

Simulação - Grafo de Erdos-Renyi

Guilherme Yukio Iasunaga

2023-11-27

```
##### SIMULACAO GRAFO DE ERDOS-RENYI #####
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
library(igraph)
```

```
##
```

```
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:dplyr':
```

```
##
```

```
##      as_data_frame, groups, union
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      decompose, spectrum
```

```
## The following object is masked from 'package:base':
```

```
##
```

```
##      union
```

```
simulacao_erdos_renyi <- function(vert_n, lambda) {
```

```
  p <- lambda/(vert_n-1)
```

```

matriz <- c() # criacao para a representação em matriz

set.seed(11218195) # escolho uma semente para os numeros aleatorios

for (i in c(1:vert_n)) {

  # rodo uma bernoulli(p) para cada combinação dois a dois de arestas, para  $v > v'$ ,
  # para decidir se eles vão ter ligacao
  tem_aresta_i <- c(rep(x = 0, times = i), rbinom(n = vert_n-i, size = 1, prob = p))

  # concateno as linhas na matriz
  matriz <- cbind(matriz, tem_aresta_i)

}

# transformo a matriz num data frame e renomeio as colunas
matriz <- matriz %>% data.frame() %>% `colnames<-`(c(1:vert_n))

# pego os veretices que possuem uma aresta entre elas
pares_com_aresta <- which(matriz == 1, arr.ind = TRUE) %>% data.frame()

# coloco num vetor as arestas de forma que seja possivel plotar o grafo com a funcao graph
vetor_pares <- c()

for (i in c(1:length(pares_com_aresta[,1]))) {

  vetor_pares <- append(append(vetor_pares, pares_com_aresta[i, 1]), pares_com_aresta[i, 2])

}

# criação do grafo
grafo <- graph(edges=vetor_pares, n=vert_n, directed=FALSE)

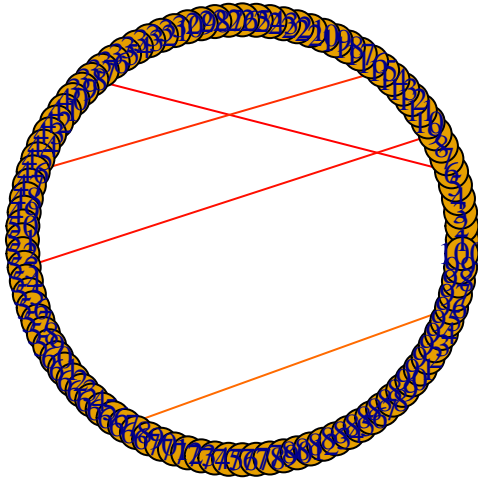
# Atribuindo cores às arestas
cores_arestas <- rainbow(vert_n)
E(grafo)$color <- cores_arestas

return(list(
  M_matriz = matriz + t(matriz),
  grau_vertices = colSums(matriz + t(matriz)) %>% data.frame() %>% `colnames<-`(c('Grau')),
  G_grafo = grafo
))
}

simulacao1 <- simulacao_erdos_renyi(vert_n = 100, lambda = 0.1)

# plota o grafico
plot(simulacao1$G_grafo, layout=layout_in_circle)

```



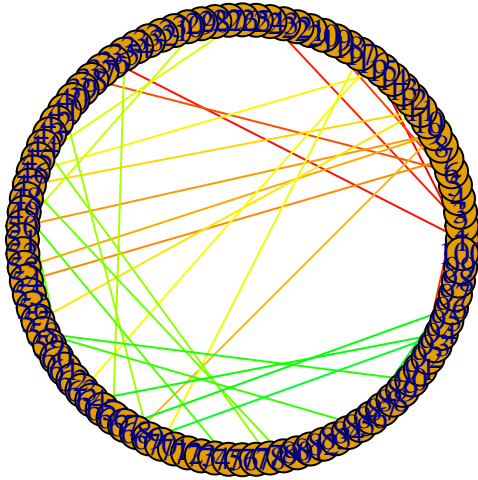
```
print(paste0("Número de arestas presentes na simulação com N = 100 e Lambda = 2.5: ", (simulacao1$grau_
```

```
## [1] "Número de arestas presentes na simulação com N = 100 e Lambda = 2.5: 8"
```

```
simulacao2 <- simulacao_erdos_renyi(vert_n = 100, lambda = 0.7)
```

```
# plota o grafico
```

```
plot(simulacao2$G_grafo, layout=layout_in_circle)
```



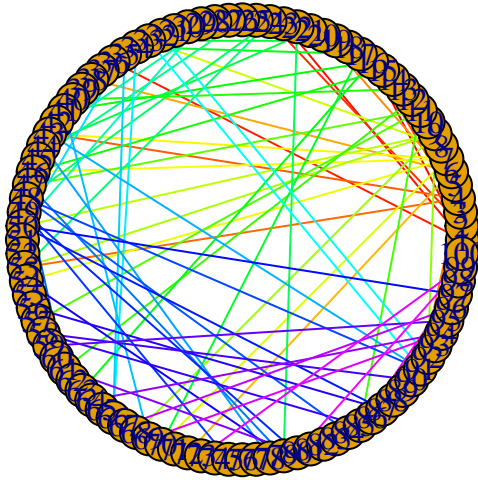
```
print(paste0("Número de arestas presentes na simulação com N = 100 e Lambda = 2.5: ", (simulacao2$grau_
```

```
## [1] "Número de arestas presentes na simulação com N = 100 e Lambda = 2.5: 40"
```

```
simulacao3 <- simulacao_erdos_renyi(vert_n = 100, lambda = 1.5)
```

```
# plota o grafico
```

```
plot(simulacao3$G_grafo, layout=layout_in_circle)
```



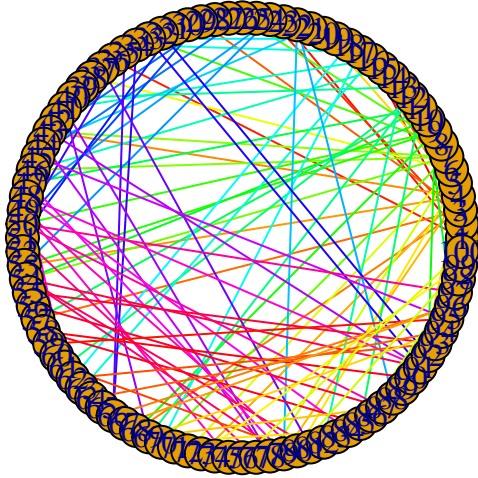
```
print(paste0("Número de arestas presentes na simulação com N = 100 e Lambda = 2.5: ", (simulacao3$grau_
```

```
## [1] "Número de arestas presentes na simulação com N = 100 e Lambda = 2.5: 88"
```

```
simulacao4 <- simulacao_erdos_renyi(vert_n = 100, lambda = 2.5)
```

```
# plota o grafico
```

```
plot(simulacao4$G_grafo, layout=layout_in_circle)
```



```
print(paste0("Número de arestas presentes na simulação com N = 100 e Lambda = 2.5: ", (simulacao4$grau_
```

```
## [1] "Número de arestas presentes na simulação com N = 100 e Lambda = 2.5: 125"
```

Pelos grafos representados graficamente, podemos observar que a medida que o valor de λ aumenta, o número de arestas no grafo também aumenta.

OBS: Devido ao número de vértices, algumas arestas ficam ocultas graficamente.