SNA Closeness 1.1 - REDE COMPLETA (full_no_zero)

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 1.1 - REDE COMPLETA (full_no_zero)

########################## 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_full_no_zero.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

```
full_no_zero<-simplify(full_no_zero) #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(full_no_zero)$incloseness <- closeness(full_no_zero, mode = "in", weights = E(full_no_zero)$full_no_ze
V(full_no_zero)$outcloseness <- closeness(full_no_zero, mode = "out", weights = E(full_no_zero)$full_no
V(full_no_zero)$totalcloseness <- closeness(full_no_zero, mode = "total", weights = E(full_no_zero)$full
```

3.1.2 Saving to Environment

```
full_no_zero_incloseness<- closeness(full_no_zero, mode = "in", weights = E(full_no_zero)$full_no_zero)
full_no_zero_outcloseness<- closeness(full_no_zero, mode = "out", weights = E(full_no_zero)$full_no_zer
full_no_zero_totalcloseness<- closeness(full_no_zero, mode = "total", weights = E(full_no_zero)$full_no
```

3.1.3 Closeness Non-normalized - in

```
summary(full_no_zero_incloseness)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000290 0.0001480 0.0001500 0.0001486 0.0001500 0.0001550
```

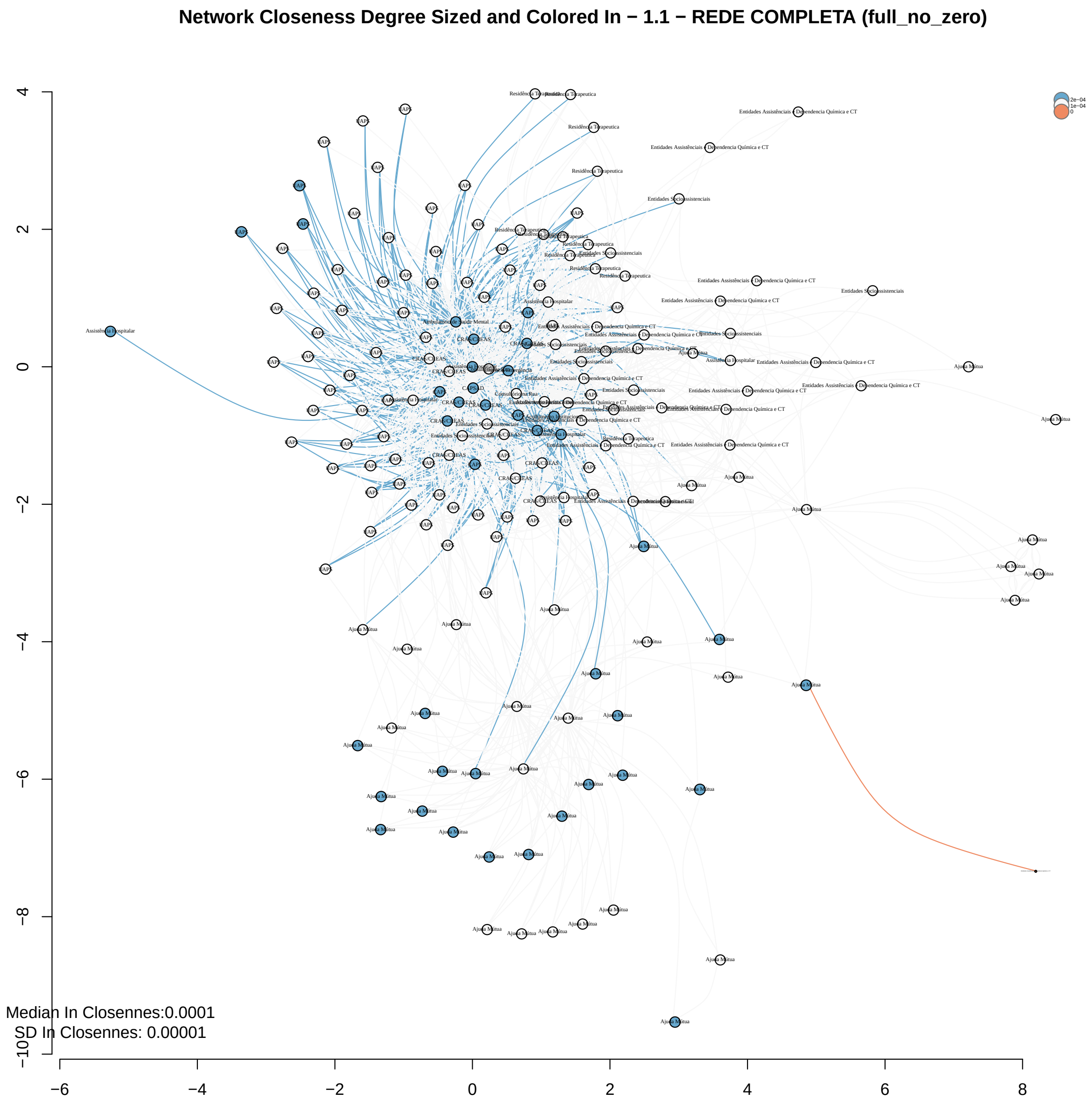
```
sd(full_no_zero_incloseness)
```

```
## [1] 9.003136e-06
```

3.2 Network Plotting Based On Non-normalized Closeness - IN

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

#Get Variable
V(full_no_zero)$full_no_zero_color_degree<-round(V(full_no_zero)$incloseness,4)

#Creating brewer pallette
vertex_full_no_zero_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(full_no_zero)$full_no_zero_color_degree)), "RdBu"))(
            length(unique(V(full_no_zero)$full_no_zero_color_degree)))

#Saving as Vertex properties
V(full_no_zero)$vertex_full_no_zero_color_degree<-
  vertex_full_no_zero_color_degree[as.numeric(
  cut(V(full_no_zero)$full_no_zero_color_degree,
      breaks=length(unique(V(full_no_zero)$full_no_zero_color_degree))))]

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

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


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

```r
     )
axis(1)
axis(2)


#Solving Problems with legend rendering
a<-V(full_no_zero)$full_no_zero_color_degree
b<-V(full_no_zero)$vertex_full_no_zero_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 - 1.1 - REDE COMPLETA (full_no_zero)", sub = "Sou
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median In Closennes:%.4f\nSD In Closennes: %.5f",
             median(closeness(full_no_zero, mode="in", weights = E(full_no_zero)$full_no_zero)),
             sd(closeness(full_no_zero, mode="in", weights = E(full_no_zero)$full_no_zero))
             )
       )
```



Source: from authors

3.2.1 Closeness Non-normalized - OUT

```
summary(full_no_zero_outcloseness)
```

```
##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000029 0.001236 0.001543 0.001270 0.001600 0.002110
```

```
sd(full_no_zero_outcloseness)
```

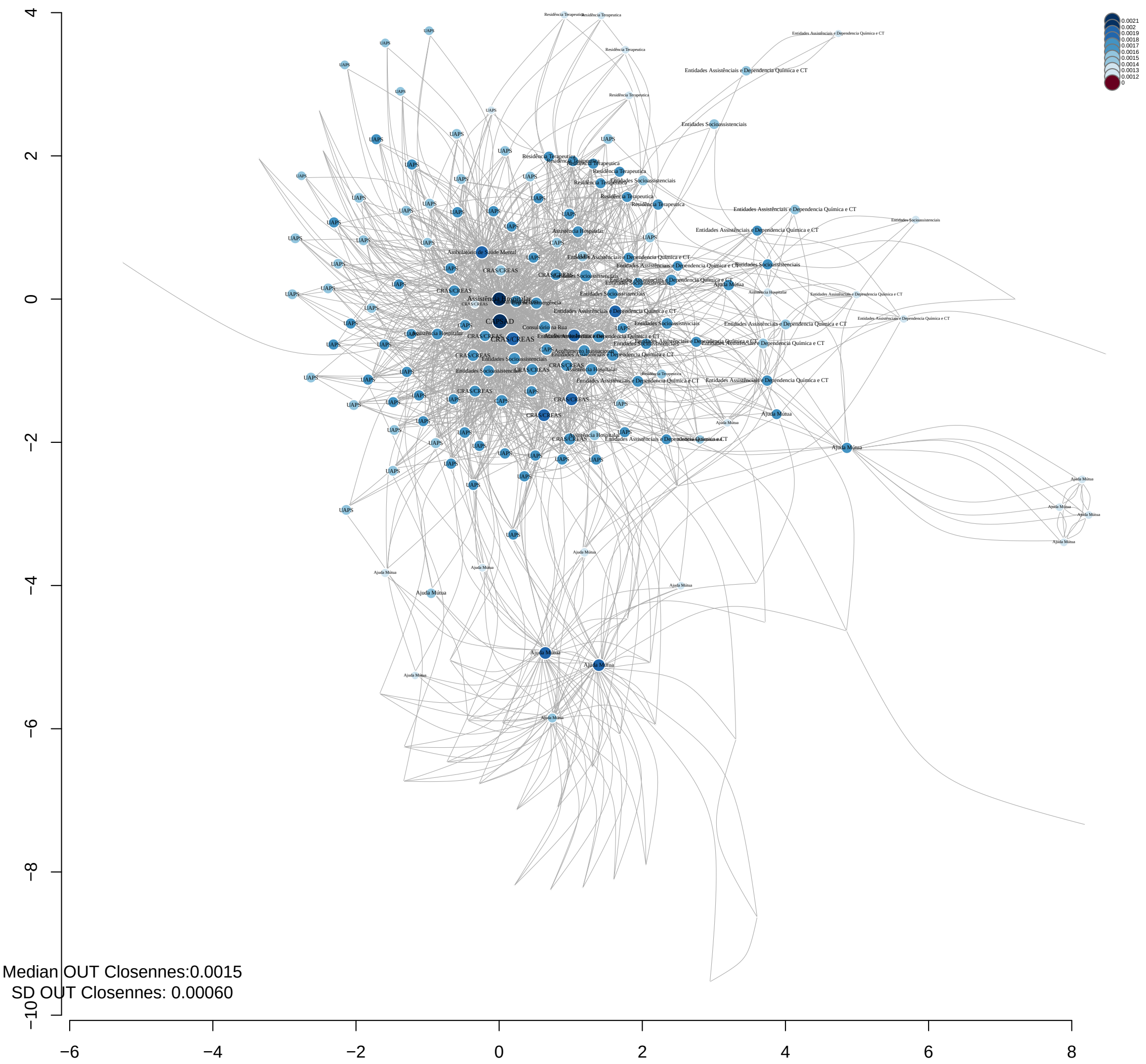
```
## [1] 0.0006047786
```

3.3 Network Plotting Based On Non-normalized Closeness - OUT

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

#Get Variable
V(full_no_zero)$full_no_zero_color_degree<-round(V(full_no_zero)$outcloseness,4)

#Creating brewer pallette
vertex_full_no_zero_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(full_no_zero)$full_no_zero_color_degree)), "RdBu"))(
            length(unique(V(full_no_zero)$full_no_zero_color_degree)))

#Saving as Vertex properties
V(full_no_zero)$vertex_full_no_zero_color_degree<-
  vertex_full_no_zero_color_degree[as.numeric(
  cut(V(full_no_zero)$full_no_zero_color_degree,
      breaks=length(unique(V(full_no_zero)$full_no_zero_color_degree))))]

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

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


#PLotting
plot(full_no_zero,
     layout=co,
     #edge.color=V(full_no_zero)$vertex_full_no_zero_color_degree[edge.start],
     edge.arrow.size=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="out"),
     edge.width=E(full_no_zero)$weight/2*mean(E(full_no_zero)$weight),
     edge.curved = TRUE,
     vertex.color=V(full_no_zero)$vertex_full_no_zero_color_degree,
     vertex.size=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="out")*10^4,
     vertex.frame.color="white",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(full_no_zero,"LABEL_COR"),
     vertex.label.cex=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, 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(full_no_zero)$full_no_zero_color_degree
b<-V(full_no_zero)$vertex_full_no_zero_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 - 1.1 - REDE COMPLETA (full_no_zero)", sub = "S
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median OUT Closennes:%.4f\nSD OUT Closennes: %.5f",
             median(closeness(full_no_zero, mode="out", weights = E(full_no_zero)$full_no_zero)),
             sd(closeness(full_no_zero, mode="out", weights = E(full_no_zero)$full_no_zero))
             )
       )
```

Network Closeness Degree Sized and Colored OUT − 1.1 − REDE COMPLETA (full_no_zero)



Source: from authors

3.3.1 Closeness Non-normalized - ALL

```
summary(full_no_zero_totalcloseness)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001529 0.002278 0.002451 0.002411 0.002564 0.003968
```

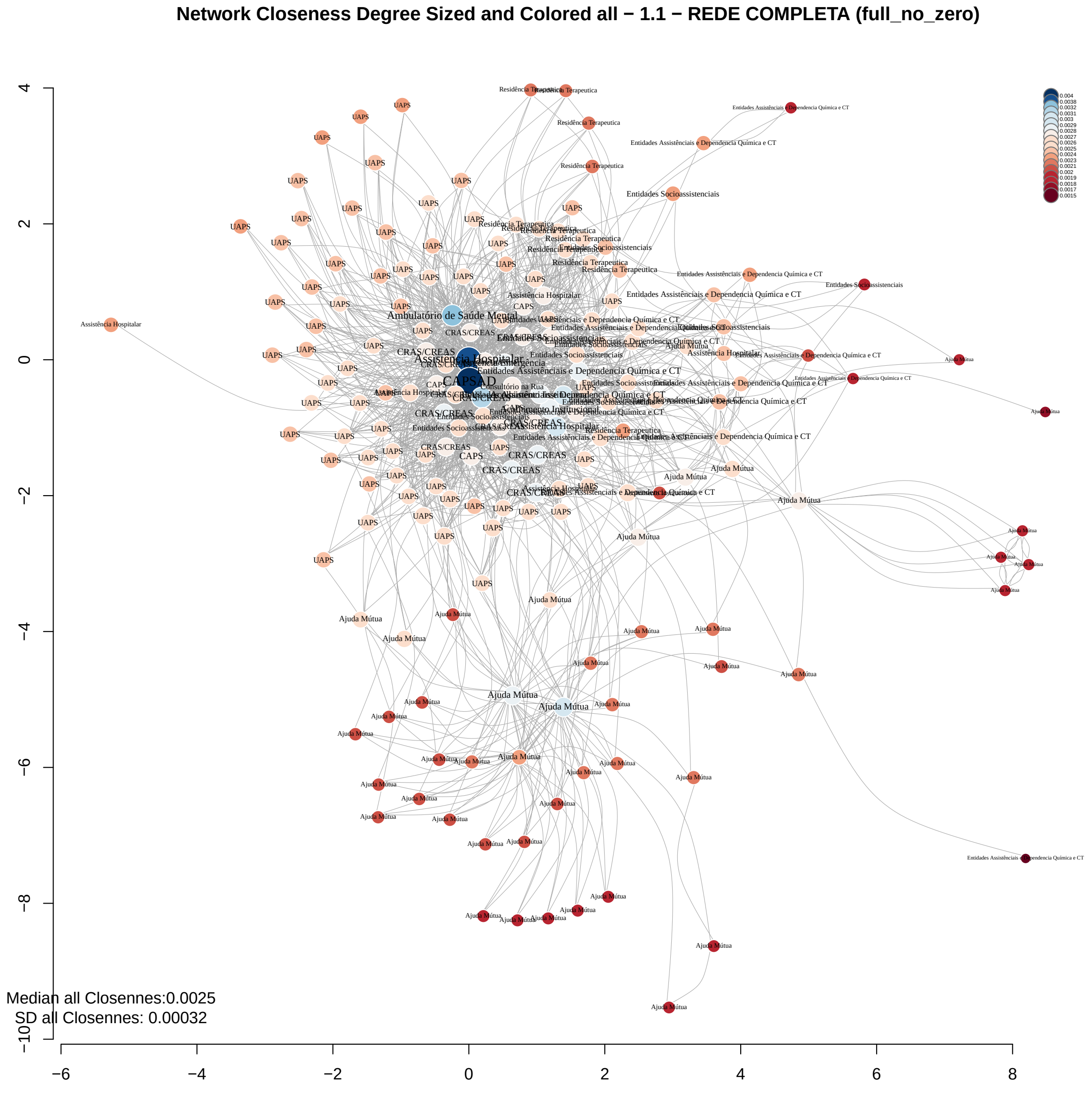
```
sd(full_no_zero_totalcloseness)
```

```
## [1] 0.0003229738
```

3.4 Network Plotting Based On Non-normalized Closeness - ALL

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

#Get Variable
V(full_no_zero)$full_no_zero_color_degree<-round(V(full_no_zero)$allcloseness,4)

#Creating brewer pallette
vertex_full_no_zero_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(full_no_zero)$full_no_zero_color_degree)), "RdBu"))(
            length(unique(V(full_no_zero)$full_no_zero_color_degree)))

#Saving as Vertex properties
V(full_no_zero)$vertex_full_no_zero_color_degree<-
  vertex_full_no_zero_color_degree[as.numeric(
  cut(V(full_no_zero)$full_no_zero_color_degree,
      breaks=length(unique(V(full_no_zero)$full_no_zero_color_degree))))]

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

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


#PLotting
plot(full_no_zero,
     layout=co,
     #edge.color=V(full_no_zero)$vertex_full_no_zero_color_degree[edge.start],
     edge.arrow.size=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="all"),
     edge.width=E(full_no_zero)$weight/2*mean(E(full_no_zero)$weight),
     edge.curved = TRUE,
     vertex.color=V(full_no_zero)$vertex_full_no_zero_color_degree,
     vertex.size=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="all")*10^4,
     vertex.frame.color="white",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(full_no_zero,"LABEL_COR"),
     vertex.label.cex=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, 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
```

```
a<-V(full_no_zero)$full_no_zero_color_degree
b<-V(full_no_zero)$vertex_full_no_zero_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 - 1.1 - REDE COMPLETA (full_no_zero)", sub = "S
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median all Closennes:%.4f\nSD all Closennes: %.5f",
             median(closeness(full_no_zero, mode="all", weights = E(full_no_zero)$full_no_zero)),
             sd(closeness(full_no_zero, mode="all", weights = E(full_no_zero)$full_no_zero))
             )
       )
```



Source: from authors

3.5 Closeness Normalized

3.5.1 Saving to Igraph object

```
V(full_no_zero)$incloseness_n <- closeness(full_no_zero, mode = "in",, weights = E(full_no_zero)$full_no
V(full_no_zero)$outcloseness_n <- closeness(full_no_zero, mode = "out", normalized = T, weights = E(full
V(full_no_zero)$totalcloseness_n <- closeness(full_no_zero, mode = "total", normalized = T, weights = E
```

3.5.2 Saving to Environment

```
full_no_zero_incloseness_n<- closeness(full_no_zero, mode = "in", normalized = T, weights = E(full_no_ze
full_no_zero_outcloseness_n<- closeness(full_no_zero, mode = "out", normalized = T, weights = E(full_no_
full_no_zero_totalcloseness_n<- closeness(full_no_zero, mode = "total", normalized = T, weights = E(full
```

3.5.3 Closeness Normalized - IN

```
summary(full_no_zero_incloseness_n)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005348 0.027540 0.027830 0.027620 0.027880 0.028910
```

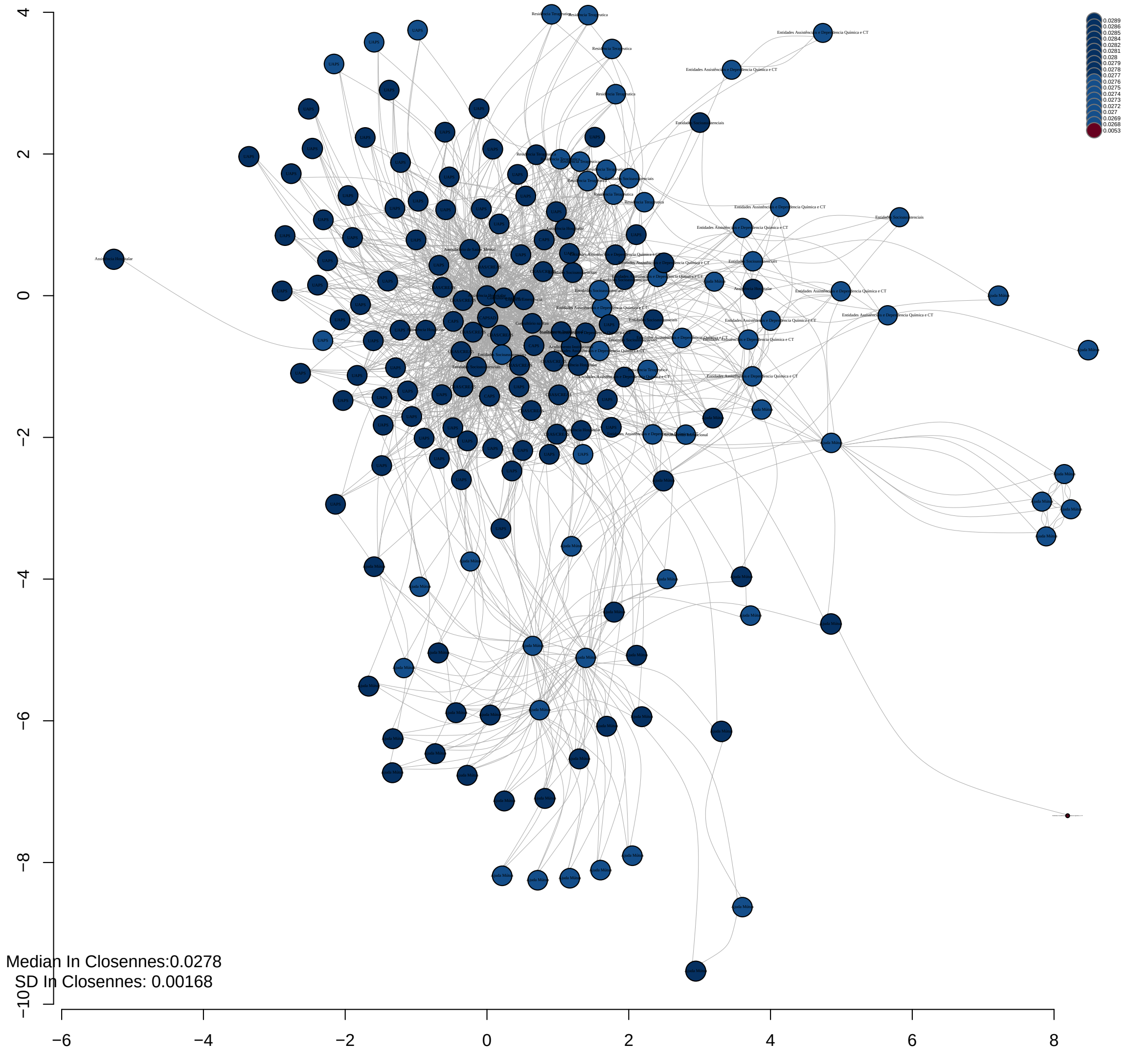
```
sd(full_no_zero_incloseness_n)
```

```
## [1] 0.001675712
```

3.6 Network Plotting Based On Normalized Closeness - IN

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

#Get Variable
V(full_no_zero)$full_no_zero_color_degree<-round(V(full_no_zero)$incloseness_n,4)

#Creating brewer pallette
vertex_full_no_zero_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(full_no_zero)$full_no_zero_color_degree)), "RdBu"))(
            length(unique(V(full_no_zero)$full_no_zero_color_degree)))

#Saving as Vertex properties
V(full_no_zero)$vertex_full_no_zero_color_degree<-
  vertex_full_no_zero_color_degree[as.numeric(
  cut(V(full_no_zero)$full_no_zero_color_degree,
      breaks=length(unique(V(full_no_zero)$full_no_zero_color_degree))))]

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

# Fixing ego
minC <- rep(-Inf, vcount(full_no_zero))
maxC <- rep(Inf, vcount(full_no_zero))
minC[1] <- maxC[1] <- 0
co <- layout_with_fr(full_no_zero, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(fu
#PLotting
plot(full_no_zero,
     layout=co,
     #edge.color=V(full_no_zero)$vertex_full_no_zero_color_degree[edge.start],
     edge.arrow.size=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="in",normalize
     edge.width=E(full_no_zero)$weight/10*mean(E(full_no_zero)$weight),
     edge.curved = TRUE,
     vertex.color=V(full_no_zero)$vertex_full_no_zero_color_degree,
     vertex.size=(closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="in",normalized =
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(full_no_zero,"LABEL_COR"),
     vertex.label.cex=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="in",normaliz
     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(full_no_zero)$full_no_zero_color_degree
b<-V(full_no_zero)$vertex_full_no_zero_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 - 1.1 - REDE COMPLETA (full_no_zero)", sub = "Sou
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median In Closennes:%.4f\nSD In Closennes: %.5f",
             median(closeness(full_no_zero, mode="in", weights = E(full_no_zero)$full_no_zero, normalize
             sd(closeness(full_no_zero, mode="in", weights = E(full_no_zero)$full_no_zero, normalized =
             )
       )
```

Network Closeness Degree Sized Normalized In − 1.1 − REDE COMPLETA (full_no_zero)



Source: from authors

###Closeness Normalized - OUT

```r
summary(full_no_zero_outcloseness_n)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005348 0.229800 0.287000 0.236300 0.297600 0.392400
```

```r
sd(full_no_zero_outcloseness_n)
```

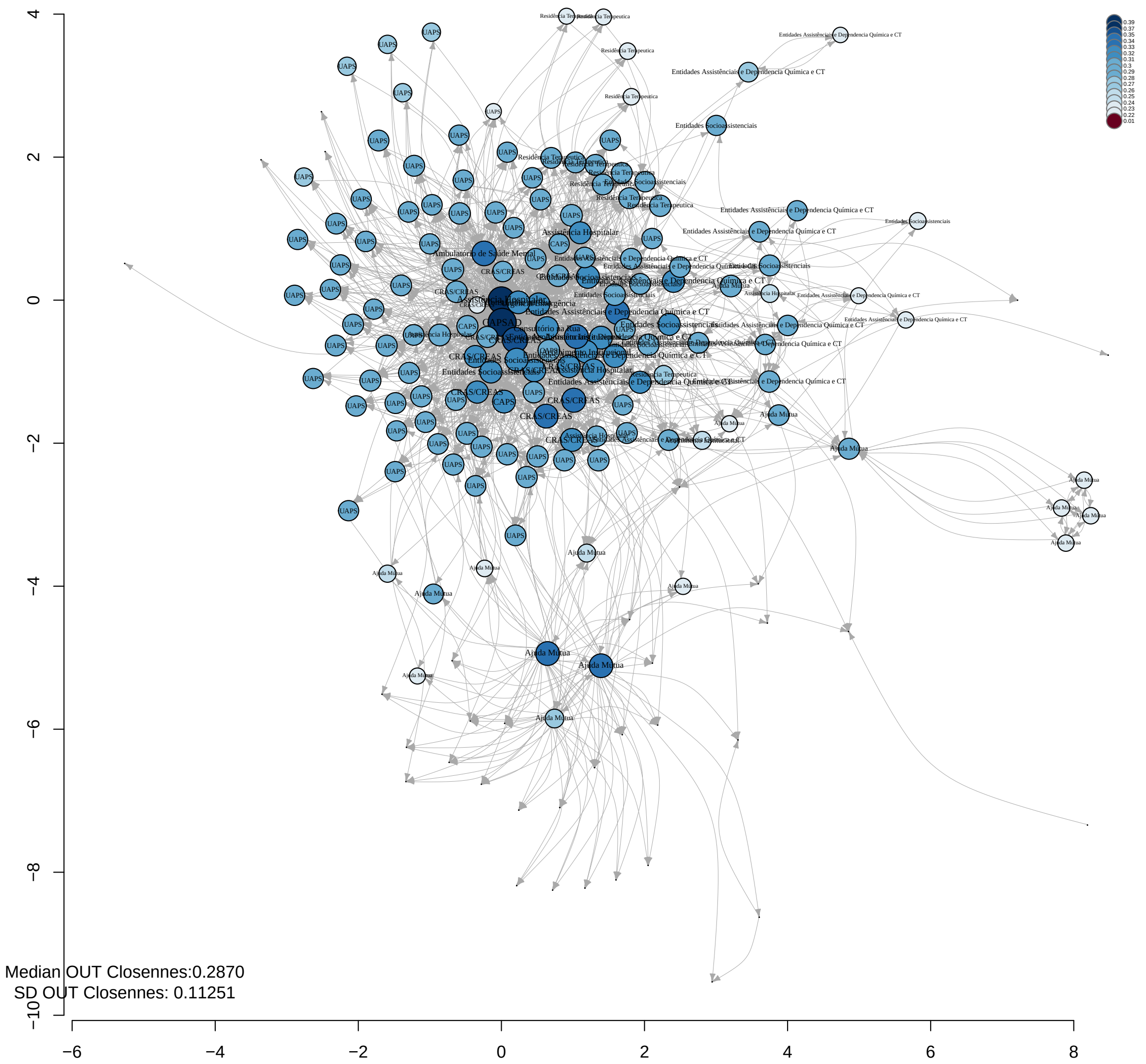
```
## [1] 0.1125115
```

3.7 Network Plotting Based On Normalized Closeness - OUT

```r
V(full_no_zero)$outcloseness_n<-closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="ou

#Get Variable
V(full_no_zero)$full_no_zero_color_degree<-round(V(full_no_zero)$outcloseness_n,2)

#Creating brewer pallette
vertex_full_no_zero_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(full_no_zero)$full_no_zero_color_degree)), "RdBu"))(
            length(unique(V(full_no_zero)$full_no_zero_color_degree)))

#Saving as Vertex properties
V(full_no_zero)$vertex_full_no_zero_color_degree<-
  vertex_full_no_zero_color_degree[as.numeric(
  cut(V(full_no_zero)$full_no_zero_color_degree,
      breaks=length(unique(V(full_no_zero)$full_no_zero_color_degree))))]

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

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


#PLotting
plot(full_no_zero,
     layout=co,
     #edge.color=V(full_no_zero)$vertex_full_no_zero_color_degree[edge.start],
     edge.arrow.size=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="out",normali
     edge.width=E(full_no_zero)$weight/10*mean(E(full_no_zero)$weight),
     edge.curved = TRUE,
     vertex.color=V(full_no_zero)$vertex_full_no_zero_color_degree,
     vertex.size=(closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="out",normalized
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(full_no_zero,"LABEL_COR"),
     vertex.label.cex=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="out",normali
     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(full_no_zero)$full_no_zero_color_degree
b<-V(full_no_zero)$vertex_full_no_zero_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 - 1.1 - REDE COMPLETA (full_no_zero)", sub = "So
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median OUT Closennes:%.4f\nSD OUT Closennes: %.5f",
             median(closeness(full_no_zero, mode="out", weights = E(full_no_zero)$full_no_zero, normali
             sd(closeness(full_no_zero, mode="out", weights = E(full_no_zero)$full_no_zero, normalized
             )
       )
```

Network Closeness Degree Sized Normalized OUT – 1.1 – REDE COMPLETA (full_no_zero)



Source: from authors

3.7.1 Closeness Normalized - ALL

```
summary(full_no_zero_totalcloseness_n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2844  0.4237  0.4559  0.4484  0.4769  0.7381
```

```
sd(full_no_zero_totalcloseness_n)
```

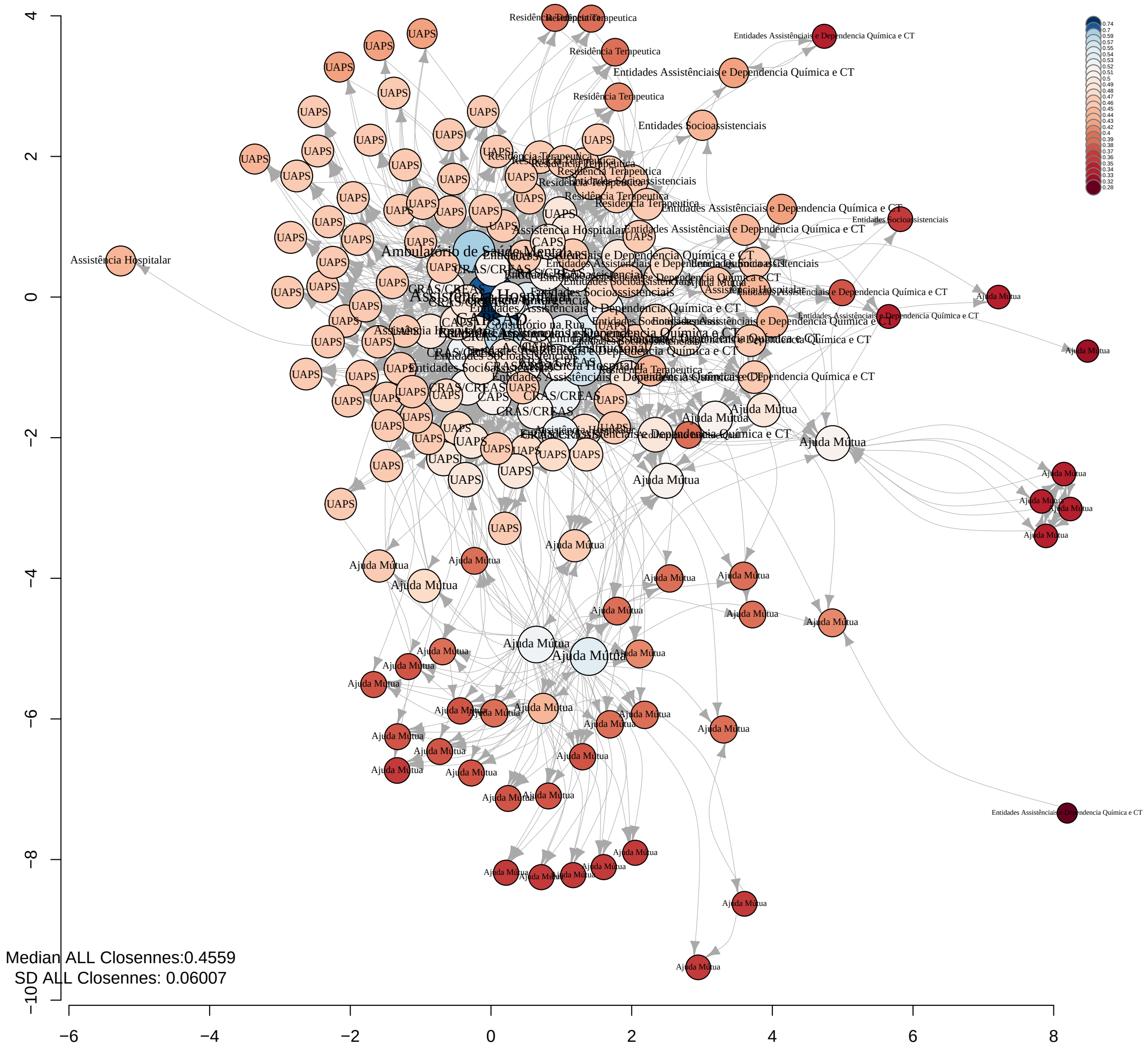
```
## [1] 0.06006723
```

3.8 Network Plotting Based On Normalized Closeness - ALL

```
V(full_no_zero)$allcloseness_n<-closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="al

#Get Variable
V(full_no_zero)$full_no_zero_color_degree<-round(V(full_no_zero)$allcloseness_n,2)

#Creating brewer pallette
vertex_full_no_zero_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(full_no_zero)$full_no_zero_color_degree)), "RdBu"))(
            length(unique(V(full_no_zero)$full_no_zero_color_degree)))

#Saving as Vertex properties
V(full_no_zero)$vertex_full_no_zero_color_degree<-
  vertex_full_no_zero_color_degree[as.numeric(
  cut(V(full_no_zero)$full_no_zero_color_degree,
      breaks=length(unique(V(full_no_zero)$full_no_zero_color_degree))))]

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

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


#PLotting
plot(full_no_zero,
     layout=co,
     #edge.color=V(full_no_zero)$vertex_full_no_zero_color_degree[edge.start],
     edge.arrow.size=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="all",normali
     edge.width=E(full_no_zero)$weight/10*mean(E(full_no_zero)$weight),
     edge.curved = TRUE,
     vertex.color=V(full_no_zero)$vertex_full_no_zero_color_degree,
     vertex.size=(closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="all",normalized
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(full_no_zero,"LABEL_COR"),
     vertex.label.cex=closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode="all",normali
     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(full_no_zero)$full_no_zero_color_degree
b<-V(full_no_zero)$vertex_full_no_zero_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 - 1.1 - REDE COMPLETA (full_no_zero)", sub = "Sou
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median ALL Closennes:%.4f\nSD ALL Closennes: %.5f",
             median(closeness(full_no_zero, mode="all", weights = E(full_no_zero)$full_no_zero, normali
             sd(closeness(full_no_zero, mode="all", weights = E(full_no_zero)$full_no_zero, normalized =
             )
       )
```

Network Closeness Degree Sized Normalized ALL − 1.1 − REDE COMPLETA (full_no_zero)



Source: from authors

3.9 Closeness Normalized

3.9.1 Saving to Igraph object

```
V(full_no_zero)$incloseness_n <- closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode =
V(full_no_zero)$outcloseness_n <- closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode =
V(full_no_zero)$totalcloseness_n <- closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode
```

3.10 Centralization Closseness

```
V(full_no_zero)$full_no_zero_centr_closeness<- centralization.closeness(full_no_zero)$res
full_no_zero_centr_closeness<- centralization.closeness(full_no_zero)$res
full_no_zero_centr_closeness_all<- centralization.closeness(full_no_zero)
```

3.10.1 Centralization

```
full_no_zero_centr_closeness_all$centralization
```

```
## [1] 0.15777
```

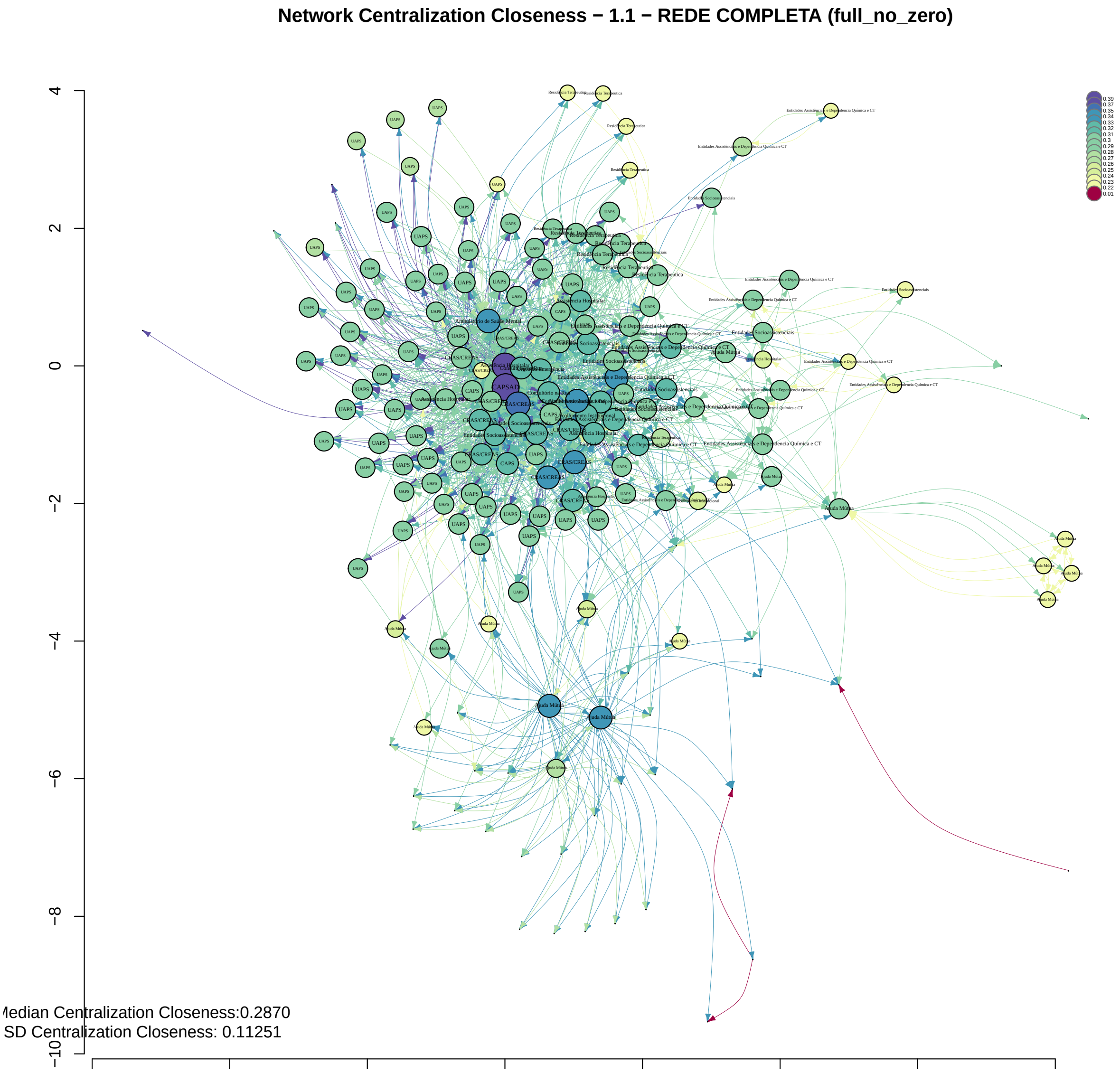
3.10.2 Theoretical Max

```
full_no_zero_centr_closeness_all$theoretical_max
```

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

3.11 Network Plotting Based On Centralization Closeness

```
V(full_no_zero)$full_no_zero_centr_closeness<- centralization.closeness(full_no_zero)$res

#Get Variable
V(full_no_zero)$full_no_zero_color_degree<-round(V(full_no_zero)$full_no_zero_centr_closeness,2)

#Creating brewer pallette
vertex_full_no_zero_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
          V(full_no_zero)$full_no_zero_color_degree)), "Spectral"))(
            length(unique(V(full_no_zero)$full_no_zero_color_degree)))

#Saving as Vertex properties
V(full_no_zero)$vertex_full_no_zero_color_degree<-
  vertex_full_no_zero_color_degree[as.numeric(
  cut(V(full_no_zero)$full_no_zero_color_degree,
      breaks=length(unique(V(full_no_zero)$full_no_zero_color_degree))))]

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

# Fixing ego
minC <- rep(-Inf, vcount(full_no_zero))
maxC <- rep(Inf, vcount(full_no_zero))
minC[1] <- maxC[1] <- 0
co <- layout_with_fr(full_no_zero, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(fu
#PLotting
plot(full_no_zero,
     layout=co,
     edge.color=V(full_no_zero)$vertex_full_no_zero_color_degree[edge.start],
     edge.arrow.size=centralization.closeness(full_no_zero)$res,
     edge.width=E(full_no_zero)$weight/10*mean(E(full_no_zero)$weight),
     edge.curved = TRUE,
     vertex.color=V(full_no_zero)$vertex_full_no_zero_color_degree,
     vertex.size=centralization.closeness(full_no_zero)$res*100,
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(full_no_zero,"LABEL_COR"),
     vertex.label.cex=centralization.closeness(full_no_zero)$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(full_no_zero)$full_no_zero_color_degree
b<-V(full_no_zero)$vertex_full_no_zero_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 - 1.1 - REDE COMPLETA (full_no_zero)", sub = "Source: from au
  text(
    x=range(co[,1])[1],
    y=range(co[,2])[1],
      labels = sprintf(
             "Median Centralization Closeness:%.4f\nSD Centralization Closeness: %.5f",
            median(centralization.closeness(full_no_zero)$res),
            sd(centralization.closeness(full_no_zero)$res)
            )
      )
```



Source: from authors

4 Closeness Dinamic Table

4.1 Getting Closeness Measures

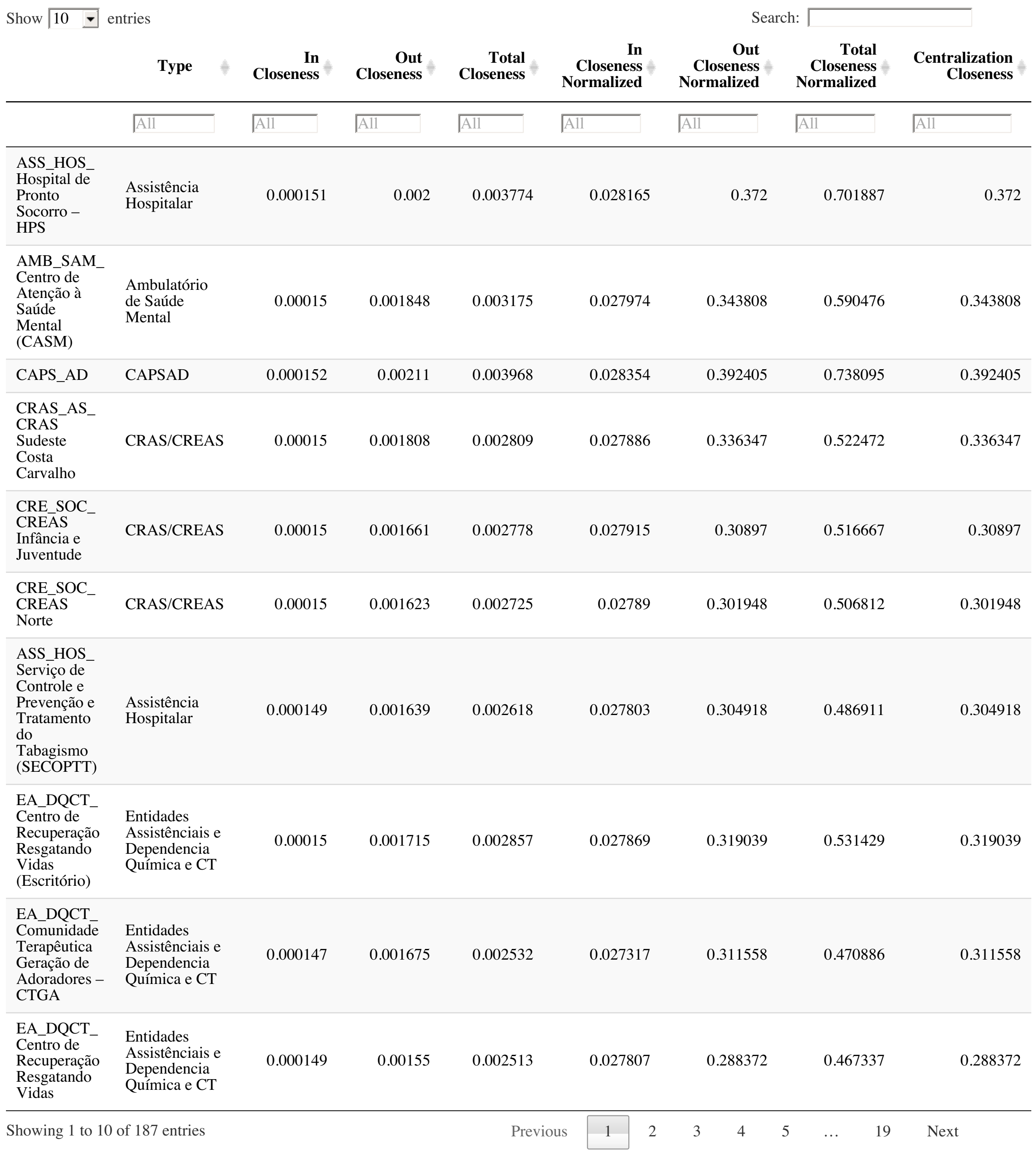
```
full_no_zero_incloseness<- closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode = "in")
full_no_zero_outcloseness<- closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode = "out
full_no_zero_totalcloseness<- closeness(full_no_zero, weights = E(full_no_zero)$full_no_zero, mode = "to
full_no_zero_incloseness_n<- closeness(full_no_zero,weights = E(full_no_zero)$full_no_zero, mode = "in"
full_no_zero_outcloseness_n<- closeness(full_no_zero,weights = E(full_no_zero)$full_no_zero, mode = "ou
full_no_zero_totalcloseness_n<- closeness(full_no_zero,weights = E(full_no_zero)$full_no_zero, mode = "
full_no_zero_centr_closeness <- centralization.closeness(full_no_zero)$res %>% round(6)
```

4.2 Creating a datagrame of measures

```
full_no_zero_df_closseness <- data.frame(
full_no_zero_incloseness,
full_no_zero_outcloseness,
full_no_zero_totalcloseness,
full_no_zero_incloseness_n,
full_no_zero_outcloseness_n,
full_no_zero_totalcloseness_n,
full_no_zero_centr_closeness) %>% round(6)

#Adding type
full_no_zero_df_closseness <-cbind(full_no_zero_df_closseness, V(full_no_zero)$LABEL_COR)

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

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

4.3 General tabel - DT

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



	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	0.000151	0.002	0.003774	0.028165	0.372	0.701887	0.372
AMB_SAM_ Centro de Atenção à Saúde Mental (CASM)	Ambulatório de Saúde Mental	0.00015	0.001848	0.003175	0.027974	0.343808	0.590476	0.343808
CAPS_AD	CAPSAD	0.000152	0.00211	0.003968	0.028354	0.392405	0.738095	0.392405
CRAS_AS_ CRAS Sudeste Costa Carvalho	CRAS/CREAS	0.00015	0.001808	0.002809	0.027886	0.336347	0.522472	0.336347
CRE_SOC_ CREAS Infância e Juventude	CRAS/CREAS	0.00015	0.001661	0.002778	0.027915	0.30897	0.516667	0.30897
CRE_SOC_ CREAS Norte	CRAS/CREAS	0.00015	0.001623	0.002725	0.02789	0.301948	0.506812	0.301948
ASS_HOS_ Serviço de Controle e Prevenção e Tratamento do Tabagismo (SECOPTT)	Assistência Hospitalar	0.000149	0.001639	0.002618	0.027803	0.304918	0.486911	0.304918
EA_DQCT_ Centro de Recuperação Resgatando Vidas (Escritório)	Entidades Assistênciais e Dependencia Química e CT	0.00015	0.001715	0.002857	0.027869	0.319039	0.531429	0.319039
EA_DQCT_ Comunidade Terapêutica Geração de Adoradores – CTGA	Entidades Assistênciais e Dependencia Química e CT	0.000147	0.001675	0.002532	0.027317	0.311558	0.470886	0.311558
EA_DQCT_ Centro de Recuperação Resgatando Vidas	Entidades Assistênciais e Dependencia Química e CT	0.000149	0.00155	0.002513	0.027807	0.288372	0.467337	0.288372

Showing 1 to 10 of 187 entries

4.4 Aggregating data from previous table - mean

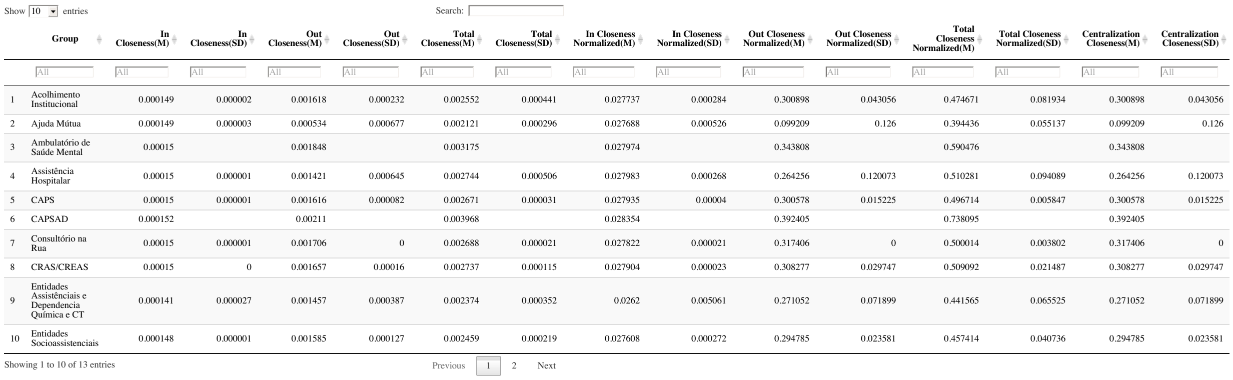
```
aggdata_mean <-aggregate(full_no_zero_df_closseness, by=list(full_no_zero_df_closseness$Type), FUN=mean

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(full_no_zero_df_closseness, by=list(full_no_zero_df_closseness$Type), FUN=sd, na

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	0.000149	0.000002	0.001618	0.000232	0.002552	0.000441	0.027737	0.000284	0.300898	0.043056	0.474671	0.081934	0.300898	0.043056
2	Ajuda Mútua	0.000149	0.000003	0.000534	0.000677	0.002121	0.000296	0.027688	0.000526	0.099209	0.126	0.394436	0.055137	0.099209	0.126
3	Ambulatório de Saúde Mental	0.00015		0.001848		0.003175		0.027974		0.343808		0.590476		0.343808	
4	Assistência Hospitalar	0.00015	0.000001	0.001421	0.000645	0.002744	0.000506	0.027983	0.000268	0.264256	0.120073	0.510281	0.094089	0.264256	0.120073
5	CAPS	0.00015	0.000001	0.001616	0.000082	0.002671	0.000031	0.027935	0.00004	0.300578	0.015225	0.496714	0.005847	0.300578	0.015225
6	CAPSAD	0.000152		0.00211		0.003968		0.028354		0.392405		0.738095		0.392405	
7	Consultório na Rua	0.00015	0.000001	0.001706	0	0.002688	0.000021	0.027822	0.000021	0.317406	0	0.500014	0.003802	0.317406	0
8	CRAS/CREAS	0.00015	0	0.001657	0.00016	0.002737	0.000115	0.027904	0.000023	0.308277	0.029747	0.509092	0.021487	0.308277	0.029747
9	Entidades Assistênciais e Dependencia Química e CT	0.000141	0.000027	0.001457	0.000387	0.002374	0.000352	0.0262	0.005061	0.271052	0.071899	0.441565	0.065525	0.271052	0.071899
10	Entidades Socioassistenciais	0.000148	0.000001	0.001585	0.000127	0.002459	0.000219	0.027608	0.000272	0.294785	0.023581	0.457414	0.040736	0.294785	0.023581

Showing 1 to 10 of 13 entries

5 Saving objects with new variables and changes

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