128300 0.136100 0.153000
```

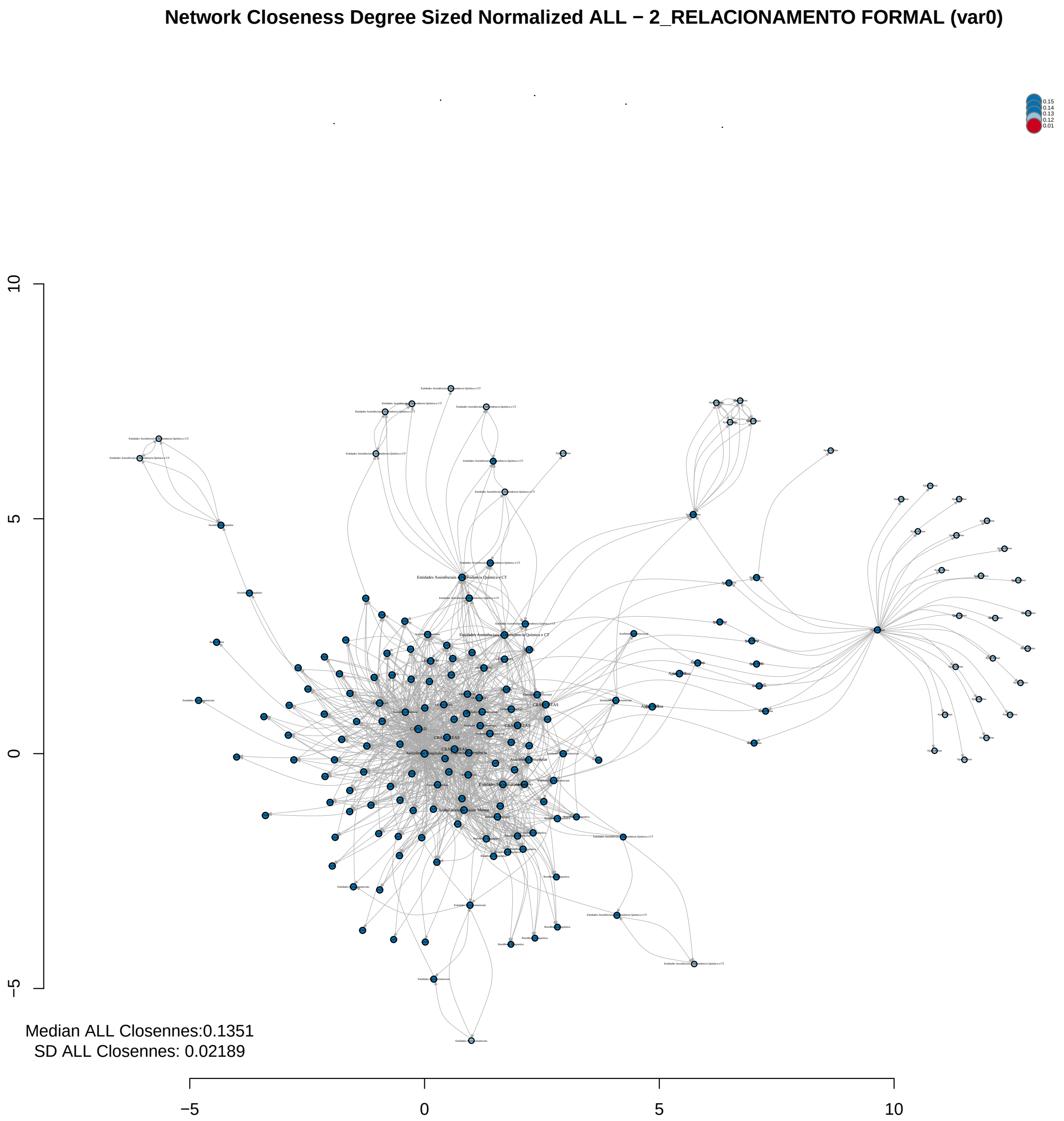
```
sd(var0_totalcloseness_n)
```

```
## [1] 0.02188651
```

3.8 Network Plotting Based On Normalized Closeness - ALL

```
V(var0)$allcloseness_n<-closeness(var0, weights = E(var0)$var0, mode="all", normalized = T)

#Get Variable
V(var0)$var0_color_degree<-round(V(var0)$allcloseness_n,2)

#Creating brewer pallette
vertex_var0_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(var0)$var0_color_degree)), "RdBu"))(
            length(unique(V(var0)$var0_color_degree)))

#Saving as Vertex properties
V(var0)$vertex_var0_color_degree<-
  vertex_var0_color_degree[as.numeric(
  cut(V(var0)$var0_color_degree,
      breaks=length(unique(V(var0)$var0_color_degree))))]

set.seed(123)
#Plotting based only on degree measures
edge.start <- ends(var0, es=E(var0), names=F)[,1]

# Fixing ego
minC <- rep(-Inf, vcount(var0))
maxC <- rep(Inf, vcount(var0))
minC[1] <- maxC[1] <- 0
co <- layout_with_fr(var0, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(var0)$weig


#PLotting
plot(var0,
     layout=co,
     #edge.color=V(var0)$vertex_var0_color_degree[edge.start],
     edge.arrow.size=closeness(var0, weights = E(var0)$var0, mode="all",normalized = T),
     edge.width=E(var0)$weight/10*mean(E(var0)$weight),
     edge.curved = TRUE,
     vertex.color=V(var0)$vertex_var0_color_degree,
     vertex.size=(closeness(var0, weights = E(var0)$var0, mode="all",normalized = T))*100,
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(var0,"LABEL_COR"),
     vertex.label.cex=closeness(var0, weights = E(var0)$var0, mode="all",normalized = T)*1.5,
     vertex.label.dist=0,
     rescale=F,
     xlim=range(co[,1]),
     ylim=range(co[,2])
     )
axis(1)
axis(2)


#Solving Problems with legend rendering
```

```
a<-V(var0)$var0_color_degree
b<-V(var0)$vertex_var0_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#Adding Legend
legend(x=range(co[,1])[2],
       y=range(co[,2])[2],
       legend=as.character(f),
       pch=21,
       col = "#777777",
       pt.bg=as.character(g),
       pt.cex=2,
       bty="n",
       ncol=1,
       lty=1,
       cex = .3)

#Adding Title
  title("Network Closeness Degree Sized Normalized ALL - 2_RELACIONAMENTO FORMAL (var0)", sub = "Source
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median ALL Closennes:%.4f\nSD ALL Closennes: %.5f",
             median(closeness(var0, mode="all", weights = E(var0)$var0, normalized = T)),
             sd(closeness(var0, mode="all", weights = E(var0)$var0, normalized = T))
             )
       )
```



Network Closeness Degree Sized Normalized ALL − 2_RELACIONAMENTO FORMAL (var0)

Source: from authors

3.9 Closeness Normalized

3.9.1 Saving to Igraph object

```
V(var0)$incloseness_n <- closeness(var0, weights = E(var0)$var0, mode = "in", normalized = T) %>% round
V(var0)$outcloseness_n <- closeness(var0, weights = E(var0)$var0, mode = "out", normalized = T) %>% rou
V(var0)$totalcloseness_n <- closeness(var0, weights = E(var0)$var0, mode = "total", normalized = T) %>%
```

3.10 Centralization Closeness

```
V(var0)$var0_centr_closeness<- centralization.closeness(var0)$res
var0_centr_closeness<- centralization.closeness(var0)$res
var0_centr_closeness_all<- centralization.closeness(var0)
```

3.10.1 Centralization

```
var0_centr_closeness_all$centralization
```

```
## [1] 0.06834656
```

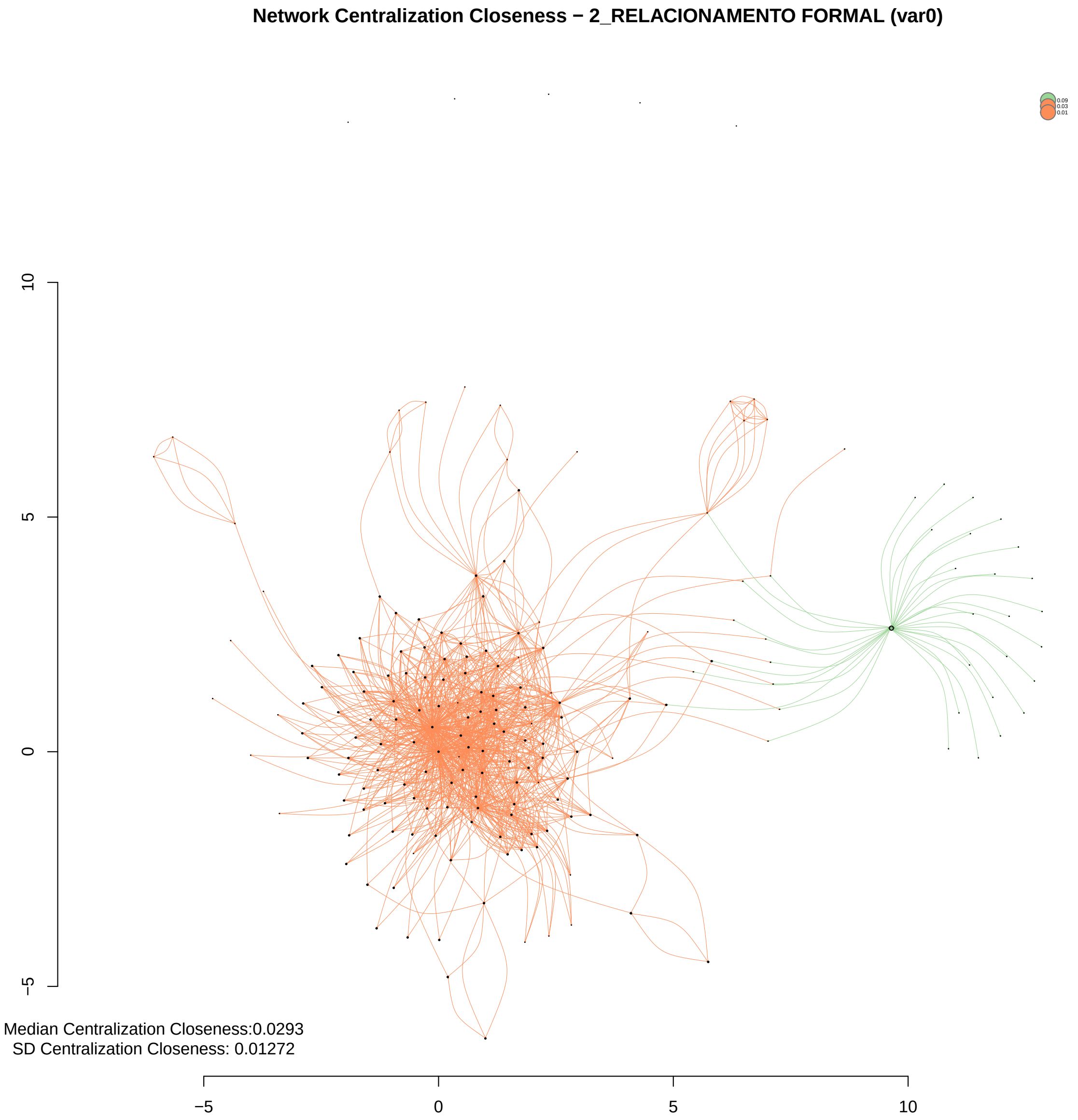
3.10.2 Theoretical Max

```
var0_centr_closeness_all$theoretical_max
```

```
## [1] 185.0053
```

3.11 Network Plotting Based On Centralization Closeness

```
V(var0)$var0_centr_closeness<- centralization.closeness(var0)$res

#Get Variable
V(var0)$var0_color_degree<-round(V(var0)$var0_centr_closeness,2)

#Creating brewer pallette
vertex_var0_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(var0)$var0_color_degree)), "Spectral"))(
            length(unique(V(var0)$var0_color_degree)))

#Saving as Vertex properties
V(var0)$vertex_var0_color_degree<-
  vertex_var0_color_degree[as.numeric(
  cut(V(var0)$var0_color_degree,
      breaks=length(unique(V(var0)$var0_color_degree))))]

set.seed(123)
#Plotting based only on degree measures
edge.start <- ends(var0, es=E(var0), names=F)[,1]

# Fixing ego
minC <- rep(-Inf, vcount(var0))
maxC <- rep(Inf, vcount(var0))
minC[1] <- maxC[1] <- 0
co <- layout_with_fr(var0, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(var0)$weig
#PLotting
plot(var0,
     layout=co,
     edge.color=V(var0)$vertex_var0_color_degree[edge.start],
     edge.arrow.size=centralization.closeness(var0)$res,
     edge.width=E(var0)$weight/10*mean(E(var0)$weight),
     edge.curved = TRUE,
     vertex.color=V(var0)$vertex_var0_color_degree,
     vertex.size=centralization.closeness(var0)$res*100,
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(var0,"LABEL_COR"),
     vertex.label.cex=centralization.closeness(var0)$res,
     vertex.label.dist=0,
     rescale=F,
     xlim=range(co[,1]),
     ylim=range(co[,2])
     )
axis(1)
axis(2)


#Solving Problems with legend rendering
a<-V(var0)$var0_color_degree
b<-V(var0)$vertex_var0_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#Adding Legend
legend(x=range(co[,1])[2],
       y=range(co[,2])[2],
       legend=as.character(f),
       pch=21,
       col = "#777777",
       pt.bg=as.character(g),
       pt.cex=2,
       bty="n",
       ncol=1,
       lty=1,
       cex = .3)

#Adding Title
  title("Network Centralization Closeness - 2_RELACIONAMENTO FORMAL (var0)", sub = "Source: from author
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median Centralization Closeness:%.4f\nSD Centralization Closeness: %.5f",
             median(centralization.closeness(var0)$res),
             sd(centralization.closeness(var0)$res)
             )
       )
```



Source: from authors

4 Closeness Dinamic Table

4.1 Getting Closeness Measures

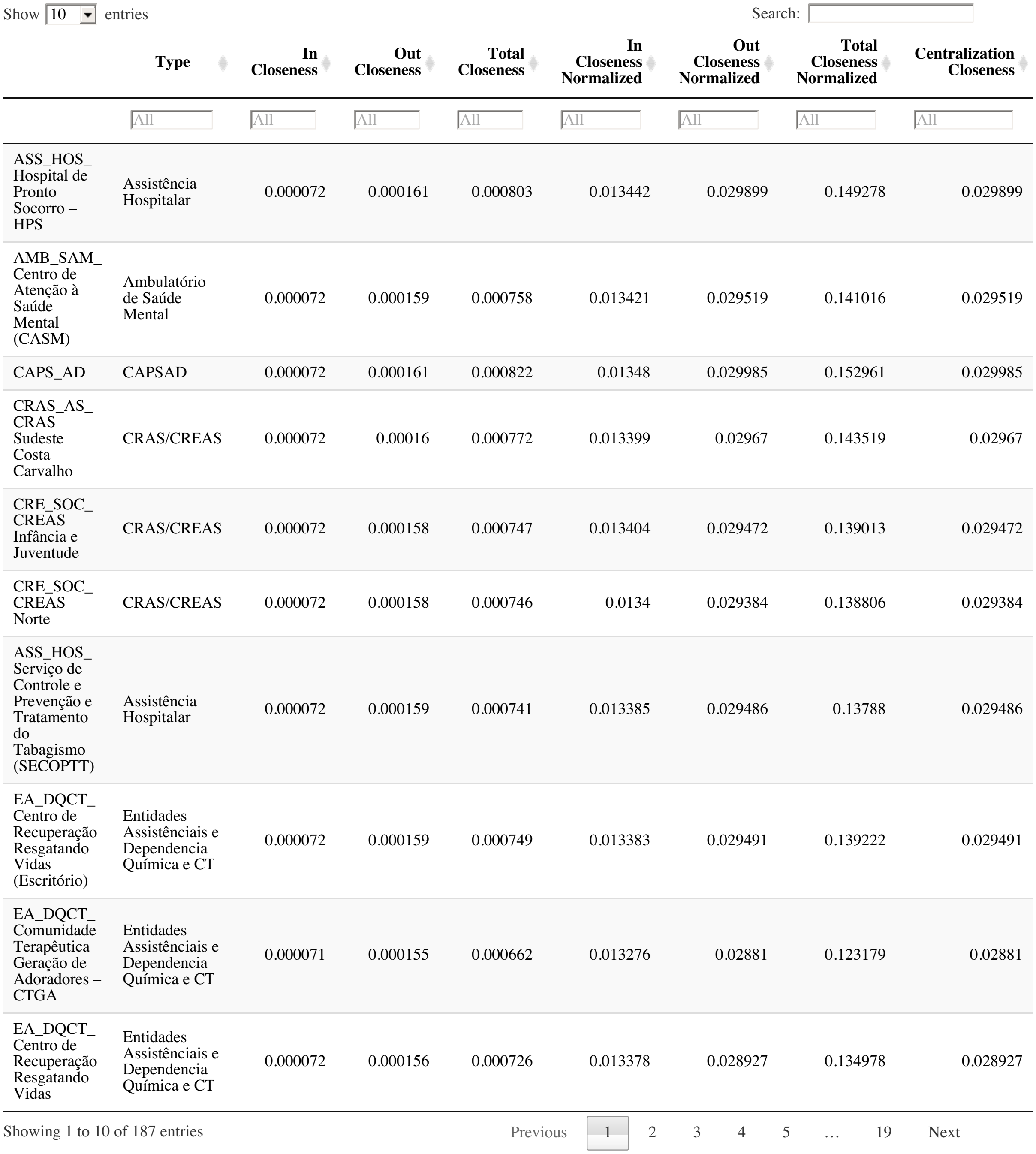
```
var0_incloseness<- closeness(var0, weights = E(var0)$var0, mode = "in") %>% round(6)
var0_outcloseness<- closeness(var0, weights = E(var0)$var0, mode = "out") %>% round(6)
var0_totalcloseness<- closeness(var0, weights = E(var0)$var0, mode = "total") %>% round(6)
var0_incloseness_n<- closeness(var0,weights = E(var0)$var0, mode = "in", normalized = T) %>% round(6)
var0_outcloseness_n<- closeness(var0,weights = E(var0)$var0, mode = "out", normalized = T) %>% round(6)
var0_totalcloseness_n<- closeness(var0,weights = E(var0)$var0, mode = "total", normalized = T) %>% roun
var0_centr_closeness <- centralization.closeness(var0)$res %>% round(6)
```

4.2 Creating a datagrame of measures

```
var0_df_closseness <- data.frame(
var0_incloseness,
var0_outcloseness,
var0_totalcloseness,
var0_incloseness_n,
var0_outcloseness_n,
var0_totalcloseness_n,
var0_centr_closeness) %>% round(6)

#Adding type
var0_df_closseness <-cbind(var0_df_closseness, V(var0)$LABEL_COR)

#Adding names
names(var0_df_closseness) <- c("In Closeness", "Out Closeness", "Total Closeness","In Closeness Normali

#Ordering Variables
var0_df_closseness<-var0_df_closseness[c("Type","In Closeness", "Out Closeness", "Total Closeness","In C
```

4.3 General tabel - DT

```
datatable(var0_df_closseness, filter = 'top')
```



Show 10 entries

Search:

	Type	In Closeness	Out Closeness	Total Closeness	In Closeness Normalized	Out Closeness Normalized	Total Closeness Normalized	Centralization Closeness
	All	All	All	All	All	All	All	All
ASS_HOS_ Hospital de Pronto Socorro – HPS	Assistência Hospitalar	0.000072	0.000161	0.000803	0.013442	0.029899	0.149278	0.029899
AMB_SAM_ Centro de Atenção à Saúde Mental (CASM)	Ambulatório de Saúde Mental	0.000072	0.000159	0.000758	0.013421	0.029519	0.141016	0.029519
CAPS_AD	CAPSAD	0.000072	0.000161	0.000822	0.01348	0.029985	0.152961	0.029985
CRAS_AS_ CRAS Sudeste Costa Carvalho	CRAS/CREAS	0.000072	0.00016	0.000772	0.013399	0.02967	0.143519	0.02967
CRE_SOC_ CREAS Infância e Juventude	CRAS/CREAS	0.000072	0.000158	0.000747	0.013404	0.029472	0.139013	0.029472
CRE_SOC_ CREAS Norte	CRAS/CREAS	0.000072	0.000158	0.000746	0.0134	0.029384	0.138806	0.029384
ASS_HOS_ Serviço de Controle e Prevenção e Tratamento do Tabagismo (SECOPTT)	Assistência Hospitalar	0.000072	0.000159	0.000741	0.013385	0.029486	0.13788	0.029486
EA_DQCT_ Centro de Recuperação Resgatando Vidas (Escritório)	Entidades Assistênciais e Dependencia Química e CT	0.000072	0.000159	0.000749	0.013383	0.029491	0.139222	0.029491
EA_DQCT_ Comunidade Terapêutica Geração de Adoradores – CTGA	Entidades Assistênciais e Dependencia Química e CT	0.000071	0.000155	0.000662	0.013276	0.02881	0.123179	0.02881
EA_DQCT_ Centro de Recuperação Resgatando Vidas	Entidades Assistênciais e Dependencia Química e CT	0.000072	0.000156	0.000726	0.013378	0.028927	0.134978	0.028927

Showing 1 to 10 of 187 entries

Previous 1 2 3 4 5 … 19 Next

4.4 Aggregating data from previous table - mean

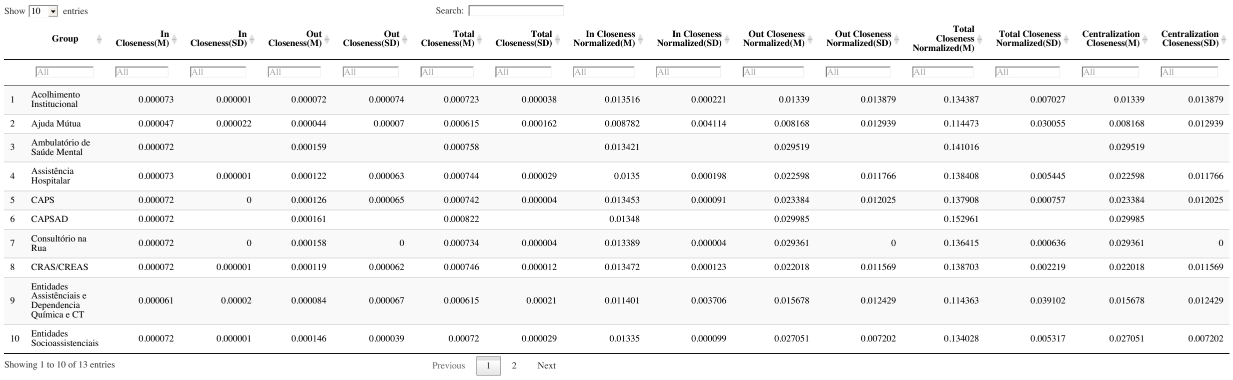
```
aggdata_mean <-aggregate(var0_df_closseness, by=list(var0_df_closseness$Type), FUN=mean, na.rm=TRUE)

names(aggdata_mean) <- c("Group","Type","In Closeness(M)", "Out Closeness(M)", "Total Closeness(M)","In

#Removing Type variable
aggdata_mean<-aggdata_mean[,-c(2)]
```

4.5 Aggregating data from previous table - sd

```
aggdata_sd <-aggregate(var0_df_closseness, by=list(var0_df_closseness$Type), FUN=sd, na.rm=TRUE)

names(aggdata_sd) <- c("Group","Type","In Closeness(SD)", "Out Closeness(SD)", "Total Closeness(SD)","I

#Removing Type variable
aggdata_sd<-aggdata_sd[,-c(2)]

#Merging mean and standart deviation
total_table <- merge(aggdata_mean,aggdata_sd,by="Group")

#Rounding
Group<-total_table[,c(1)] #Keeping group
total_table<-total_table[,-c(1)] %>% round(6) #Rouding
total_table<-cbind(Group,total_table) #Binding toghter

#Organizing Variabels
total_table<-total_table[c("Group","In Closeness(M)", "In Closeness(SD)", "Out Closeness(M)", "Out Clos
```

4.6 Plotting final table with round for Closseness

```
datatable(total_table, filter = 'top')
```



Show 10 entries Search:

	Group	In Closeness(M)	In Closeness(SD)	Out Closeness(M)	Out Closeness(SD)	Total Closeness(M)	Total Closeness(SD)	In Closeness Normalized(M)	In Closeness Normalized(SD)	Out Closeness Normalized(M)	Out Closeness Normalized(SD)	Total Closeness Normalized(M)	Total Closeness Normalized(SD)	Centralization Closeness(M)	Centralization Closeness(SD)
	All	All	All	All	All	All	All	All	All	All	All	All	All	All	All
1	Acolhimento Institucional	0.000073	0.000001	0.000072	0.000074	0.000723	0.000038	0.013516	0.000221	0.01339	0.013879	0.134387	0.007027	0.01339	0.013879
2	Ajuda Mútua	0.000047	0.000022	0.000044	0.00007	0.000615	0.000162	0.008782	0.004114	0.008168	0.012939	0.114473	0.030055	0.008168	0.012939
3	Ambulatório de Saúde Mental	0.000072		0.000159		0.000758		0.013421		0.029519		0.141016		0.029519	
4	Assistência Hospitalar	0.000073	0.000001	0.000122	0.000063	0.000744	0.000029	0.0135	0.000198	0.022598	0.011766	0.138408	0.005445	0.022598	0.011766
5	CAPS	0.000072	0	0.000126	0.000065	0.000742	0.000004	0.013453	0.000091	0.023384	0.012025	0.137908	0.000757	0.023384	0.012025
6	CAPSAD	0.000072		0.000161		0.000822		0.01348		0.029985		0.152961		0.029985	
7	Consultório na Rua	0.000072	0	0.000158	0	0.000734	0.000004	0.013389	0.000004	0.029361	0	0.136415	0.000636	0.029361	0
8	CRAS/CREAS	0.000072	0.000001	0.000119	0.000062	0.000746	0.000012	0.013472	0.000123	0.022018	0.011569	0.138703	0.002219	0.022018	0.011569
9	Entidades Assistênciais e Dependencia Química e CT	0.000061	0.00002	0.000084	0.000067	0.000615	0.00021	0.011401	0.003706	0.015678	0.012429	0.114363	0.039102	0.015678	0.012429
10	Entidades Socioassistenciais	0.000072	0.000001	0.000146	0.000039	0.00072	0.000029	0.01335	0.000099	0.027051	0.007202	0.134028	0.005317	0.027051	0.007202

Showing 1 to 10 of 13 entries

5 Saving objects with new variables and changes

```
save.image("~/SNArRDJF/Robject/3_closeness_var0.RData")
```