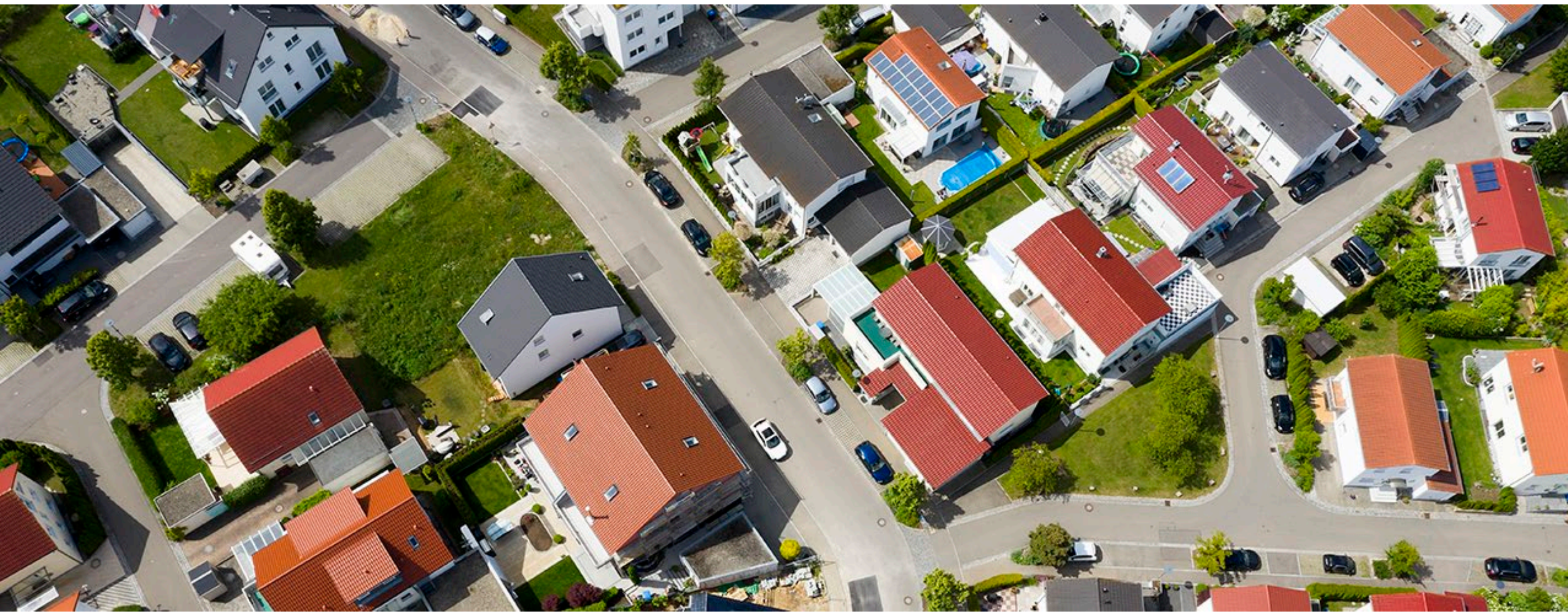
ISCTE INSTITUTO UNIVERSITÁRIO DE LISBOA

Ano Letivo 2023/2024

15 de dezembro de 2023

Rede de Contactos Sociais

Trabalho 1



UC Análise de Redes

Licenciatura Ciência de Dados

CDC1 e CDC2

Docentes

Maria João Lopes

Catarina Nunes

André Silvestre Nº104532

Eliane Gabriel Nº103303

Maria João Lourenço Nº104716

Margarida Pereira Nº105877

Umeima Mahomed Nº 99239

Índice

Introdução

Redes Sociais

As redes são estruturas complexas que podem ser utilizadas para modelar uma vasta gama de sistemas naturais e artificiais. No contexto das ciências sociais, as redes podem ser utilizadas para representar relações entre indivíduos, organizações ou entidades coletivas. [1]

Neste sentido, foi proposto no âmbito da UC de *Análise de Redes* inserida na licenciatura de Ciência de Dados, este trabalho que visa estudar uma rede não orientada que representa os contactos sociais diretos entre os habitantes de uma zona residencial. Os nodos da rede representam os habitantes da zona residencial e cada ligação representa um contacto social direto entre dois habitantes.

A análise da rede irá permitir-nos obter informações sobre a estrutura e a dinâmica das relações sociais entre os habitantes da zona residencial. Os resultados obtidos poderão ter aplicações práticas, por exemplo, na identificação de grupos sociais ou na compreensão da propagação de informações ou comportamentos.

Questão 1

Considerando toda a rede que representa os contactos sociais diretos entre habitantes de uma zona residencial, foram calculadas diversas métricas que permitem a sua análise. Dada a dimensão da rede, estas foram calculadas recorrendo à *package* `igraph` do *R*.

Dimensão e Densidade da Rede

Esta rede é **não orientada** e é constituída por 787 **nodos** e 1197 **ligações**. Cada **nodo** representa um habitante de uma zona residencial e cada **ligação** representa os contactos sociais diretos entre esses habitantes.

Graficamente a rede pode ser representada tal como visível na **Figura 1**.

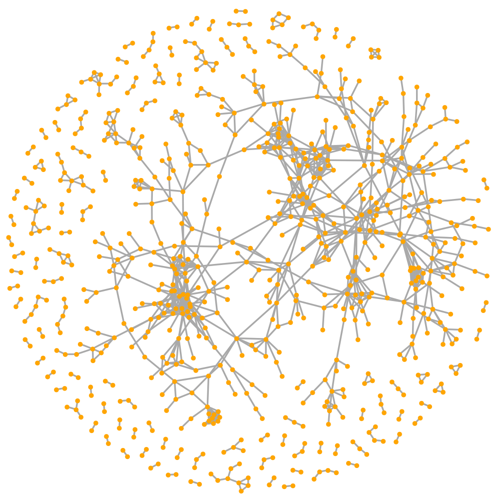


Figura 1 | Representação gráfica da rede.

Relativamente à **densidade**, d, que relaciona o número de ligações existentes numa rede com o número máximo possível de ligações, é obtida em redes não orientadas por:

$$d = \frac{L}{L_{max}} = \frac{L}{\frac{N(N-1)}{2}} = \frac{2L}{N(N-1)}$$

Neste caso, a densidade obtida é

$$\frac{2 \times 1197}{787 \times 786} = 0.003870142$$

Pode-se afirmar que uma rede é **esparsa** se L for muito inferior a L_{max}.

Nesta rede o número de ligações, L, é 1197 e o número máximo de ligações, L_{max}, é de $\frac{N(N-1)}{2} = \frac{787 \times 786}{2} = 309291$.

Como $1197 \ll 309291$ e $d \ll 1$, conclui-se que a rede é **esparsa**, tal como seria de esperar uma vez que, em geral, as redes reais são esparsas.

Deste modo, a densidade sugere que os habitantes da zona residencial não têm muitos contactos sociais diretos uns com os outros.

Grau dos Nodos da Rede

Quanto ao grau médio desta rede não orientada, este é calculado por:

$$< k > = \frac{\sum_i k_i}{N} = \frac{2 \times L}{N} = \frac{2 \times 1197}{787} = 3.041931$$

sendo k_i o grau de um nodo i, o nº de ligações incidentes nesse nodo.

Este valor pode ainda ser dado por:

$$< k > = d(N-1) = 0.003870142 \times (787-1) = 3.041931$$

Sendo o valor de $< k > \approx 3.042$, conclui-se que, em média, cada habitante estará aproximadamente conectado a 3 outros habitantes.

A distribuição de graus está apresentada na **Tabela 1** e posteriormente na **Figura 2**.

Graus	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
n	309	156	101	63	41	26	20	21	14	12	8	8	3	1	0	2	2

Tabela 1 | Distribuição do grau da rede.

Através da **Tabela 1**, vemos que a distribuição de graus é assimétrica, com a maioria dos habitantes a ter um grau baixo (1, 2 ou 3) e poucos são os que têm graus elevados, ou seja, o círculo social é relativamente pequeno.

Esta observação sugere que a rede é composta por uma grande maioria de habitantes com um círculo social relativamente pequeno, e por uma minoria de habitantes com um círculo social mais alargado.

Conexidade da Rede

A rede não é conexa, ou seja, não existe, pelo menos um caminho entre qualquer par de habitantes. O número de componentes conexas é 104, a dimensão mínima e máxima das componentes conexas são 2 e 496, respetivamente, ou seja, existem subredes isoladas e uma componente gigante com cerca de $\frac{2}{3}$ dos nodos da rede. (Fig.1)

Associação de Grau da Rede

A associação de grau ocorre quando os nodos de maior grau estão ligados principalmente a nodos de grau elevado e os de menor grau estão unidos principalmente a nodos de grau reduzido.

Medindo a associação baseada no grau através do coeficiente de correlação de *Pearson*, ρ ,o valor obtido foi 0.4765607. Trata-se de um valor positivo e maior que 0.3, logo é aceite como significativo de que a rede é **associativa**.

Aquando do cálculo da associatividade, tendo por base o grau médio adjacente de cada nodo, considera-se:

$$k_{nn}(i) = \frac{1}{k_i} \sum_j a_{ij} k_j$$

em que $a_{ij} = 1$ se i e j são adjacentes e 0 caso contrário.

Seja também a função $< k_{nn}(k) >$ que consiste na média dos valores $k_{nn}(i)$ calculada para os nodos i com grau igual a k:

- Se esta função é **crescente** então a rede **é associativa**.
- Se a função é **decrescente** então a rede **é não associativa**.

Aplicando esta teoria à rede em estudo, observa-se através do *output* produzido de **`knn(rede)$knnk`** no *R* que a função está a crescer monotonamente, tendo os valores $k_{nn}(1) = 2.76$ e $k_{nn}(17) = 9.26$, demonstrativos desse crescimento. Assim, uma vez mais, conclui-se que esta rede é **associativa**.

Interpretando este facto ao contexto da rede em estudo, podemos aferir que habitantes com graus semelhantes têm a tendência a se conectar entre si. Isso sugere uma estrutura de rede em que a popularidade ou centralidade dos nodos influencia as suas conexões, o que é uma característica de rede associativa.

Caminhos Mais Curtos e Diâmetro da Rede

A média dos comprimentos dos caminhos mais curtos (caminho entre um par de nodos com menor comprimento) de uma rede não orientada é dada por:

$$< l >= \frac{2\sum_{i,j} l_{ij}}{N(N-1)}$$

em que l_{ij} é o comprimento do caminho mais curto entre i e j, e N representa o número de nodos da rede.

Porém, como existem nodos sem caminhos mais curtos, pode considerar-se $l_{ij} = \infty$ e $(1 / l_{ij}) = 0$, e

$$< \boldsymbol{l} >= \left(\frac{2 \times \left(\sum_{i,j} 1/l_{ij}\right)}{N(N-1)}\right)^{-1}$$

Assim, para a rede em estudo, este valor pode ser calculado como:

$$< \boldsymbol{l} >= \left(\frac{2 \times 38984}{787 \times (787-1)}\right)^{-1} \approx 7.9$$

Utilizando a função `mean_distance`, o valor obtido foi de 7.914034, confirmando o calculado no *R*.

Quanto ao diâmetro da rede, l_{max}, este é o caminho mais longo dos caminhos mais curtos entre 2 nodos ($\max_{i,j} l_{i,j}$) sendo este 21.

Deste modo a distância média **não** é considerada **pequena**, dado que não é de todo próxima do valor de $\log N = \log 787 \approx 2.896$.

Esta conclusão ($< l > \sim log\ N$ indica distâncias pequenas) decorre do facto de a distância média ser **pequena** se cresce muito lentamente com o número de nodos da rede, e daí compara-se a distância média com o valor do $\log_{10} x$ do número de nodos, que é uma função que cresce muito lentamente.

Triângulos da Rede

Numa rede social, podem observar-se diversos triângulos – conjuntos de três nodos em que existe uma ligação entre cada par de nodos. Sendo esta rede num contexto semelhante, observa-se que no total existem 2307 triângulos.

Para estudar este valor, pode-se calcular o coeficiente de *Clustering* médio dos nodos, $< C >$, que define a média das frações de pares de nodos adjacentes a cada nodo da rede que estão ligados entre si, isto é, a média dos $C(i)$ dos nodos.

No cálculo desta média podem considerar-se apenas os nodos com $grau > 1$ ou todos os nodos da rede. Considerando a abordagem do grau ser superior a 1, a média é dada por:

$$< C >= \frac{\sum_{i:k_i>1} C(i)}{N_{k>1}}$$

Assim, na rede em estudo, este valor é calculado como:

$$< C > = \frac{210.9147}{478} = 0.44$$

De acordo com a interpretação probabilística, $< C >$ é a probabilidade de dois vizinhos de um nodo selecionado aleatoriamente se ligarem um ao outro. [1]

Adicionalmente, pode-se definir o coeficiente de *Clustering* da rede, que considera as subredes conexas de dimensão 3 (ternos conexos). No caso de existirem 3 ligações, é possível identificar um triângulo, ou seja, a cada triângulo estão associados 3 ternos conexos, cada um centrado num nodo.

Assim, este coeficiente é dado por:

$$C = \frac{Nº\ de\ ternos\ conexos\ fechados}{Nº\ total\ de\ ternos\ conexos} = \frac{3\ \times Nº\ de\ Triângulos}{Nº\ total\ de\ ternos\ conexos} = \frac{traço(A^3)}{\sum_i k_i(k_i - 1)}$$

Aplicando à rede, este coeficiente é de $C = 0.4199126$.

Sendo este valor superior a 0.3, podemos indicar que a rede tem um número significativo de que existem muitos triângulos.

Heterogeneidade da Rede

A amplitude de variação na distribuição dos graus dos nodos dá uma ideia sobre a heterogeneidade.

Representado graficamente a distribuição de grau (Fig.2)

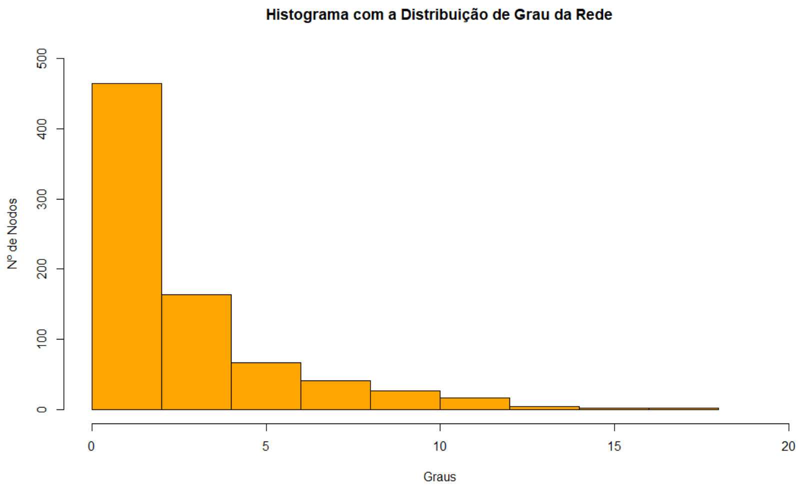


Figura 2 | Representação gráfica da distribuição de graus da rede.

Neste gráfico (Fig.2) verifica-se a existência de uma cauda longa ou pesada, relevando que a rede tem heterogeneidade nos valores dos graus.

Para se corroborar esta observação, pode-se considerar o parâmetro $\boldsymbol{K}$ dado por:

$$K = \frac{\langle k^2 \rangle}{\langle k \rangle^2}$$

Tendo sido alcançado o valor $K = 1.837585$.

Dado que este valor é bastante superior a 1, a rede é heterogénea (em grau) e indicia a existência de *hubs*, ou seja, nodos da rede com graus significativamente maiores que a média.

Desta forma, salienta-se que alguns habitantes têm um número significativamente maior de contactos sociais que a média.

Decomposição de *core* da Rede

Em redes de grande dimensão, é útil considerar apenas a **parte mais densa** *(core)*. O grau dos nodos pode ser utilizado para dividir a rede em partes, designadas por **conchas** *(shells)*, e baseadas na estrutura e localização periféricas da rede.

Para obter a parte mais densa da rede, baseada no grau dos nodos, começa-se por remover os nodos com menor grau. O **Algoritmo de Decomposição** $\boldsymbol{k-core}$ ***decomposition*** é um algoritmo iterativo que começa por iniciar o valor de k a 0 e permite decompor a rede, sendo bastante **útil para remover os nodos periféricos.**

Após realizarmos a decomposição *core* da rede no *R*, representamos tabelarmente a distribuição das conchas (**Tabela 2**) e graficamente a rede resultante da decomposição, com indicação do core e em tamanho proporcional ao valor de $k - core$. (**Fig. 3**).

Conchas	1	2	3	4	5	6	7	8
Nº de Nodos	382	201	97	47	41	0	0	19

Tabela 2 | Distribuição das conchas formandas na decomposição da rede.

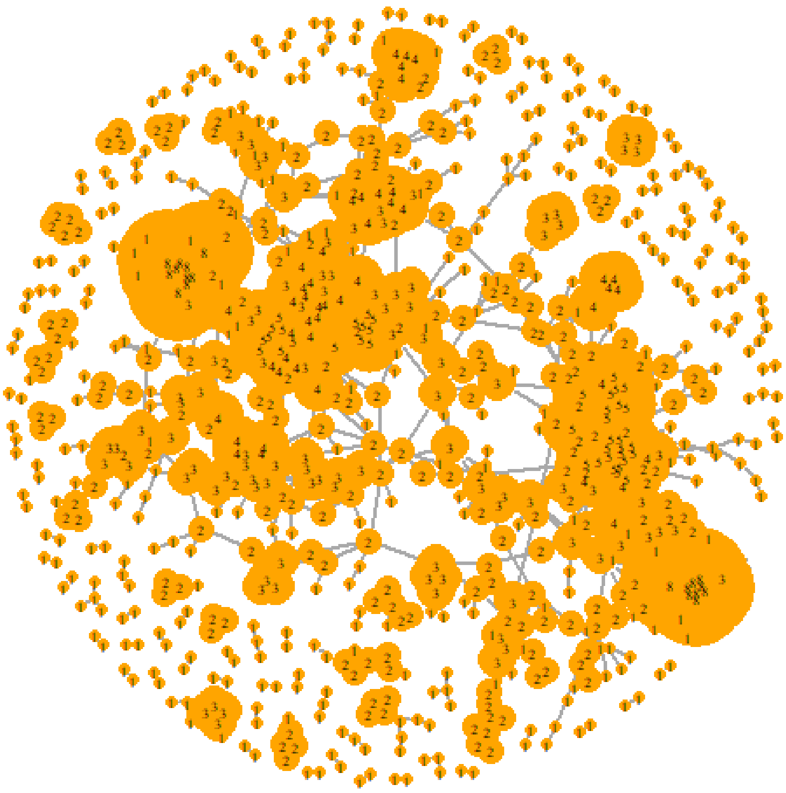


Figura 3 | Representação gráfica da decomposição da rede.

Com a análise dos resultados obtidos, constata-se que existem 8 conchas, pelo que existem diferentes níveis de nuclearidade na rede. Os nodos que pertencem a conchas menores são mais periféricos, existindo em maior número (382 nodos); já os nodos com maior concha são os mais nucleares da rede, existindo apenas 19.

No contexto da rede em questão, os nodos com menor concha podem ser pessoas que vivem sozinhas, que trabalham fora da zona residencial ou que não socializam com os vizinhos.

Quanto aos nodos com maior concha, podem ser pessoas que são muito sociáveis, que vivem em famílias numerosas ou que participam em atividades sociais na zona residencial.

Questão 2

Dimensão e Densidade da Componente Gigante

Replicando todas as definições e considerações realizadas na Questão 1, faremos um resumo mais breve dos resultados obtidos nesta questão, na qual focaremos o nosso estudo à componente gigante, tal como pedido.

O número de **nodos** e de **ligações** da componente gigante (Fig.4) é $N = 496$ e $L = 984$, respetivamente.

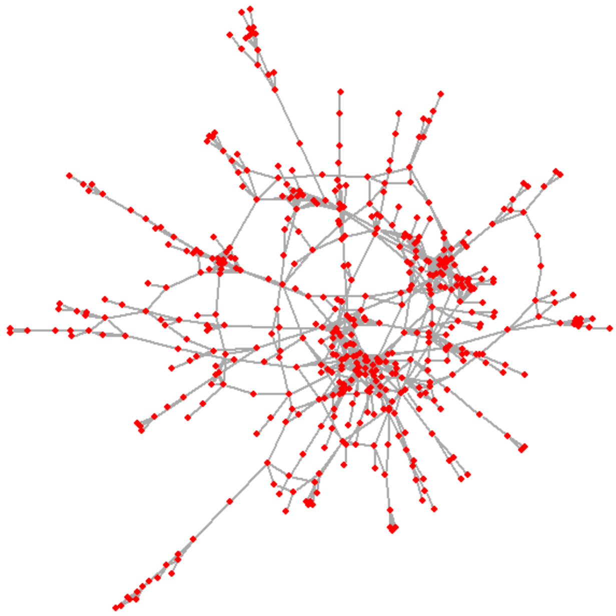


Figura 4 | Representação gráfica da componente gigante.

A **densidade** é de $d = 0.00801564$. Na subrede o número de ligações, L, é 984 e o número máximo de ligações, L_{max} , é de $\frac{N(N-1)}{2} = \frac{496 \times 495}{2} = 122760$.

Como $984 << 122760$ e $d \ << 1$, conclui-se que a rede é **esparsa**, ou seja, não está densamente conectada.

Deste modo, a densidade sugere que os habitantes da zona residencial da componente gigante não têm muitos contactos sociais diretos uns com os outros.

Grau dos Nodos da Subrede

O **grau médio** é de $< k > = 3.967742$, ou seja, em média, cada habitante está conectado a cerca de 4 outros habitantes.

A distribuição de graus é a apresentada na **Tabela 3** e **Figura 5**.

Graus	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
N	116	86	79	60	38	26	20	21	14	12	8	8	3	1	0	2	2

Tabela 3 | Distribuição do grau da componente gigante.

Através da **Tabela 3**, vemos que a distribuição de graus é assimétrica, sendo que a maioria dos habitantes têm baixos graus (1, 2, 3 ou 4), e poucos são os que têm graus elevados, ou seja, a maioria dos habitantes tem um número reduzido de conexões sociais diretas.

Esta observação sugere que a subrede é composta por uma grande maioria de habitantes com um círculo social relativamente pequeno, e por uma minoria de habitantes com um círculo social mais alargado.

Associação de Grau da Subrede

O valor obtido de associação de grau foi de $\rho = 0.345145$, logo habitantes com graus semelhantes, tendem a se conectar. Trata-se de um valor positivo e maior que 0.3, logo é aceite como significativo de que a subrede é **associativa**.

Através do *output* produzido de `knn(giant_component)$knnk` no *R*, verifica-se que a função está a crescer monotonamente, tendo os valores $k_{nn}(1) = 4.91$ e $k_{nn}(17) = 9.26$, demonstrativos desse crescimento. Assim, uma vez mais, conclui-se que esta rede é **associativa**.

Interpretando este facto ao contexto da subrede em estudo, podemos aferir que habitantes com graus semelhantes têm a tendência a se conectar entre si. Isso sugere uma estrutura de subrede em que a popularidade ou centralidade dos nodos influencia as suas conexões, o que é uma característica de rede associativa.

Caminhos Mais Curtos e Diâmetro da Subrede

A média dos comprimentos dos caminhos mais curtos (caminho entre um par de nodos com menor comprimento) obtido na subrede é de $< l > = 7.93344$. Deste modo a distância média **não** é considerada **pequena**, dado que é um valor não próximo de $\log N = \log 496 \approx 2.695$.

Quanto ao diâmetro da subrede, l_{max}, este é o caminho mais longo entre 2 nodos ($\max_{i,j} l_{i,j}$) sendo este **21**.

Triângulos da Subrede

Observa-se que no total existem 2229 triângulos. Na subrede em estudo, o valor do **coeficiente de *Clustering* médio dos nodos, $< C >$** , obtido foi de $< C > \approx 0.44$ De acordo com a interpretação probabilística, este valor refere-se à probabilidade de dois vizinhos de um nodo selecionado aleatoriamente se ligarem um ao outro.

Aplicando à subrede, este coeficiente é de $C \approx 0.42$. Sendo este valor superior a 0.3, podemos indicar que existem **muitos triângulos**.

Heterogeneidade da Subrede

Representado graficamente a distribuição de grau da subrede (**Fig.5**), verifica-se a existência de uma cauda longa ou pesada, revelando que a rede tem **heterogeneidade** nos valores dos graus.

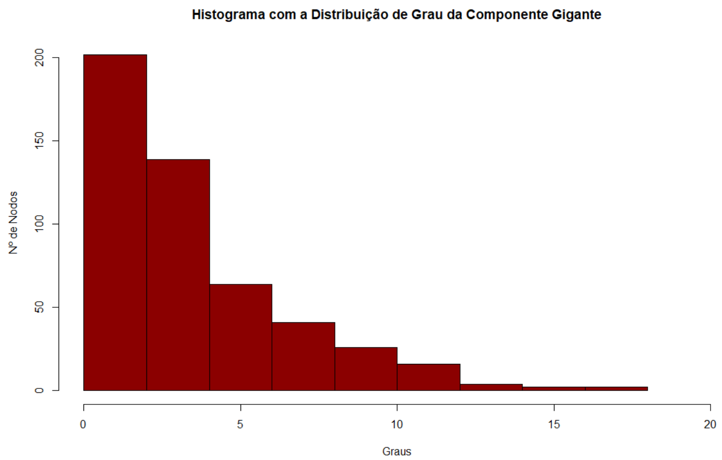


Figura 5 | Representação gráfica da distribuição de graus da subrede.

Para se corroborar esta observação, pode-se considerar o parâmetro de heterogeneidade da subrede, no qual se obteve $\boldsymbol{K} = 1.612086$.

Dado que este valor é bastante superior a 1, a subrede é heterogénea (em grau) e indicia a existência de *hubs*.

Decomposição de *core* da Subrede

Após realizarmos a decomposição *core* da componente gigante no *R*, representamos tabelarmente a distribuição das conchas (Tabela 4) e graficamente a rede resultante da decomposição, com indicação do core e em tamanho proporcional ao valor de $k - core$. (Fig. 6).

Conchas	1	2	3	4	5	6	7	8
Nº de Nodos	149	151	89	47	41	0	0	19

Tabela 4 | Distribuição das conchas formadas na decomposição da subrede.

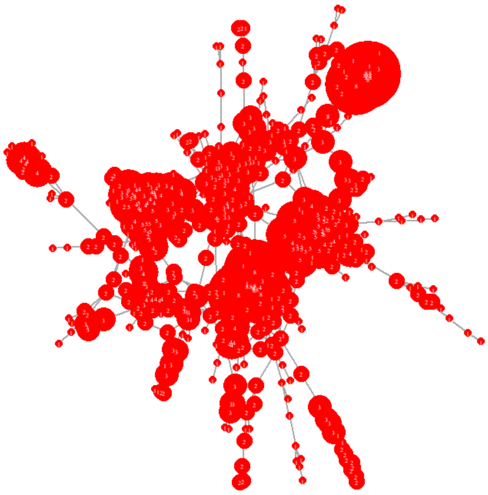


Figura 6 | Representação gráfica da decomposição da rede.

Com a análise dos resultados obtidos, uma vez mais, constata-se que existem 8 conchas, pelo que existem diferentes níveis de nuclearidade na rede. Os nodos que pertencem a conchas menores (1 e 2) são mais periféricos, existindo em maior número (300 nodos); já os nodos com maior concha são os mais nucleares da rede, existindo apenas 19.

A decomposição de *core* da subrede sugere que a rede dos contactos sociais diretos da zona residencial é relativamente dispersa. A maioria dos habitantes tem poucos contactos sociais diretos. No entanto, há um pequeno grupo de pessoas que são muito sociáveis e que têm um papel importante na ligação entre os diferentes residentes.

Questão 3

	Rede Completa	Componente Gigante
Nº de Nodos *(N)*	787	496
Nº de Ligações *(L)*	1197	984
Densidade *(d)*	0.0039	0.008
Grau Médio *(< k>)*	3.0419	3.9677
Coeficiente de Pearson *(p)*	0.4766	0.3451
Distância Média C+c*(< l>)*	7.914	7.933
Diâmetro *(l_{max})*	21	21
Nº de Triângulos	2307	2229
Coeficiente de Cluster médio dos nodos *(< C>)*	0.4412	0.4392
Coeficiente de Cluster da (sub)rede *(C)*	0.4199	0.4198
Heterogeneidade *(K)*	1.84	1.61
Nº de Conchas	8	8

Tabela 5 | Comparação dos resultados da rede e da sua componente gigante.

Comparação entre a Rede e a sua Componente Gigante

Dimensão e Densidade

A rede toda tem $N = 787$ nós, enquanto a componente gigante tem $N = 496$. Isto significa que a componente gigante representa cerca de 63% da rede toda. A rede toda tem $L = 1197$ ligações, enquanto a componente gigante tem $L = 984$. Isto significa que a componente gigante representa cerca de 82% das ligações da rede.

A densidade da rede é de $d = 0.0039$, enquanto a densidade da sua componente gigante é de $d = 0.008$. Isto significa que a componente gigante é ligeiramente mais densa do que a rede, ou seja, representa as pessoas com mais contacto desta zona habitacional, no contexto da rede.

Grau dos Nodos

O grau médio da rede é de $< k > = 3.0419$, enquanto o da sua componente gigante é de $< k > = 3.9677$. Isto significa que os nodos da componente gigante têm, em média, mais ligações do que os nodos da rede, ou seja, os habitantes desta subrede têm mais contactos sociais diretos.

Associação de Grau

O coeficiente de *Pearson* da rede é de $\rho = 0.4766$, enquanto que o da sua componente gigante é de $\rho = 0.3451$, sendo ambas associativas. Isto significa que a componente gigante é ligeiramente menos homogénea do que a rede, ou seja, os nodos com graus mais altos tendem a estar menos ligados uns aos outros.

Caminhos Mais Curtos e Diâmetro

A distância média da rede é de $< l > = 7.914$, enquanto a distância média da sua componente gigante é de $< l > = 7.933$. Isto significa que a distância média entre dois nodos é ligeiramente semelhante na rede e na componente gigante.

O diâmetro da rede é de $l_{max} = 21$, enquanto o diâmetro da sua componente gigante é de $l_{max} = 21$. Isso significa que, em ambas as redes, existe sempre um caminho de, no máximo, 21 ligações entre dois nodos aleatórios.

Triângulos

O número de triângulos da rede é de 2307, enquanto o número de triângulos da sua componente gigante é de 2229. Isso significa que, na componente gigante, existem mais triângulos, o que indica que os nodos estão mais interligados uns aos outros.

O coeficiente de *clustering* da rede é de $C = 0.4199$, enquanto o da componente gigante é de $C = 0.4198$. Isso significa que, em ambos, os nodos estão agrupados de forma similar.

Heterogeneidade

A heterogeneidade da rede é de $K = 1.84$, enquanto a da componente gigante é de $K = 1.61$, havendo heterogeneidade, em ambas, indicando a existência de *hubs*. Porém a diferença de valores significa que, na componente gigante, a distribuição de grau está mais concentrada à volta de um determinado valor, em comparação com a rede.

Decomposição de *core*

O número de conchas é de 8 para a rede e também para a sua componente gigante. Isso significa que, em ambas as redes, existem nodos com graus semelhantes.

Em suma, os resultados indicam que a componente gigante da rede é mais densa, interligada e agrupada do que a rede completa. Isso significa que há uma parte dos habitantes da zona residencial que estão mais conectados uns aos outros, formando uma rede mais compacta e densa.

Tal como esperado, as conclusões tiradas sobre a rede não diferem muito em relação à sua subrede, sendo esta última, a maior parte da rede (cerca de 63%).

Bibliografia

[1] Barabási, A.-L. (2016). Network Science, 1st edition, Cambridge University Press: Cambridge.

[2] Ognyanova, K. (2016). *Introduction to R and network analysis*. https://kateto.net/wp-content/uploads/2018/03/R%20for%20Networks%20Workshop%20-%20Ognyanova%20-%202018.pdf

[3] igraph. (n.d.). *Function reference*. R.igraph.org. https://r.igraph.org/reference/index.html

Apêndice

Trabalho 1 | AR

Rede de Contactos Sociais

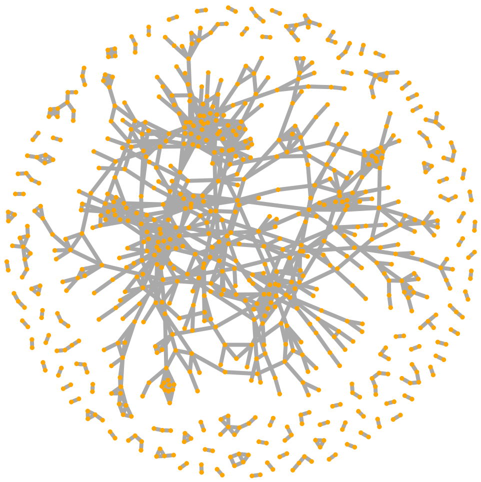
André Silvestre Nº104532 Eliane Gabriel Nº103303
Maria João Lourenço Nº104716 Margarida Pereira Nº105877
Umeima Mahomed Nº 99239

15 de dezembro de 2023

Questão 1

```r
# Importar a rede de um ficheiro .txt
rede <- read_graph("trab_links.txt", format = "edgelist", directed=F)

# Representar graficamente a rede
plot(rede, vertex.color= "orange", vertex.label=NA, vertex.size=2,
     vertex.frame.color=NA, edge.width= 2)
```



Nº de Nodos e de Ligações

```r
cat("A rede tem", vcount(rede) ,'nodos e', ecount(rede), 'ligações.')
```

```
## A rede tem 787 nodos e 1197 ligações.
```

Densidade

```r
edge_density(rede)
```

```
## [1] 0.003870142
```

Graus dos Nodos

```r
# Grau de cada nó
grau <- data.frame("Grau (k_i)"=degree(rede,mode='all'))
grau_transposto <- as.data.frame(t(grau))
colnames(grau_transposto) <- 1:vcount(rede) # Cada coluna é um nodo
grau_transposto
```

```
##             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 26
## Grau..k_i. 9 4 4 7 1 1 1 3 4  6  4  4  3  1  1  2  1  1  3  1  3  2  3  2  2  1
##            27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
## Grau..k_i.  2  1  5  5  1  5  1  1  1  2  6  2  5  2  2  2  1  1  4  2  2  1  1
##            50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## Grau..k_i.  1  3  3  2  2  1  1  1  7  6  1  1  1  8  1  2  2  2  6  3  1  5  1
##            73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
## Grau..k_i.  1  3  1  1  3  1  4  3  4  1  1  5  4  1  1  1  4  2  1  1  5  1  4
##            96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113
## Grau..k_i.  3  1  1  3   1   4   2   2   3  11   2   2   3   1   2   1   1   1
##            114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130
## Grau..k_i.   1   5   4   1   2   1   1   2   1   5   3   3  12   4   2   2   1
##            131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
## Grau..k_i.   1   1   1   1   1   1   1  10   1   6   1   2   1   2   2   1   1
##            148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164
## Grau..k_i.   5   1   1   1   1   2   1   2   1   1   1   1   1   3   2   4   8
##            165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181
## Grau..k_i.   5  12   8   7  10  10  11  10   9   4  17   6   8  12  13   4   3
##            182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
## Grau..k_i.   6   3   8   7  13   9   7   4   4   4   2   1   5   3   8  13   8
##            199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
## Grau..k_i.   6   7  10  17  11   9   9   4   6   4   7   5   1   4   1   5   3
##            216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232
## Grau..k_i.   9   6   6   4   1   2   1   5   3   7  10   5   4   3   5  10   5
##            233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249
## Grau..k_i.   6   5   2   6   6   8   8   8   8   8   9   2   8  10   5   9   7
##            250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266
## Grau..k_i.   1   7   4   5   3   6   6   4   7   2  12   5   6   3  11   6   7
```

```
##             267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283
## Grau..k_i.    2   3   6   3   8  12  10   9   3   2  12   7   2   2   8   4   2
##             284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
## Grau..k_i.   10   7   9  11   8   5   6   2   1   2   5   1   8   4   2  16   1
##             301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317
## Grau..k_i.    3   1   1   1   3   1   3   1   1   4   1   2   2   8   8   5   9
##             318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334
## Grau..k_i.    5   2   9   1   2   2   2   7   2   3   3   1   6   7   3   1   1
##             335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
## Grau..k_i.    1   1   2   4  11   4   5   1   1  10   3   3   4   3   8   3   4
##             352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368
## Grau..k_i.    4   1   1   4  12   2   3   3   2   2   3   1   1   1   2   2   2
##             369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385
## Grau..k_i.   11   2   2  14  12   1   2   3   1   2  16   4   6   6   9   5   1
##             386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402
## Grau..k_i.    1   4   5   1   5   2  11   5   1   1   4   4   5   3   1   1   2
##             403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419
## Grau..k_i.    4   4   2   4   1   2   2   1   4   9   2   2   2   1   2   2  10
##             420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436
## Grau..k_i.    7   1   1   2   5   2   1   1   2   3   3   1   1   1   3   3   3
##             437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453
## Grau..k_i.    3   3   2   5   4   1   1   1   3   7   3   3   1   1   5   1   2
##             454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470
## Grau..k_i.    1   4   4   1   3   3   2   2   6   4   3   5   4   1   7   1   2
##             471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487
## Grau..k_i.    1   1   2   2   2   2   3   2   3   1   1   2   1   1   3   1   1
##             488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504
## Grau..k_i.    5   2   2   6   2   1   2   1   1   1   1   3   3   2   2   4   5
##             505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521
## Grau..k_i.    2   4   3   6   1   1   1   2   4   3   2   1   1   1   2   1   1
##             522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538
## Grau..k_i.    3   3   1   1   2   2   3   1   3   2   2   1   2   2   1   1   2
##             539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555
## Grau..k_i.    1   1   3   1   3   5   1   1   1   2   1   1   2   1   2   1   1
##             556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572
## Grau..k_i.    1   1   2   1   1   1   4   1   1   4   1   1   1   1   1   1   2
##             573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589
## Grau..k_i.    3   2   1   1   2   3   1   3   2   1   1   1   2   1   1   1   4
##             590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606
## Grau..k_i.    2   2   1   2   2   1   2   2   1   3   1   2   3   1   1   3   7
##             607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623
## Grau..k_i.    3   2   3   2   2   1   1   1   1   2   1   1   1   1   1   1   2
##             624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640
## Grau..k_i.    1   3   1   5   2   1   1   4   3   1   2   1   2   1   1   1   2
##             641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657
## Grau..k_i.    1   1   1   2   2   2   3   3   3   3   1   2   2   2   2   1   1
##             658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674
## Grau..k_i.    2   1   1   1   1   1   2   2   8   1   3   1   1   1   2   1   1
##             675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691
## Grau..k_i.    1   1   1   2   1   1   3   1   3   1   1   1   1   2   2   3   1
##             692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708
## Grau..k_i.    3   2   1   2   3   4   1   1   1   2   3   1   2   1   1   1   3
##             709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725
## Grau..k_i.    1   1   1   2   1   1   1   1   1   3   2   1   1   1   1   1   2
```

```
##            726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742
## Grau..k_i.   1   5   3   1   1   1   1   1   3   3   1   1   1   1   1   1   2
##            743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759
## Grau..k_i.   1   1   1   1   1   1   2   1   1   2   1   1   3   1   4   1   3
##            760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776
## Grau..k_i.   3   1   1   4   1   1   4   2   1   1   1   1   4   1   2   4   1
##            777 778 779 780 781 782 783 784 785 786 787
## Grau..k_i.   1   1   1   3   1   4   1   3   1   1   1
```

Grau Médio

```r
# Grau médio
cat("<k> =",mean(degree(rede,mode='all')))
```

```
## <k> = 3.041931
```

Distribuição dos Graus

```r
# Distribuição de frequências do grau
knitr::kable(t(table(factor(degree(rede,mode='all'), levels = 1:max(degree(rede,mode='all'))))))
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
309	156	101	63	41	26	20	21	14	12	8	8	3	1	0	2	2

Extra - Representação dos nodos proporcional ao grau

```r
# Representação dos nodos proporcional ao grau
plot(rede, vertex.size=degree(rede,mode="all")*2, vertex.color= "orange",
     vertex.label=NA, vertex.frame.color=NA, edge.width= 2)
```

Rede Conexa ou Não Conexa?

```
# Verificar se a rede é conexa
is_connected <- is.connected(rede)
cat("A rede é conexa?", is_connected, "\n")
```

```
## A rede é conexa? FALSE
```

```
if (!is_connected) {
  # Se a rede não for conexa, calcular o nº de componentes conexas e as dim. máx e mín
  components <- clusters(rede)
  num_components <- components$no
  min_size <- min(components$csize)
  max_size <- max(components$csize)
  cat("Número de componentes conexas:", num_components, "\n")
  cat("Dimensão mínima das componentes conexas:", min_size, "\n")
  cat("Dimensão máxima das componentes conexas:", max_size, "\n")
}
```

```
## Número de componentes conexas: 104
## Dimensão mínima das componentes conexas: 2
## Dimensão máxima das componentes conexas: 496
```

Associação de Grau

```r
# Coeficiente de Assortatividade - Coeficiente de correlação de Pearson para os graus
dos nodos adjacentes.
assortativity_degree(rede)
```

```
## [1] 0.4765607
```

```r
# Determina os valores da função knn - Média do grau dos vizinhos
knn(rede)$knnk
```

```
##  [1] 2.763754 3.429487 4.435644 5.067460 5.580488 6.089744 6.764286 7.946429
##  [9] 7.746032 8.275000 7.943182 8.270833 8.230769 7.857143      NaN 7.906250
## [17] 9.264706
```

Média dos Comprimentos dos Caminhos + Curtos

```r
# Calcular a média dos caminhos mais curtos pela fórmula
caminhos_mais_curto <- distances(rede, algorithm = 'dijkstra')
soma_inversos <- sum(1 / caminhos_mais_curto[caminhos_mais_curto > 0])
((2 * soma_inversos) / (vcount(rede) * (vcount(rede) - 1)))^(-1)
```

```
## [1] 7.995324
```

```r
# Média da Distância - Ignora as distâncias de caminhos inexistentes.
mean_distance(rede)
```

```
## [1] 7.914034
```

Diâmetro da Rede

```r
diameter(rede)
```

```
## [1] 21
```

Estudo dos Triângulos da Rede

```r
# Determinar o nº de triângulos da rede
sum(count_triangles(rede))
```

```
## [1] 2307
```

```r
# Determinar a média do coeficiente de clustering para nodos com grau > 1
mean(transitivity(rede, type = "local")[(degree(rede) > 1)], na.rm = TRUE)
```

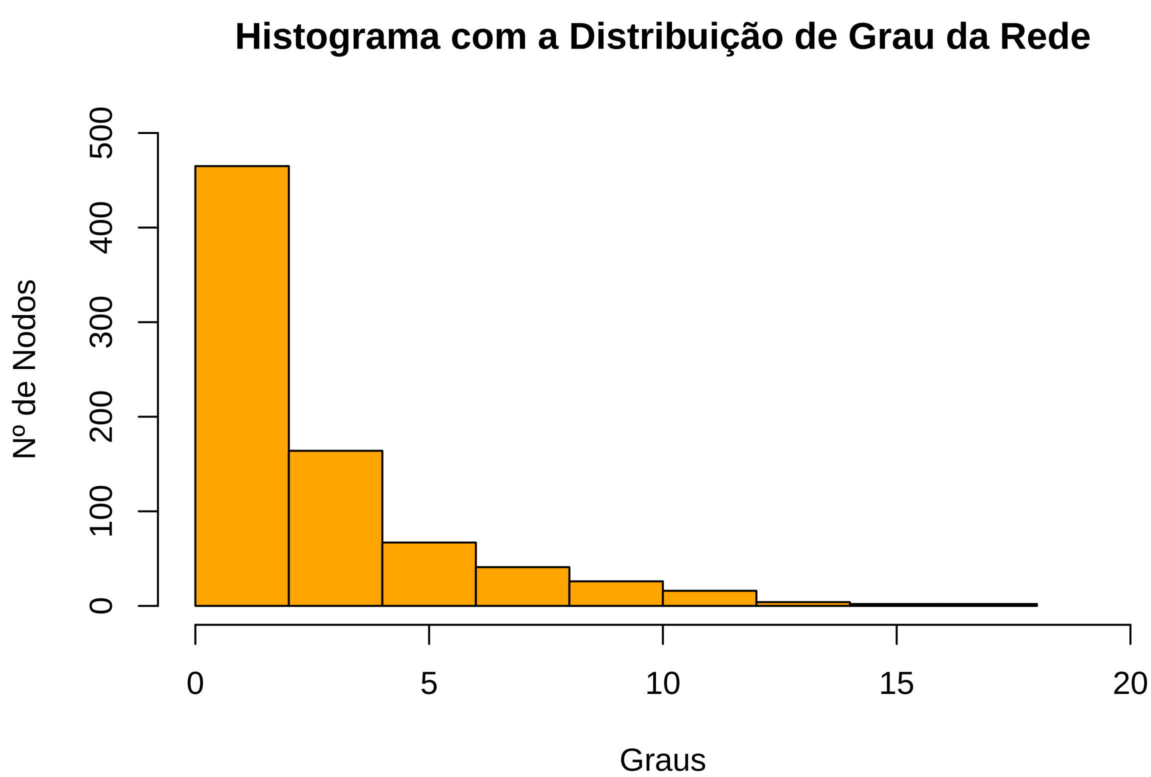
```
## [1] 0.4412442
```

```r
# Determinar o coeficiente de clustering da rede
transitivity(rede,type="global")
```

```
## [1] 0.4199126
```

Parâmetro de Heterogenidade

```r
deg <- degree(rede,mode="total")
hist(deg, col = 'orange',
     main = 'Histograma com a Distribuição de Grau da Rede',
     xlab = 'Graus',
     ylab = 'Nº de Nodos',
     ylim = c(0,500),
     xlim= c(0,20))
```




```r
ht <- mean(deg*deg)/(mean(deg)^2)
ht
```

```
## [1] 1.837585
```

Estudo dos ***Hubs*** da Rede

```r
# closeness(rede)        # Centralidade de proximidade

# centr_clo(rede)        # Normalização

# betweenness(rede)      # Centralidade

# edge.betweenness(rede) # Intermediação de Ligações
```

Decomposição da Rede (Core Decomposition)

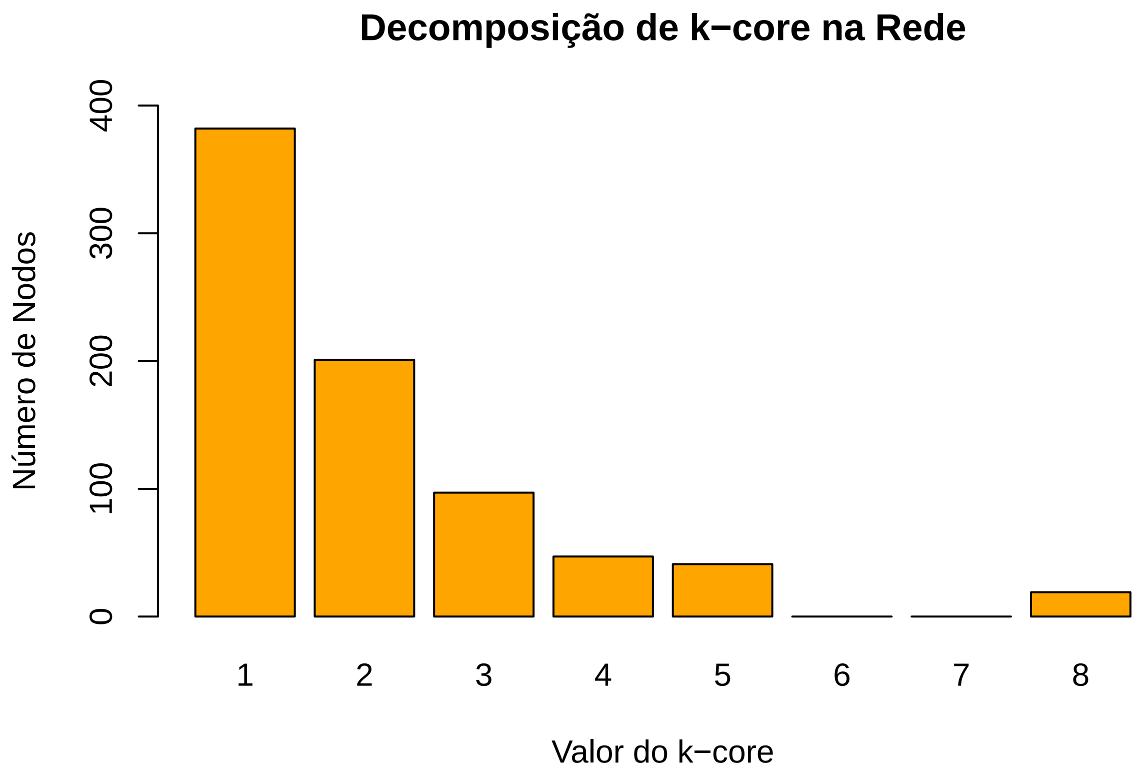
```r
# Decomposição de core da rede
coreness <- coreness(rede, mode="all")     # Escreve num vector o valor do k-core de cada nodo.
cat("Conchas (shells) na rede:")
```

```
## Conchas (shells) na rede:
```

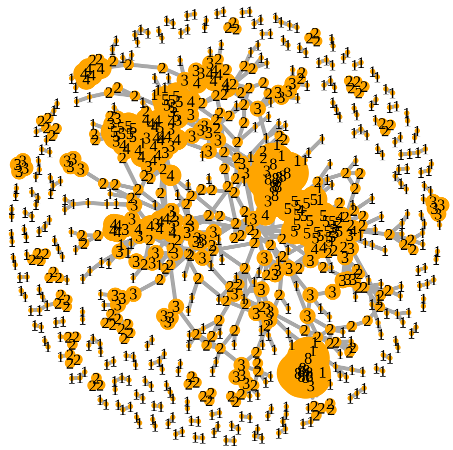
```r
knitr::kable(t(table(factor(coreness, levels = 1:8))))
```

1	2	3	4	5	6	7	8
382	201	97	47	41	0	0	19

```r
# Representação gráfica do vector kc
barplot(table(factor(coreness, levels = 1:8)),
        col = 'orange',
        main = "Decomposição de k-core na Rede",
        xlab = "Valor do k-core",
        ylab = "Número de Nodos",
        ylim = c(0,400))
```




```r
# Representa os nodos com indicação do core e em tamanho proporcional ao valor de kc.
plot(rede,
     vertex.size=coreness*10/4,
     vertex.label=coreness,
     vertex.label.color=c("black"),
     vertex.label.cex = 0.5,
     vertex.color= "orange",
     vertex.frame.color=NA,
     edge.width= 2)
```



10

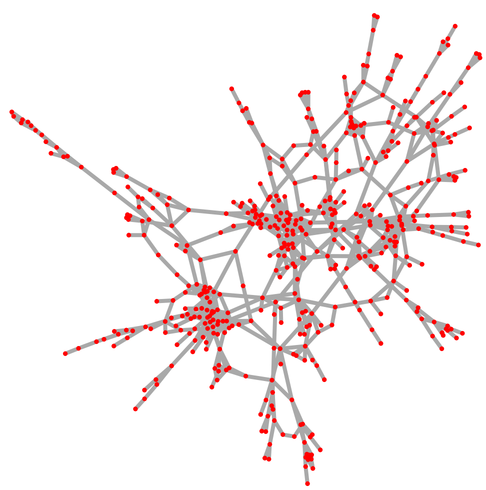
Questão 2

Componente Gigante

```
# components(rede) # A componente gigante é a 1ª do csize

# Encontrar a componente gigante da rede
giant_component <- induced.subgraph(rede, which(components$membership == which.max(components$csize)))

# Representar graficamente a componente gigante da rede
plot(giant_component,
     vertex.color= "red",
     vertex.label=NA,
     vertex.size=2,
     vertex.frame.color=NA,
     edge.width= 2)
```



Nº de Nodos e de Ligações da Subrede Componente Gigante da Rede

```
cat("Número de nós na componente gigante:", vcount(giant_component), "\n")
```

```
## Número de nós na componente gigante: 496
```

```r
cat("Número de ligações na componente gigante:", ecount(giant_component), "\n")
```

```
## Número de ligações na componente gigante: 984
```

Densidade da Componente Gigante

```r
edge_density(giant_component)
```

```
## [1] 0.00801564
```

Graus dos Nodos da Componente Gigante

```r
# Grau de cada nó
grau_cg <- data.frame("Grau (k_i)"=degree(giant_component,mode='all'))
grau_transposto_cg <- as.data.frame(t(grau_cg))
colnames(grau_transposto_cg) <- 1:vcount(giant_component) # Cada coluna é um nodo
grau_transposto_cg
```

```
##             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 26
## Grau..k_i. 9 4 4 7 1 3 4 6 4  4  3  1  1  2  3  1  5  5  1  5  1  2  6  2  5  4
##            27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
## Grau..k_i.  2  2  1  2  1  1  7  6  1  8  6  3  1  5  3  3  1  4  3  4  1  1  4
##            50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## Grau..k_i.  2  5  1  3  1  4  2  2  3 11  2  2  3  1  2  5  4  2  1  5 12  4  2
##            73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
## Grau..k_i.  2  1  1  1  1 10  1  6  1  2  5  1  1  1  1  2  1  3  2  4  8  5 12
##            96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113
## Grau..k_i.  8  7 10 10  11  10   9   4  17   6   8  12  13   4   3   6   3   8
##            114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130
## Grau..k_i.   7  13   9   7   4   4   4   2   1   5   3   8  13   8   6   7  10
##            131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147
## Grau..k_i.  17  11   9   9   4   6   4   7   5   1   4   1   5   3   9   6   6
##            148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164
## Grau..k_i.   4   1   2   1   5   3   7  10   5   4   3   5  10   5   6   5   2
##            165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181
## Grau..k_i.   6   6   8   8   8   8   8   9   2   8  10   5   9   7   1   7   4
##            182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
## Grau..k_i.   5   3   6   6   4   7   2  12   5   6   3  11   6   7   2   3   6
##            199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
## Grau..k_i.   3   8  12  10   9   3   2  12   7   2   2   8   4   2  10   7   9
##            216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232
## Grau..k_i.  11   8   5   6   5   1   8   4   2  16   1   3   1   3   1   3   4
##            233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249
## Grau..k_i.   1   2   2   8   8   5   9   5   2   9   2   2   7   2   3   3   1
##            250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266
## Grau..k_i.   6   7   3   1   1   2   4  11   4   5   1   1  10   3   3   4   3
##            267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283
## Grau..k_i.   8   3   4   4   1   1   4  12   2   3   3   2   2   3   1  11   2
```

```
##              284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
## Grau..k_i.     2  14  12   1   2  16   4   6   6   9   5   4   5   1  11   5   1
##              301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317
## Grau..k_i.     1   4   4   5   3   2   4   4   2   4   1   2   2   1   4   9   2
##              318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334
## Grau..k_i.     1  10   7   3   3   3   3   3   2   5   4   1   7   3   3   5   1
##              335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
## Grau..k_i.     2   1   4   4   1   3   3   2   2   6   4   3   5   4   1   7   1
##              352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368
## Grau..k_i.     2   1   2   2   2   3   1   3   5   6   2   3   3   2   2   4   5
##              369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385
## Grau..k_i.     2   4   3   6   1   2   4   3   2   1   2   3   3   3   1   3   5
##              386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402
## Grau..k_i.     2   1   2   2   1   4   1   1   1   3   2   2   3   1   3   2   4
##              403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419
## Grau..k_i.     2   2   1   2   2   1   3   3   1   1   3   7   3   2   2   1   1
##              420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436
## Grau..k_i.     3   1   5   1   4   3   1   2   1   2   1   1   1   1   2   8   3
##              437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453
## Grau..k_i.     1   2   1   1   2   1   3   1   1   1   3   2   3   4   1   1   1
##              454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470
## Grau..k_i.     2   3   2   1   1   1   3   2   1   5   3   3   3   2   1   1   1
##              471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487
## Grau..k_i.     1   3   1   4   3   3   4   1   4   2   1   1   4   1   2   4   1
##              488 489 490 491 492 493 494 495 496
## Grau..k_i.     1   1   3   1   4   1   3   1   1
```

Grau Médio da Componente Gigante

```r
# Grau médio
cat("<k> =",mean(degree(giant_component,mode='all')))
```

```
## <k> = 3.967742
```

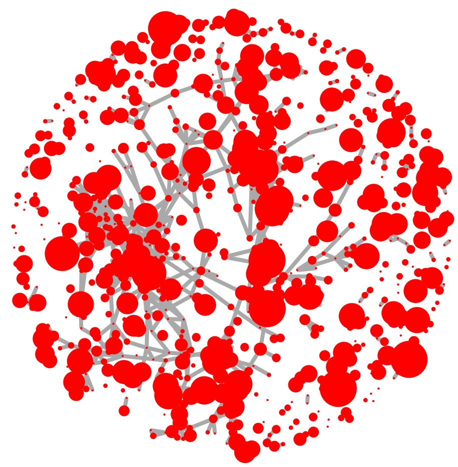
Distribuição dos Graus da Componente Gigante

```r
# Distribuição de frequências do grau.
knitr::kable(t(table(factor(degree(giant_component,mode='all'),
                            levels = 1:max(degree(giant_component,mode='all'))))))
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
116	86	79	60	38	26	20	21	14	12	8	8	3	1	0	2	2

Extra - Representação dos nodos proporcional ao grau da Componente Gigante

```r
# Representação dos nodos proporcional ao grau
plot(rede,
     vertex.size=degree(giant_component,mode="all")*1,
     vertex.color= "red",
     vertex.label=NA,
     vertex.frame.color=NA,
     edge.width= 2)
```



Associação de Grau da Componente Gigante

```r
# Coeficiente de Assortatividade - Coeficiente de correlação de Pearson para os graus
dos nodos adjacentes.
assortativity_degree(giant_component)
```

```
## [1] 0.345145
```

```r
#  Determina os valores da função knn - Média do grau dos vizinhos
knn(giant_component)$knnk
```

```
##  [1] 4.913793 4.616279 4.953586 5.195833 5.842105 6.089744 6.764286 7.946429
##  [9] 7.746032 8.275000 7.943182 8.270833 8.230769 7.857143      NaN 7.906250
## [17] 9.264706
```

Média dos Comprimentos dos Caminhos + Curtos da Componente Gigante

```
# Média da Distância - Ignora as distâncias de caminhos inexistentes.
mean_distance(giant_component)
```

```
## [1] 7.933447
```

Diâmetro da rede da Componente Gigante

```
diameter(giant_component)
```

```
## [1] 21
```

Estudo dos Triângulos da rede

```
# Determinar o nº de triângulos da componente gigante
sum(count_triangles(giant_component))
```

```
## [1] 2229
```

```
# Determinar a média do coeficiente de clustering para nodos com grau > 1 da componente gigante
mean(transitivity(giant_component, type = "local")[(degree(giant_component) > 1)], na.rm = TRUE)
```

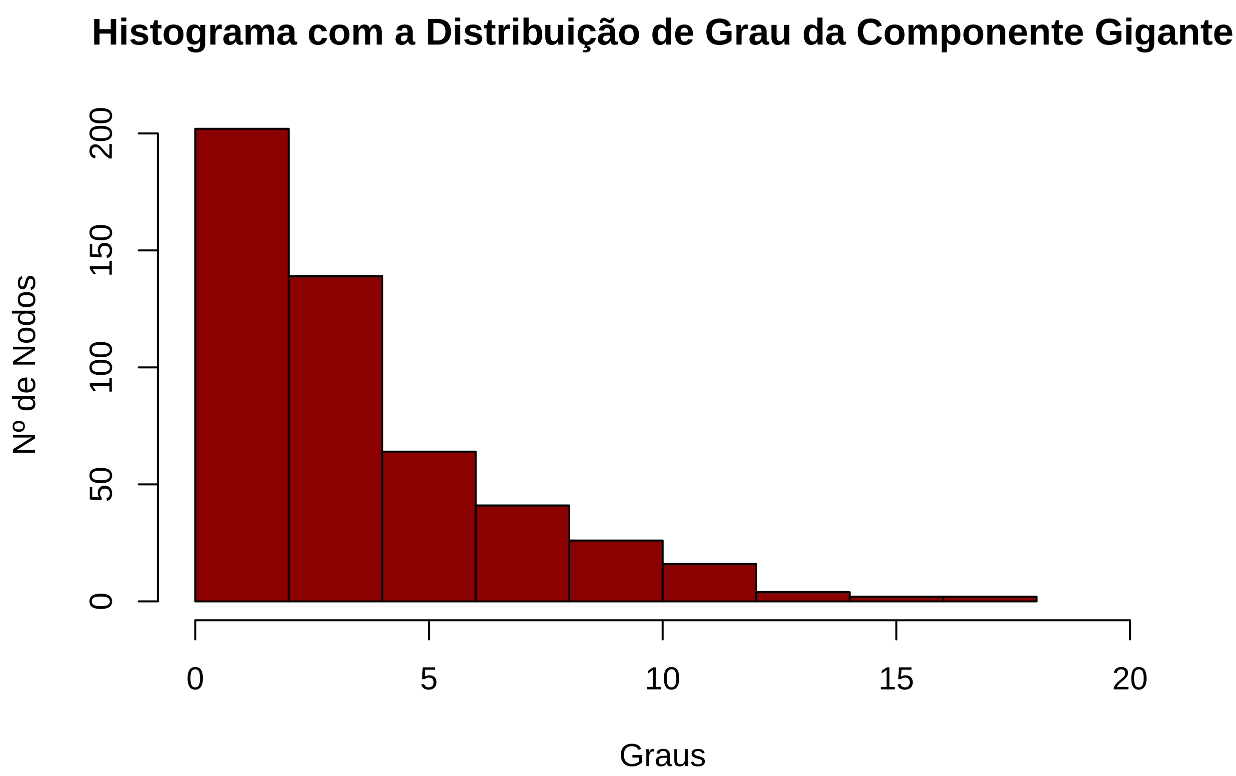
```
## [1] 0.4391616
```

```
# Determinar o coeficiente de clustering da componente gigante
transitivity(giant_component,type="global")
```

```
## [1] 0.419774
```

Parâmetro de Heterogenidade da Componente Gigante

```
deg_giant <- degree(giant_component, mode="total")
hist(deg_giant,
     col = 'darkred',
     main = 'Histograma com a Distribuição de Grau da Componente Gigante',
     xlab = 'Graus',
     ylab = 'Nº de Nodos',
     xlim = c(0,20))
```




```
ht_giant <- mean(deg_giant*deg_giant)/(mean(deg_giant)^2)
ht_giant
```

```
## [1] 1.612086
```

Estudo dos *Hubs* da Componente Gigante

```
# closeness(giant_component)        # Centralidade de proximidade

# centr_clo(giant_component)        # Normalização

# betweenness(giant_component)      # Centralidade

# edge.betweenness(giant_component) # Intermediação de Ligações
```

Decomposição (*Core Decomposition*) da Componente Gigante

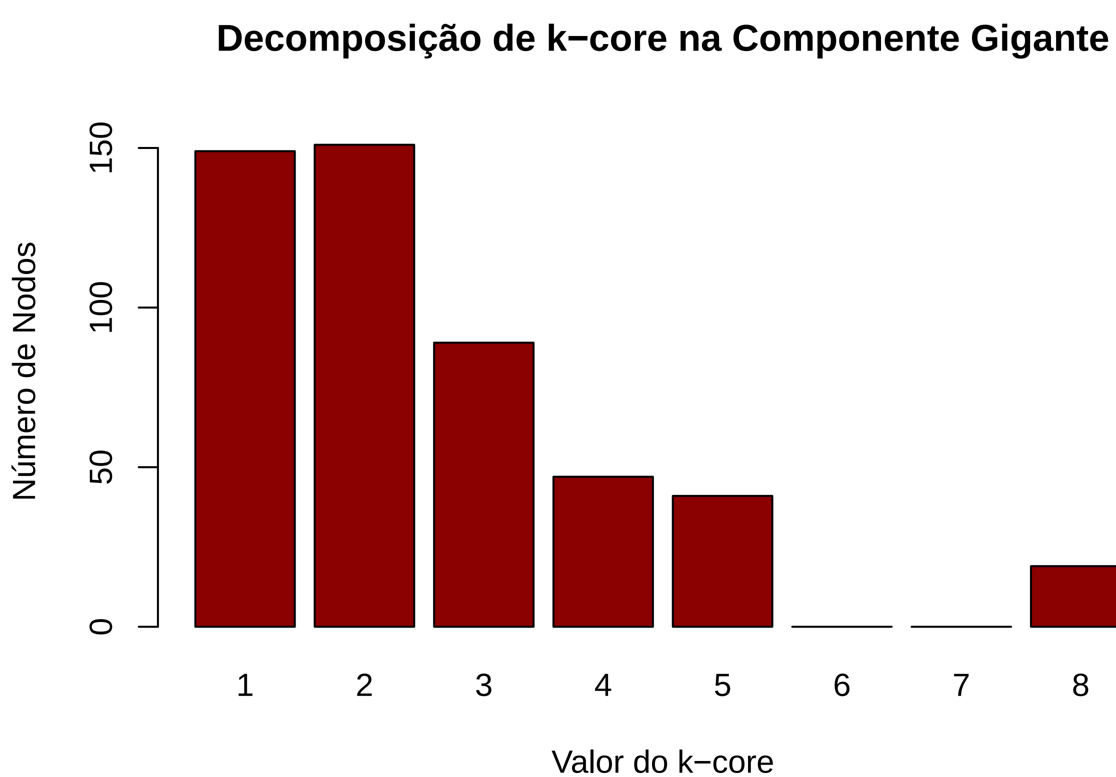
```
# Decomposição de core da componente gigante
coreness_giant <- coreness(giant_component, mode="all") # Escreve num vector o valor do k-core de cada
cat("Conchas (shells) na componente gigante:")
```

```
## Conchas (shells) na componente gigante:
```

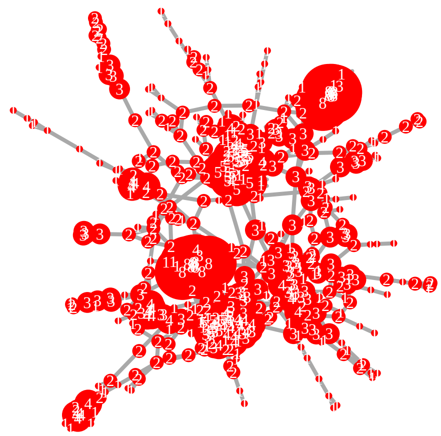
```r
knitr::kable(t(table(factor(coreness_giant, levels = 1:max(coreness_giant)))))
```

1	2	3	4	5	6	7	8
149	151	89	47	41	0	0	19

```r
# Representação gráfica do vector kc da Componente Gigante
barplot(table(factor(coreness_giant, levels = 1:max(coreness_giant))),
        col = 'darkred',
        main = "Decomposição de k-core na Componente Gigante",
        xlab = "Valor do k-core",
        ylab = "Número de Nodos",
        ylim=c(0,160))
```




```r
# Representa os nodos com indicação do core e em tamanho proporcional ao valor de kc.
plot(giant_component,
     vertex.size=coreness_giant*10/3,
     vertex.label=coreness_giant,
     vertex.label.color=c("white"),
     vertex.label.cex = 0.5,
     vertex.color= "red",
     vertex.frame.color=NA, edge.width= 2)
```



Questão 3

```r
# Q3: Comparação entre a rede completa e a componente gigante

# Criar um dataframe comparativo dos resultados
df <- data.frame(
  Nodos = c('**Nº de Nodos *(***$N$ ***)***', vcount(rede), vcount(giant_component)),
  Ligacoes = c('**Nº de Ligações *(***$L$***)***', ecount(rede), ecount(giant_component)),
  Densidade = c('**Densidade *(***$d$ ***)***', round(edge_density(rede), 4),
              round(edge_density(giant_component), 4)),
  Grau_Medio = c('**Grau Médio *(***$<k>$***)***',
               round(mean(degree(rede, mode = "all")), 4),
               round(mean(degree(giant_component, mode = "all")), 4)),
  Coeficiente_de_Pearson = c('**Coeficiente de Pearson *(***$p$***)***',
                           round(assortativity_degree(rede), 4),
                           round(assortativity_degree(giant_component), 4)),
  Media_Distancia = c('**Distância Média C+c*(***$<l>$***)***',
                    round(mean_distance(rede), 3),
                    round(mean_distance(giant_component), 3)),
  Diametro = c('**Diâmetro *(***$l_{max}$***)***',
             diameter(rede), diameter(giant_component)),
  Triangulos = c('**Nº de Triângulos**', sum(count_triangles(rede)),
               sum(count_triangles(giant_component))),

  # Coeficiente de Cluster médio dos nodos (<C>)
  Coef_custering_medio = c('**Coeficiente de Cluster médio dos nodos *(***$<C>$***)***',
                         round(mean(
                           transitivity(rede,
                                        type = "local")[(degree(rede) > 1)],
                           na.rm = TRUE), 4),
                         round(mean(
                           transitivity(giant_component,
                                        type = "local")[(degree(giant_component) > 1)],
                           na.rm = TRUE), 4)),

  # Coeficiente de Cluster da rede (C)
  Coef_custering_rede = c('**Coeficiente de Cluster da (sub)rede *(***$C$***)***',
                          round(transitivity(rede, type="global"), 4),
                          round(transitivity(giant_component, type="global"), 4)),
  Heterogeneidade = c('**Heterogeneidade *(***$K$***)***', round(ht, 2), round(ht_giant, 2)),
  N_Conchas = c('**Nº de Conchas**', max(coreness(rede, mode="all")),
              max(coreness(giant_component, mode="all")))

)

# Criar a tabela flextable
ftable <- flextable(as.data.frame(t(df))) %>% colformat_md()

# Personalizar a tabela
set_flextable_defaults(fonts_ignore=TRUE)
ftable <- border_remove(x = ftable) %>%
  hline(i= 1, part = "header", border = fp_border(color = "gray", width = 2)) %>%
  hline_bottom(part = "body", border = fp_border(color = "grey", width = 1)) %>%
```

```
  vline(j=1:2, border = fp_border(color = "white", width = 5)) %>%
  align(j= 2:3, align = "center", part = "all") %>%
  bg(j = 2, bg = "#ED7D31", part = "header") %>%
  bg(j = 3, bg = "darkred", part = "header") %>%
  color(j = 2:3, color = "white", part = "header") %>%
  set_header_labels(V1 = "", V2 = "Rede Completa", V3 = "Componente Gigante") %>%
  bold(bold = TRUE, part = "header") %>%
  autofit()
ftable
```