

# SNA Closeness 27\_ATIVIDADE

*Leonardo Martins*

*17 de julho de 2016*

## Contents

<b>1</b>	<b>27_ATIVIDADE</b>	<b>2</b>
<b>2</b>	<b>Loading objects generated with previous script</b>	<b>2</b>
2.1	Reload packages . . . . .	2
2.2	Adding phantom tools . . . . .	2
2.3	Setting a random seed - this is a good strategy to keep the same graph pattern layout in a new report generation . . . . .	2
2.4	Simplify Graph - removing loops and duple edges . . . . .	3
<b>3</b>	<b>Closeness - centrality based on distance to others in the graph</b>	<b>3</b>
3.1	Closeness Non-normalized . . . . .	3
3.2	Network Plotting Based On Non-normalized Closeness - IN . . . . .	4
3.3	Network Plotting Based On Non-normalized Closeness - OUT . . . . .	7
3.4	Network Plotting Based On Non-normalized Closeness - ALL . . . . .	10
3.5	Closeness Normalized . . . . .	12
3.6	Network Plotting Based On Normalized Closeness - IN . . . . .	13
3.7	Network Plotting Based On Normalized Closeness - OUT . . . . .	16
3.8	Network Plotting Based On Normalized Closeness - ALL . . . . .	19
3.9	Closeness Normalized . . . . .	21
3.10	Centralization Closeness . . . . .	22
3.11	Network Plotting Based On Centralization Closeness . . . . .	22
<b>4</b>	<b>Closeness Dinamic Table</b>	<b>24</b>
4.1	Getting Closeness Measures . . . . .	24
4.2	Creating a datagramme of measures . . . . .	25
4.3	General tabel - DT . . . . .	25
4.4	Aggregating data from previous table - mean . . . . .	26
4.5	Aggregating data from previous table - sd . . . . .	27
4.6	Plotting final table with round for Closeness . . . . .	27

SNA Descriptive 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 27\_ATIVIDADE

##### 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/Robjct/2_degree_atividade.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 duple edges

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

#### 3.1.2 Saving to Environment

```
atividade_incloseness<- closeness(atividade, mode = "in", weights = E(atividade)$atividade) %>% round(6)
atividade_outcloseness<- closeness(atividade, mode = "out", weights = E(atividade)$atividade) %>% round(6)
atividade_totalcloseness<- closeness(atividade, mode = "total", weights = E(atividade)$atividade) %>% round(6)
```

#### 3.1.3 Closeness Non-normalized - in

```
summary(atividade_incloseness)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## 0.0000290 0.0001280 0.0001290 0.0001272 0.0001300 0.0001350
```

```
sd(atividade_incloseness)
```

```
## [1] 1.070926e-05
```

### 3.2 Network Plotting Based On Non-normalized Closeness - IN

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

#Get Variable
V(atividade)$atividade_color_degree<-round(V(atividade)$incloseness,4)

#Creating brewer palette
vertex_atividade_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
    V(atividade)$atividade_color_degree)), "RdBu"))(
    length(unique(V(atividade)$atividade_color_degree)))

#Saving as Vertex properties
V(atividade)$vertex_atividade_color_degree<-
  vertex_atividade_color_degree[as.numeric(
    cut(V(atividade)$atividade_color_degree,
      breaks=length(unique(V(atividade)$atividade_color_degree))))]

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

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

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

```

    )
axis(1)
axis(2)

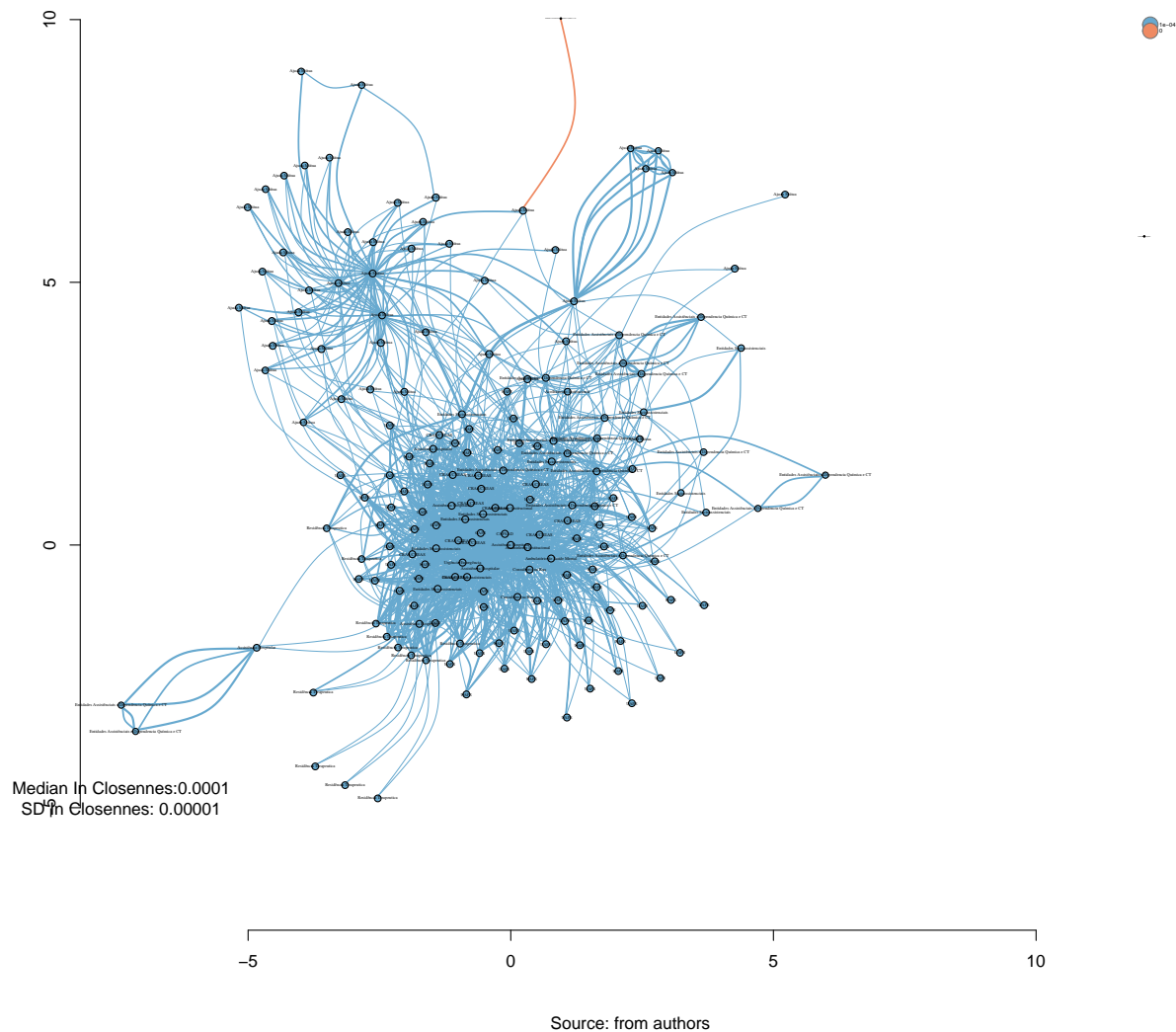
#Solving Problems with legend rendering
a<-V(atividade)$atividade_color_degree
b<-V(atividade)$vertex_atividade_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#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 - 27_ATIVIDADE", sub = "Source: from authors ")
text(
  x=range(co[,1])[1],
  y=range(co[,2])[1],
  labels = sprintf(
    "Median In Closennes:%.4f\nSD In Closennes: %.5f",
    median(closeness(atividade, mode="in", weights = E(atividade)$atividade)),
    sd(closeness(atividade, mode="in", weights = E(atividade)$atividade))
  )
)

```

### Network Closeness Degree Sized and Colored In – 27\_ATIVIDADE



#### 3.2.1 Closeness Non-normalized - OUT

```
summary(atividade_outcloseness)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## 0.0000290 0.0005300 0.0007820 0.0006511 0.0008990 0.0011040
```

```
sd(atividade_outcloseness)
```

```
## [1] 0.0003428904
```

### 3.3 Network Plotting Based On Non-normalized Closeness - OUT

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

#Get Variable
V(atividade)$atividade_color_degree<-round(V(atividade)$outcloseness,4)

#Creating brewer palette
vertex_atividade_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
    V(atividade)$atividade_color_degree)), "RdBu"))(
    length(unique(V(atividade)$atividade_color_degree)))

#Saving as Vertex properties
V(atividade)$vertex_atividade_color_degree<-
  vertex_atividade_color_degree[as.numeric(
    cut(V(atividade)$atividade_color_degree,
      breaks=length(unique(V(atividade)$atividade_color_degree))))]

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

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

#Plotting
plot(atividade,
  layout=co,
  #edge.color=V(atividade)$vertex_atividade_color_degree[edge.start],
  edge.arrow.size=closeness(atividade, weights = E(atividade)$atividade, mode="out"),
  edge.width=E(atividade)$weight/2*mean(E(atividade)$weight),
  edge.curved = TRUE,
  vertex.color=V(atividade)$vertex_atividade_color_degree,
  vertex.size=closeness(atividade, weights = E(atividade)$atividade, mode="out")*10^4,
  vertex.frame.color="white",
  vertex.label.color="black",
  vertex.label=get.vertex.attribute(atividade,"LABEL_COR"),
  vertex.label.cex=closeness(atividade, weights = E(atividade)$atividade, 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(atividade)$atividade_color_degree
b<-V(atividade)$vertex_atividade_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)

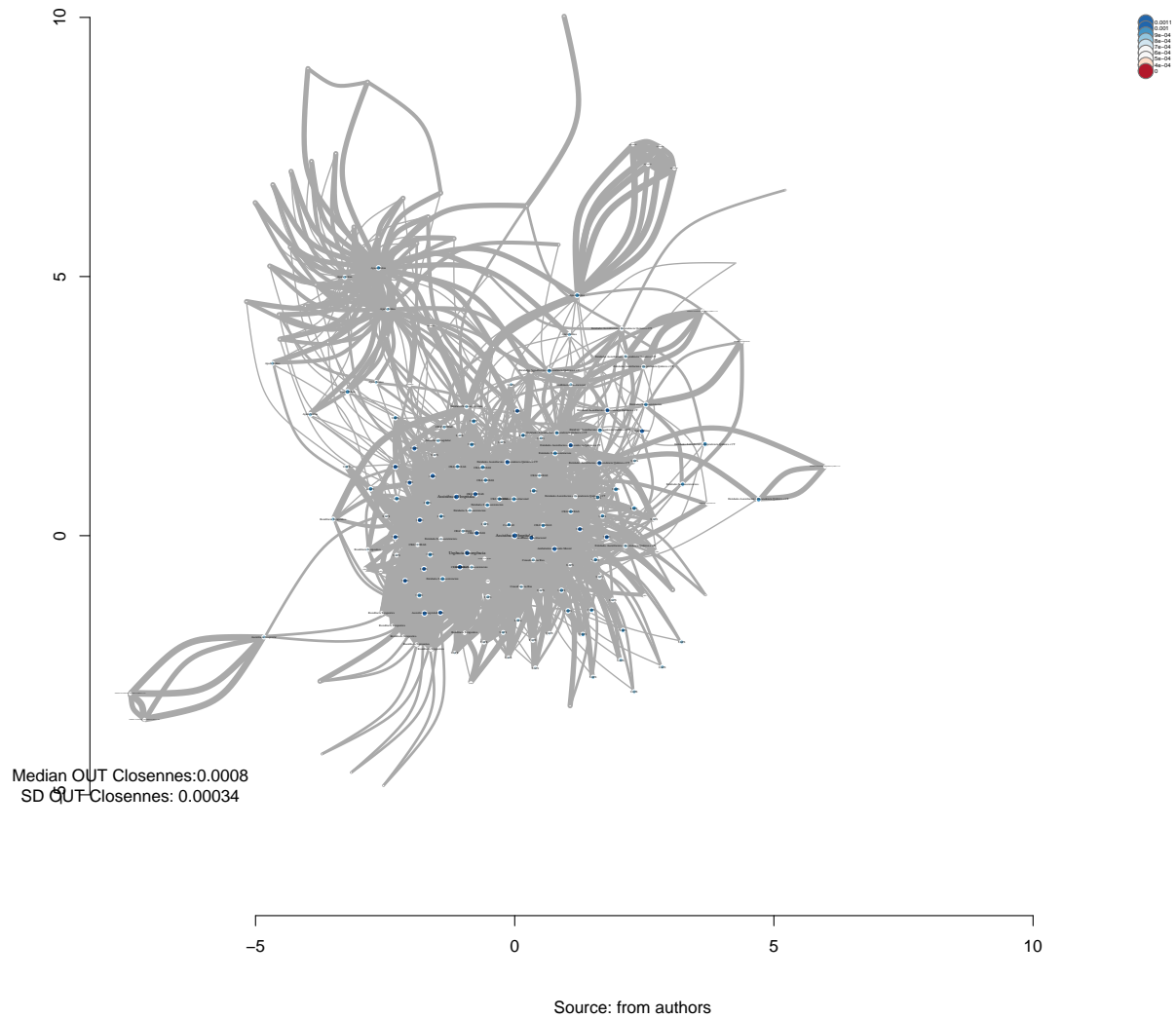
#Adding Legend
legend(x=range(co[,1])[2],
      y=range(co[,2])[2],
      legend=as.character(f),
      pch=21,
      col = "#777777",
      pt.bg=as.character(g),
      pt.cex=2,
      bty="n",
      ncol=1,
      lty=1,
      cex = .3)

#Adding Title
title("Network Closeness Degree Sized and Colored OUT - 27_ATIVIDADE", sub = "Source: from authors ")
text(
  x=range(co[,1])[1],
  y=range(co[,2])[1],
  labels = sprintf(
    "Median OUT Closennes: %.4f\nSD OUT Closennes: %.5f",
    median(closeness(atividade, mode="out", weights = E(atividade)$atividade)),
    sd(closeness(atividade, mode="out", weights = E(atividade)$atividade))
  )
)

```



### Network Closeness Degree Sized and Colored OUT – 27\_ATIVIDADE



#### 3.3.1 Closeness Non-normalized - ALL

```
summary(atividade_totalcloseness)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.000029 0.001093 0.001225 0.001192 0.001342 0.001650
```

```
sd(atividade_totalcloseness)
```

```
## [1] 0.0002215581
```

### 3.4 Network Plotting Based On Non-normalized Closeness - ALL

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

#Get Variable
V(atividade)$atividade_color_degree<-round(V(atividade)$allcloseness,4)

#Creating brewer palette
vertex_atividade_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
    V(atividade)$atividade_color_degree)), "RdBu"))(
    length(unique(V(atividade)$atividade_color_degree)))

#Saving as Vertex properties
V(atividade)$vertex_atividade_color_degree<-
  vertex_atividade_color_degree[as.numeric(
    cut(V(atividade)$atividade_color_degree,
      breaks=length(unique(V(atividade)$atividade_color_degree))))]

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

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

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

```

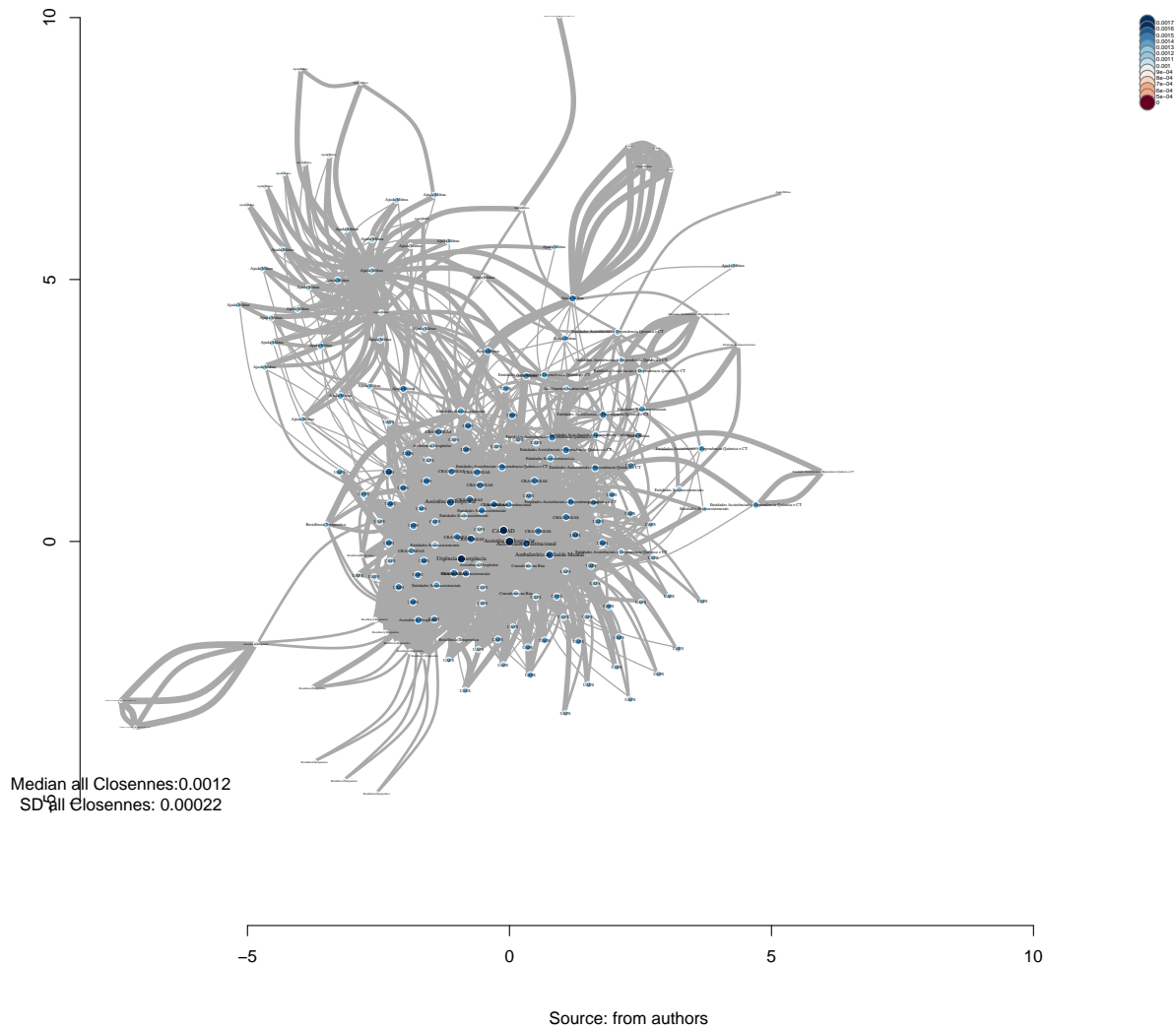
a<-V(atividade)$atividade_color_degree
b<-V(atividade)$vertex_atividade_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#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 - 27_ATIVIDADE", sub = "Source: from authors ")
text(
  x=range(co[,1])[1],
  y=range(co[,2])[1],
  labels = sprintf(
    "Median all Closennes: %.4f\nSD all Closennes: %.5f",
    median(closeness(atividade, mode="all", weights = E(atividade)$atividade)),
    sd(closeness(atividade, mode="all", weights = E(atividade)$atividade))
  )
)

```

## Network Closeness Degree Sized and Colored all – 27\_ATIVIDADE



### 3.5 Closeness Normalized

#### 3.5.1 Saving to Igraph object

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

### 3.5.2 Saving to Environment

```
atividade_incloseness_n<- closeness(atividade, mode = "in", normalized = T, weights = E(atividade)$ativ  
atividade_outcloseness_n<- closeness(atividade, mode = "out", normalized = T, weights = E(atividade)$at  
atividade_totalcloseness_n<- closeness(atividade, mode = "total", normalized = T, weights = E(atividade)
```

### 3.5.3 Closeness Normalized - IN

```
summary(atividade_incloseness_n)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
## 0.005348 0.023760 0.023950 0.023650 0.024160 0.025080
```

```
sd(atividade_incloseness_n)
```

```
## [1] 0.001994514
```

## 3.6 Network Plotting Based On Normalized Closeness - IN

```
V(atividade)$incloseness_n<-closeness(atividade, weights = E(atividade)$atividade, mode="in", normalized  
  
#Get Variable  
V(atividade)$atividade_color_degree<-round(V(atividade)$incloseness_n,4)  
  
#Creating brewer palette  
vertex_atividade_color_degree<-  
  colorRampPalette(brewer.pal(length(unique(  
    V(atividade)$atividade_color_degree)), "RdBu"))(  
    length(unique(V(atividade)$atividade_color_degree)))  
  
#Saving as Vertex properties  
V(atividade)$vertex_atividade_color_degree<-  
  vertex_atividade_color_degree[as.numeric(  
    cut(V(atividade)$atividade_color_degree,  
      breaks=length(unique(V(atividade)$atividade_color_degree))))]  
  
set.seed(123)  
#Plotting based only on degree measures  
edge.start <- ends(atividade, es=E(atividade), names=F)[,1]  
  
# Fixing ego  
minC <- rep(-Inf, vcount(atividade))  
maxC <- rep(Inf, vcount(atividade))  
minC[1] <- maxC[1] <- 0  
co <- layout_with_fr(atividade, niter=10^4, minx=minC, maxx=maxC,miny=minC, maxy=maxC, weights = E(ativ
```

```

#Plotting
plot(atividade,
     layout=co,
     #edge.color=V(atividade)$vertex_atividade_color_degree[edge.start],
     edge.arrow.size=closeness(atividade, weights = E(atividade)$atividade, mode="in",normalized = T),
     edge.width=E(atividade)$weight/10*mean(E(atividade)$weight),
     edge.curved = TRUE,
     vertex.color=V(atividade)$vertex_atividade_color_degree,
     vertex.size=(closeness(atividade, weights = E(atividade)$atividade, mode="in",normalized = T))*100,
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(atividade,"LABEL_COR"),
     vertex.label.cex=closeness(atividade, weights = E(atividade)$atividade, mode="in",normalized = T)*,
     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(atividade)$atividade_color_degree
b<-V(atividade)$vertex_atividade_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)

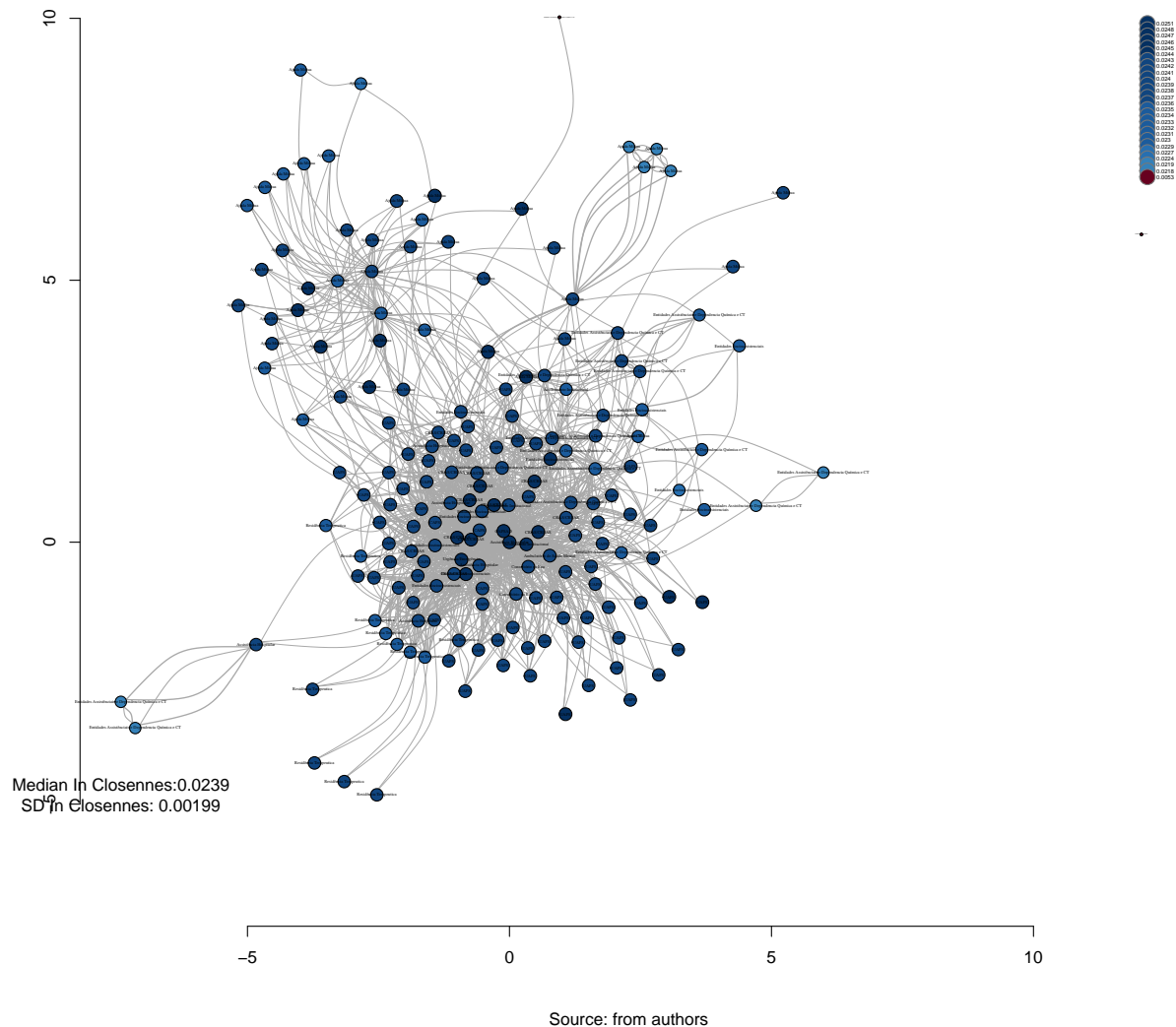
#Adding Legend
legend(x=range(co[,1])[2],
      y=range(co[,2])[2],
      legend=as.character(f),
      pch=21,
      col = "#777777",
      pt.bg=as.character(g),
      pt.cex=2,
      bty="n",
      ncol=1,
      lty=1,
      cex = .3)

#Adding Title
title("Network Closeness Degree Sized Normalized In - 27_ATIVIDADE", sub = "Source: from authors ")
text(
  x=range(co[,1])[1],
  y=range(co[,2])[1],
  labels = sprintf(
    "Median In Closennes: %.4f\nSD In Closennes: %.5f",
    median(closeness(atividade, mode="in", weights = E(atividade)$atividade, normalized = T)),

```

```
sd(closeness(atividade, mode="in", weights = E(atividade)$atividade, normalized = T))
)
```

Network Closeness Degree Sized Normalized In - 27\_ATIVIDADE



### Closeness Normalized - OUT

```
summary(atividade_outcloseness_n)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.005348 0.098540 0.145400 0.121100 0.167300 0.205300
```

```
sd(atividade_outcloseness_n)
```

```
## [1] 0.06379751
```

### 3.7 Network Plotting Based On Normalized Closeness - OUT

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

#Get Variable
V(atividade)$atividade_color_degree<-round(V(atividade)$outcloseness_n,2)

#Creating brewer palette
vertex_atividade_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
    V(atividade)$atividade_color_degree)), "RdBu"))(
    length(unique(V(atividade)$atividade_color_degree)))

#Saving as Vertex properties
V(atividade)$vertex_atividade_color_degree<-
  vertex_atividade_color_degree[as.numeric(
    cut(V(atividade)$atividade_color_degree,
      breaks=length(unique(V(atividade)$atividade_color_degree))))]

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

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

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

#Solving Problems with legend rendering
```



```

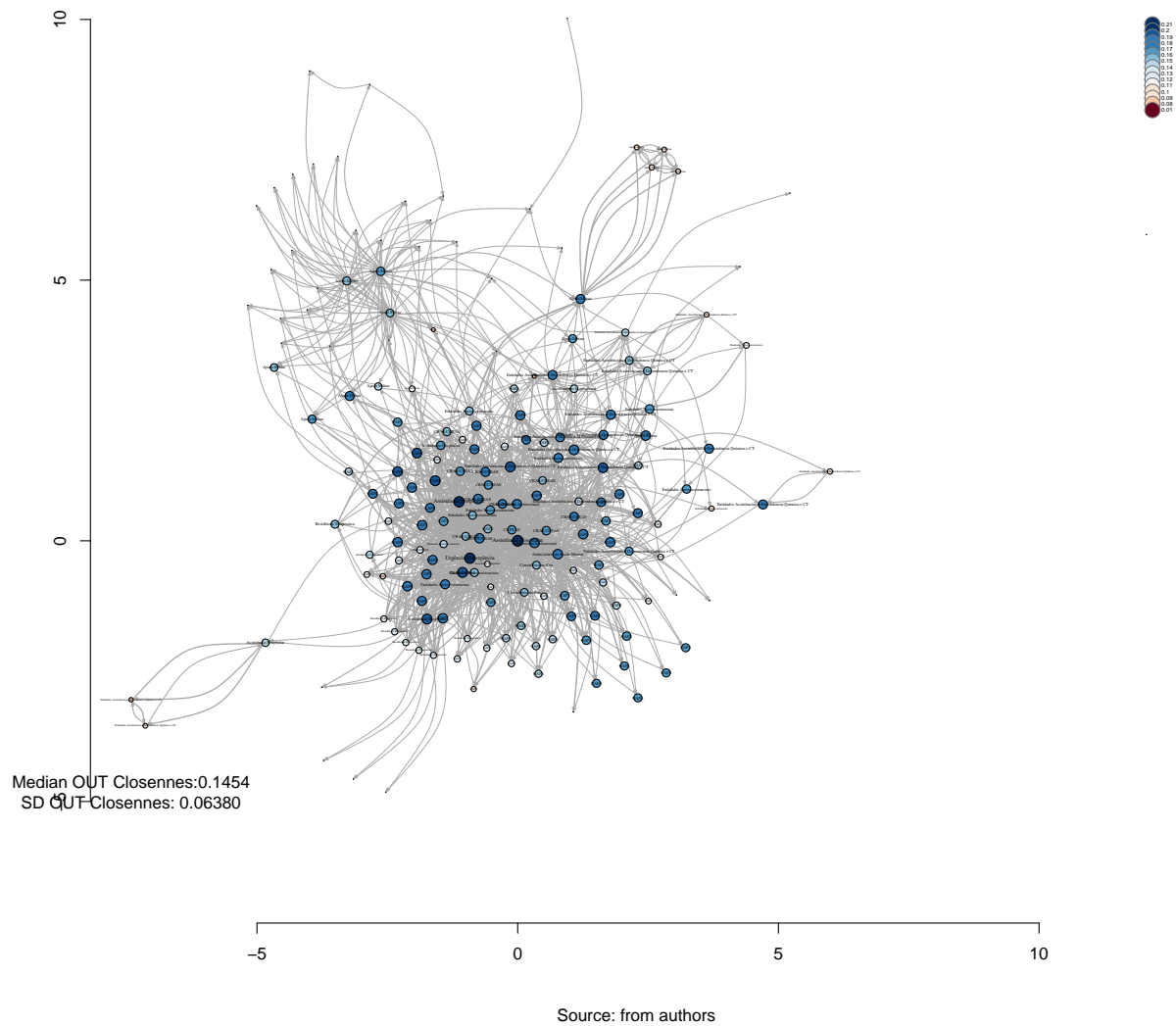
a<-V(atividade)$atividade_color_degree
b<-V(atividade)$vertex_atividade_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#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 - 27_ATIVIDADE", sub = "Source: from authors ")
text(
  x=range(co[,1])[1],
  y=range(co[,2])[1],
  labels = sprintf(
    "Median OUT Closennes: %.4f\nSD OUT Closennes: %.5f",
    median(closeness(atividade, mode="out", weights = E(atividade)$atividade, normalized = T))
    sd(closeness(atividade, mode="out", weights = E(atividade)$atividade, normalized = T))
  )
)

```

### Network Closeness Degree Sized Normalized OUT – 27\_ATIVIDADE



#### 3.7.1 Closeness Normalized - ALL

```
summary(atividade_totalcloseness_n)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.005348 0.203400 0.227900 0.221700 0.249700 0.306900
```

```
sd(atividade_totalcloseness_n)
```

```
## [1] 0.04121422
```

### 3.8 Network Plotting Based On Normalized Closeness - ALL

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

#Get Variable
V(atividade)$atividade_color_degree<-round(V(atividade)$allcloseness_n,2)

#Creating brewer palette
vertex_atividade_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
    V(atividade)$atividade_color_degree)), "RdBu"))(
    length(unique(V(atividade)$atividade_color_degree)))

#Saving as Vertex properties
V(atividade)$vertex_atividade_color_degree<-
  vertex_atividade_color_degree[as.numeric(
    cut(V(atividade)$atividade_color_degree,
      breaks=length(unique(V(atividade)$atividade_color_degree))))]

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

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

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

#Solving Problems with legend rendering
```

```

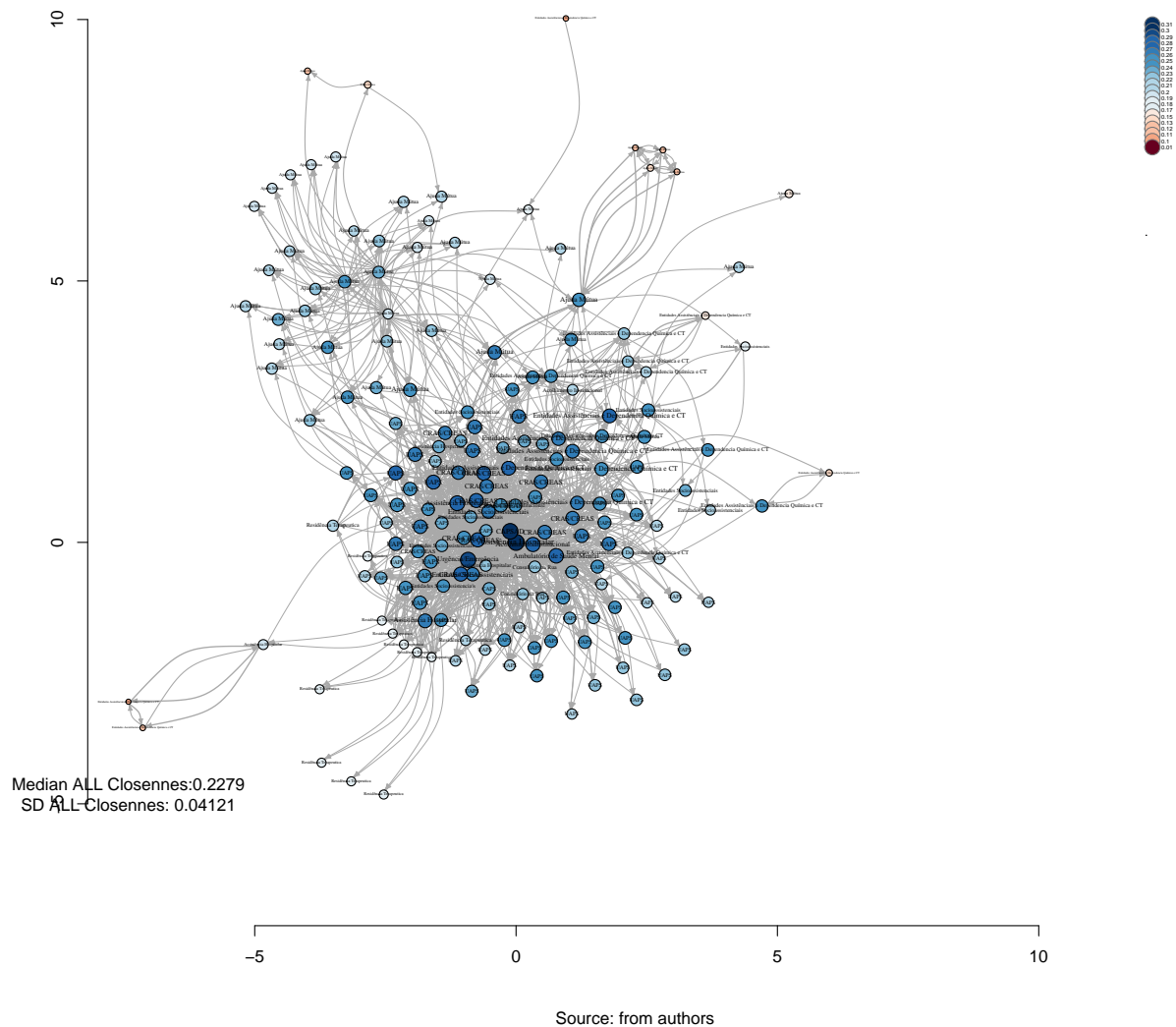
a<-V(atividade)$atividade_color_degree
b<-V(atividade)$vertex_atividade_color_degree
c<-table(a,b)
d<-as.data.frame(c)
e<-subset(d, d$Freq>0)
e<-e[order(e$a,decreasing=T),]
f<-t(e$a)
g<-t(e$b)

#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 - 27_ATIVIDADE", sub = "Source: from authors ")
text(
  x=range(co[,1])[1],
  y=range(co[,2])[1],
  labels = sprintf(
    "Median ALL Closennes: %.4f\nSD ALL Closennes: %.5f",
    median(closeness(atividade, mode="all", weights = E(atividade)$atividade, normalized = T))
    sd(closeness(atividade, mode="all", weights = E(atividade)$atividade, normalized = T))
  )
)

```

## Network Closeness Degree Sized Normalized ALL – 27\_ATIVIDADE



### 3.9 Closeness Normalized

#### 3.9.1 Saving to Igraph object

```
V(atividade)$incloseness_n <- closeness(atividade, weights = E(atividade)$atividade, mode = "in", normalized = TRUE)
V(atividade)$outcloseness_n <- closeness(atividade, weights = E(atividade)$atividade, mode = "out", normalized = TRUE)
V(atividade)$totalcloseness_n <- closeness(atividade, weights = E(atividade)$atividade, mode = "total", normalized = TRUE)
```

### 3.10 Centralization Closeness

```
V(atividade)$atividade_central_closeness<- centralization.closeness(atividade)$res
atividade_central_closeness<- centralization.closeness(atividade)$res
atividade_central_closeness_all<- centralization.closeness(atividade)
```

#### 3.10.1 Centralization

```
atividade_central_closeness_all$centralization
```

```
## [1] 0.1023336
```

#### 3.10.2 Theoretical Max

```
atividade_central_closeness_all$theoretical_max
```

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

### 3.11 Network Plotting Based On Centralization Closeness

```
V(atividade)$atividade_central_closeness<- centralization.closeness(atividade)$res

#Get Variable
V(atividade)$atividade_color_degree<-round(V(atividade)$atividade_central_closeness,2)

#Creating brewer palette
vertex_atividade_color_degree<-
  colorRampPalette(brewer.pal(length(unique(
    V(atividade)$atividade_color_degree)), "Spectral"))(
    length(unique(V(atividade)$atividade_color_degree)))

#Saving as Vertex properties
V(atividade)$vertex_atividade_color_degree<-
  vertex_atividade_color_degree[as.numeric(
    cut(V(atividade)$atividade_color_degree,
      breaks=length(unique(V(atividade)$atividade_color_degree))))]

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

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

```

#Plotting
plot(atividade,
     layout=co,
     edge.color=V(atividade)$vertex_atividade_color_degree[edge.start],
     edge.arrow.size=centralization.closeness(atividade)$res,
     edge.width=E(atividade)$weight/10*mean(E(atividade)$weight),
     edge.curved = TRUE,
     vertex.color=V(atividade)$vertex_atividade_color_degree,
     vertex.size=centralization.closeness(atividade)$res*100,
     vertex.frame.color="black",
     vertex.label.color="black",
     vertex.label=get.vertex.attribute(atividade,"LABEL_COR"),
     vertex.label.cex=centralization.closeness(atividade)$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(atividade)$atividade_color_degree
b<-V(atividade)$vertex_atividade_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)

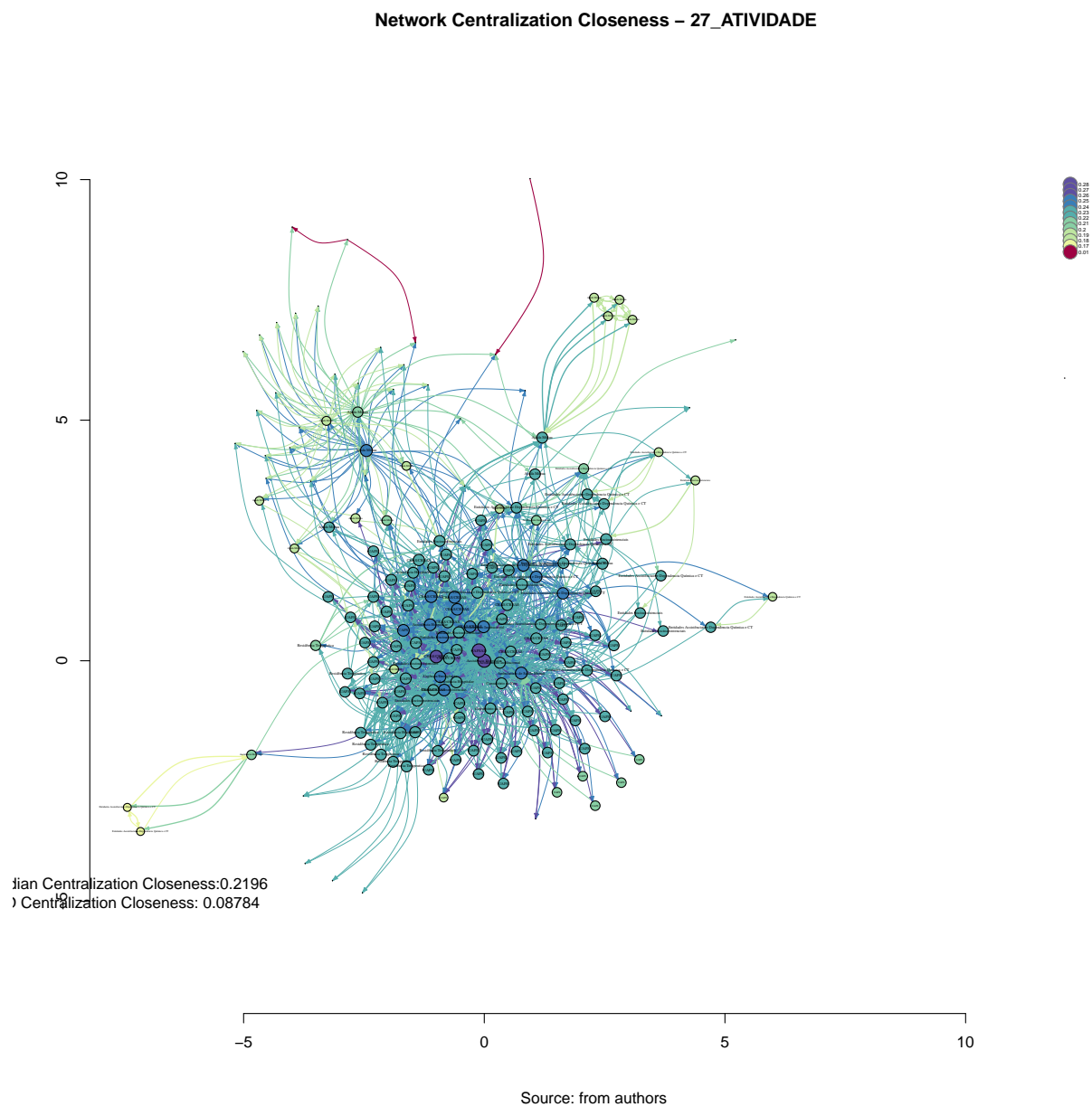
#Adding Title
title("Network Centralization Closeness - 27_ATIVIDADE", sub = "Source: from authors ")
text(
  x=range(co[,1])[1],
  y=range(co[,2])[1],
  labels = sprintf(
    "Median Centralization Closeness:%.4f\nSD Centralization Closeness: %.5f",

```

```

median(centralization.closeness(atividade)$res),
sd(centralization.closeness(atividade)$res)
)
)

```



## 4 Closeness Dinamic Table

### 4.1 Getting Closeness Measures



```

atividade_incloseness<- closeness(atividade, weights = E(atividade)$atividade, mode = "in") %>% round(6)
atividade_outcloseness<- closeness(atividade, weights = E(atividade)$atividade, mode = "out") %>% round(6)
atividade_totalcloseness<- closeness(atividade, weights = E(atividade)$atividade, mode = "total") %>% round(6)
atividade_incloseness_n<- closeness(atividade, weights = E(atividade)$atividade, mode = "in", normalized = TRUE) %>% round(6)
atividade_outcloseness_n<- closeness(atividade, weights = E(atividade)$atividade, mode = "out", normalized = TRUE) %>% round(6)
atividade_totalcloseness_n<- closeness(atividade, weights = E(atividade)$atividade, mode = "total", normalized = TRUE) %>% round(6)
atividade_central_closeness <- centralization.closeness(atividade)$res %>% round(6)

```

## 4.2 Creating a datagramme of measures

```

atividade_df_closeness <- data.frame(
  atividade_incloseness,
  atividade_outcloseness,
  atividade_totalcloseness,
  atividade_incloseness_n,
  atividade_outcloseness_n,
  atividade_totalcloseness_n,
  atividade_central_closeness) %>% round(6)

#Adding type
atividade_df_closeness <- cbind(atividade_df_closeness, V(atividade)$LABEL_COR)

#Adding names
names(atividade_df_closeness) <- c("In Closeness", "Out Closeness", "Total Closeness", "In Closeness Normalized", "Out Closeness Normalized", "Total Closeness Normalized", "Central Closeness")

#Ordering Variables
atividade_df_closeness<-atividade_df_closeness[c("Type", "In Closeness", "Out Closeness", "Total Closeness", "In Closeness Normalized", "Out Closeness Normalized", "Total Closeness Normalized", "Central Closeness")]

```

## 4.3 General tabel - DT

```

datatable(atividade_df_closeness, filter = 'top')

```

Show 10 entries

Search:

	Type	In Closeness	Out Closeness	Total Closeness	In Closeness Normalized	Out Closeness Normalized	Total Closeness Normalized	Centralization Closeness
	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>
ASS_HOS_ Hospital de Pronto Socorro – HPS	Assistência Hospitalar	0.000132	0.001104	0.00165	0.024606	0.205298	0.306931	0.271137
AMB_SAM_ Centro de Atenção à Saúde Mental (CASM)	Ambulatório de Saúde Mental	0.000131	0.000989	0.001493	0.02432	0.183976	0.277612	0.24767
CAPS_AD	CAPSAD	0.000134	0.000864	0.001605	0.02484	0.160622	0.298555	0.277198
CRAS_AS_ CRAS Sudeste Costa Carvalho	CRAS/CREAS	0.000131	0.000949	0.001435	0.024346	0.176471	0.266858	0.248996
CRE_SOC_ CREAS Infância e Juventude	CRAS/CREAS	0.000131	0.000969	0.001451	0.024387	0.180233	0.269956	0.228501
CRE_SOC_ CREAS Norte	CRAS/CREAS	0.000131	0.000892	0.001379	0.024342	0.165923	0.256552	0.227384
ASS_HOS_ Serviço de Controle e Prevenção e Tratamento do Tabagismo (SECOPTT)	Assistência Hospitalar	0.000128	0.000579	0.001167	0.02388	0.107764	0.217036	0.231631
EA_DQCT_ Centro de Recuperação Resgatando Vidas (Escritório)	Entidades Assistenciais e Dependencia Química e CT	0.00013	0.000899	0.001374	0.024175	0.167266	0.255495	0.24031
EA_DQCT_ Comunidade Terapêutica Geração de Adoradores – CTGA	Entidades Assistenciais e Dependencia Química e CT	0.000123	0.000833	0.001156	0.022895	0.155	0.215029	0.223558
EA_DQCT_ Centro de Recuperação Resgatando Vidas	Entidades Assistenciais e Dependencia Química e CT	0.000129	0.000943	0.001332	0.023923	0.175472	0.24767	0.223827

Showing 1 to 10 of 187 entries

Previous  2 3 4 5 ... 19 Next

#### 4.4 Aggregating data from previous table - mean

```
aggdata_mean <- aggregate(atividade_df_closeness, by=list(atividade_df_closeness$Type), FUN=mean, na.rm=TRUE)
names(aggdata_mean) <- c("Group", "Type", "In Closeness(M)", "Out Closeness(M)", "Total Closeness(M)", "In Closeness Normalized", "Out Closeness Normalized", "Total Closeness Normalized", "Centralization Closeness")
#Removing Type variable
aggdata_mean<-aggdata_mean[,-c(2)]
```

## 4.5 Aggregating data from previous table - sd

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

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

#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 toghther

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

## 4.6 Plotting final table with round for Closeness

```
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)
1	Acolhimento Institucional	0.000129	0.000004	0.000898	0.000107	0.00129	0.000179	0.023951	0.00069	0.167018	0.019921	0.239881	0.033263	0.224164	0.022102
2	Ajuda Mútua	0.000128	0.000004	0.000281	0.000352	0.001086	0.000207	0.023734	0.00076	0.052217	0.065503	0.201982	0.038488	0.077408	0.095968
3	Ambulatório de Saúde Mental	0.000131		0.000989		0.001493		0.02432		0.183976		0.277612		0.24767	
4	Assistência Hospitalar	0.000115	0.000038	0.000773	0.000376	0.001151	0.000539	0.021451	0.007106	0.143723	0.069961	0.214073	0.100268	0.200684	0.088911
5	CAPS	0.000129	0.000001	0.000794	0.000146	0.00124	0.000028	0.024013	0.000182	0.14762	0.027073	0.230568	0.005102	0.228223	0.010423
6	CAPSAD	0.000134		0.000864		0.001605		0.02484		0.160622		0.298555		0.277198	
7	Consultório na Rua	0.00013	0	0.000793	0	0.00118	0.000023	0.024107	0.00002	0.147502	0	0.219384	0.00439	0.224367	0
8	CRAS/CREAS	0.000131	0	0.00088	0.00009	0.001397	0.000041	0.024367	0.000045	0.163617	0.016741	0.259816	0.007564	0.230962	0.018425
9	Entidades Assistenciais e Dependência Química e CT	0.00012	0.000023	0.000768	0.000255	0.001123	0.000321	0.022377	0.004217	0.142953	0.047492	0.208824	0.059672	0.202043	0.05302
10	Entidades Socioassistenciais	0.000128	0.000004	0.000801	0.000131	0.001238	0.000139	0.023809	0.000641	0.149052	0.024286	0.230349	0.025775	0.22336	0.01446

Showing 1 to 10 of 13 entries

Previous 1 2 Next

## 5 Saving objects with new variables and changes

```
save.image("~/SNArRDJF/Robobject/3_closeness_atividade.RData")
```