Tarefa Aula 1 – Exploração de SNA em R

Novembro de 2019

Table of Contents

[Alunos / ID (matrícula) 1](#_Toc24314967)

[1 Enunciado (parte 1) 2](#_Toc24314968)

[1.1 Explorando as rotinas de exemplo para rede one mode 2](#_Toc24314969)

[1.2 Aplicando modificações na rede one mode e analisando os resultados 7](#_Toc24314970)

[1.3 Explorando as rotinas de exemplo para rede two mode 10](#_Toc24314971)

[1.4 Aplicando modificações na rede two mode e analisando os resultados 13](#_Toc24314972)

[1.5 Analisando os resultados 17](#_Toc24314973)

[2 Enunciado (parte 2) 17](#_Toc24314974)

[2.1 Analisando agrupamento do tipo hierárquico aglomerativo 17](#_Toc24314975)

[2.2 Visualizando os clusters 21](#_Toc24314976)

[2.3 Checando as estatísticas 23](#_Toc24314977)

# Alunos / ID (matrícula)

* [Daniel Campos](mailto:daniel.ferraz.campos@gmail.com) / A57635769
* [Leandro Daniel](mailto:contato@leandrodaniel.com) / A57622988
* [Rodrigo Goncalves](mailto:rodrigo.goncalves@me.com) / A57566093
* [Ygor Lima](mailto:ygor_redesocial@hotmail.com) / A57549661

# Enunciado (parte 1)

* 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\_Berrini T1.xlsx e Rede Two Mode\_Tarefa Aula 1\_Berrini T1.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.

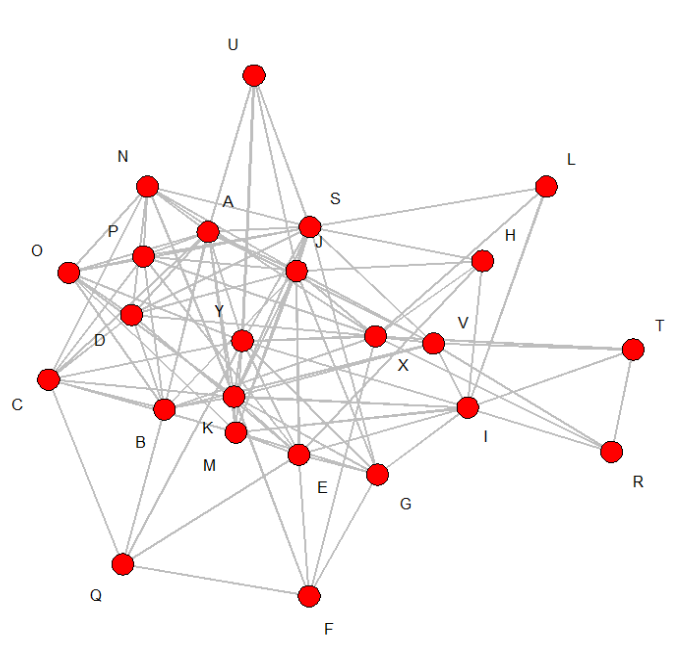
## Explorando as rotinas de exemplo para rede one mode

Como primeiro passo para o exercício proposto, vamos carregar os dados da planilha Excel **Rede One Mode\_Tarefa Aula 1\_Berrini T1.xlsx**. Incialmente, removeremos a primeira coluna que possui os nomes dos nós e vamos atribui-la às linhas.

# carregando a planilha com os dados do exercício  
rede\_one\_mode <- read\_xlsx("../../data/raw/Rede One Mode\_Tarefa Aula 1\_Berrini T1.xlsx",   
 sheet = "Exemplo Rede") %>% as.data.frame()  
  
# removendo a coluna com os nomes dos nós  
grede\_one\_mode <- rede\_one\_mode[,2:25]  
grede\_one\_mode <- sapply(grede\_one\_mode, as.integer)  
  
# atribuindo o nome dos nós às linhas  
rownames(grede\_one\_mode) <- rede\_one\_mode[,1]

Podemos então, visualizar a rede sob a forma de um grafo.

gplot(grede\_one\_mode,   
 gmode = "graph",   
 displaylabels = TRUE,   
 edge.col = "gray",   
 usearrows = FALSE)



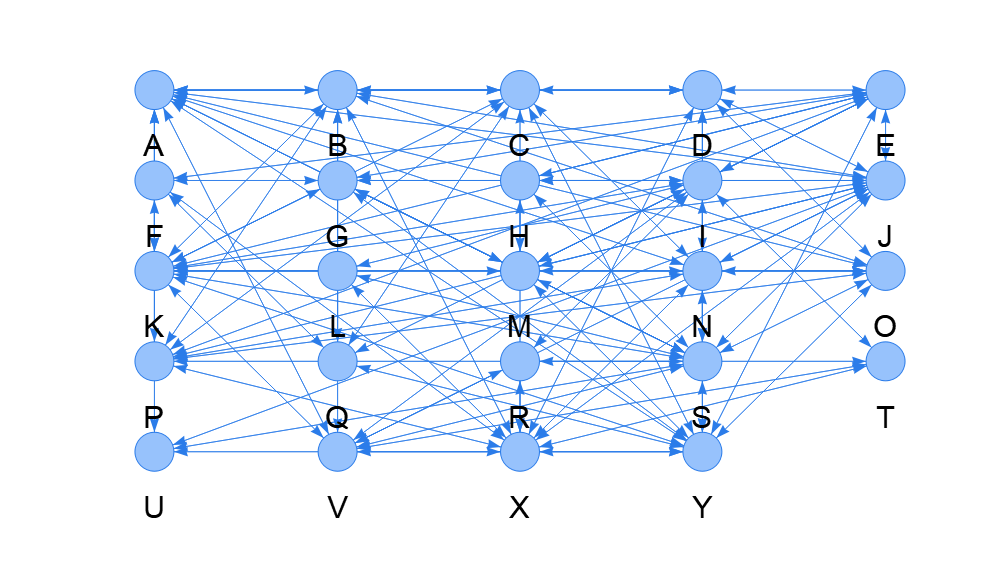
O **R** oferece várias packages para manipulação e geração de visualizações de redes. Abaixo, vemos um exemplo utilizando a library **visNetwork** com layouts alternativos e interativos.

* **layout\_on\_grid**;
* **layout\_in\_circle**;
* **layout\_with\_gem**;
* **layout\_as\_tree**.

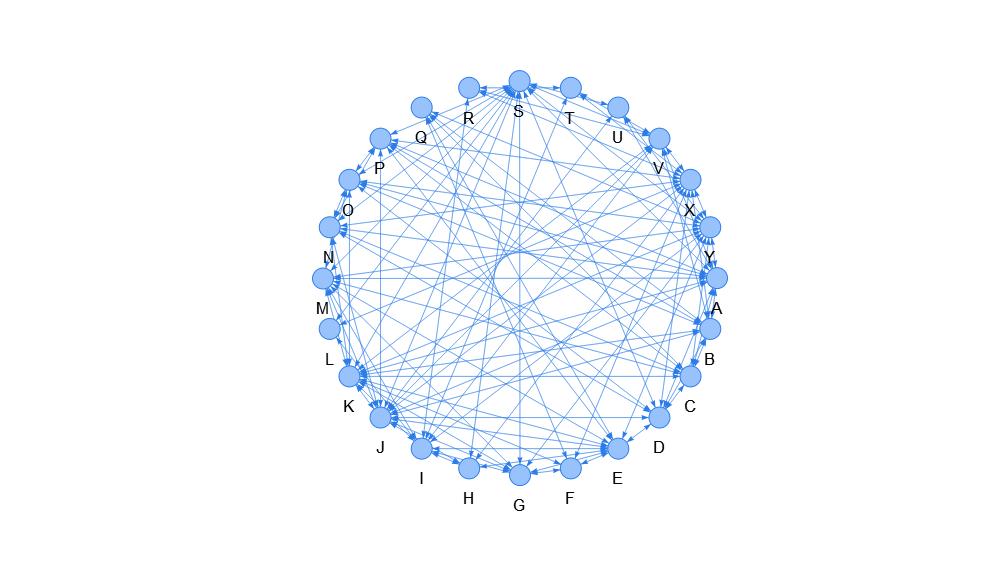
Podemos, inclusíve, adicionar a opção para fazer o highlight dos nós relacionados ao selecionar um nó específico.

g\_one\_mode <- igraph::graph\_from\_adjacency\_matrix(as.matrix(grede\_one\_mode))  
  
p <- visIgraph(g\_one\_mode) %>%  
 visOptions(highlightNearest = list(enabled = TRUE,   
 degree = 1,   
 algorithm = 'hierarchical')) %>%   
 visNodes(font = list(size = 40, color = "black"))

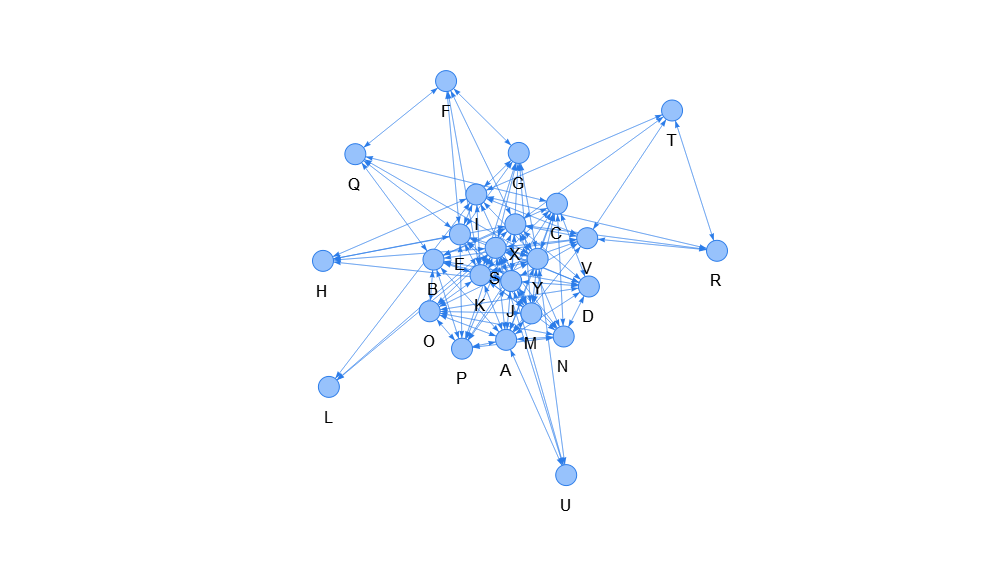
p %>% visIgraphLayout(layout = 'layout\_on\_grid', type = "full")



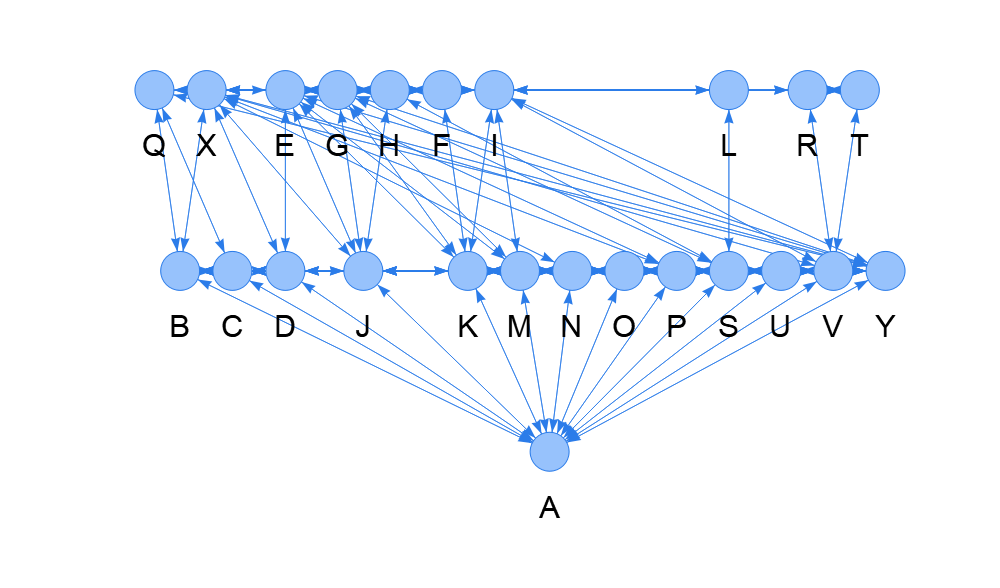
p %>% visIgraphLayout(layout = 'layout\_in\_circle')



p %>% visIgraphLayout(layout = 'layout\_with\_gem')



p %>% visIgraphLayout(layout = 'layout\_as\_tree', type = "full")



A **SNA**, do inglês Systems Network Analysis (Análise de Sistemas de Redes), dispõe de um conjunto de métricas que auxiliam na compreensão da dinâmica dos relacionamentos existentes nas redes analisadas. Dentre elas, destacam-se as métricas de Centralidade (Centrality), que permitem a avaliação da importância de um nó (individuo/ação/postagem) dentro da rede. Na sequência, vamos explorar as métricas de centralidade: **degree**, **closeness** e **betweenness**.

A **Centralidade de Grau** (Degree Centrality) representa o número de ligações que um nó possui (degree), dividido pelo número de ligações possíveis.

Definimos como **Centralidade de Proximidade** (Closeness Centrality) a medida do comprimento médio dos caminhos mais curtos de um vértice para cada um dos outros vértices de um grafo.

A chamada **Centralidade de Intermediação** (Betweenness Centrality) representa a quantidade de vezes que um determinado nó aparece no caminho geodésico entre dois nós da rede.

A seguir, exploraremos as métricas de centralidade para a rede one mode.

metricas\_one\_mode <- cbind(label = rownames(grede\_one\_mode),  
 degree = degree(grede\_one\_mode, gmode = "graph", cmode = "indegree"),  
 closeness = round(closeness(grede\_one\_mode, gmode = "graph"), 4),  
 betweenness = round(betweenness(grede\_one\_mode, gmode = "graph"), 4)) %>%   
 as\_tibble %>% type\_convert(col\_types = cols(.default = col\_double(),  
 label = col\_character()))  
  
metricas\_one\_mode <- arrange(metricas\_one\_mode, desc(`betweenness`))  
  
table\_output <- flextable(as.data.frame(metricas\_one\_mode))  
table\_output <- fontsize(table\_output, part = "all", size = 9)  
table\_output <- autofit(table\_output)  
table\_output

| label | degree | closeness | betweenness |
| --- | --- | --- | --- |
| X | 12.000 | 0.676 | 29.532 |
| S | 14.000 | 0.719 | 20.194 |
| I | 10.000 | 0.639 | 18.419 |
| Y | 12.000 | 0.676 | 16.260 |
| K | 14.000 | 0.719 | 15.603 |
| J | 14.000 | 0.719 | 14.838 |
| E | 11.000 | 0.657 | 12.517 |
| V | 9.000 | 0.622 | 11.349 |
| A | 13.000 | 0.697 | 8.310 |
| B | 10.000 | 0.639 | 7.649 |
| C | 9.000 | 0.561 | 4.386 |
| M | 10.000 | 0.639 | 3.953 |
| D | 9.000 | 0.622 | 3.371 |
| P | 9.000 | 0.622 | 3.371 |
| G | 8.000 | 0.605 | 2.426 |
| N | 9.000 | 0.622 | 2.320 |
| F | 5.000 | 0.548 | 2.033 |
| O | 9.000 | 0.590 | 1.996 |
| Q | 5.000 | 0.523 | 1.686 |
| H | 5.000 | 0.548 | 0.893 |
| L | 3.000 | 0.511 | 0.434 |
| R | 4.000 | 0.500 | 0.167 |
| T | 4.000 | 0.500 | 0.167 |
| U | 4.000 | 0.511 | 0.125 |

## Aplicando modificações na rede one mode e analisando os resultados

Decidimos aplicar uma transofmação na rede one mode original onde setamos uma matriz de vizinhança completamente randomica para avaliarmo os resultados.

Abaixo, a comparação de um recorte da matriz de vizinhança original com a matriz de vizinhança completamente randomizada.

grede\_one\_mode\_modificada <- grede\_one\_mode  
  
for (i in 1:24) {  
 grede\_one\_mode\_modificada[, i] <- as.integer(sample(0:1, 24, replace = TRUE))  
}  
  
# rede one mode original  
table\_output <- flextable(as.data.frame(grede\_one\_mode[1:5, 1:5] %>% as.matrix))  
table\_output <- fontsize(table\_output, part = "all", size = 9)  
table\_output <- autofit(table\_output)  
table\_output

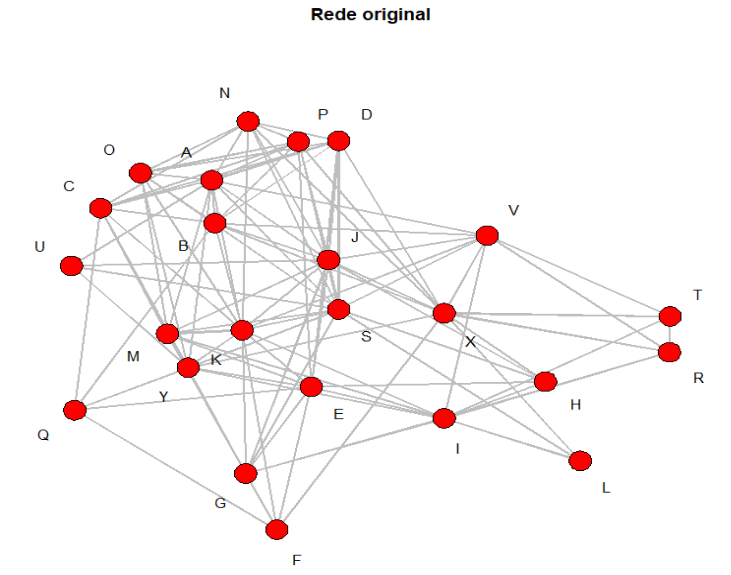
| A | B | C | D | E |
| --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |

# rede one mode modificada de forma randômica  
table\_output <- flextable(as.data.frame(grede\_one\_mode\_modificada[1:5, 1:5] %>% as.matrix))  
table\_output <- fontsize(table\_output, part = "all", size = 9)  
table\_output <- autofit(table\_output)  
table\_output

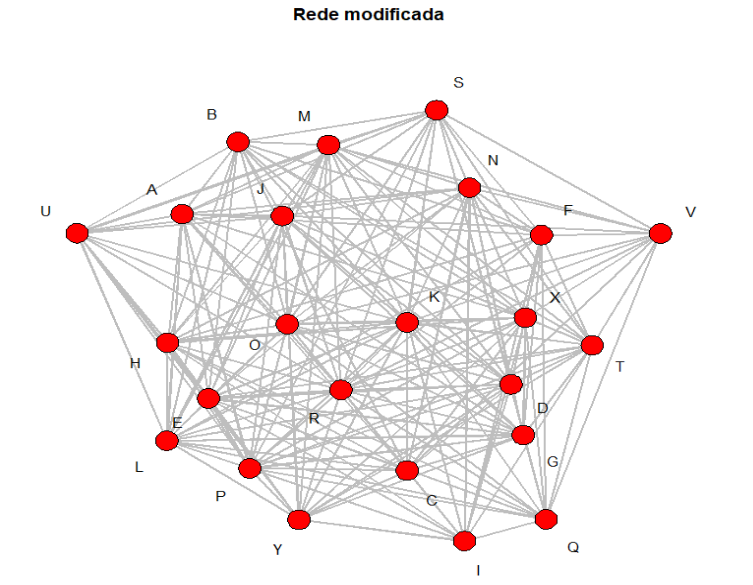
| A | B | C | D | E |
| --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |

Vizualizamos o plot da rede one mode modificada (completamente randomizada) em comparação com a rede original.

gplot(grede\_one\_mode,   
 main = 'Rede original',   
 gmode = "graph",   
 displaylabels = TRUE,   
 edge.col = "gray",   
 usearrows = FALSE)



gplot(grede\_one\_mode\_modificada,   
 main = 'Rede modificada',   
 gmode = "graph",   
 displaylabels = TRUE,   
 edge.col = "gray",   
 usearrows = FALSE)



A seguir, exploraremos as métricas de centralidade para a rede one mode modificada.

metricas\_one\_mode\_modificada <- cbind(label = rownames(grede\_one\_mode\_modificada),  
 degree = degree(grede\_one\_mode\_modificada,  
 gmode = "graph", cmode = "indegree"),  
 closeness = round(closeness(grede\_one\_mode\_modificada,   
 gmode = "graph"), 4),  
 betweenness = round(betweenness(grede\_one\_mode\_modificada,   
 gmode = "graph"), 4)) %>%   
 as\_tibble %>% type\_convert(col\_types = cols(.default = col\_double(),  
 label = col\_character()))  
  
metricas\_one\_mode\_modificada <- arrange(metricas\_one\_mode\_modificada, desc(`betweenness`))  
  
table\_output <- flextable(as.data.frame(metricas\_one\_mode\_modificada))  
table\_output <- fontsize(table\_output, part = "all", size = 9)  
table\_output <- autofit(table\_output)  
table\_output

| label | degree | closeness | betweenness |
| --- | --- | --- | --- |
| N | 16.000 | 0.885 | 12.847 |
| J | 13.000 | 0.793 | 8.803 |
| P | 12.000 | 0.885 | 8.702 |
| C | 15.000 | 0.852 | 8.471 |
| V | 13.000 | 0.885 | 7.186 |
| R | 10.000 | 0.821 | 6.630 |
| K | 10.000 | 0.793 | 6.467 |
| O | 10.000 | 0.793 | 6.139 |
| G | 12.000 | 0.885 | 6.056 |
| F | 15.000 | 0.821 | 5.794 |
| A | 13.000 | 0.821 | 5.712 |
| T | 12.000 | 0.742 | 5.475 |
| S | 10.000 | 0.852 | 5.317 |
| H | 13.000 | 0.852 | 5.281 |
| L | 11.000 | 0.742 | 4.706 |
| I | 10.000 | 0.767 | 4.692 |
| M | 8.000 | 0.793 | 4.499 |
| X | 9.000 | 0.767 | 4.487 |
| Y | 11.000 | 0.767 | 4.304 |
| E | 13.000 | 0.767 | 4.226 |
| U | 10.000 | 0.742 | 3.893 |
| D | 10.000 | 0.767 | 3.726 |
| B | 8.000 | 0.793 | 3.584 |
| Q | 8.000 | 0.697 | 3.003 |

## Explorando as rotinas de exemplo para rede two mode

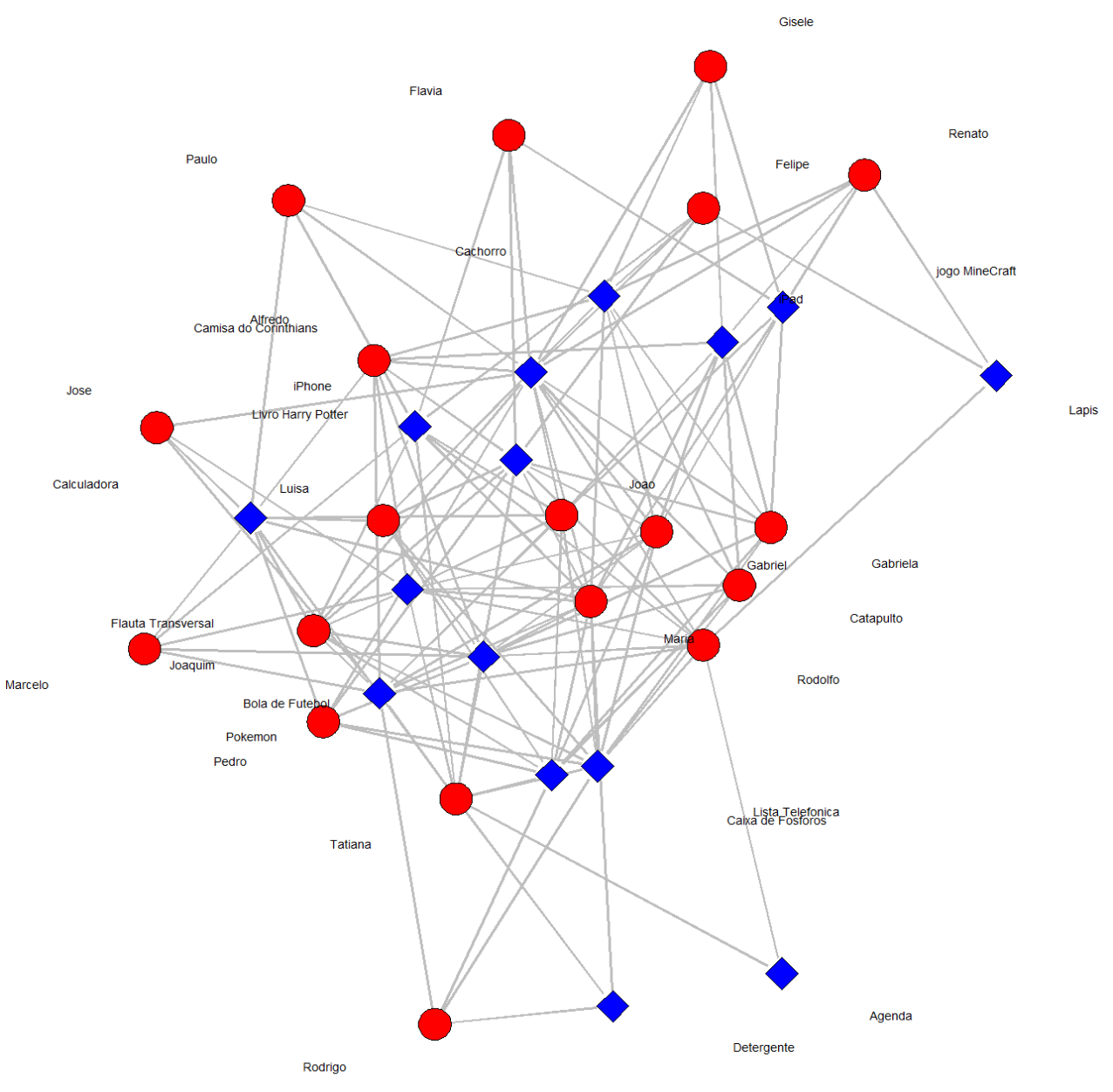
Como primeiro passo para o exercício proposto, vamos carregar os dados da planilha Excel **Rede Two Mode\_Tarefa Aula 1\_Berrini T1.xlsx**. Faremos um tratamento semelhante ao realizado para a rede anterior.

# carregando a planilha com os dados do exercício  
rede\_two\_mode <- read\_xlsx("../../data/raw/Rede Two Mode\_Tarefa Aula 1\_Berrini T1.xlsx",   
 sheet = "Rede 2-mode") %>% as.data.frame

names(rede\_two\_mode)[1] <- 'nome'  
  
# removendo a coluna com os nomes dos nós  
grede\_two\_mode <- rede\_two\_mode[, 2:16]  
grede\_two\_mode <- sapply(grede\_two\_mode, as.integer)  
  
# atribuindo o nome dos nós às linhas  
rownames(grede\_two\_mode) <- rede\_two\_mode[,1]

Visualizando a rede sob a forma de um grafo.

gplot(grede\_two\_mode, gmode = "twomode",   
 displaylabels = TRUE,  
 edge.col = "gray",   
 label.cex = 0.7,   
 usearrows = FALSE)



A seguir, exploraremos as métricas de centralidade para a rede two mode.

metricas\_two\_mode <- cbind(labels = append(rownames(grede\_two\_mode),  
 names(rede\_two\_mode)[-1]),  
 degree = degree(grede\_two\_mode, gmode = "twomode", cmode = "indegree"),  
 closeness = round(closeness(grede\_two\_mode, gmode = "twomode"), 4),  
 betweenness = round(betweenness(grede\_two\_mode, gmode = "twomode"), 4)) %>%   
 as\_tibble %>% type\_convert(col\_types = cols(.default = col\_double(),  
 labels = col\_character()))  
  
metricas\_two\_mode <- arrange(metricas\_two\_mode, desc(`betweenness`))  
  
table\_output <- flextable(as.data.frame(metricas\_two\_mode))  
table\_output <- fontsize(table\_output, part = "all", size = 9)  
table\_output <- autofit(table\_output)  
table\_output

| labels | degree | closeness | betweenness |
| --- | --- | --- | --- |
| Camisa do Corinthians | 16.000 | 0.623 | 163.800 |
| Maria | 13.000 | 0.600 | 106.774 |
| Rodolfo | 9.000 | 0.524 | 80.173 |
| Tatiana | 10.000 | 0.508 | 69.597 |
| Livro Harry Potter | 11.000 | 0.524 | 58.610 |
| Lista Telefonica | 11.000 | 0.524 | 55.605 |
| Pokemon | 10.000 | 0.508 | 54.485 |
| Cachorro | 10.000 | 0.492 | 53.434 |
| iPhone | 9.000 | 0.492 | 51.121 |
| Bola de Futebol | 11.000 | 0.524 | 49.606 |
| Calculadora | 10.000 | 0.492 | 45.799 |
| Gabriel | 10.000 | 0.541 | 45.025 |
| Flauta Transversal | 10.000 | 0.508 | 42.350 |
| Caixa de Fosforos | 10.000 | 0.492 | 41.931 |
| Joao | 9.000 | 0.524 | 40.261 |
| Joaquim | 9.000 | 0.524 | 28.877 |
| iPad | 8.000 | 0.465 | 27.547 |
| Alfredo | 8.000 | 0.508 | 27.200 |
| Gabriela | 8.000 | 0.508 | 26.810 |
| jogo MineCraft | 7.000 | 0.452 | 25.162 |
| Felipe | 5.000 | 0.452 | 19.352 |
| Renato | 5.000 | 0.429 | 19.139 |
| Catapulto | 7.000 | 0.492 | 18.945 |
| Luisa | 7.000 | 0.492 | 17.922 |
| Pedro | 6.000 | 0.478 | 11.141 |
| Marcelo | 5.000 | 0.440 | 7.026 |
| Rodrigo | 4.000 | 0.398 | 6.781 |
| Paulo | 4.000 | 0.440 | 6.733 |
| Flavia | 4.000 | 0.440 | 6.600 |
| Lapis | 3.000 | 0.388 | 5.423 |
| Jose | 4.000 | 0.452 | 4.999 |
| Gisele | 4.000 | 0.418 | 4.645 |
| Detergente | 3.000 | 0.407 | 4.225 |
| Agenda | 2.000 | 0.379 | 0.899 |

## Aplicando modificações na rede two mode e analisando os resultados

Assim como modificamos a rede one mode para uma rede randomica vamos aplicar a mesma transformação para a rede two mode.

grede\_two\_mode\_modificada <- grede\_two\_mode  
  
for (i in 1:15) {  
 grede\_two\_mode\_modificada[, i] <- as.integer(sample(0:1, 19, replace = TRUE))  
}  
  
# rede one mode original (primeiras 6 linhas e colunas)  
table\_output <- flextable((as.data.frame(grede\_two\_mode[1:5, 1:5] %>% as.matrix)))  
table\_output <- fontsize(table\_output, part = "all", size = 9)  
table\_output <- autofit(table\_output)  
table\_output

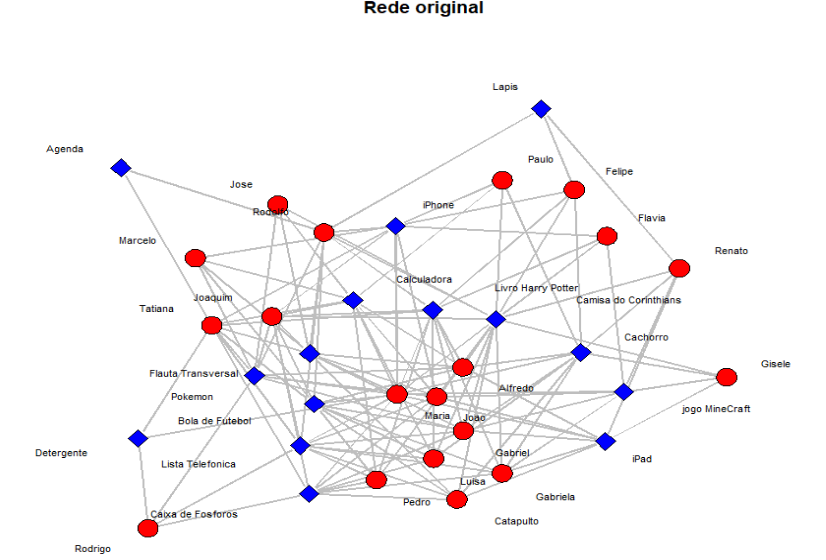
| iPhone | iPad | Livro Harry Potter | jogo MineCraft | Camisa do Corinthians |
| --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 |

# rede one mode modificada de forma randomica (primeiras 6 linhas e colunas)'  
table\_output <- flextable(as.data.frame(grede\_two\_mode\_modificada[1:5, 1:5] %>% as.matrix))  
table\_output <- fontsize(table\_output, part = "all", size = 9)  
table\_output <- autofit(table\_output)  
table\_output

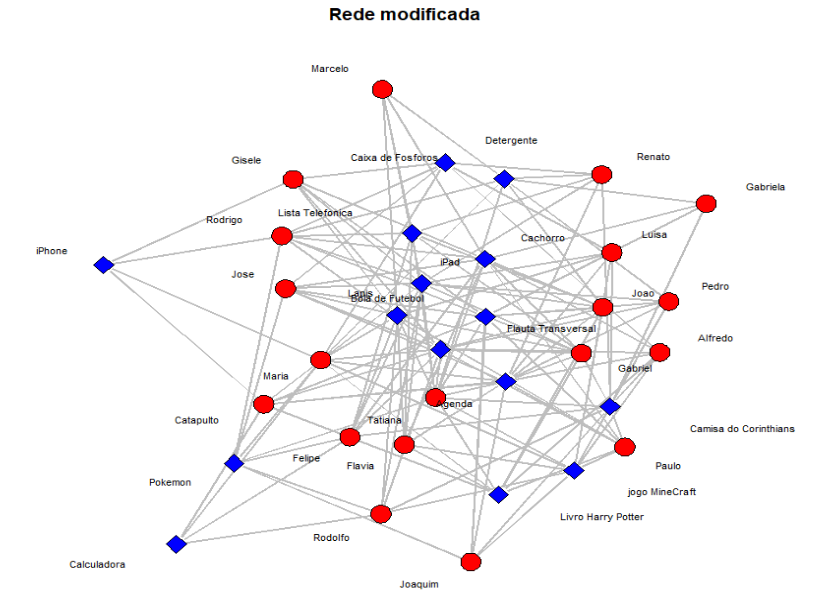
| iPhone | iPad | Livro Harry Potter | jogo MineCraft | Camisa do Corinthians |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Vizualizamos o plot da rede one mode modificada (completamente randomizada) em comparação com a rede original.

gplot(grede\_two\_mode,   
 main = 'Rede original',   
 gmode = "twomode",   
 displaylabels = TRUE,  
 edge.col = "gray",   
 label.cex = 0.7,   
 usearrows = FALSE)



gplot(grede\_two\_mode\_modificada,   
 main = 'Rede modificada',   
 gmode = "twomode",   
 displaylabels = TRUE,  
 edge.col = "gray",   
 label.cex = 0.7,   
 usearrows = FALSE)



metricas\_two\_mode\_modificada <- cbind(labels = append(rownames(grede\_two\_mode\_modificada),  
 names(rede\_two\_mode)[-1]),  
 degree = degree(grede\_two\_mode\_modificada,  
 gmode = "twomode", cmode = "indegree"),  
 closeness = round(closeness(grede\_two\_mode\_modificada,  
 gmode = "twomode"), 4),  
 betweenness = round(betweenness(grede\_two\_mode\_modificada,  
 gmode = "twomode"), 4)) %>%   
 as\_tibble %>% type\_convert(col\_types = cols(.default = col\_double(),  
 labels = col\_character()))  
  
metricas\_two\_mode\_modificada <- arrange(metricas\_two\_mode\_modificada, desc(`betweenness`))  
  
table\_output <- flextable(as.data.frame(metricas\_two\_mode\_modificada))  
table\_output <- fontsize(table\_output, part = "all", size = 9)  
table\_output <- autofit(table\_output)  
table\_output

| labels | degree | closeness | betweenness |
| --- | --- | --- | --- |
| Maria | 11.000 | 0.559 | 71.380 |
| Camisa do Corinthians | 12.000 | 0.541 | 70.727 |
| Lapis | 12.000 | 0.541 | 67.517 |
| Joaquim | 11.000 | 0.559 | 65.446 |
| jogo MineCraft | 11.000 | 0.524 | 60.410 |
| Jose | 10.000 | 0.541 | 60.087 |
| Flauta Transversal | 10.000 | 0.508 | 58.259 |
| Detergente | 10.000 | 0.508 | 57.523 |
| Pokemon | 10.000 | 0.508 | 55.534 |
| Livro Harry Potter | 9.000 | 0.492 | 53.341 |
| Calculadora | 9.000 | 0.492 | 45.189 |
| Cachorro | 9.000 | 0.492 | 44.493 |
| Paulo | 9.000 | 0.524 | 43.053 |
| Agenda | 9.000 | 0.492 | 36.852 |
| Renato | 8.000 | 0.508 | 35.883 |
| Marcelo | 8.000 | 0.492 | 31.144 |
| Gabriela | 7.000 | 0.492 | 26.251 |
| Catapulto | 7.000 | 0.492 | 26.176 |
| iPhone | 7.000 | 0.465 | 26.102 |
| Lista Telefonica | 7.000 | 0.465 | 25.319 |
| Alfredo | 7.000 | 0.492 | 24.597 |
| Caixa de Fosforos | 7.000 | 0.465 | 23.492 |
| Joao | 7.000 | 0.492 | 23.159 |
| Tatiana | 7.000 | 0.492 | 22.892 |
| iPad | 6.000 | 0.452 | 20.617 |
| Luisa | 5.000 | 0.465 | 14.658 |
| Flavia | 6.000 | 0.478 | 13.077 |
| Rodolfo | 6.000 | 0.478 | 12.678 |
| Rodrigo | 5.000 | 0.440 | 12.378 |
| Gabriel | 5.000 | 0.452 | 11.642 |
| Felipe | 5.000 | 0.465 | 11.329 |
| Bola de Futebol | 5.000 | 0.440 | 8.626 |
| Gisele | 5.000 | 0.440 | 8.336 |
| Pedro | 4.000 | 0.429 | 7.836 |

## Analisando os resultados

Assim como modificamos a rede one mode para uma rede randomica vamos aplicar a mesma transformação para a rede two mode.

Observando a métrica de **betweenness**, constatamos que, por estamos utilizando uma geração randômica de interações, não temos nenhum node se destacando com um elo importante para a rede.

Na mesma linha, a métrica de **closeness** mostra que a medida do comprimento médio dos caminhos mais curtos de um vértice para um dos outros vértices do grafo ficam similares.

# Enunciado (parte 2)

* **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.*

## Analisando agrupamento do tipo hierárquico aglomerativo

Como primeiro passo para o exercício proposto, vamos carregar novamente os dados da planilha Excel **Rede Two Mode\_Tarefa Aula 1\_Berrini T1.xlsx**.

# carregando a planilha com os dados do exercício  
rede\_two\_mode <- read\_xlsx("../../data/raw/Rede Two Mode\_Tarefa Aula 1\_Berrini T1.xlsx",   
 sheet = "Rede 2-mode") %>% as.data.frame

## New names:  
## \* `` -> ...1

names(rede\_two\_mode)[1] <- 'nome'  
  
# atribuindo o nome dos nós às linhas  
rownames(rede\_two\_mode) <- rede\_two\_mode$NA.

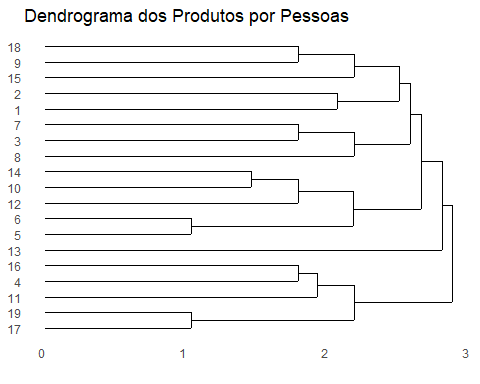
O **dendrograma** é um diagrama de árvore que exibe os grupos formados por agrupamento de observações em cada passo e em seus níveis de similaridade. O nível de similaridade é medido ao longo do eixo vertical (alternativamente, você pode exibir o nível de distância) e as diferentes observações são listadas ao longo do eixo horizontal.

Utilizaremos o dendrograma para visualizar como os agrupamentos são formados em cada passo e para avaliar os níveis de similaridade (ou distância) dos agrupamentos que são formados.

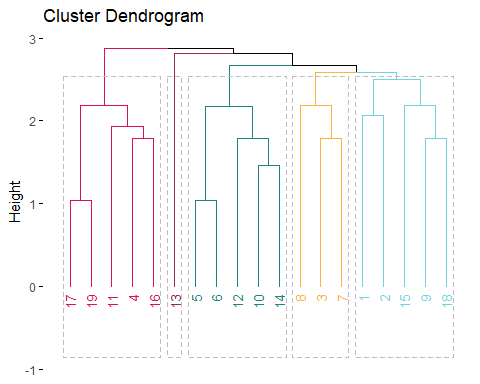
# criando o cluster hierárquico (Hierarchical Clustering)  
hc <- hclust(dist(rede\_two\_mode), "average")

## Warning in dist(rede\_two\_mode): NAs introduced by coercion

hcdata <- dendro\_data(hc)  
  
# plotando dendrograma (vertical)  
ggdendrogram(hcdata, rotate = TRUE, size = 2) +   
 labs(title = "Dendrograma dos Produtos por Pessoas")



# plotando dendrograma com cores, para k = 5, usando a library factoextra  
fviz\_dend(hc,   
 k = 5,   
 cex = 0.7,   
 k\_colors = c("#d81159", "#8f2d56", "#218380", "#fbb13c", "#73d2de"),   
 rect = TRUE)



Para exibir os níveis de similaridade (ou distância), mantenha seu ponteiro sobre uma linha horizontal no dendrograma. O padrão de como a similaridade ou os valores de distância mudam de um passo para outro pode ajudar a escolher o agrupamento final dos seus dados. O passo onde os valores mudam abruptamente podem identificar um bom ponto para definir o agrupamento final.

A decisão sobre o agrupamento final também é chamada corte do dendrograma. O corte do dendrograma é similar ao desenho de uma linha por todo o dendrograma para especificar o agrupamento final. Você também pode comparar os dendrogramas a agrupamentos finais diferentes para determinar qual agrupamento final faz mais sentido para os dados.

Analisando o dendrograma é possível verificar que 5 clusters parece ser uma boa escolha para o número de clusters, pois existe uma diferença significativa na altura das quebras das observações.

# "cortando" a árvore em 3 grupos  
grupos <- cutree(hc, k = 5)  
grupos

## [1] 1 1 2 3 4 4 2 2 1 4 3 4 5 4 1 3 3 1 3

# criando um dataframe para obter os nomes dos nodes  
nodes <- data.frame(id = seq(1:34),  
 label = c(rede\_two\_mode$nome, names(rede\_two\_mode)[-1]))  
# identifica os grupos  
nodes$group <- as.character(c(grupos, rep(6, 15)))  
# faz a contagem de relacionamentos entre os nodes, altera o nome das colunas e  
# seleciona apenas os nodes com alguma interação entre si  
edges <- pivot\_longer(data = rede\_two\_mode, cols = -nome, names\_to = 'to', values\_to = 'weight')  
names(edges) <- c('from', 'to', 'weight')  
edges <- filter(edges, weight > 0)  
edges$from <- as.integer(plyr::mapvalues(edges$from, nodes$label, nodes$id, warn\_missing = FALSE))  
edges$to <- as.integer(plyr::mapvalues(edges$to, nodes$label, nodes$id, warn\_missing = FALSE))  
visNetwork(nodes = nodes, edges = edges) %>%   
 visOptions(highlightNearest = list(enabled = TRUE,   
 degree = 0.2,   
 algorithm = 'hierarchical')) %>%   
 visNodes(font = list(size = 20, color = "black"), fixed = FALSE, physics = FALSE) %>%  
 visGroups(groupname = '1', color = '#d81159') %>%  
 visGroups(groupname = '2', color = '#8f2d56') %>%  
 visGroups(groupname = '3', color = '#218380') %>%  
 visGroups(groupname = '4', color = '#fbb13c') %>%  
 visGroups(groupname = '5', color = '#73d2de') %>%  
 visGroups(groupname = '6', color = '#202020', shape = 'triangle', size = 20)

A close up of a logo

Description automatically generated

## Visualizando os clusters

O método AGNES (Agglomerative Nesting) é um dos tipos mais comums de métodos de cluster hierárquico usado para agrupar objetos em clusters com base em sua similaridade.

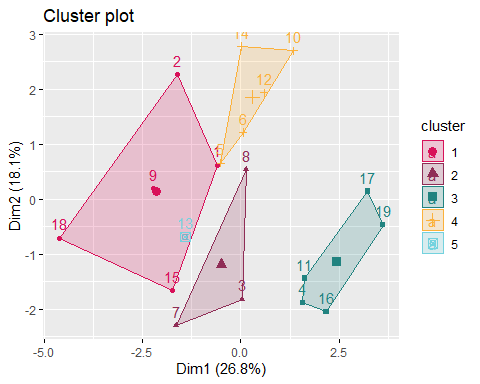
O algoritmo começa tratando cada observação como um cluster. Em seguida, pares de clusters são mesclados sucessivamente até que todos os clusters tenham sido mesclados em um grande cluster contendo todos os objetos.

Também podemos visualizar a classificação das observações utilizando os dois principais componentes da técnica de redução de dimensionalidade PCA (Principal Component Analysis).

# determinando a distância euclidiana  
cluster\_dist <- dist(rede\_two\_mode, method = "euclidean")

## Warning in dist(rede\_two\_mode, method = "euclidean"): NAs introduced by  
## coercion

# encontrando os clusters utilizando AGENS com o método WARD  
cluster\_agnes\_ward <- agnes(cluster\_dist,   
 diss = TRUE,   
 metric = 'euclidian',   
 method = 'ward')  
  
# plotando os clusters  
fviz\_cluster(list(data = rede\_two\_mode[,-1],   
 cluster = cutree(cluster\_agnes\_ward, k = 5)),   
 show.clust.cent = TRUE) +  
 scale\_colour\_manual(values = c("#d81159", "#8f2d56", "#218380", "#fbb13c", "#73d2de")) +  
 scale\_fill\_manual(values = c("#d81159", "#8f2d56", "#218380", "#fbb13c", "#73d2de"))



Assim como fizemos para o método AGNES podemos visualizar a classificação, para o método k-means, das observações utilizando os dois principais componentes da técnica de redução de dimensionalidade PCA (Principal Component Analysis).

O método de classificação por K-means é um dos algoritmos de aprendizado de máquina não supervisionados mais simples e populares.

O algoritmo começa com um primeiro grupo de centroides selecionados aleatoriamente, que são usados como pontos de partida para cada cluster e, em seguida, executa cálculos iterativos para otimizar as posições dos centroides.

Para cada iteração o algoritmo calcula a métrica de distância selecionada entre cada observação e o centroide classificando cada observação ao cluster cujo centroide está mais próximo.

Ele interrompe a criação e otimização de clusters quando:

* Os centroides se estabilizaram - não há alteração em seus valores porque o agrupamento foi bem-sucedido.
* O número definido de iterações foi alcançado.

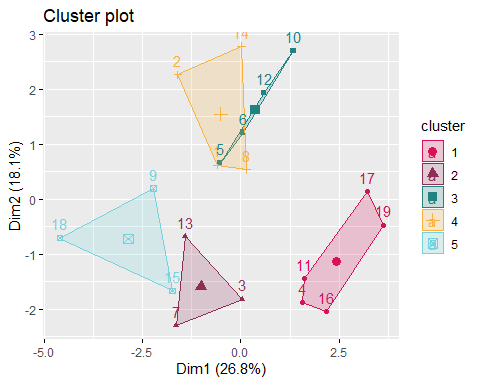
O parâmetro mais importante para o método de k-means é a quantidade de clusters que o algoritmo irá utilizar.

Para encontrar a melhor quantidade de clusters podemos observar o comportamento de duas métricas, withinss e betweenss, que medem as distâncias intra-cluster e inter-clusters de cada observação e centroide dos clusters.

A melhor quantidade de k é dada no ponto onde existe uma estabilização entre as métricas.

Observamos que os métodos classificaram as observações em clusters diferentes.

# encontrando kmeans, para k = 5  
cluster\_kmeans <- kmeans(rede\_two\_mode[,-1], centers = 5, nstart = 100)  
  
# plotando os clusters  
fviz\_cluster(list(data = rede\_two\_mode[,-1],   
 cluster = cluster\_kmeans$cluster),  
 show.clust.cent = TRUE) +  
 scale\_colour\_manual(values = c("#d81159", "#8f2d56", "#218380", "#fbb13c", "#73d2de")) +  
 scale\_fill\_manual(values = c("#d81159", "#8f2d56", "#218380", "#fbb13c", "#73d2de"))



## Checando as estatísticas

Podemos analisar todas as métricas do cluster com o método **ward** utilizando a função **cluster.stats**.

# estatísticas do cluster com AGNES, com método WARD  
cluster.stats(cluster\_dist, cutree(cluster\_agnes\_ward, k = 5))

## $n  
## [1] 19  
##   
## $cluster.number  
## [1] 5  
##   
## $cluster.size  
## [1] 5 3 5 5 1  
##   
## $min.cluster.size  
## [1] 1  
##   
## $noisen  
## [1] 0  
##   
## $diameter  
## [1] 2.921187 2.309401 2.309401 2.529822 NA  
##   
## $average.distance  
## [1] 2.324492 2.054616 1.980107 1.913976 NaN  
##   
## $median.distance  
## [1] 2.187496 2.065591 2.065591 1.927223 NA  
##   
## $separation  
## [1] 1.788854 2.065591 2.065591 1.788854 2.529822  
##   
## $average.toother  
## [1] 2.812774 2.680176 2.879056 2.687038 2.862085  
##   
## $separation.matrix  
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0.000000 2.065591 2.529822 1.788854 2.529822  
## [2,] 2.065591 0.000000 2.065591 2.065591 2.529822  
## [3,] 2.529822 2.065591 0.000000 2.065591 2.921187  
## [4,] 1.788854 2.065591 2.065591 0.000000 2.529822  
## [5,] 2.529822 2.529822 2.921187 2.529822 0.000000  
##   
## $ave.between.matrix  
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0.000000 2.582117 3.139002 2.619831 2.838327  
## [2,] 2.582117 0.000000 2.690909 2.734983 2.842776  
## [3,] 3.139002 2.690909 0.000000 2.709397 2.992067  
## [4,] 2.619831 2.734983 2.709397 0.000000 2.767447  
## [5,] 2.838327 2.842776 2.992067 2.767447 0.000000  
##   
## $average.between  
## [1] 2.777851  
##   
## $average.within  
## [1] 2.069818  
##   
## $n.between  
## [1] 138  
##   
## $n.within  
## [1] 33  
##   
## $max.diameter  
## [1] 2.921187  
##   
## $min.separation  
## [1] 1.788854  
##   
## $within.cluster.ss  
## [1] 31.14667  
##   
## $clus.avg.silwidths  
## 1 2 3 4 5   
## 0.05777471 0.15803368 0.23741491 0.24597002 0.00000000   
##   
## $avg.silwidth  
## [1] 0.1673631  
##   
## $g2  
## NULL  
##   
## $g3  
## NULL  
##   
## $pearsongamma  
## [1] 0.5931752  
##   
## $dunn  
## [1] 0.6123724  
##   
## $dunn2  
## [1] 1.110831  
##   
## $entropy  
## [1] 1.500365  
##   
## $wb.ratio  
## [1] 0.7451148  
##   
## $ch  
## [1] 3.780101  
##   
## $cwidegap  
## [1] 2.065591 2.065591 2.065591 1.788854 0.000000  
##   
## $widestgap  
## [1] 2.065591  
##   
## $sindex  
## [1] 1.788854  
##   
## $corrected.rand  
## NULL  
##   
## $vi  
## NULL

Finalmente, vemos abaixo as métricas gerais do método de classificação **k-means** aplicado a este dataset.

# estatísticas do cluster com KMEANS  
cluster.stats(cluster\_dist, cluster\_kmeans$cluster)

## $n  
## [1] 19  
##   
## $cluster.number  
## [1] 5  
##   
## $cluster.size  
## [1] 5 3 4 4 3  
##   
## $min.cluster.size  
## [1] 3  
##   
## $noisen  
## [1] 0  
##   
## $diameter  
## [1] 2.309401 2.732520 2.309401 2.732520 2.309401  
##   
## $average.distance  
## [1] 1.980107 2.350399 1.841848 2.211893 2.054616  
##   
## $median.distance  
## [1] 2.065591 2.529822 1.927223 2.187496 2.065591  
##   
## $separation  
## [1] 2.065591 2.065591 1.460593 1.460593 1.788854  
##   
## $average.toother  
## [1] 2.879056 2.761570 2.642762 2.686725 2.809381  
##   
## $separation.matrix  
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0.000000 2.065591 2.065591 2.529822 2.529822  
## [2,] 2.065591 0.000000 2.529822 2.065591 2.065591  
## [3,] 2.065591 2.529822 0.000000 1.460593 1.788854  
## [4,] 2.529822 2.065591 1.460593 0.000000 2.065591  
## [5,] 2.529822 2.065591 1.788854 2.065591 0.000000  
##   
## $ave.between.matrix  
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0.000000 2.791873 2.659351 2.910282 3.217545  
## [2,] 2.791873 0.000000 2.848099 2.772620 2.580961  
## [3,] 2.659351 2.848099 0.000000 2.424046 2.701399  
## [4,] 2.910282 2.772620 2.424046 0.000000 2.578474  
## [5,] 3.217545 2.580961 2.701399 2.578474 0.000000  
##   
## $average.between  
## [1] 2.757723  
##   
## $average.within  
## [1] 2.070029  
##   
## $n.between  
## [1] 143  
##   
## $n.within  
## [1] 28  
##   
## $max.diameter  
## [1] 2.73252  
##   
## $min.separation  
## [1] 1.460593  
##   
## $within.cluster.ss  
## [1] 30.86222  
##   
## $clus.avg.silwidths  
## 1 2 3 4 5   
## 0.24280004 0.06536266 0.21387112 0.05539007 0.18063915   
##   
## $avg.silwidth  
## [1] 0.1594237  
##   
## $g2  
## NULL  
##   
## $g3  
## NULL  
##   
## $pearsongamma  
## [1] 0.5587743  
##   
## $dunn  
## [1] 0.5345225  
##   
## $dunn2  
## [1] 1.031334  
##   
## $entropy  
## [1] 1.59027  
##   
## $wb.ratio  
## [1] 0.7506296  
##   
## $ch  
## [1] 3.847199  
##   
## $cwidegap  
## [1] 2.065591 2.529822 1.788854 2.065591 2.065591  
##   
## $widestgap  
## [1] 2.529822  
##   
## $sindex  
## [1] 1.460593  
##   
## $corrected.rand  
## NULL  
##   
## $vi  
## NULL