Análise de Redes Sociais e Text Mining - Tarefa 1

Rafael Gustavo Furlan, Ricardo Squassina Lee

• Explore as rotinas Exemplo Rede.R e Exemplo Rede Two Mode.R . Rode os códigos na plataforma R utilizando como base as tabelas Rede One Mode_Tarefa Aula 1_Paulista T4.xlsx e Rede Two Mode_Tarefa Aula 1_Paulista T4.xlsx. (atenção: não são as mesmas bases trabalhadas em sala).

• Faça pequenas modificações na tabela e veja seus resultados.

• Inclua outras análises em seu código (usando as extensões sna, network ou igraph) e comente os resultados (seja criativo!).

• Compile as saídas dos códigos (conteúdo das variáveis, gráficos, tabelas) em um documento Word (usando o modelo deste documento) e comente seus resultados (principalmente as medidas de centralidade), análises, potenciais implicações gerenciais, etc, conforme discutido em sala na Aula 1.

• Desafio: Baseado na tabela da Rede Two Mode desta tarefa, faça uma análise de agrupamento (cluster analysis) do tipo hierárquico aglomerativo (dendrograma) das pessoas ou dos produtos adquiridos por elas, levando em consideração apenas a estrutura de relações entre elas. Comente como implementou e discuta os resultados, comparando com a rede construída. Utilize a plataforma R e o script de exemplo de uso de Cluster Analysis em R.

Dica: após a seleção dos grupos, desenhe a rede e represente os nós das pessoas (ou produtos) com cores de acordo com o grupo correspondente.

Lendo os arquivos

```
# carregando os arquivos
#Rede_One_Mode <- read_excel("Rede One Mode_Tarefa Aula 1_Paulista T4.xlsx")
#Rede_Two_Mode <- read_excel("Rede Two Mode_Tarefa Aula 1_Paulista T4.xlsx")

Rede_One_Mode <- read.csv(file = "Rede One Mode_Tarefa Aula 1_Paulista
T4.csv",
                          header = TRUE)
Rede_Two_Mode <- read.csv(file = "Rede Two Mode_Tarefa Aula 1_Paulista
T4.csv",
                          encoding = "UTF-8",
                          header = TRUE)
```

Explore as rotinas Exemplo Rede.R e Exemplo Rede Two Mode.R . Rode os códigos na plataforma R utilizando como base as tabelas Rede One Mode_Tarefa Aula 1_Paulista T4.xlsx e Rede Two Mode_Tarefa Aula 1_Paulista T4.xlsx. (atenção: não são as mesmas bases trabalhadas em sala).

```r
# explorando Rede One Mode
Rede_One_Mode
```

```
##    ï..Label A B C D E F G H I J K L M N O P
## 1         A 0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1
## 2         B 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## 3         C 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## 4         D 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
## 5         E 0 0 0 1 0 1 1 1 1 1 1 0 1 0 0 1
## 6         F 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
## 7         G 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 0
## 8         H 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0
## 9         I 0 0 0 0 1 0 1 1 0 0 1 1 1 0 0 0
## 10        J 1 1 0 1 1 0 1 1 0 0 1 0 1 1 0 1
## 11        K 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 0
## 12        L 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
## 13        M 1 0 1 0 1 0 1 0 1 1 1 0 0 0 1 0
## 14        N 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## 15        O 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## 16        P 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
```

```r
grede_one_mode <- Rede_One_Mode[,2:17]
grede_one_mode
```

```
##    A B C D E F G H I J K L M N O P
## 1  0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1
## 2  1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## 3  1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## 4  1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
## 5  0 0 0 1 0 1 1 1 1 1 1 0 1 0 0 1
## 6  0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
## 7  0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 0
## 8  0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0
## 9  0 0 0 0 1 0 1 1 0 0 1 1 1 0 0 0
## 10 1 1 0 1 1 0 1 1 0 0 1 0 1 1 0 1
## 11 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 0
## 12 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
## 13 1 0 1 0 1 0 1 0 1 1 1 0 0 0 1 0
## 14 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## 15 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## 16 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
```

```r
rownames(grede_one_mode) <- Rede_One_Mode[,1]
grede_one_mode
```

```
##   A B C D E F G H I J K L M N O P
## A 0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1
## B 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## C 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## D 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
## E 0 0 0 1 0 1 1 1 1 1 1 0 1 0 0 1
```

```
## F 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
## G 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 0
## H 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0
## I 0 0 0 0 1 0 1 1 0 0 1 1 1 0 0 0
## J 1 1 0 1 1 0 1 1 0 0 1 0 1 1 0 1
## K 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 0
## L 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## M 1 0 1 0 1 0 1 0 1 1 1 0 0 0 1 0
## N 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## O 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## P 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
```

```
print("sna::degree")
```

```
## [1] "sna::degree"
```

```
sna::degree(grede_one_mode,gmode="graph",cmode="indegree")
```

```
##  [1]  9  7  7  7  9  3  6  3  6 10 11  1  8  7  7  7
```

```
print("sna::closeness")
```

```
## [1] "sna::closeness"
```

```
sna::closeness(grede_one_mode,gmode="graph")
```

```
##  [1] 0.6818182 0.6250000 0.6000000 0.6250000 0.7142857 0.5357143 0.6250000
##  [8] 0.5172414 0.6250000 0.7142857 0.7894737 0.3947368 0.6818182 0.6250000
## [15] 0.6000000 0.6250000
```

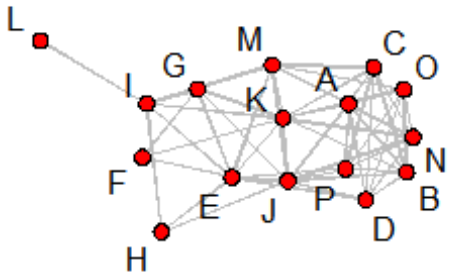
```
print("sna::betweenness")
```

```
## [1] "sna::betweenness"
```

```
sna::betweenness(grede_one_mode,gmode="graph")
```

```
##  [1]  1.953846  1.010989  1.371429  2.079121 13.177289  0.000000  1.733333
##  [8]  0.400000 15.307692 10.481685 18.699634  0.000000  6.323443  1.010989
## [15]  1.371429  2.079121
```

```
gplot(grede_one_mode,gmode="graph",displaylabels =
TRUE,edge.col="gray",usearrows=FALSE)
```




```
#gplot3d(grede_one_mode)

# Explorando Rede Two Mode
Rede_Two_Mode
```

```
##    X.U.FEFF. iPhone iPad Livro.Harry.Potter jogo.MineCraft
## 1       João      1    1                   0              1
## 2      Maria      1    1                   1              1
## 3       José      0    0                   0              0
## 4      Paulo      1    0                   0              0
## 5      Pedro      0    0                   1              0
## 6      Luisa      0    0                   1              0
## 7    Marcelo      1    0                   0              0
## 8    Alfredo      0    1                   1              0
## 9    Joaquim      1    0                   1              0
## 10  Gabriela      0    1                   1              1
## 11    Flávia      1    0                   1              1
## 12 Catapulto      0    1                   0              0
## 13   Rodrigo      0    0                   0              0
## 14   Gabriel      0    1                   1              1
## 15   Rodolfo      1    0                   1              0
##    Camisa.do.Corinthians Bola.de.Futebol Flauta.Transversal
## 1                      1               0                  1
## 2                      1               1                  1
## 3                      1               0                  1
## 4                      1               0                  0
## 5                      1               1                  0
```

```
## 6                     1               1                 0
## 7                     0               1                 1
## 8                     1               1                 1
## 9                     1               1                 1
## 10                    1               1                 0
## 11                    1               0                 0
## 12                    1               1                 1
## 13                    0               0                 0
## 14                    1               1                 1
## 15                    1               1                 1
##    Lista.Telefônica Caixa.de.Fósforos Calculadora Detergente
## 1                 1                 1           1          0
## 2                 1                 1           1          1
## 3                 0                 0           1          0
## 4                 0                 0           1          0
## 5                 1                 1           1          0
## 6                 1                 1           1          0
## 7                 0                 0           1          0
## 8                 0                 0           1          0
## 9                 1                 1           1          0
## 10                1                 1           0          0
## 11                0                 0           0          0
## 12                1                 1           0          0
## 13                1                 1           0          1
## 14                1                 1           0          0
## 15                1                 0           0          0
```

```
grede_two_mode <- Rede_Two_Mode[,2:12]
grede_two_mode
```

```
##    iPhone iPad Livro.Harry.Potter jogo.MineCraft Camisa.do.Corinthians
## 1       1    1                  0              1                     1
## 2       1    1                  1              1                     1
## 3       0    0                  0              0                     1
## 4       1    0                  0              0                     1
## 5       0    0                  1              0                     1
## 6       0    0                  1              0                     1
## 7       1    0                  0              0                     0
## 8       0    1                  1              0                     1
## 9       1    0                  1              0                     1
## 10      0    1                  1              1                     1
## 11      1    0                  1              1                     1
## 12      0    1                  0              0                     1
## 13      0    0                  0              0                     0
## 14      0    1                  1              1                     1
## 15      1    0                  1              0                     1
##    Bola.de.Futebol Flauta.Transversal Lista.Telefônica Caixa.de.Fósforos
## 1                0                  1                1                 1
## 2                1                  1                1                 1
## 3                0                  1                0                 0
```

```
## 4                0                 0                 0                 0
## 5                1                 0                 1                 1
## 6                1                 0                 1                 1
## 7                1                 1                 0                 0
## 8                1                 1                 0                 0
## 9                1                 1                 1                 1
## 10               1                 0                 1                 1
## 11               0                 0                 0                 0
## 12               1                 1                 1                 1
## 13               0                 0                 1                 1
## 14               1                 1                 1                 1
## 15               1                 1                 1                 0
##    Calculadora Detergente
## 1            1          0
## 2            1          1
## 3            1          0
## 4            1          0
## 5            1          0
## 6            1          0
## 7            1          0
## 8            1          0
## 9            1          0
## 10           0          0
## 11           0          0
## 12           0          0
## 13           0          1
## 14           0          0
## 15           0          0
```

```
rownames(grede_two_mode) <- Rede_Two_Mode[,1]

print("sna::degree")
```

```
## [1] "sna::degree"
```

```
sna::degree(grede_two_mode,gmode="twomode",cmode="indegree")
```

```
##  [1]  8 11  3  3  6  6  4  6  8  7  4  6  3  8  6  7  6  9  5 13 10  9 10
## [24]  9  9  2
```

```
print("sna::closeness")
```

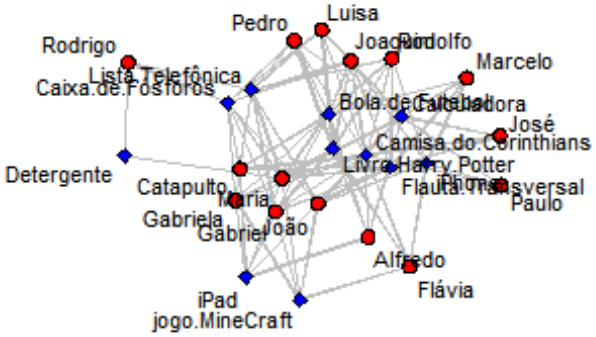
```
## [1] "sna::closeness"
```

```
sna::closeness(grede_two_mode,gmode="twomode")
```

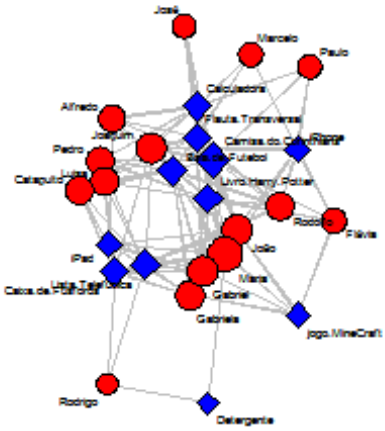
```
##  [1] 0.5555556 0.6410256 0.4385965 0.4385965 0.5102041 0.5102041 0.4545455
##  [8] 0.4901961 0.5555556 0.5319149 0.4545455 0.5102041 0.3846154 0.5555556
## [15] 0.5102041 0.4901961 0.4716981 0.5319149 0.4545455 0.6410256 0.5555556
## [22] 0.5319149 0.5555556 0.5319149 0.5319149 0.4098361
```

```
print("sna::betweenness")
```

```
## [1] "sna::betweenness"

sna::betweenness(grede_two_mode,gmode="twomode")

##  [1] 31.841411 89.083242  2.029950  2.532392 11.714679 11.714679  5.136318
##  [8] 12.241745 25.300294 18.405570  6.296868 12.023228  5.705013 25.175721
## [15] 12.798892 26.387388 10.194285 30.358487  8.844409 89.770501 38.863806
## [22] 36.500161 47.969415 38.313380 41.961685  2.836483

gplot(grede_two_mode,gmode="twomode",displaylabels = TRUE,
      edge.col="gray",label.cex = 0.7,usearrows=FALSE)
```




```
# Aprimorando a representacao da rede
gplot(grede_two_mode,gmode="twomode",displaylabels = TRUE,
      edge.col="gray",label.cex = 0.4,usearrows=FALSE,
      vertex.cex = sna::closeness(grede_two_mode,gmode="twomode")*4)
```




```
#gplot3d(grede_two_mode,gmode = "twomode")
```

Faça pequenas modificações na tabela e veja seus resultados.

```
grede_one_mode_a <- grede_one_mode
grede_one_mode_a$A <- 1
grede_one_mode_a
```

```
##   A B C D E F G H I J K L M N O P
## A 1 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1
## B 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## C 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## D 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
## E 1 0 0 1 0 1 1 1 1 1 1 0 1 0 0 1
## F 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
## G 1 0 0 0 1 1 0 0 1 1 1 0 1 0 0 0
## H 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0
## I 1 0 0 0 1 0 1 1 0 0 1 1 1 0 0 0
## J 1 1 0 1 1 0 1 1 0 0 1 0 1 1 0 1
## K 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 0
## L 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## M 1 0 1 0 1 0 1 0 1 1 1 0 0 0 1 0
## N 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## O 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## P 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
```

```
sna::degree(grede_one_mode_a,gmode="graph",cmode="indegree")
```

```
##  [1] 15  7  7  7  9  3  6  3  6 10 11  1  8  7  7  7
```

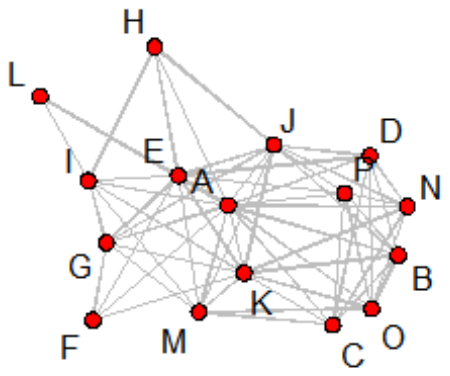
```
sna::closeness(grede_one_mode_a,gmode="graph")
```

```
##  [1] 1.0000000 0.6521739 0.6521739 0.6521739 0.7500000 0.5769231 0.6521739
##  [8] 0.5769231 0.6521739 0.7500000 0.7894737 0.5357143 0.6818182 0.6521739
## [15] 0.6521739 0.6521739
```

```
sna::betweenness(grede_one_mode_a,gmode="graph")
```

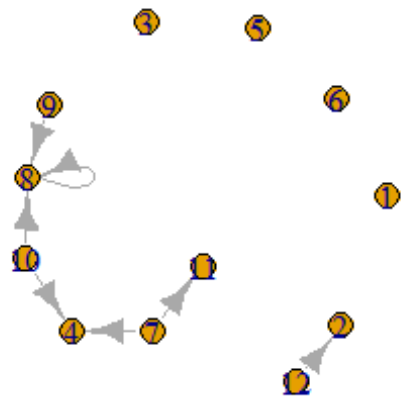
```
##  [1] 12.6519231  0.9340659  1.3714286  1.7252747 10.2112637  0.0000000
##  [7]  1.4916667  0.2833333 10.4455128  8.0432234 13.4855311  0.0000000
## [13]  4.3260073  0.9340659  1.3714286  1.7252747
```

```
gplot(grede_one_mode_a,gmode="graph",displaylabels =
TRUE,edge.col="gray",usearrows=FALSE)
```

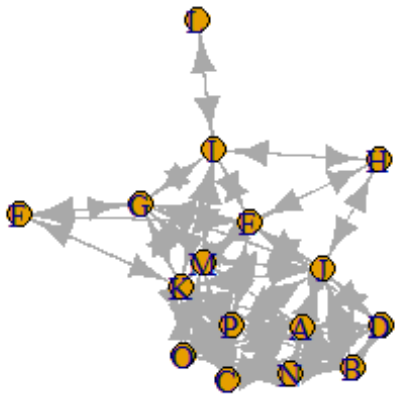


Exercicio de transformação de grede_one_mode, coluna A somente, com ligações criadas aleatóriamente, para observar o comportamnto do grafo.

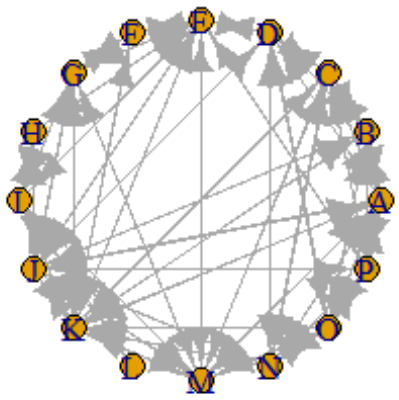
```
grede_two_mode_a <- make_graph(edges = c(grede_one_mode$A--
grede_one_mode$B++grede_one_mode$C--grede_one_mode$D--grede_one_mode$E--
grede_one_mode$F--grede_one_mode$G--grede_one_mode$H--grede_one_mode$I--
grede_one_mode$J--grede_one_mode$K--grede_one_mode$L--grede_one_mode$M--
grede_one_mode$N--grede_one_mode$O--grede_one_mode$P)+1,n = 5)

plot(grede_two_mode_a,gmode = "twomode")
```



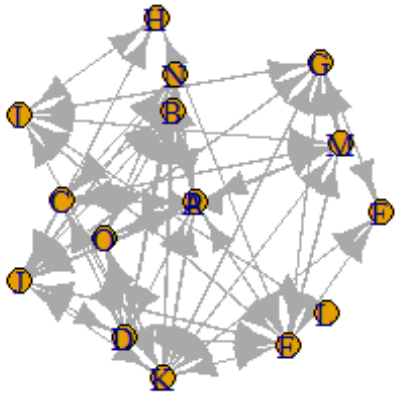

```
########################## igraph - One-Mode
# gera a um objeto graph
g1 <- graph.adjacency(as.matrix(grede_one_mode), weighted=NULL, mode =
"directed")
summary(g1)

## IGRAPH 137e9fb DN-- 16 108 --
## + attr: name (v/c)

layout1 <- layout.fruchterman.reingold(g1)
layout2 <- layout.circle(g1)
layout3 <- layout.sphere(g1)
layout4 <- layout.random(g1)
layout5 <- layout.reingold.tilford(g1)

## Warning in layout_as_tree(structure(list(16, TRUE, c(0, 0, 0, 0, 0, 0, 0,
:
## At structural_properties.c:3338 :graph contains a cycle, partial result is
## returned

layout6 <- layout.kamada.kawai(g1)
layout7 <- layout.lgl(g1)


# plot a gragh using the parameters in the layout
plot(g1, layout=layout1)
```



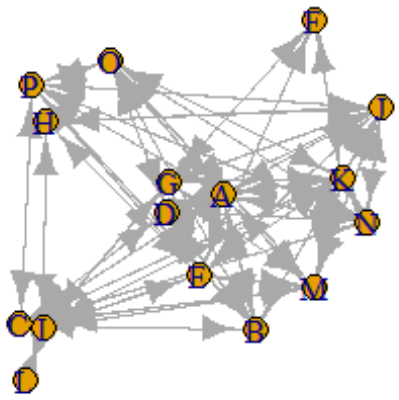
```
plot(g1, layout=layout2)
```



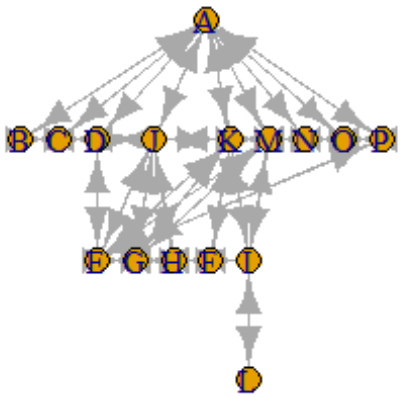
```
plot(g1, layout=layout3)
```



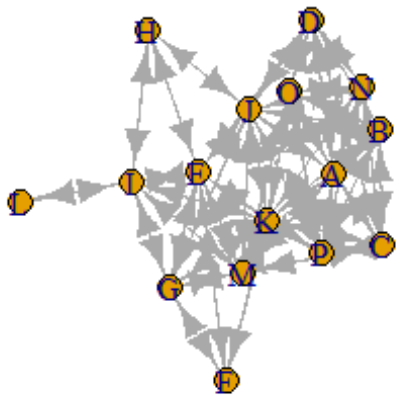
```
plot(g1, layout=layout4)
```



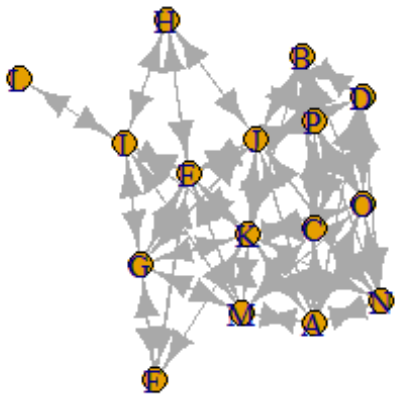
```
plot(g1, layout=layout5)
```



```
plot(g1, layout=layout6)
```



```
plot(g1, layout=layout7)
```



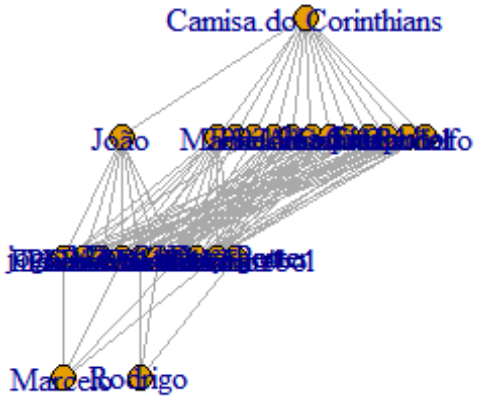
```
########################## igraph - Two-Mode
# gera a um objeto graph
g2 <- graph_from_incidence_matrix(data.matrix(grede_two_mode))
summary(g2)

## IGRAPH 13cf770 UN-B 26 89 --
## + attr: type (v/l), name (v/c)

layout1 <- layout.fruchterman.reingold(g2)
layout2 <- layout.circle(g2)
layout3 <- layout.sphere(g2)
layout4 <- layout.random(g2)
layout5 <- layout.reingold.tilford(g2)
layout6 <- layout.kamada.kawai(g2)
layout7 <- layout.lgl(g2)


# plot a gragh using the parameters in the layout
plot(g2, layout=layout1)
```


```
plot(g2, layout=layout2)
```

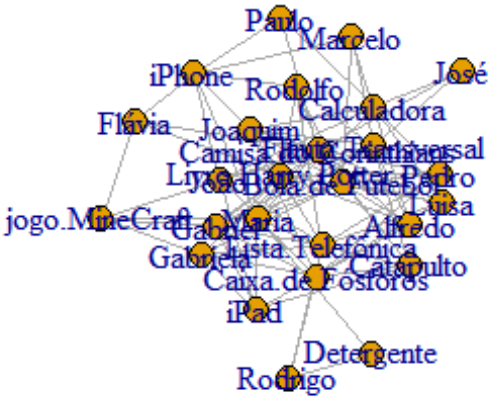

```
plot(g2, layout=layout3)
```


```
plot(g2, layout=layout4)
```


```
plot(g2, layout=layout5)
```




```
plot(g2, layout=layout6)
```




```
plot(g2, layout=layout7)
```

Compile as saídas dos códigos (conteúdo das variáveis, gráficos, tabelas) em um documento Word (usando o modelo deste documento) e comente seus resultados (principalmente as medidas de centralidade), análises, potenciais implicações gerenciais, etc, conforme discutido em sala na Aula 1.

One-Mode

```
# Explorando a rede
sna::degree(grede_one_mode,gmode="graph",cmode="indegree")

##  [1]  9  7  7  7  9  3  6  3  6 10 11  1  8  7  7  7

sna::closeness(grede_one_mode,gmode="graph")

##  [1] 0.6818182 0.6250000 0.6000000 0.6250000 0.7142857 0.5357143 0.6250000
##  [8] 0.5172414 0.6250000 0.7142857 0.7894737 0.3947368 0.6818182 0.6250000
## [15] 0.6000000 0.6250000

sna::betweenness(grede_one_mode,gmode="graph")

##  [1]  1.953846  1.010989  1.371429  2.079121 13.177289  0.000000  1.733333
##  [8]  0.400000 15.307692 10.481685 18.699634  0.000000  6.323443  1.010989
## [15]  1.371429  2.079121

sna::bicomponent.dist(grede_one_mode) # retorna os bicomponentes de um
gráfico de entrada, juntamente com a distribuição de tamanho e as informações
de associação.
```

```
## $members
## $members$`1`
##  [1]  1  2  3  4  5  6  7  8  9 10 11 13 14 15 16
##
##
## $membership
##  [1]  1  1  1  1  1  1  1  1  1  1  1 NA  1  1  1  1
##
## $csize
##  1
## 15
##
## $cdist
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
```

```
sna::bicomponent.dist(grede_one_mode, symmetrize = c("strong", "weak"))
```

```
## $members
## $members$`1`
##  [1]  1  2  3  4  5  6  7  8  9 10 11 13 14 15 16
##
##
## $membership
##  [1]  1  1  1  1  1  1  1  1  1  1  1 NA  1  1  1  1
##
## $csize
##  1
## 15
##
## $cdist
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
```

```
sna::components(grede_one_mode,connected="weak")
```

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

```
sna::components(grede_one_mode,connected="strong")
```

```
## Node 1, Reach 16, Total 16
## Node 2, Reach 16, Total 32
## Node 3, Reach 16, Total 48
## Node 4, Reach 16, Total 64
## Node 5, Reach 16, Total 80
## Node 6, Reach 16, Total 96
## Node 7, Reach 16, Total 112
## Node 8, Reach 16, Total 128
## Node 9, Reach 16, Total 144
## Node 10, Reach 16, Total 160
## Node 11, Reach 16, Total 176
```

```
## Node 12, Reach 16, Total 192
## Node 13, Reach 16, Total 208
## Node 14, Reach 16, Total 224
## Node 15, Reach 16, Total 240
## Node 16, Reach 16, Total 256

## [1] 1
```

```
sna::cug.test(grede_one_mode,gtrans,cmode="size")
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: size
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0.5635359
## Pr(X>=Obs): 0.028
## Pr(X<=Obs): 0.972
```

```
sna::cug.test(grede_one_mode,gtrans,cmode="edges")
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: edges
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0.5635359
## Pr(X>=Obs): 0
## Pr(X<=Obs): 1
```

```
sna::cug.test(grede_one_mode,gtrans,cmode="dyad.census")
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: dyad.census
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0.5635359
## Pr(X>=Obs): 0.001
## Pr(X<=Obs): 0.999
```

```
sna::diag.remove(grede_one_mode)
```

```
##    A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P
## A NA  1  1  1  0  0  0  0  0  1  1  0  1  1  1  1
## B  1 NA  1  1  0  0  0  0  0  1  1  0  0  0  1  1
## C  1  1 NA  1  0  0  0  0  0  0  1  0  1  1  0  1
## D  1  1  1 NA  1  0  0  0  0  1  0  0  0  1  1  0
## E  0  0  0  1 NA  1  1  1  1  1  1  0  1  0  0  1
## F  0  0  0  0  1 NA  1  0  0  0  1  0  0  0  0  0
## G  0  0  0  0  1  1 NA  0  1  1  1  0  1  0  0  0
## H  0  0  0  0  1  0  0 NA  1  1  0  0  0  0  0  0
## I  0  0  0  0  1  0  1  1 NA  0  1  1  1  0  0  0
## J  1  1  0  1  1  0  1  1  0 NA  1  0  1  1  0  1
## K  1  1  1  0  1  1  1  0  1  1 NA  0  1  1  1  0
## L  0  0  0  0  0  0  0  0  1  0  0 NA  0  0  0  0
## M  1  0  1  0  1  0  1  0  1  1  1  0 NA  0  1  0
## N  1  0  1  1  0  0  0  0  0  1  1  0  0 NA  1  1
## O  1  1  0  1  0  0  0  0  0  0  1  0  1  1 NA  1
## P  1  1  1  0  1  0  0  0  0  1  0  0  0  1  1 NA
```

```
sna::efficiency(grede_one_mode)
```

```
## [1] 0.5866667
```

```
sna::gden(grede_one_mode)
```

```
## [1] 0.45
```

```
sna::grecip(grede_one_mode)
```

```
## Mut
##   1
```

```
sna::gt(grede_one_mode)
```

```
##   A B C D E F G H I J K L M N O P
## A 0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1
## B 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## C 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## D 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
## E 0 0 0 1 0 1 1 1 1 1 1 0 1 0 0 1
## F 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
## G 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 0
## H 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0
## I 0 0 0 0 1 0 1 1 0 0 1 1 1 0 0 0
## J 1 1 0 1 1 0 1 1 0 0 1 0 1 1 0 1
## K 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 0
## L 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## M 1 0 1 0 1 0 1 0 1 1 1 0 0 0 1 0
## N 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1
## O 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1
## P 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 0
```

```
sna::gtrans(grede_one_mode)
```

```
## [1] 0.5635359

#sna::gvectorize(grede_one_mode)
sna::infocent(grede_one_mode)

##  [1] 2.9635895 2.7071595 2.7143948 2.7239353 3.0547850 1.8037716 2.5818548
##  [8] 1.8213167 2.4989845 3.1584245 3.2659082 0.7842137 2.9525181 2.7071595
## [15] 2.7143948 2.7239353
```

Two-Mode

```
# Explorando a rede
sna::degree(grede_two_mode,gmode="graph",cmode="indegree")

##  [1]  8 11  3  3  6  6  4  6  8  7  4  6  3  8  6  7  6  9  5 13 10  9 10
## [24]  9  9  2

sna::closeness(grede_two_mode,gmode="graph")

##  [1] 0.5555556 0.6410256 0.4385965 0.4385965 0.5102041 0.5102041 0.4545455
##  [8] 0.4901961 0.5555556 0.5319149 0.4545455 0.5102041 0.3846154 0.5555556
## [15] 0.5102041 0.4901961 0.4716981 0.5319149 0.4545455 0.6410256 0.5555556
## [22] 0.5319149 0.5555556 0.5319149 0.5319149 0.4098361

sna::betweenness(grede_two_mode,gmode="graph")

##  [1] 15.920705 44.541621  1.014975  1.266196  5.857340  5.857340  2.568159
##  [8]  6.120872 12.650147  9.202785  3.148434  6.011614  2.852506 12.587860
## [15]  6.399446 13.193694  5.097142 15.179244  4.422205 44.885250 19.431903
## [22] 18.250080 23.984708 19.156690 20.980843  1.418242

sna::bicomponent.dist(grede_two_mode) # retorna os bicomponentes de um
gráfico de entrada, juntamente com a distribuição de tamanho e as informações
de associação.

## $members
## $members$`1`
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26
##
##
## $membership
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## $csize
##  1
## 26
##
## $cdist
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
## 26
##  1
```

```
sna::bicomponent.dist(grede_two_mode, symmetrize = c("strong", "weak"))
```

```
## $members
## $members$`1`
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26
##
##
## $membership
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## $csize
##  1
## 26
##
## $cdist
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## 26
##  1
```

```
sna::components(grede_two_mode,connected="weak")
```

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

```
sna::components(grede_two_mode,connected="strong")
```

```
## Node 1, Reach 26, Total 26
## Node 2, Reach 26, Total 52
## Node 3, Reach 26, Total 78
## Node 4, Reach 26, Total 104
## Node 5, Reach 26, Total 130
## Node 6, Reach 26, Total 156
## Node 7, Reach 26, Total 182
## Node 8, Reach 26, Total 208
## Node 9, Reach 26, Total 234
## Node 10, Reach 26, Total 260
## Node 11, Reach 26, Total 286
## Node 12, Reach 26, Total 312
## Node 13, Reach 26, Total 338
## Node 14, Reach 26, Total 364
## Node 15, Reach 26, Total 390
## Node 16, Reach 26, Total 416
## Node 17, Reach 26, Total 442
## Node 18, Reach 26, Total 468
## Node 19, Reach 26, Total 494
## Node 20, Reach 26, Total 520
## Node 21, Reach 26, Total 546
```

```
## Node 22, Reach 26, Total 572
## Node 23, Reach 26, Total 598
## Node 24, Reach 26, Total 624
## Node 25, Reach 26, Total 650
## Node 26, Reach 26, Total 676

## [1] 1
```

```
sna::cug.test(grede_two_mode,gtrans,cmode="size")
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: size
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0
## Pr(X>=Obs): 1
## Pr(X<=Obs): 0
```

```
sna::cug.test(grede_two_mode,gtrans,cmode="edges")
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: edges
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0
## Pr(X>=Obs): 1
## Pr(X<=Obs): 0
```

```
sna::cug.test(grede_two_mode,gtrans,cmode="dyad.census")
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: dyad.census
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0
## Pr(X>=Obs): 1
## Pr(X<=Obs): 0
```

```
sna::diag.remove(grede_two_mode)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]   NA    0    0    0    0    0    0    0    0     0     0     0     0
##  [2,]    0   NA    0    0    0    0    0    0    0     0     0     0     0
##  [3,]    0    0   NA    0    0    0    0    0    0     0     0     0     0
##  [4,]    0    0    0   NA    0    0    0    0    0     0     0     0     0
##  [5,]    0    0    0    0   NA    0    0    0    0     0     0     0     0
##  [6,]    0    0    0    0    0   NA    0    0    0     0     0     0     0
##  [7,]    0    0    0    0    0    0   NA    0    0     0     0     0     0
##  [8,]    0    0    0    0    0    0    0   NA    0     0     0     0     0
##  [9,]    0    0    0    0    0    0    0    0   NA     0     0     0     0
## [10,]    0    0    0    0    0    0    0    0    0    NA     0     0     0
## [11,]    0    0    0    0    0    0    0    0    0     0    NA     0     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0    NA     0
## [13,]    0    0    0    0    0    0    0    0    0     0     0     0    NA
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [16,]    1    1    0    1    0    0    1    0    1     0     1     0     0
## [17,]    1    1    0    0    0    0    0    1    0     1     0     1     0
## [18,]    0    1    0    0    1    1    0    1    1     1     1     0     0
## [19,]    1    1    0    0    0    0    0    0    0     1     1     0     0
## [20,]    1    1    1    1    1    1    0    1    1     1     1     1     0
## [21,]    0    1    0    0    1    1    1    1    1     1     0     1     0
## [22,]    1    1    1    0    0    0    1    1    1     0     0     1     0
## [23,]    1    1    0    0    1    1    0    0    1     1     0     1     1
## [24,]    1    1    0    0    1    1    0    0    1     1     0     1     1
## [25,]    1    1    1    1    1    1    1    1    1     0     0     0     0
## [26,]    0    1    0    0    0    0    0    0    0     0     0     0     1
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
##  [1,]     0     0     1     1     0     1     1     0     1     1     1
##  [2,]     0     0     1     1     1     1     1     1     1     1     1
##  [3,]     0     0     0     0     0     0     1     0     1     0     0
##  [4,]     0     0     1     0     0     0     1     0     0     0     0
##  [5,]     0     0     0     0     1     0     1     1     0     1     1
##  [6,]     0     0     0     0     1     0     1     1     0     1     1
##  [7,]     0     0     1     0     0     0     0     1     1     0     0
##  [8,]     0     0     0     1     1     0     1     1     1     0     0
##  [9,]     0     0     1     0     1     0     1     1     1     1     1
## [10,]     0     0     0     1     1     1     1     1     0     1     1
## [11,]     0     0     1     0     1     1     1     0     0     0     0
## [12,]     0     0     0     1     0     0     1     1     1     1     1
## [13,]     0     0     0     0     0     0     0     0     0     1     1
## [14,]    NA     0     0     1     1     1     1     1     1     1     1
## [15,]     0    NA     1     0     1     0     1     1     1     1     0
## [16,]     0     1    NA     0     0     0     0     0     0     0     0
## [17,]     1     0     0    NA     0     0     0     0     0     0     0
## [18,]     1     1     0     0    NA     0     0     0     0     0     0
## [19,]     1     0     0     0     0    NA     0     0     0     0     0
## [20,]     1     1     0     0     0     0    NA     0     0     0     0
## [21,]     1     1     0     0     0     0     0    NA     0     0     0
## [22,]     1     1     0     0     0     0     0     0    NA     0     0
```

```
## [23,]     1     1     0     0     0     0     0     0     0    NA     0
## [24,]     1     0     0     0     0     0     0     0     0     0    NA
## [25,]     0     0     0     0     0     0     0     0     0     0     0
## [26,]     0     0     0     0     0     0     0     0     0     0     0
##       [,25] [,26]
##  [1,]     1     0
##  [2,]     1     1
##  [3,]     1     0
##  [4,]     1     0
##  [5,]     1     0
##  [6,]     1     0
##  [7,]     1     0
##  [8,]     1     0
##  [9,]     1     0
## [10,]     0     0
## [11,]     0     0
## [12,]     0     0
## [13,]     0     1
## [14,]     0     0
## [15,]     0     0
## [16,]     0     0
## [17,]     0     0
## [18,]     0     0
## [19,]     0     0
## [20,]     0     0
## [21,]     0     0
## [22,]     0     0
## [23,]     0     0
## [24,]     0     0
## [25,]    NA     0
## [26,]     0    NA
```

```
sna::efficiency(grede_two_mode)
```

```
## [1] 0.7552
```

```
sna::gden(grede_two_mode)
```

```
## [1] 0.2738462
```

```
sna::grecip(grede_two_mode)
```

```
## Mut
##   1
```

```
sna::gt(grede_two_mode)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [2,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [3,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [4,]    0    0    0    0    0    0    0    0    0     0     0     0     0
```

```
##  [5,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [6,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [7,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [8,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [9,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [10,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [11,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [12,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [13,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [14,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [15,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [16,]    1    1    0    1    0    0    1    0    1     0     1     0     0
## [17,]    1    1    0    0    0    0    0    1    0     1     0     1     0
## [18,]    0    1    0    0    1    1    0    1    1     1     1     0     0
## [19,]    1    1    0    0    0    0    0    0    0     1     1     0     0
## [20,]    1    1    1    1    1    1    0    1    1     1     1     1     0
## [21,]    0    1    0    0    1    1    1    1    1     1     0     1     0
## [22,]    1    1    1    0    0    0    1    1    1     0     0     1     0
## [23,]    1    1    0    0    1    1    0    0    1     1     0     1     1
## [24,]    1    1    0    0    1    1    0    0    1     1     0     1     1
## [25,]    1    1    1    1    1    1    1    1    1     0     0     0     0
## [26,]    0    1    0    0    0    0    0    0    0     0     0     0     1
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
##  [1,]     0     0     1     1     0     1     1     0     1     1     1
##  [2,]     0     0     1     1     1     1     1     1     1     1     1
##  [3,]     0     0     0     0     0     0     1     0     1     0     0
##  [4,]     0     0     1     0     0     0     1     0     0     0     0
##  [5,]     0     0     0     0     1     0     1     1     0     1     1
##  [6,]     0     0     0     0     1     0     1     1     0     1     1
##  [7,]     0     0     1     0     0     0     0     1     1     0     0
##  [8,]     0     0     0     1     1     0     1     1     1     0     0
##  [9,]     0     0     1     0     1     0     1     1     1     1     1
## [10,]     0     0     0     1     1     1     1     1     0     1     1
## [11,]     0     0     1     0     1     1     1     0     0     0     0
## [12,]     0     0     0     1     0     0     1     1     1     1     1
## [13,]     0     0     0     0     0     0     0     0     0     1     1
## [14,]     0     0     0     1     1     1     1     1     1     1     1
## [15,]     0     0     1     0     1     0     1     1     1     1     0
## [16,]     0     1     0     0     0     0     0     0     0     0     0
## [17,]     1     0     0     0     0     0     0     0     0     0     0
## [18,]     1     1     0     0     0     0     0     0     0     0     0
## [19,]     1     0     0     0     0     0     0     0     0     0     0
## [20,]     1     1     0     0     0     0     0     0     0     0     0
## [21,]     1     1     0     0     0     0     0     0     0     0     0
## [22,]     1     1     0     0     0     0     0     0     0     0     0
## [23,]     1     1     0     0     0     0     0     0     0     0     0
## [24,]     1     0     0     0     0     0     0     0     0     0     0
## [25,]     0     0     0     0     0     0     0     0     0     0     0
## [26,]     0     0     0     0     0     0     0     0     0     0     0
##       [,25] [,26]
```

```
##  [1,]     1     0
##  [2,]     1     1
##  [3,]     1     0
##  [4,]     1     0
##  [5,]     1     0
##  [6,]     1     0
##  [7,]     1     0
##  [8,]     1     0
##  [9,]     1     0
## [10,]     0     0
## [11,]     0     0
## [12,]     0     0
## [13,]     0     1
## [14,]     0     0
## [15,]     0     0
## [16,]     0     0
## [17,]     0     0
## [18,]     0     0
## [19,]     0     0
## [20,]     0     0
## [21,]     0     0
## [22,]     0     0
## [23,]     0     0
## [24,]     0     0
## [25,]     0     0
## [26,]     0     0
```

```
sna::gtrans(grede_two_mode)
```

```
## [1] 0
```

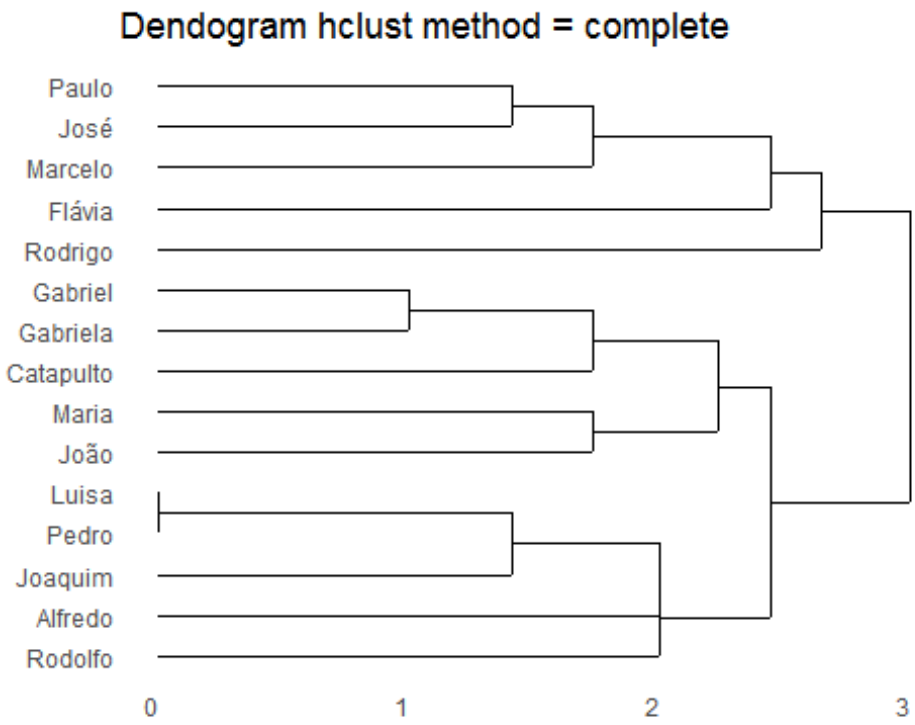
```
#sna::gvectorize(grede_two_mode)
sna::infocent(grede_two_mode)
```

```
##  [1] 3.184192 3.594628 1.891004 1.877498 2.809358 2.809358 2.239728
##  [8] 2.793615 3.195442 2.989489 2.222729 2.796286 1.720651 3.174076
## [15] 2.805486 2.927832 2.763948 3.299262 2.505820 3.761167 3.435145
## [22] 3.289203 3.441913 3.301218 3.269941 1.301833
```

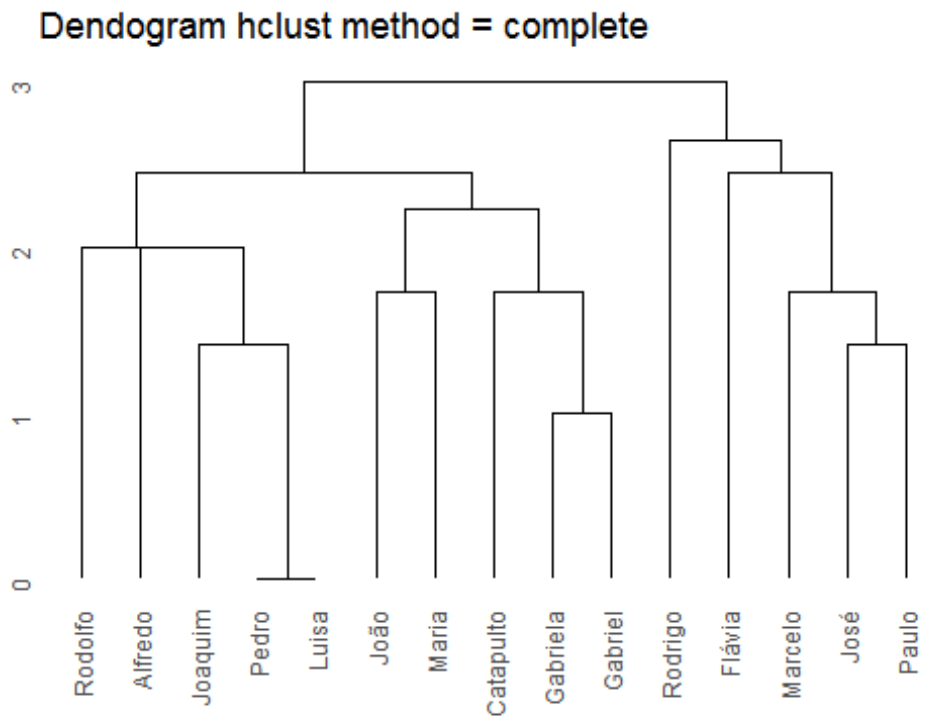
Desafio: Baseado na tabela da Rede Two Mode desta tarefa, faça uma análise de agrupamento (cluster analysis) do tipo hierárquico aglomerativo (dendrograma) das pessoas ou dos produtos adquiridos por elas, levando em consideração apenas a estrutura de relações entre elas. Comente como implementou e discuta os resultados, comparando com a rede construída. Utilize a plataforma R e o script de exemplo de uso de Cluster Analysis em R.

```
# Implementa o algoritmo hierárquico e apresenta o dendrograma
hc <- hclust(dist(grede_two_mode), "complete")  # explorar com outros métodos
de distância
```

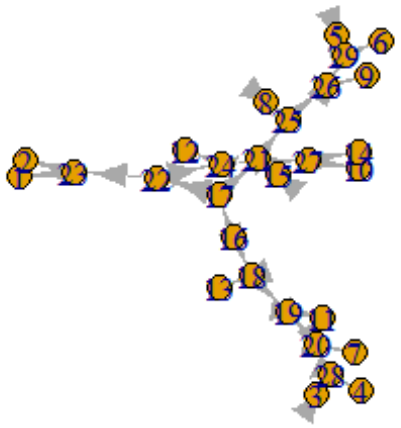
```
ggdendrogram(hc, rotate=TRUE,labels = TRUE)  + labs(title = "Dendogram hclust
method = complete")
```



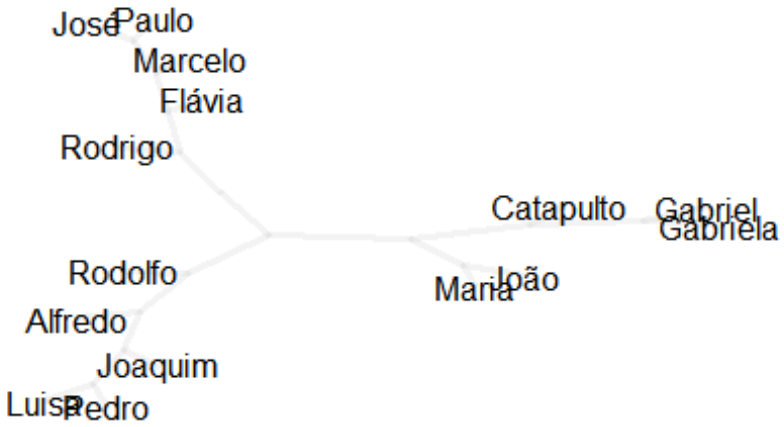

```
ggdendrogram(hc, rotate=FALSE,labels = TRUE)  + labs(title = "Dendogram
hclust method = complete")
```



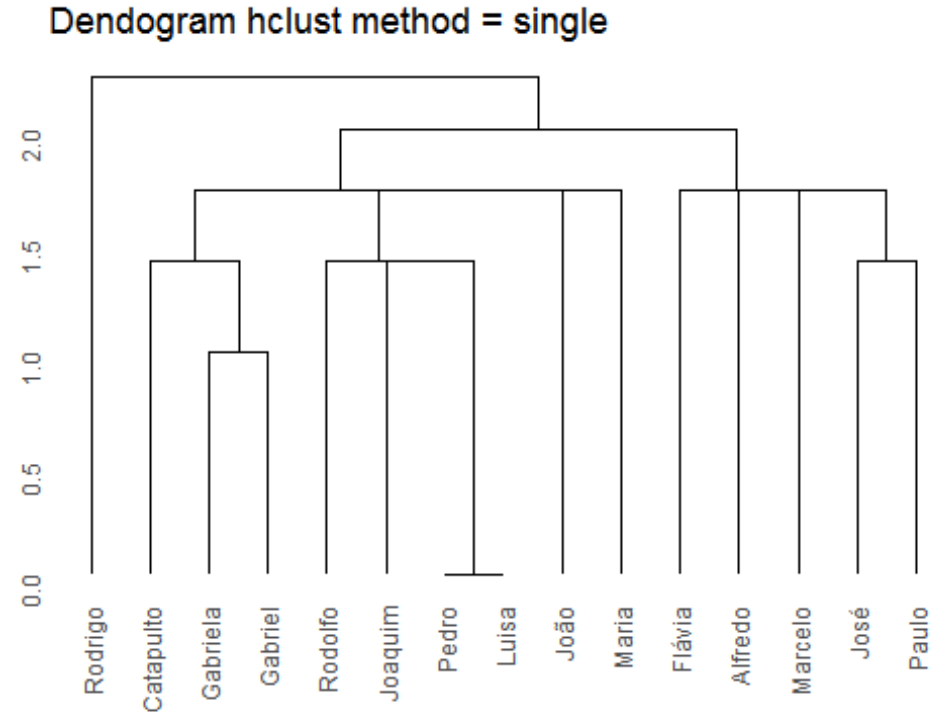

```
phylo_tree = as.phylo(hc)
graph_edges = phylo_tree$edge
graph_net = graph.edgelist(graph_edges)
plot(graph_net)
```



```
graph_layout = layout.auto(graph_net)
nobs = length(hc$labels)
{
  plot(graph_layout[,1], graph_layout[,2], type = "n", axes = FALSE,
     xlab = "", ylab = "")
  segments(
    x0 = graph_layout[graph_edges[,1],1],
    y0 = graph_layout[graph_edges[,1],2],
    x1 = graph_layout[graph_edges[,2],1],
    y1 = graph_layout[graph_edges[,2],2],
    col = "#dcdcdc55", lwd = 3.5
    )
  text(graph_layout[1:nobs,1], graph_layout[1:nobs,2],
       phylo_tree$tip.label, cex = 1, xpd = TRUE, font = 1)
}
```

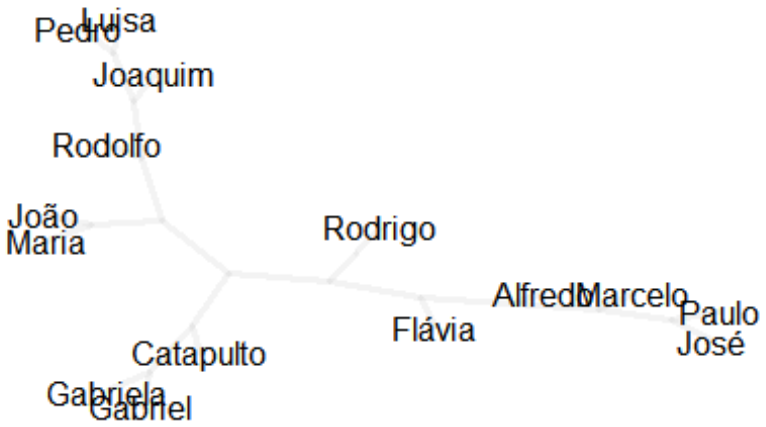



```
hc_single <- hclust(dist(grede_two_mode), method = "single")
ggdendrogram(hc_single, rotate=FALSE,labels = TRUE) + labs(title = "Dendogram
hclust method = single")
```

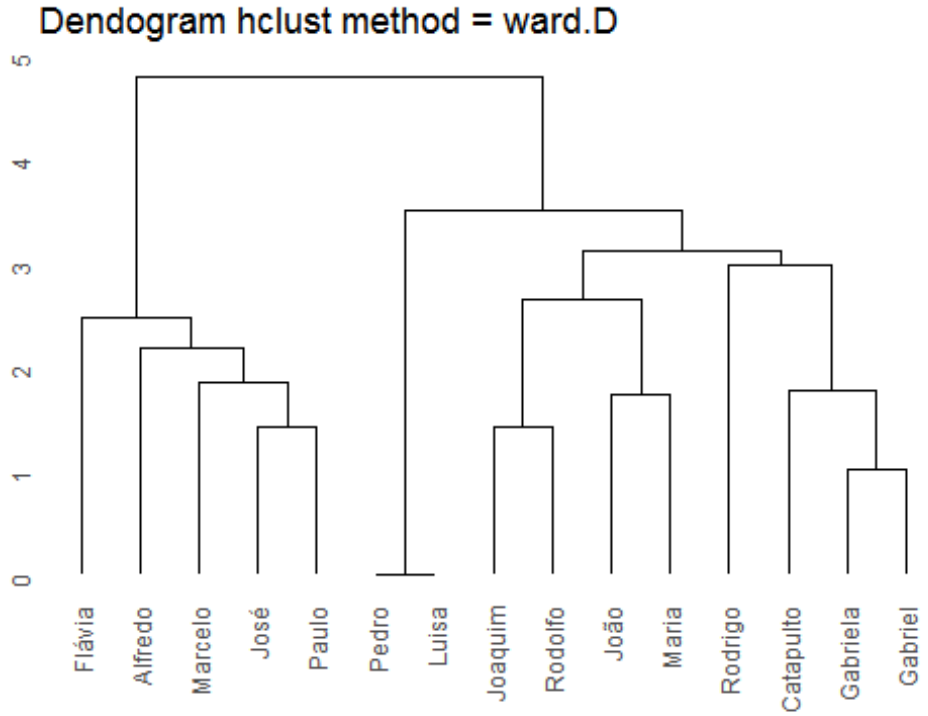


```
phylo_tree = as.phylo(hc_single)
graph_edges = phylo_tree$edge
graph_net = graph.edgelist(graph_edges)
plot(graph_net)
```

```
graph_layout = layout.auto(graph_net)
nobs = length(hc_single$labels)
{
  plot(graph_layout[,1], graph_layout[,2], type = "n", axes = FALSE,
     xlab = "", ylab = "")
  segments(
    x0 = graph_layout[graph_edges[,1],1],
    y0 = graph_layout[graph_edges[,1],2],
    x1 = graph_layout[graph_edges[,2],1],
    y1 = graph_layout[graph_edges[,2],2],
    col = "#dcdcdc55", lwd = 3.5
    )
  text(graph_layout[1:nobs,1], graph_layout[1:nobs,2],
       phylo_tree$tip.label, cex = 1, xpd = TRUE, font = 1)
}
```

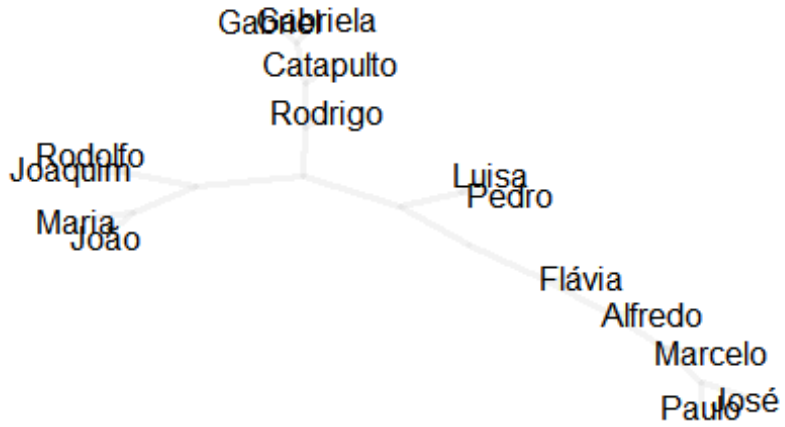



```
hc_wardD <- hclust(dist(grede_two_mode), method = "ward.D" )
ggdendrogram(hc_wardD, rotate=FALSE,labels = TRUE) + labs(title = "Dendogram
hclust method = ward.D")
```

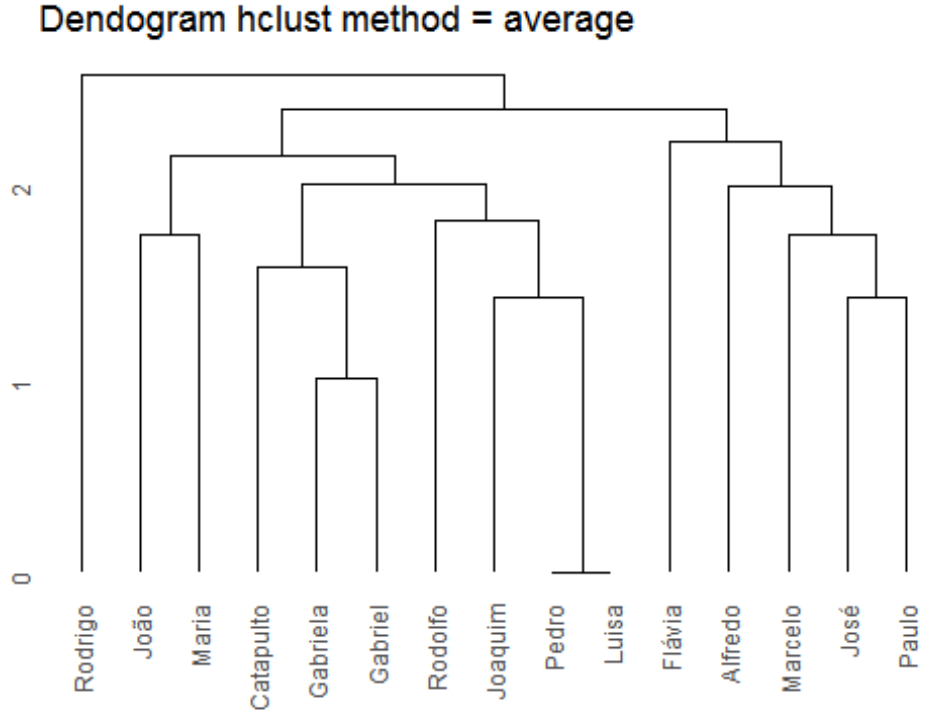


```
phylo_tree = as.phylo(hc_wardD)
graph_edges = phylo_tree$edge
graph_net = graph.edgelist(graph_edges)
plot(graph_net)
```

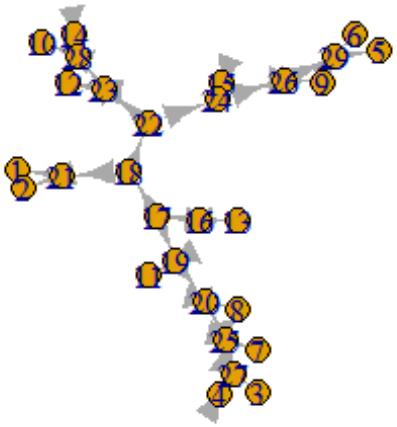
```
graph_layout = layout.auto(graph_net)
nobs = length(hc_wardD$labels)
{
  plot(graph_layout[,1], graph_layout[,2], type = "n", axes = FALSE,
     xlab = "", ylab = "")
  segments(
    x0 = graph_layout[graph_edges[,1],1],
    y0 = graph_layout[graph_edges[,1],2],
    x1 = graph_layout[graph_edges[,2],1],
    y1 = graph_layout[graph_edges[,2],2],
    col = "#dcdcdc55", lwd = 3.5
    )
  text(graph_layout[1:nobs,1], graph_layout[1:nobs,2],
       phylo_tree$tip.label, cex = 1, xpd = TRUE, font = 1)
}
```



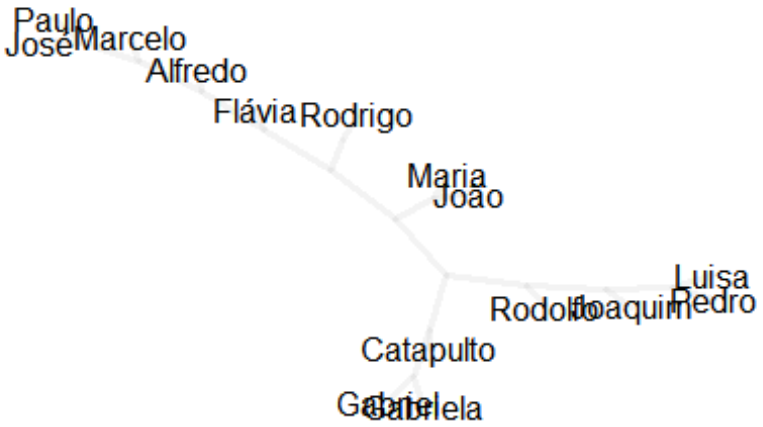

```
hc_average <- hclust(dist(grede_two_mode), method = "average" )
ggdendrogram(hc_average, rotate=FALSE,labels = TRUE) + labs(title =
"Dendogram hclust method = average")
```



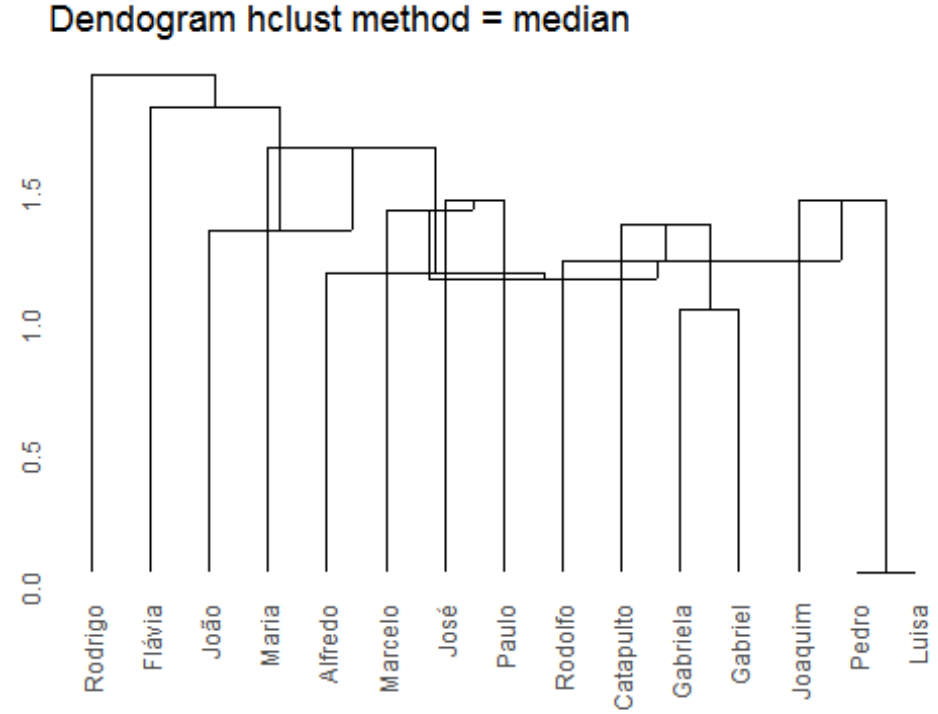
```
phylo_tree = as.phylo(hc_average)
graph_edges = phylo_tree$edge
graph_net = graph.edgelist(graph_edges)
plot(graph_net)
```



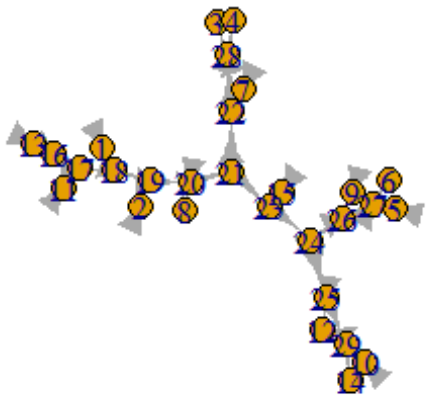
```
graph_layout = layout.auto(graph_net)
nobs = length(hc_average$labels)
{
  plot(graph_layout[,1], graph_layout[,2], type = "n", axes = FALSE,
     xlab = "", ylab = "")
  segments(
    x0 = graph_layout[graph_edges[,1],1],
    y0 = graph_layout[graph_edges[,1],2],
    x1 = graph_layout[graph_edges[,2],1],
    y1 = graph_layout[graph_edges[,2],2],
    col = "#dcdcdc55", lwd = 3.5
    )
  text(graph_layout[1:nobs,1], graph_layout[1:nobs,2],
       phylo_tree$tip.label, cex = 1, xpd = TRUE, font = 1)
}
```



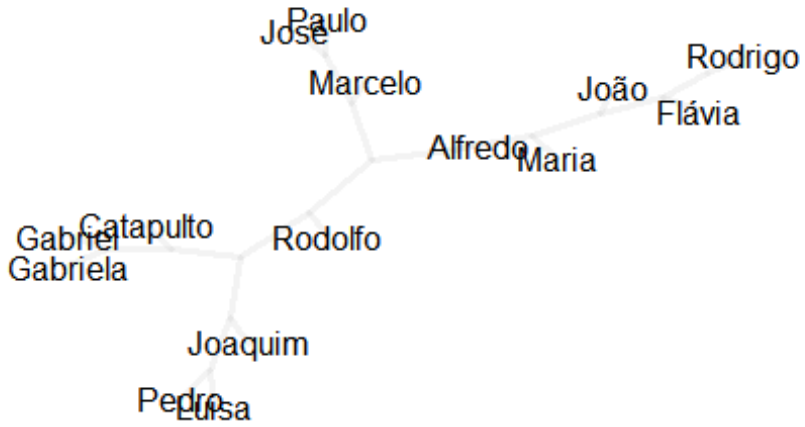

```
hc_median <- hclust(dist(grede_two_mode), method = "median" )
ggdendrogram(hc_median, rotate=FALSE,labels = TRUE) + labs(title = "Dendogram
hclust method = median")
```



```
phylo_tree = as.phylo(hc_median)
graph_edges = phylo_tree$edge
graph_net = graph.edgelist(graph_edges)
plot(graph_net)
```



```
graph_layout = layout.auto(graph_net)
nobs = length(hc$labels)
{
  plot(graph_layout[,1], graph_layout[,2], type = "n", axes = FALSE,
     xlab = "", ylab = "")
  segments(
    x0 = graph_layout[graph_edges[,1],1],
    y0 = graph_layout[graph_edges[,1],2],
    x1 = graph_layout[graph_edges[,2],1],
    y1 = graph_layout[graph_edges[,2],2],
    col = "#dcdcdc55", lwd = 3.5
    )
  text(graph_layout[1:nobs,1], graph_layout[1:nobs,2],
       phylo_tree$tip.label, cex = 1, xpd = TRUE, font = 1)
}
```


Paulo
José
Rodrigo
Marcelo
João
Flávia
Alfredo
Maria
Catapulto
Gabriel
Gabriela
Rodolfo
Joaquim
Pedro
Luísa