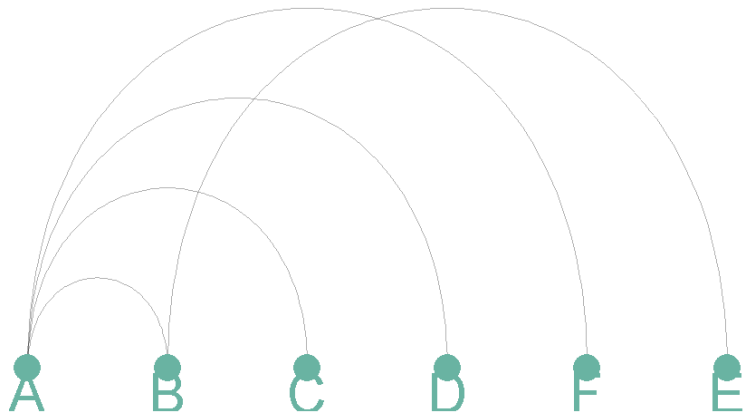


# Diagrama de Arcos

# O que é um Diagrama de Arco ?

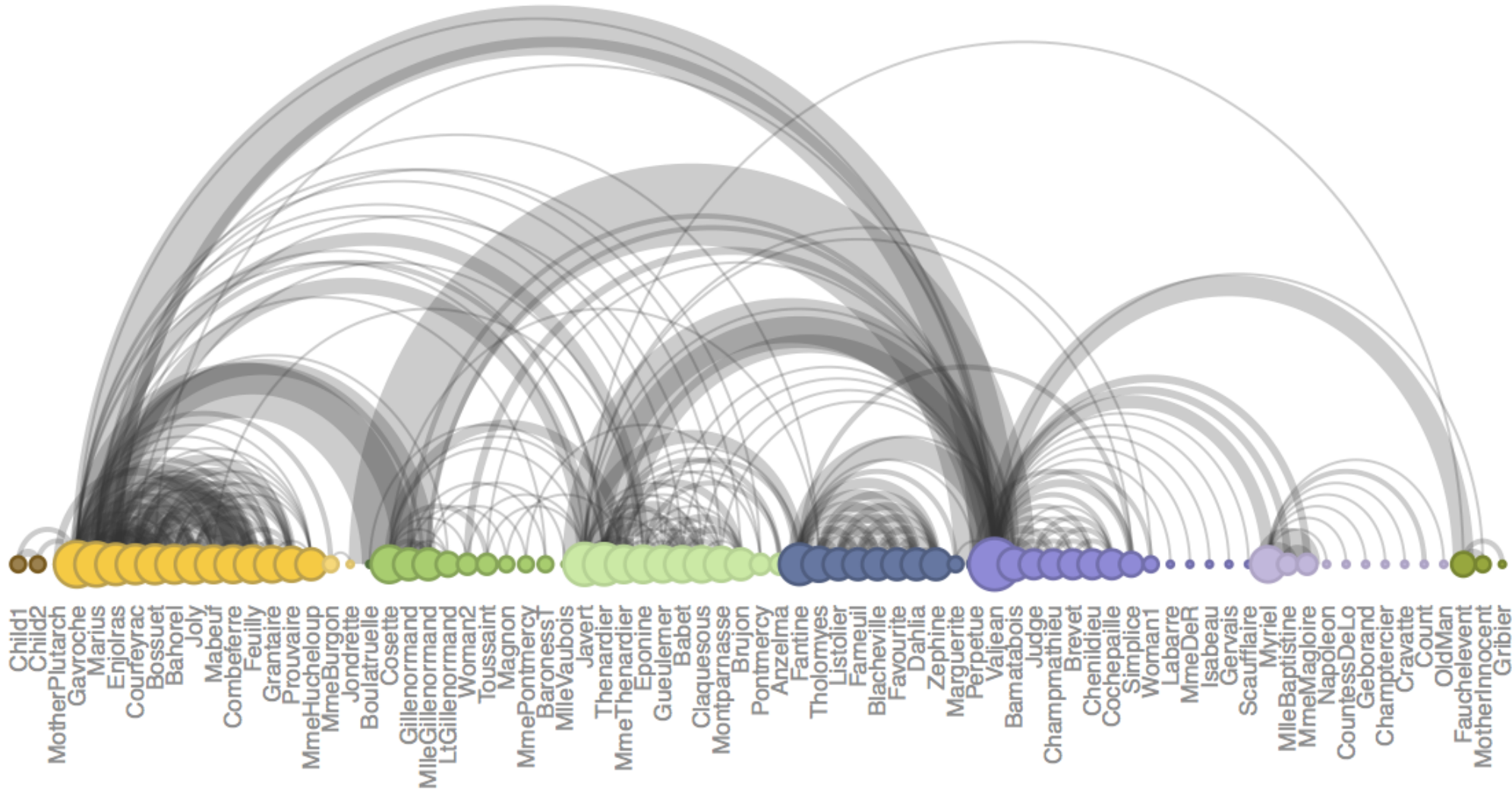
Um diagrama de arco é um tipo especial de gráfico de rede. É uma representação gráfica unidimensional para visualizar conexões entre um conjunto de entidades. A ideia principal é exibir os nós ao longo de um eixo, enquanto as conexões entre os nós são representadas com arcos.



# Gráficos interessantes

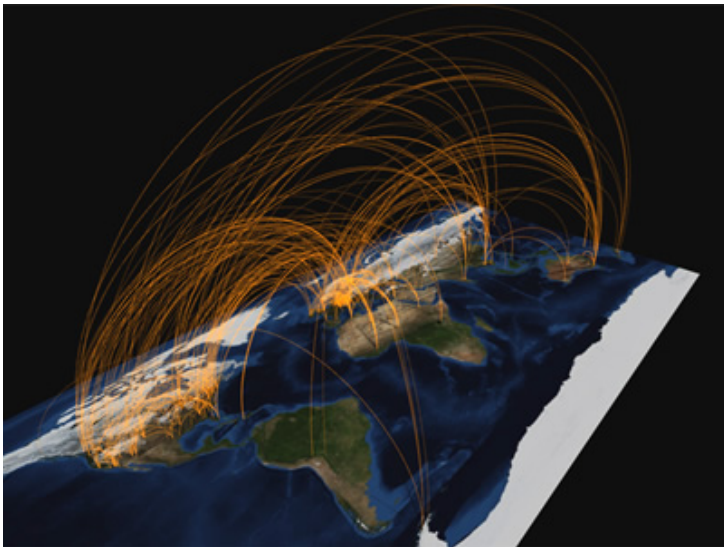
## Les Miserables

Gaston criou o diagrama de arco abaixo para visualizar os personagens do clássico de Victor Hugo, que se tornou um filme, Les Miserables. Cada personagem é conectado por um arco se eles aparecerem juntos no mesmo capítulo; quanto maior o arco, mais os personagens apareceram nos capítulos juntos. A ordem (e a cor) dos nós identificam grupos de caracteres que aparecem juntos no romance.



# Mapa de conversas do Twitter

Esta visualização foi criada por Walter Rafelsberger, essa visualização mostra as conversas de cerca de 1500 usuários do serviço de microblog Twitter. Os arcos basicamente ligam os locais dos usuários que conversam entre si. A geocodificação foi feita filtrando as informações de localização das páginas de perfil dos usuários e pesquisando-as com nomes geográficos.



# Pacote Arcdiagram

O Arcdiagram é um pacote para plotar diagramas básicos de arco em R.

## Instalação do pacote arcdiagram

```
#É preciso usar o pacote devtools para instalar o arcdiagram no R  
install.packages("devtools")  
  
#Carregando devtools  
library(devtools)  
  
#Fazendo download 'arcdiagram' usando 'install_github'  
remotes::install_github("gastonstat/arcdiagram")  
  
#load arcdiagram  
library(arcdiagram)
```

# Uso básico do pacote Arcdiagram

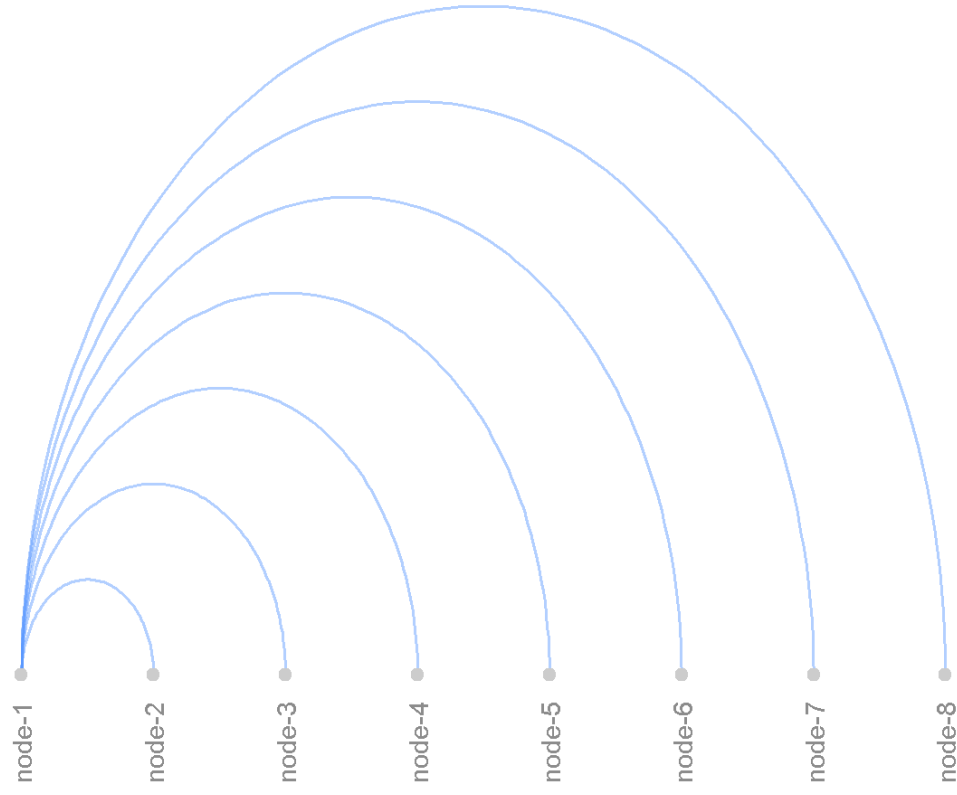
A estrutura é simples e consiste em uma função principal `arcplot()`, e uma função acessória `xynodes()`. `Arcplot` funciona da seguinte maneira, pega um objeto “edgelist” e plota as arestas como arcos. Edgelist é apenas uma matriz de duas colunas que fornece uma lista de arestas para um gráfico. `Arcplot` funciona da seguinte maneira, pega um objeto “edgelist” e plota as arestas como arcos.

```
library(arcDiagram)
# Cria um gráfico com 8 nós
star_graph = graph.star(8, mode = "out")

#Adiciona nome aos nós
V(star_graph)$name = paste("node", 1:vcount(star_graph), sep = "-")

# Extrair edgelist
star_edges = get.edgelist(star_graph)

# plot arc diagram
arcplot(star_edges)
```



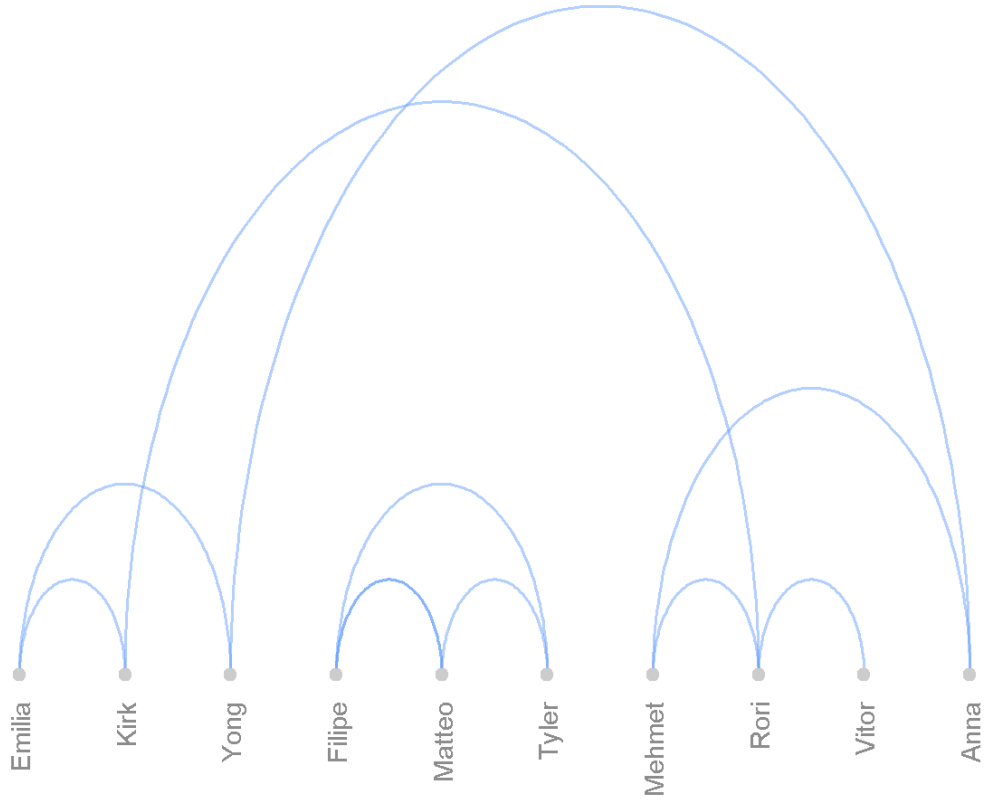


# Exemplo 2

Vamos criar uma edgelist representando projetos entre colegas de trabalho:

## Criando uma edgelist

```
lab = rbind(c("Emilia", "Kirk"), c("Emilia", "Yong"), c("Filipe", "Matteo"),  
c("Filipe", "Tyler"), c("Matteo", "Filipe"), c("Matteo", "Tyler"), c("Mehmet",  
"Rori"), c("Rori", "Kirk"), c("Rori", "Vitor"), c("Anna", "Mehmet"),  
c("Anna", "Yong"))  
  
arcplot(lab)
```



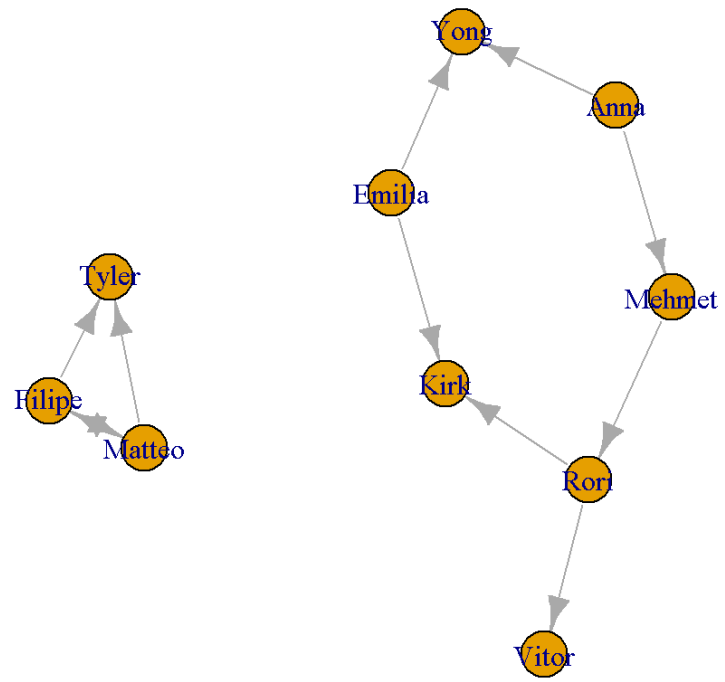
# Como ficou a edgelist

```
##      [,1]      [,2]  
## [1,] "Emilia" "Kirk"  
## [2,] "Emilia" "Yong"  
## [3,] "Filipe" "Matteo"  
## [4,] "Filipe" "Tyler"  
## [5,] "Matteo" "Filipe"  
## [6,] "Matteo" "Tyler"  
## [7,] "Mehmet" "Rori"  
## [8,] "Rori"   "Kirk"  
## [9,] "Rori"   "Vitor"  
## [10,] "Anna"  "Mehmet"  
## [11,] "Anna"  "Yong"
```

# Criando um gráfico

Se quisermos plotar nosso gráfico usando o pacote `igraph()`, primeiro precisamos criar um objeto “graph” com a função `graph.edgelist()` e, em seguida, podemos usar o método `plot()` padrão:

```
# Gráfico da edgelist  
glab = graph.edgelist(lab, directed = TRUE)  
# plot gráfico  
plot(glab)
```



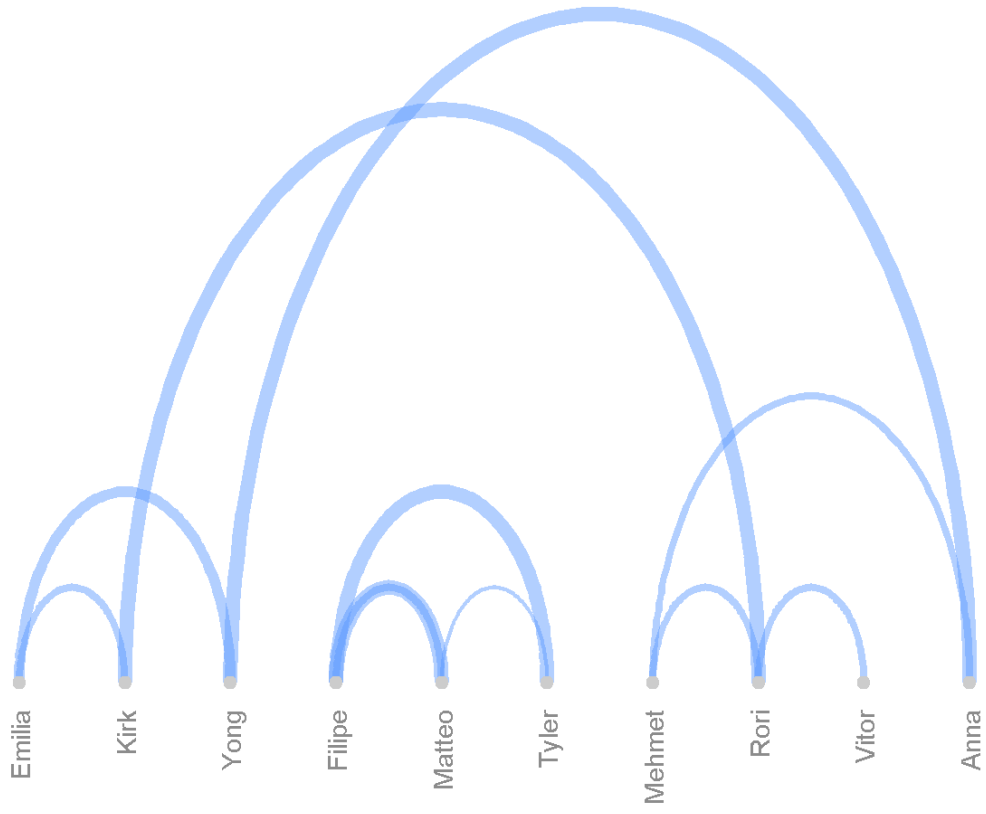
**Podemos levar em consideração o grau dos nós - o grau de um nó é o número de arestas conectadas, para isso usaremos degrees():**

```
lab_degree = degree(glab)
```

```
## Emilia   Kirk   Yong Filipe Matteo Tyler Mehmet   Rori Vitor   Anna  
##         2     2     2     3     3     2     2     3     1     2
```

**Digamos que queremos colocar pesos nas arestas para que elas reflitam algum tipo de valor. Nós podemos fazer isso atribuindo alguns números aleatórios:**

```
arcplot(lab, lwd.arcs = 1)
```





# Também podemos obter clusters dos nós usando a função `clusters()`:

```
gclus = clusters(glab)
```

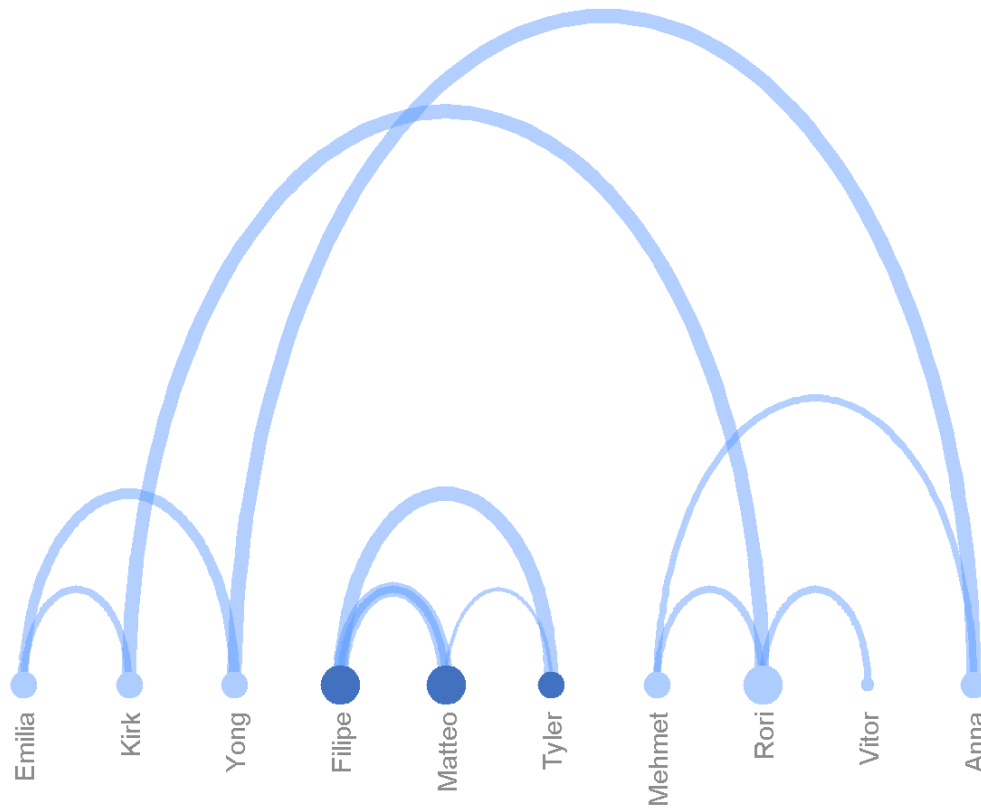
```
## $membership
## Emilia   Kirk   Yong Filipe Matteo Tyler Mehmet   Rori   Vitor   Anna
##       1       1       1       2       2       2       1       1       1
##
## $csize
## [1] 7 3
##
## $no
## [1] 2
```

# Aplicando cores que reflitam as associações de cluster:

```
# Definindo dois tons de azul  
blues = c("#adccff", "#4272bf")  
# Vetor de cores com base na associação dos clusters  
cols = blues[gclus$membership]
```

# Aplicando a função arcplot()

```
arcplot(lab, lwd.arcs = 1, cex.nodes = lab_degree,  
col.nodes = cols)
```

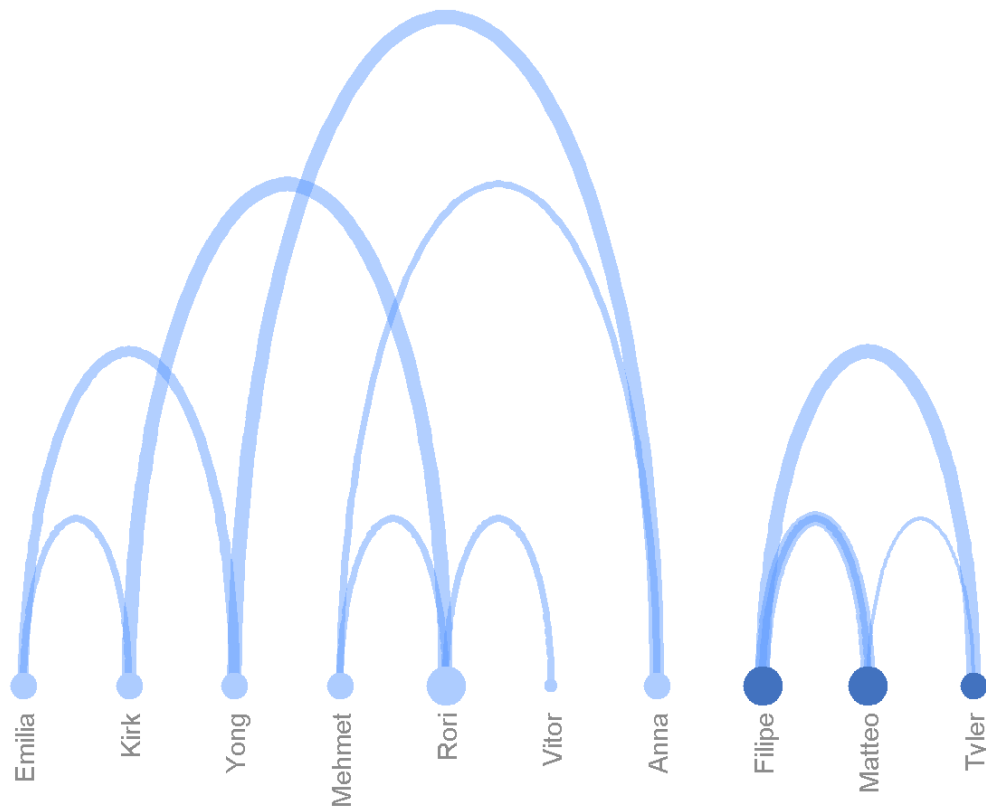


*lwd.arcs*: largura das linhas dos arcos

*cex.nodes*: tamanho dos símbolos dos nós

# Vamos ordenar os nós de acordo com os clusters:

```
arcplot(lab, lwd.arcs = 1, col.nodes = cols,  
        cex.nodes = lab_degree, bg.nodes = cols, ordering = order(gclus$membership))
```



*col.nodes*: cor dos símbolos dos nós

*bg.nodes*: cor de preenchimento para os símbolos dos nós

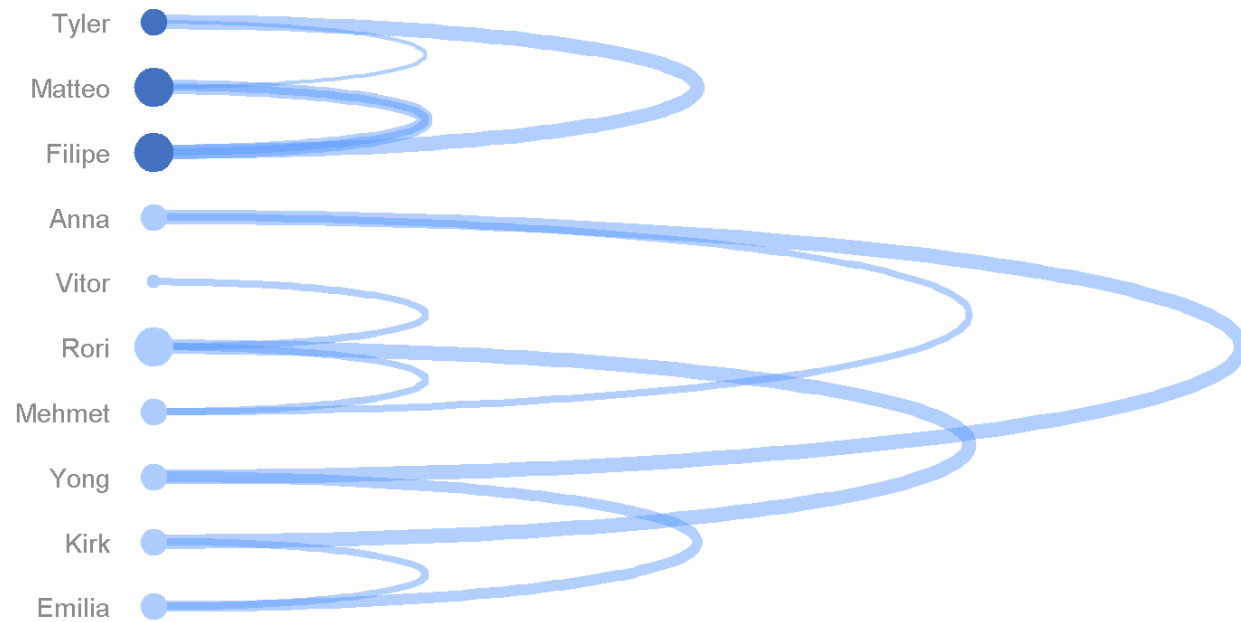
*ordering*: vetor numérico opcional que fornece a ordem dos nós



# Vamos mudar o diagrama para uma orientação vertical:

```
arcplot(lab, lwd.arcs = 1,  
cex.nodes = lab_degree, col.nodes = cols, bg.nodes = cols,  
show.nodes = TRUE, ordering = order(gclus$membership), horizontal = FALSE)
```





*show.nodes*: indica se deseja mostrar símbolo nos nós

*horizontal*: indica se deseja traçar na orientação horizontal

# Desvantagens

Os arcos são sensíveis à ordem dos nós. Quando os nós relacionados estão próximos uns dos outros, tem-se uma melhor visualização, o que ajuda a detectar clusters ou grupo de nós.