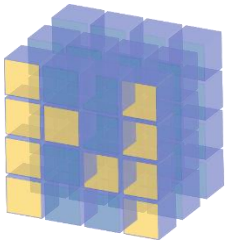
NETWORKX

Carlos J. Costa

Redes

- O estudo de redes é algo particularmente relevante.
- Redes Sociais
- Migrações
- Viagens
- Erasmus
- Investimentos
- Criminalidade

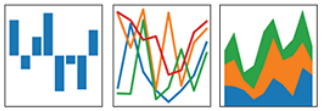
Importar

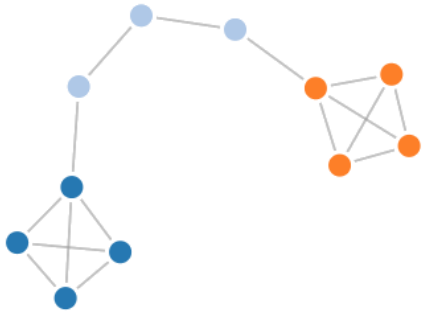


NumPy

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$





NetworkX

```
import numpy as np
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
```

matplotlib

Criar redes

```
# criar uma rede sem elementos
G = nx.Graph()
# adicionar um nó
G.add_node('Mary’)
```

Criar redes

```
# adicionar nós a partir de uma lista
G.add_nodes_from(['Mary', 'Steven', 'Alice','John’])
```

Criar redes

```
# remove node
G. remove_node('Mary')
```

```
# remove varios nós
G.remove_nodes_from(['Mary', 'Steven'])
```

Criar redes

```
# vê nós
G.nodes
```

Criar redes

```
# adiciona vários arcos (lista de tuplos)
G.add_edges_from([('Mary', 'Steven') ,
 ('Mary', 'Alice'),('Mary','John'),('Mary','Edward')])
```

Criar Redes

```
# ver os arcos de uma rede G
G.edges
```

Criar Redes

```
# adicionar arcos
# arcos são tuplos de nós (origem, destino)
# também adiciona nós se eles ainda não existirem
G.add_edge('Mary','Steven')
```

Criar Redes

- # remover arcos
- G.remove_edge('Mary','Alice')

Criar Redes

- # obter o número de nós na rede G
- G.number_of_nodes()

Criar Redes

```
# obter o número de arcos na rede G
G.number_of_edges()
```

Criar Redes

```
# obter a vizinhança de Alice
# (obtem-se um dicionario)

neighbors = G.neighbors('Alice')
print (neighbors)
```

Criar Redes

```
# obter o número de vizinhos
G.degree('Alice')
```

Criar Redes

- # gravar rede
- nx.write_edgelist(G, "parte1")

Criar Redes

```
# eliminar conteúdo da rede
G.clear()
```

Criar Redes

```
# ler dados de ficheiro para variavel de rede
G = nx.read_edgelist("parte1")
```

Criar Redes

```
#desenho simplificado
nx.draw(G)

#Outra alternativa
nx.draw(G, with_labels=True)
```

Analisar rede

- G = nx.read_edgelist("parte1")
- G.edges

Analisar rede

```
# atribuir pesos aos arcos

G.add_edge('Mary','Steven', weight=500)
G.add_edge('John','Mary', weight=10)
G.add_edge('Mary','Alice', weight=200)
```

Analisar rede

```
nx.draw(G, with_labels=True)
```

```
nx.draw(G, pos=None, arrows=True, with_labels=True)
```

Analisar rede

- # Acesso aos pesos do arcos
- G['Mary']['Steven']

Analisar rede

```
# alterar peso de arco
G['Mary']['Steven']['weight'] = 6
```

Analisar rede

#criar nova rede desta vez direcionada

dg = nx.DiGraph()

Analisar rede

```
# pode-se criar uma representação não
# direccionada da rede G

nx.to_undirected(G)
```

Analisar rede

```
# pode-se crirar uma representação
# direccionada da rede G

nx.to_directed(G)
```

Analisar rede

```
# multigraphs pode guardar diversos informações com diferentes
# propriedades sobre os mesmos arcos

MG = nx.MultiGraph()
MG.add_weighted_edges_from([(1, 2, 3.0), (1, 2, 75), (2, 3, 5)])
```

Analisar rede

```
# mostrar arcos sem pesos
MG.edges
```

Analisar rede

```
# mostrar dados de arcos com pesos
MG.edges.data('weight', default=1)
```

Analisar rede

```
# verificar o peso de um arco
MG[1][2]
```

Analisar rede

```
#gravar arcos com pesos
nx.write_weighted_edgelist(G,"parte2")
```

Modelos de redes

- import numpy as np
- import pandas as pd
- import networkx as nx
- import matplotlib.pyplot as plt

Modelos de redes

```
# Rede Barabasi-Albert (scale-free)
ba = nx.barabasi_albert_graph(50, 3)
nx.draw_spectral(ba)
```

Modelos de redes

```
# Erdos-Renyi (random) network
er = nx.erdos_renyi_graph(50, 0.1)
nx.draw_circular(er)
```

Modelos de redes

```
# Rede Watts-Strogatz (small-world)
ws = nx.watts_strogatz_graph(50, 6, 0.2)
nx.draw_circular(ws)
```

Modelos de redes

```
# random geometric graph (RGG)
rgg = nx.random_geometric_graph(200,0.125)
nx.draw(rgg)
```

Modelos de redes

```
# Grafico complete
# todos os pares de nós estão ligados com um
# unico arco

complete = nx.complete_graph(6)
nx.draw(complete)
```

Bibliografia

- https://pandas.pydata.org/
- https://pandas.pydata.org/pandas-docs/stable/getting_started/10min.html
- https://scikit-learn.org/
- https://scikit-learn.org/stable/index.html
- https://www.statsmodels.org/stable/index.html
- https://networkx.github.io/