UFRN UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

IMD INSTITUTO METRÓPOLE DIGITAL

Big Data - Foundations and Applications

Lesson #7 - Network Analysis I

Ivanovitch Silva
July, 2017

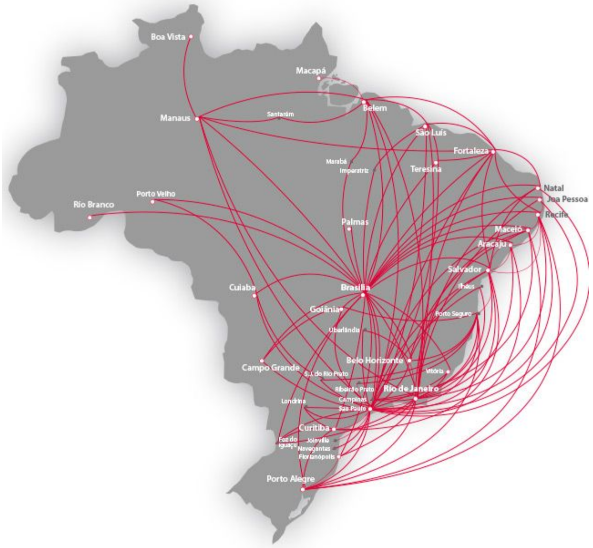
Agenda

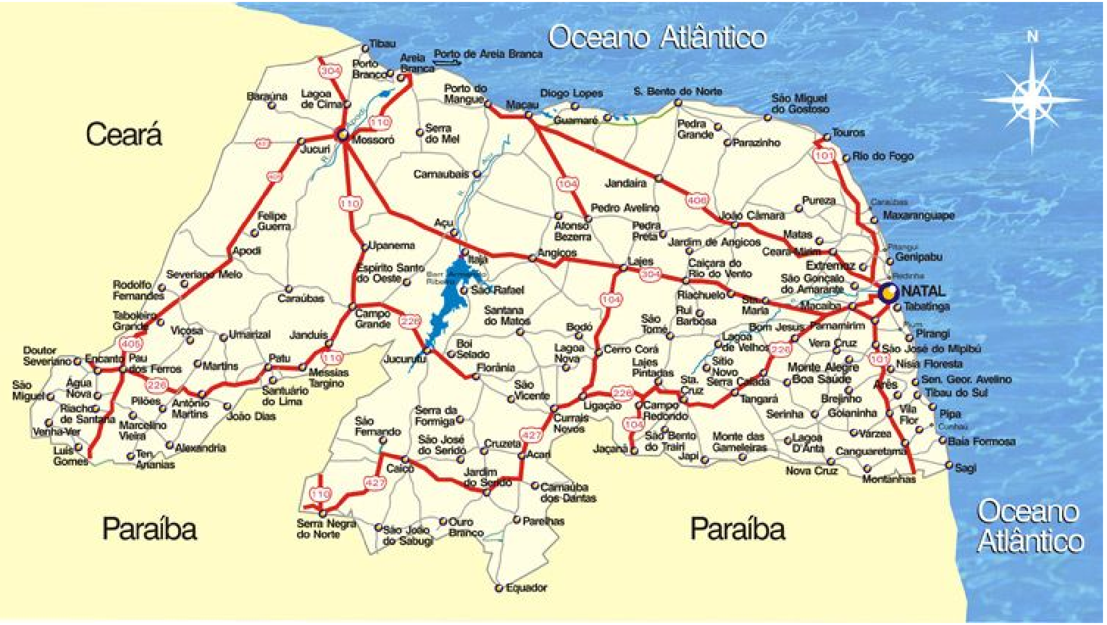
- Introduction about network analysis
- What is a network
- API
- Creating a network
- Adding nodes and edges
- Visualizing network

Previously on last class (...)

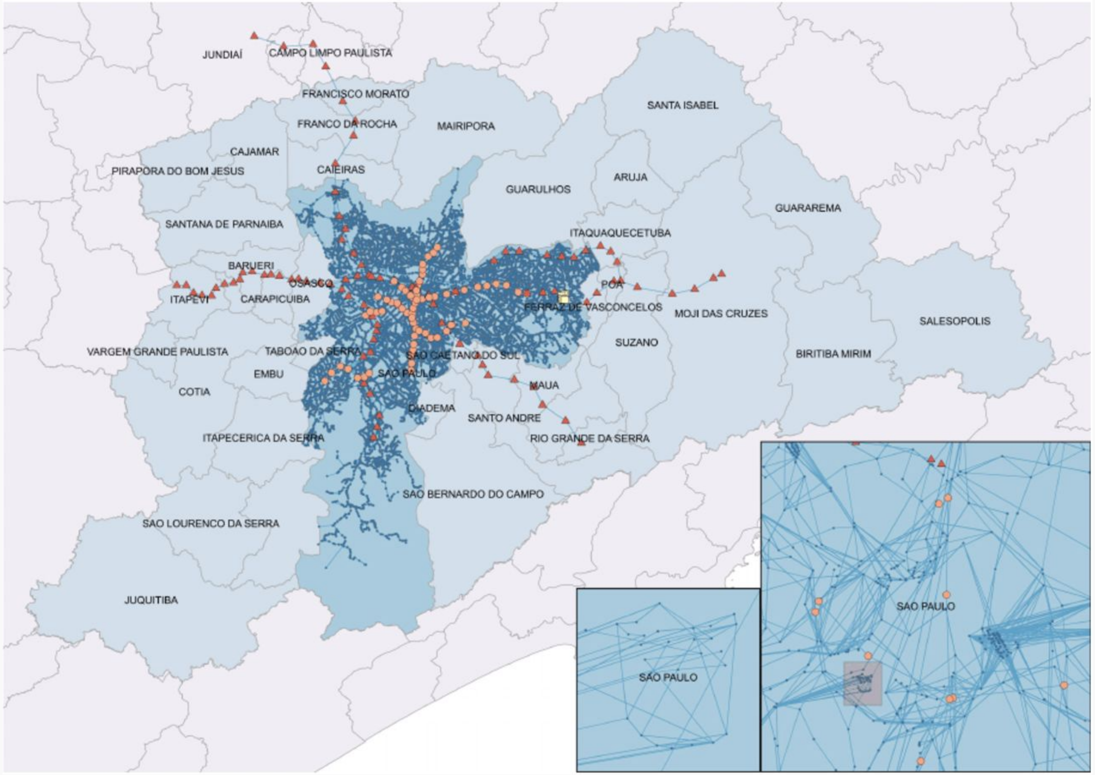
What is a network?

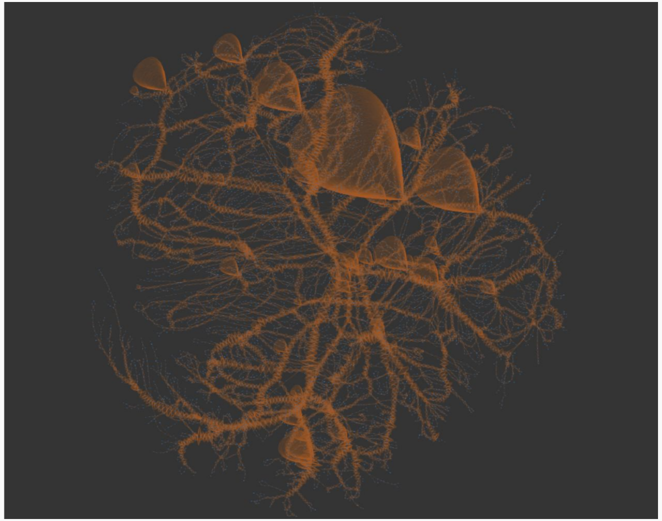
What is a network?





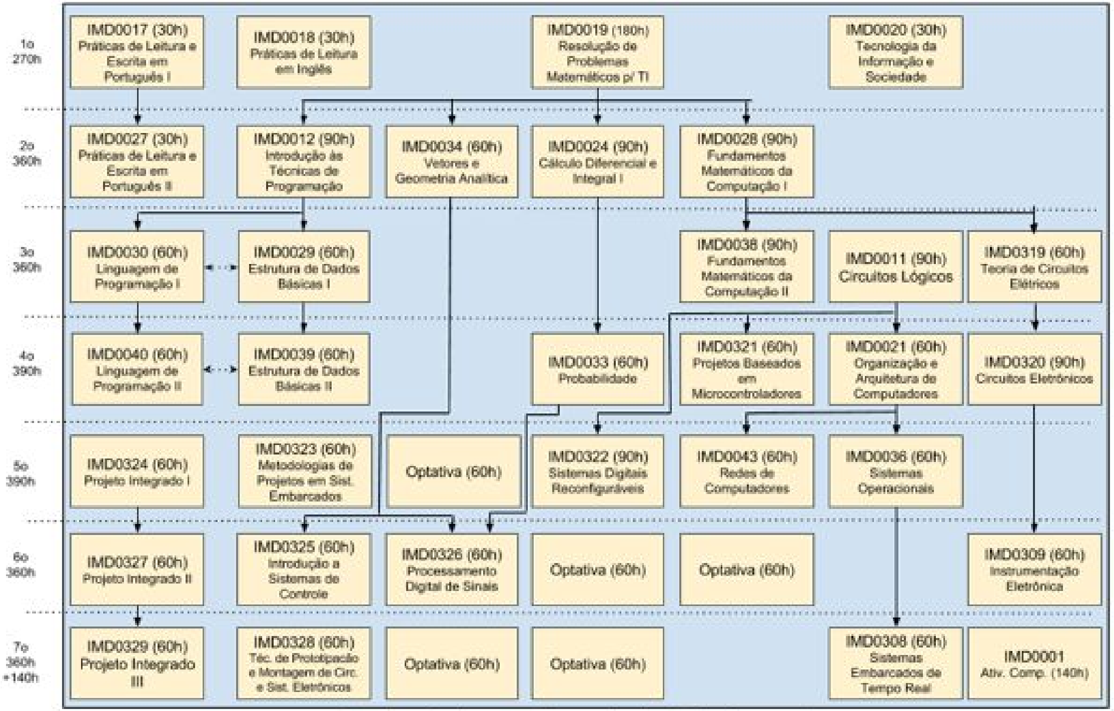
What is a network?





https://goo.gl/l8SS2X

What is a network?



1o
270h

IMD0017 (30h)
Práticas de Leitura e Escrita em Português I

IMD0018 (30h)
Práticas de Leitura em Inglês

IMD0019 (180h)
Resolução de Problemas Matemáticos p/ TI

IMD0020 (30h)
Tecnologia da Informação e Sociedade

2o
360h

IMD0027 (30h)
Práticas de Leitura e Escrita em Português II

IMD0012 (90h)
Introdução às Técnicas de Programação

IMD0034 (60h)
Vetores e Geometria Analítica

IMD0024 (90h)
Cálculo Diferencial e Integral I

IMD0028 (90h)
Fundamentos Matemáticos da Computação I

3o
360h

IMD0030 (60h)
Linguagem de Programação I

IMD0029 (60h)
Estrutura de Dados Básicas I

IMD0038 (90h)
Fundamentos Matemáticos da Computação II

IMD0011 (90h)
Circuitos Lógicos

IMD0319 (60h)
Teoria de Circuitos Elétricos

4o
390h

IMD0040 (60h)
Linguagem de Programação II

IMD0039 (60h)
Estrutura de Dados Básicas II

IMD0033 (60h)
Probabilidade

IMD0321 (60h)
Projetos Baseados em Microcontroladores

IMD0021 (60h)
Organização e Arquitetura de Computadores

IMD0320 (90h)
Circuitos Eletrônicos

5o
390h

IMD0324 (60h)
Projeto Integrado I

IMD0323 (60h)
Metodologias de Projetos em Sist. Embarcados

Optativa (60h)

IMD0322 (90h)
Sistemas Digitais Reconfiguráveis

IMD0043 (60h)
Redes de Computadores

IMD0036 (60h)
Sistemas Operacionais

6o
360h

IMD0327 (60h)
Projeto Integrado II

IMD0325 (60h)
Introdução a Sistemas de Controle

IMD0326 (60h)
Processamento Digital de Sinais

Optativa (60h)

Optativa (60h)

IMD0309 (60h)
Instrumentação Eletrônica

7o
360h
+140h

IMD0329 (60h)
Projeto Integrado III

IMD0328 (60h)
Téc. de Prototipação e Montagem de Circ. e Sist. Eletrônicos

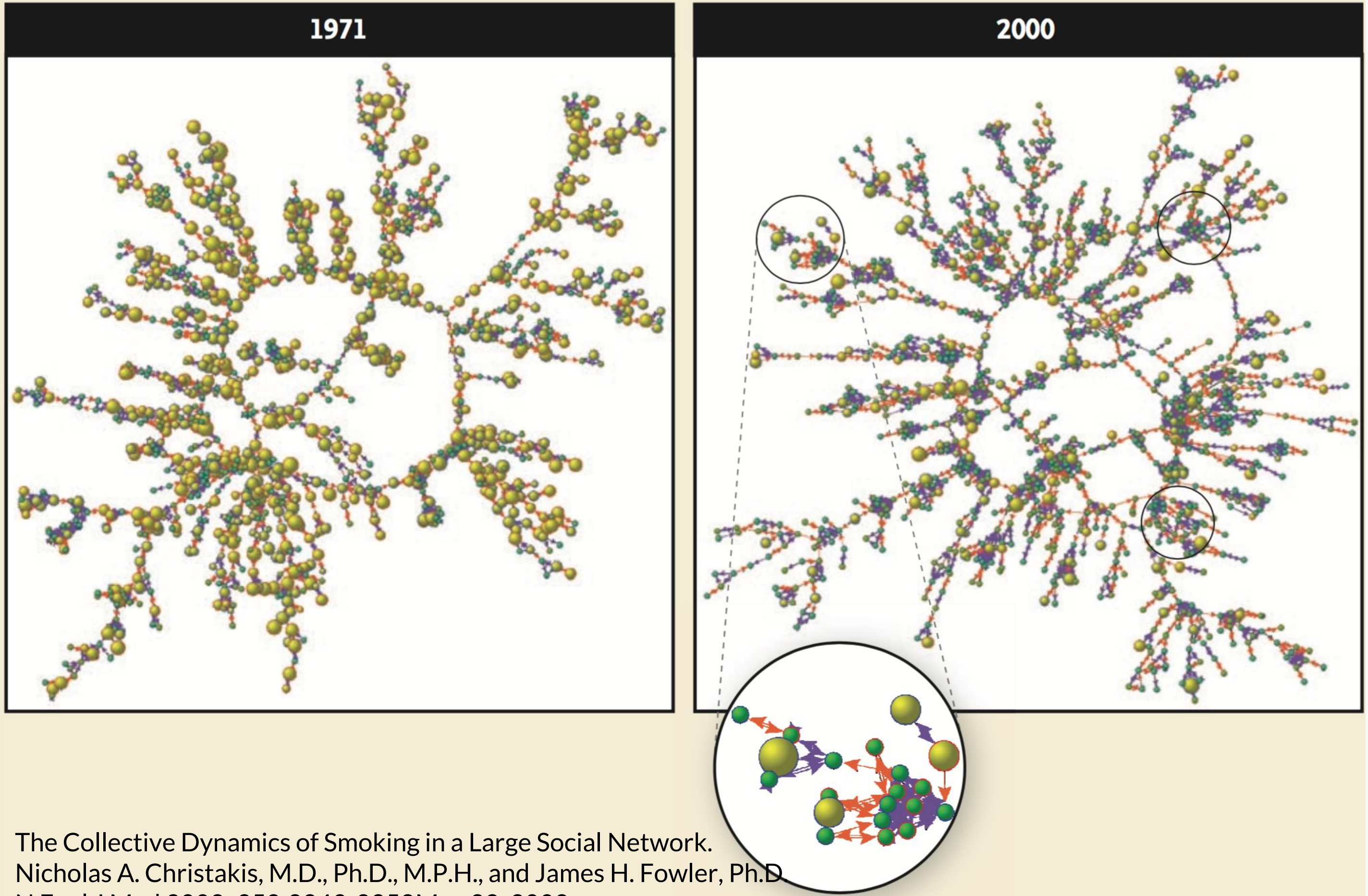
Optativa (60h)

Optativa (60h)

IMD0308 (60h)
Sistemas Embarcados de Tempo Real

IMD0001
Ativ. Comp. (140h)

Bacharelado em Tecnologia da Informação
Ênfase em Sistemas Embarcados (MT)

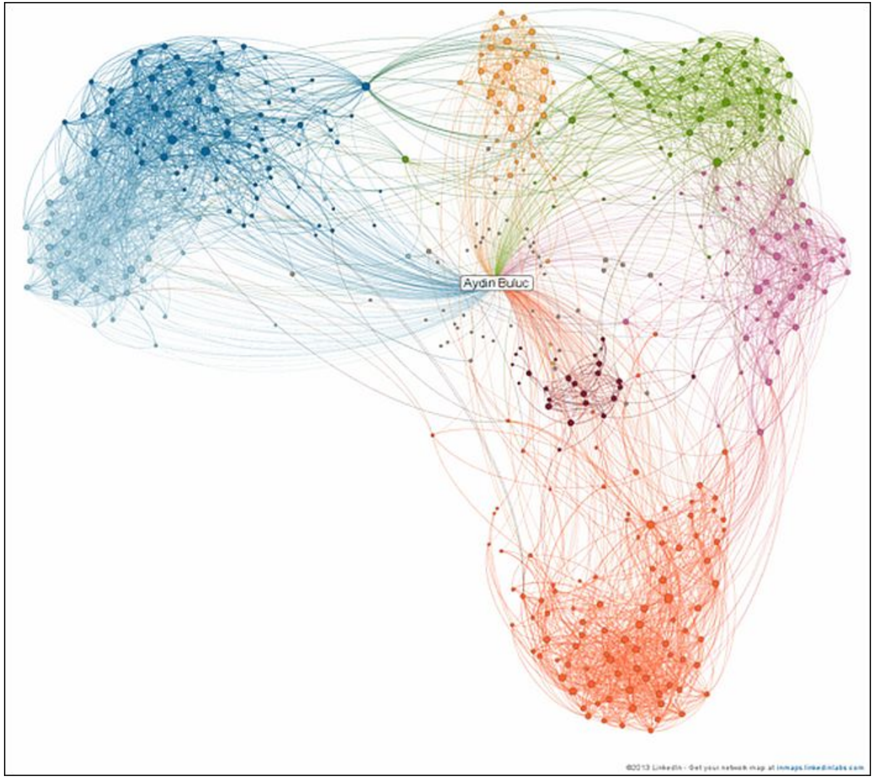


1971

2000

The Collective Dynamics of Smoking in a Large Social Network.
Nicholas A. Christakis, M.D., Ph.D., M.P.H., and James H. Fowler, Ph.D.
N Engl J Med 2008; 358:2249-2258May 22, 2008

What is a network?

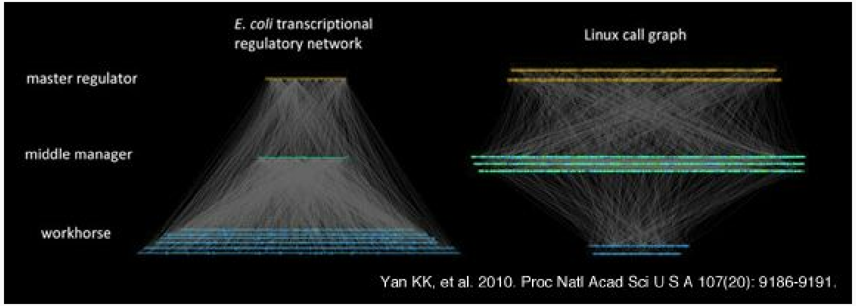


©2013 LinkedIn · Get your network map at inmaps.linkedinlabs.com

A Berkeley Lab researcher applies graph theory to find genes useful for biofuels.

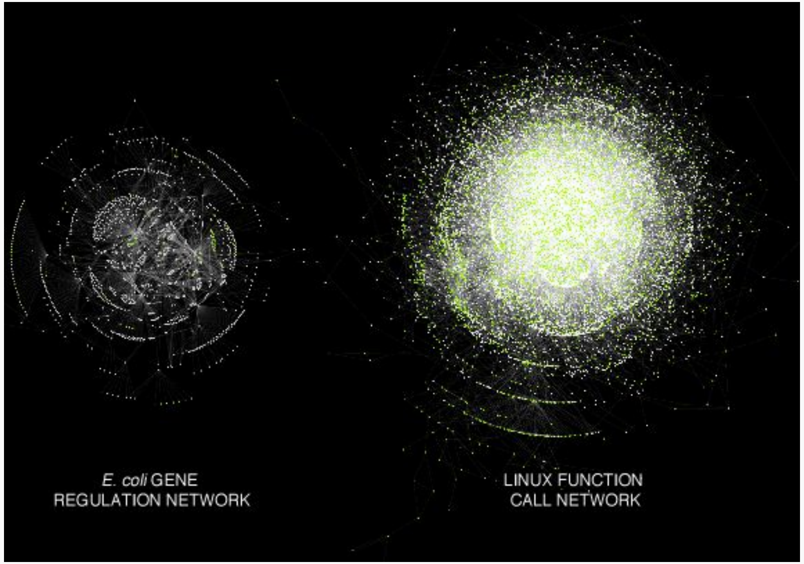
http://ascr-discovery.science.doe.gov/2013/09/sifting-genomes/

What is a network?



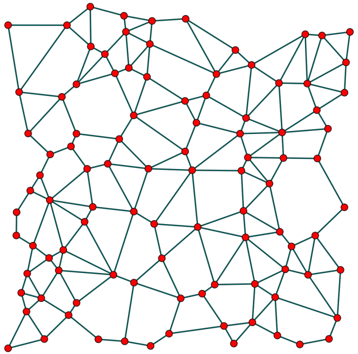
Yan KK, et al. 2010. Proc Natl Acad Sci U S A 107(20): 9186-9191.

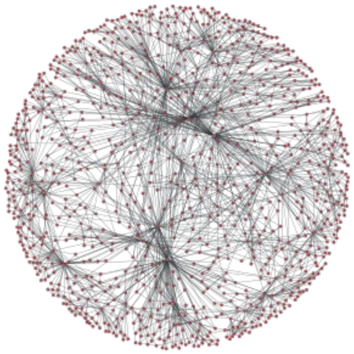
http://www.pnas.org/content/107/20/9186.abstract


E. coli GENE REGULATION NETWORK
LINUX FUNCTION CALL NETWORK

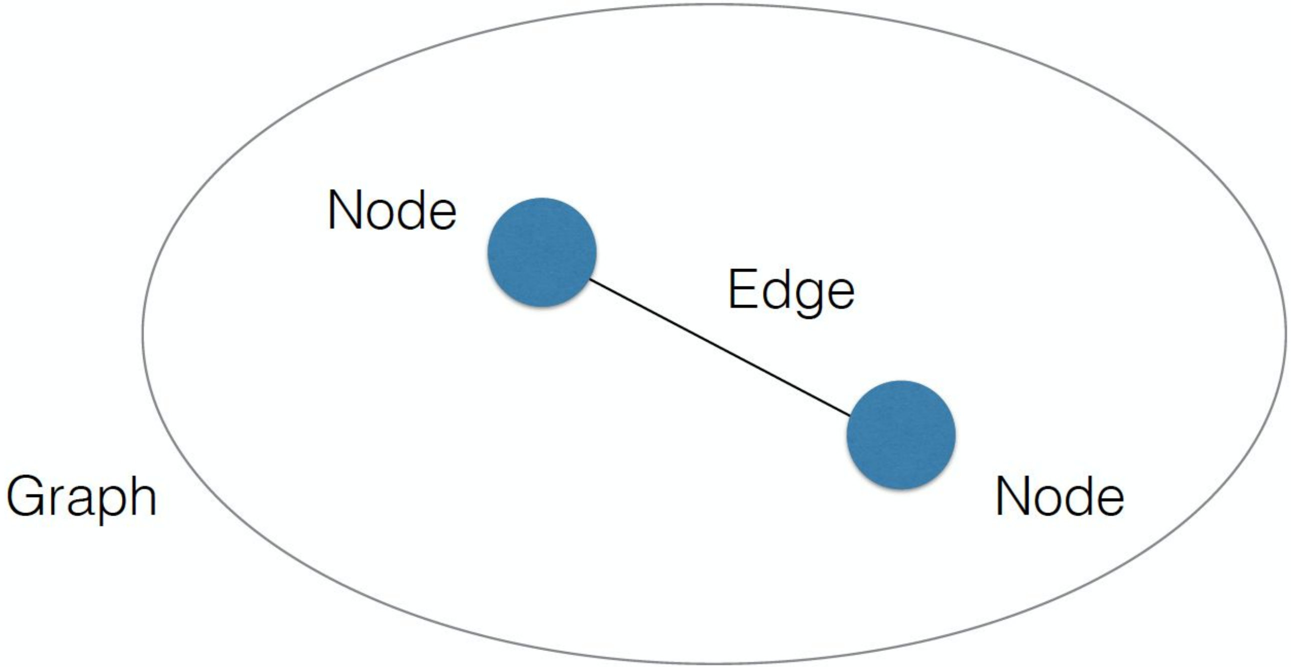
What is a network?

- A collection of points joined by lines
- Mathematically: graph
    - A gentle introduction to graph theory: https://dev.to/vaidehijoshi/a-gentle-introduction-to-graph-theory
- Representation of relationship between discrete objects
- A way of exploring data

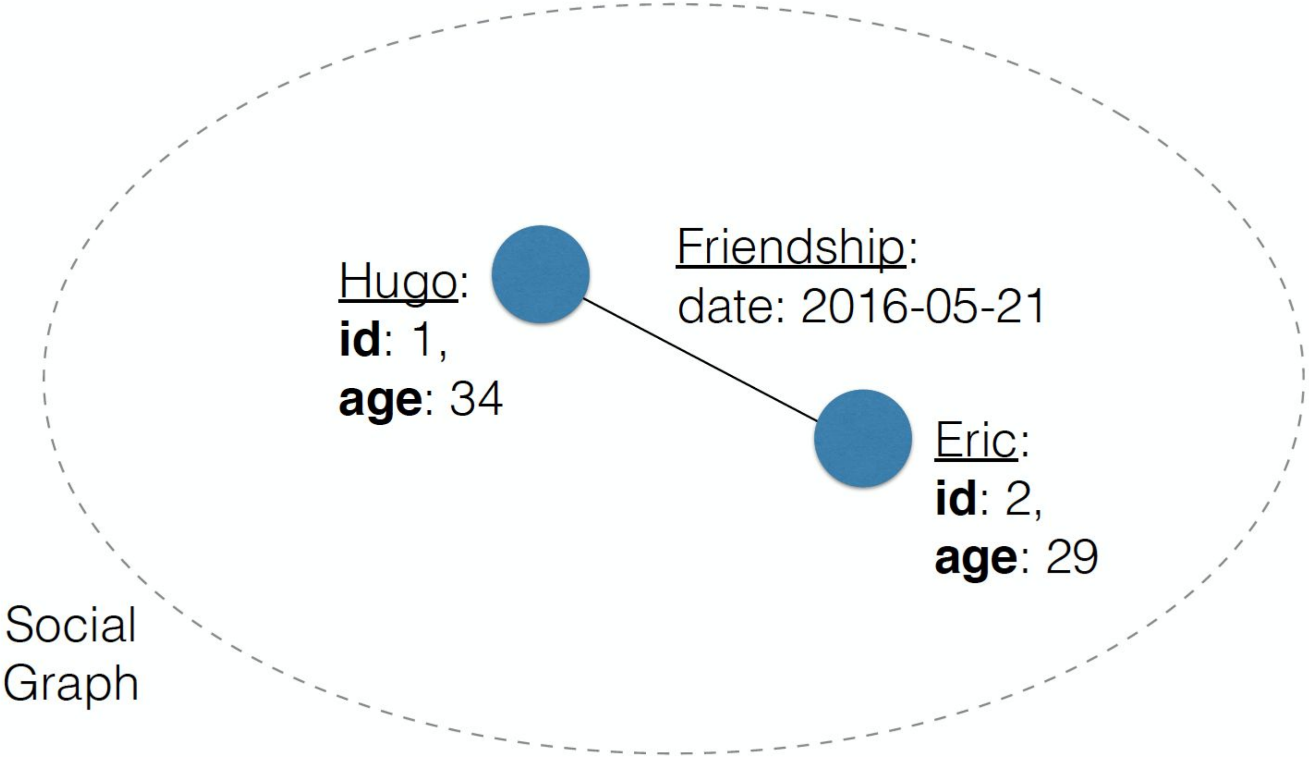




Network Structure

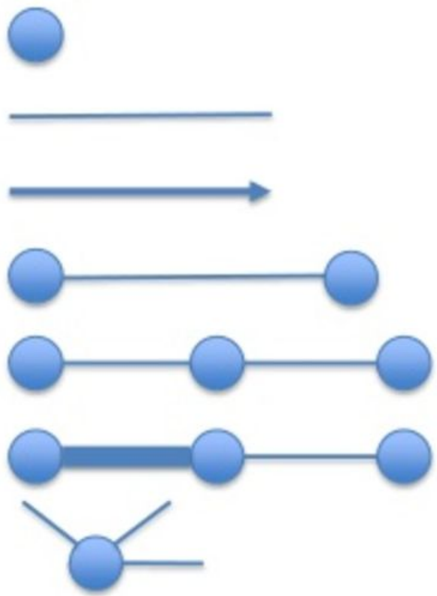


Network Structure

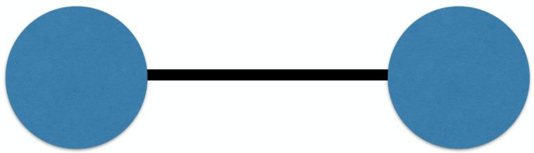


Network Structure

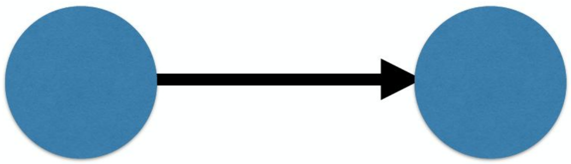
- Vertex/node
- Edge
- Directed
- Connectivity
- Path
- Weight
- Degree



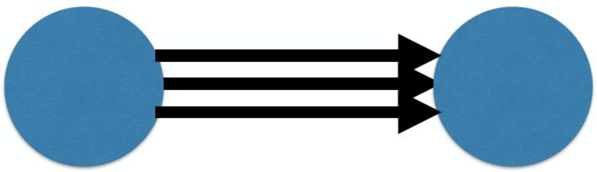
Types of Graphs



Undirected: Facebook social graph



Directed: Twitter social graph

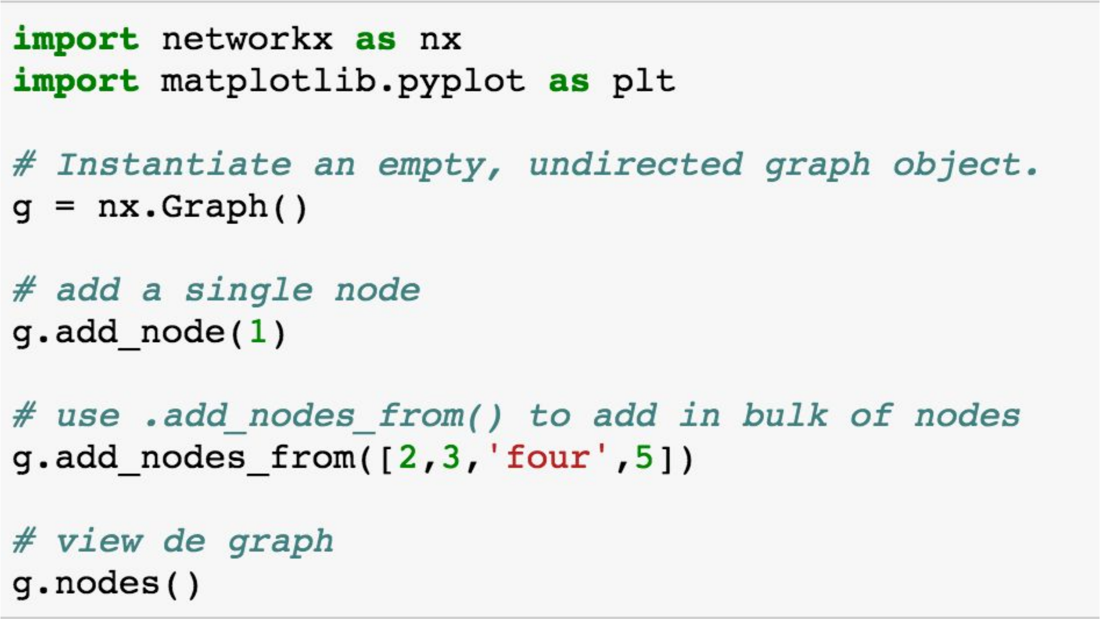


MultiDiGraph: trip records between bike sharing stations

Handle with networks: a tool perspective

- https://networkx.github.io/
- https://gephi.org/
- http://www.cytoscape.org/
- http://www.graphviz.org/

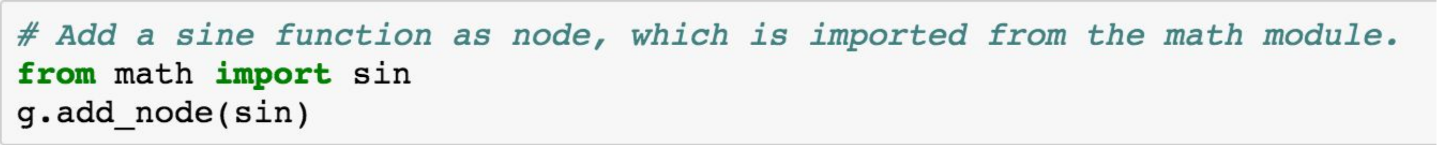
NetworkX API Basics



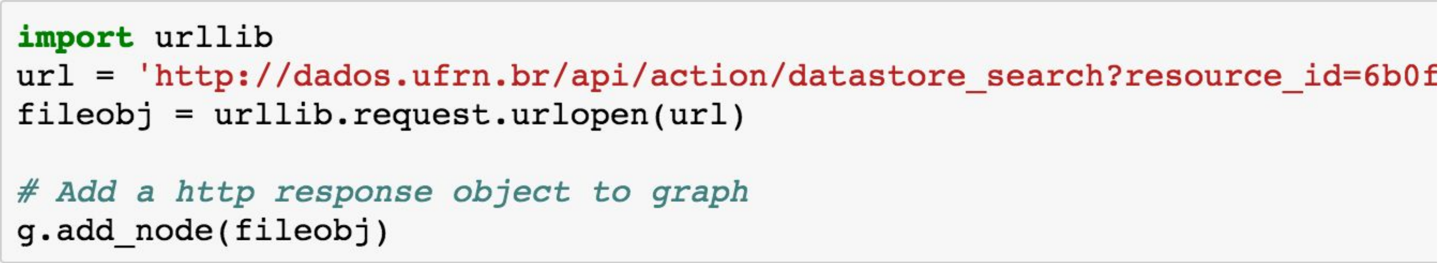
```
import networkx as nx
import matplotlib.pyplot as plt

# Instantiate an empty, undirected graph object.
g = nx.Graph()

# add a single node
g.add_node(1)

# use .add_nodes_from() to add in bulk of nodes
g.add_nodes_from([2,3,'four',5])

# view de graph
g.nodes()
```

Adding nodes

A node can be any of the so-called hashable objects - strings, numbers, files, functions, etc.

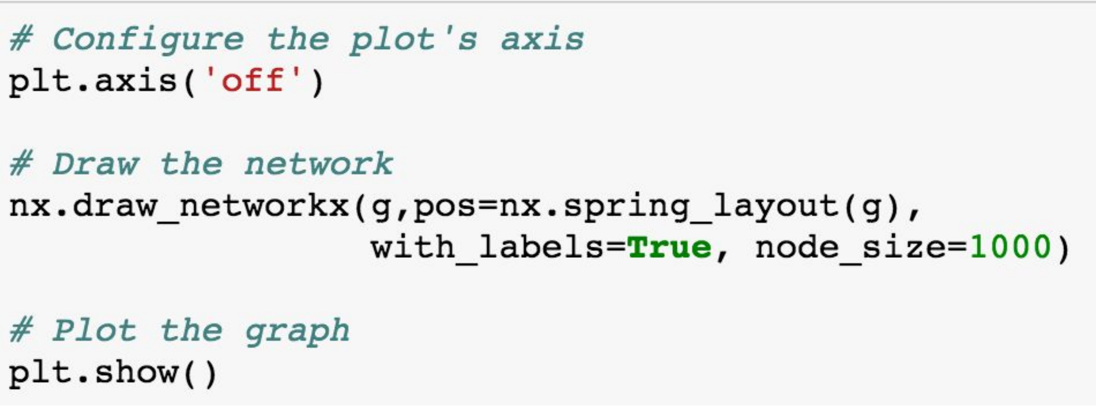


```
# Add a sine function as node, which is imported from the math module.
from math import sin
g.add_node(sin)
```

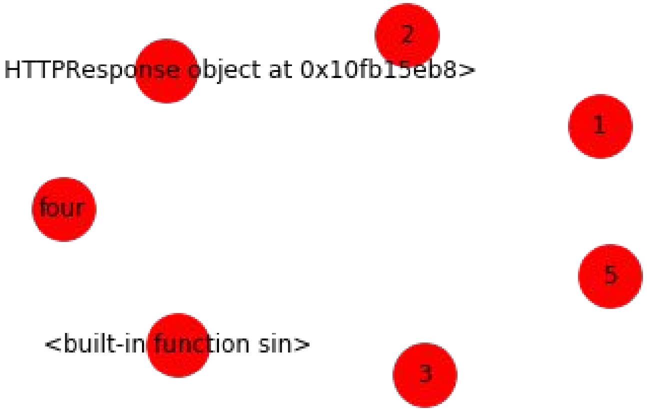


```
import urllib
url = 'http://dados.ufrn.br/api/action/datastore_search?resource_id=6b0f
fileobj = urllib.request.urlopen(url)

# Add a http response object to graph
g.add_node(fileobj)
```

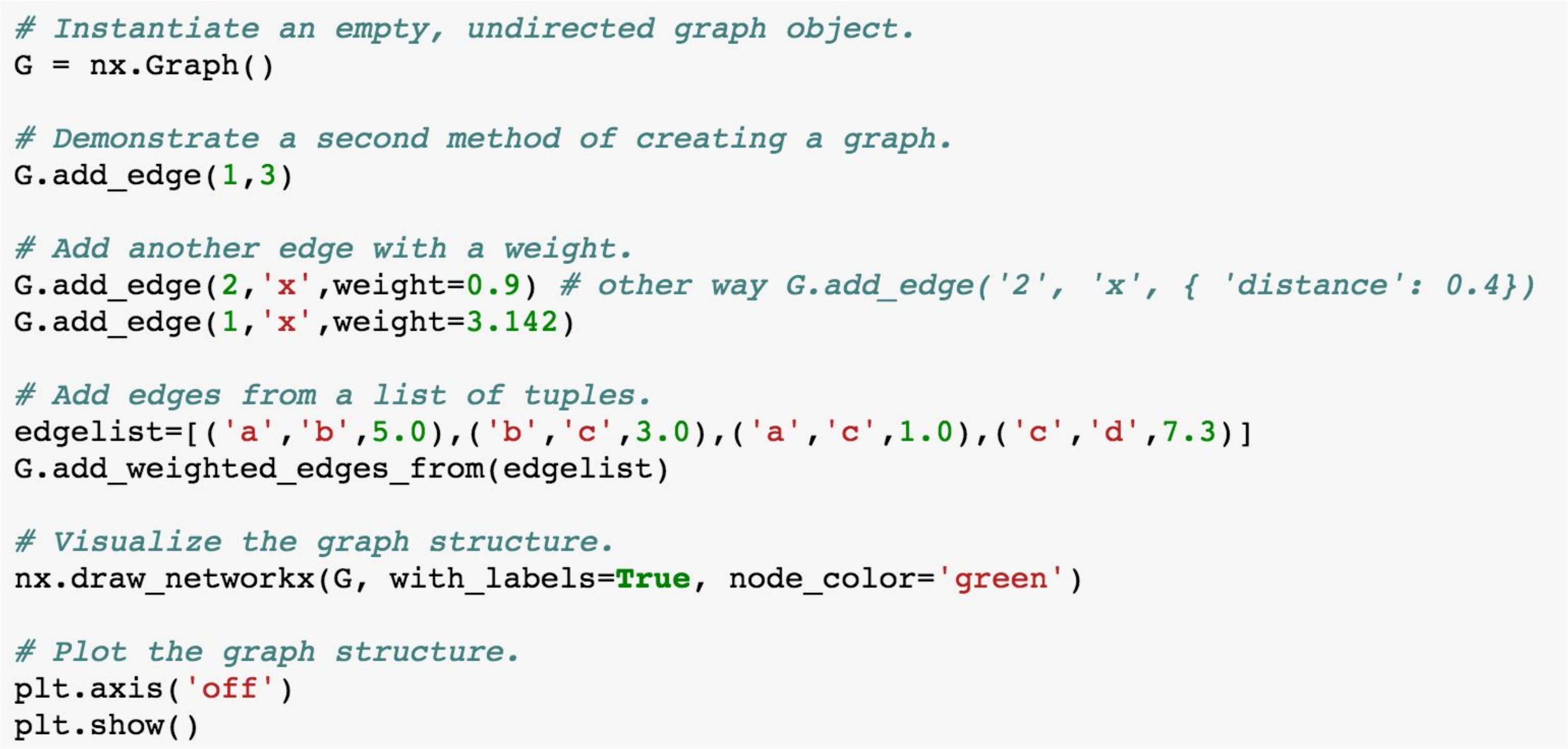
Visualize the graph structure



```
# Configure the plot's axis
plt.axis('off')

# Draw the network
nx.draw_networkx(g,pos=nx.spring_layout(g),
                 with_labels=True, node_size=1000)

# Plot the graph
plt.show()
```

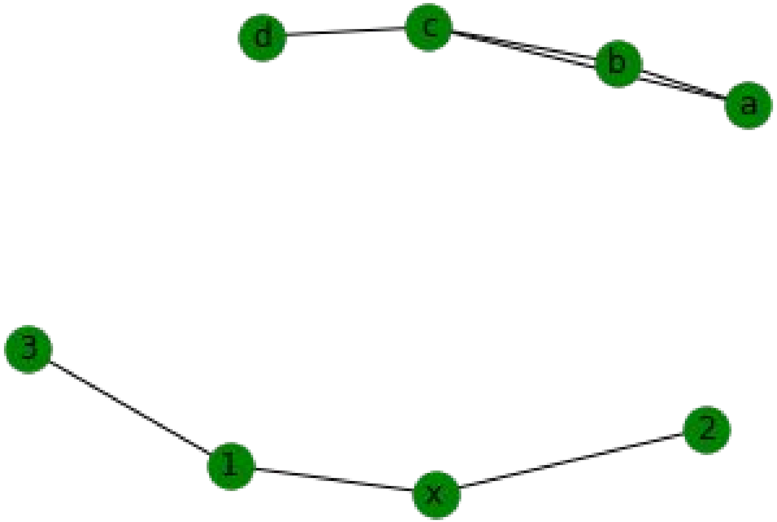


Adding edges

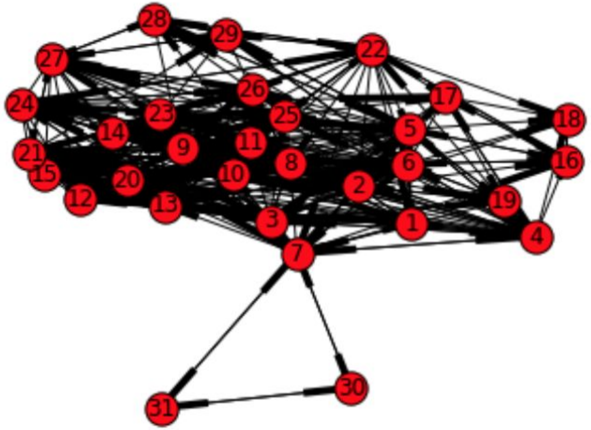


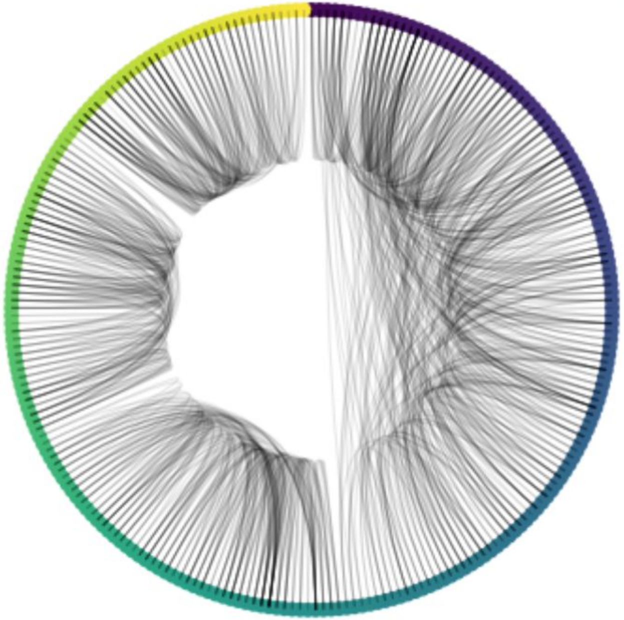
```
# Instantiate an empty, undirected graph object.
G = nx.Graph()

# Demonstrate a second method of creating a graph.
G.add_edge(1,3)

# Add another edge with a weight.
G.add_edge(2,'x',weight=0.9) # other way G.add_edge('2', 'x', { 'distance': 0.4})
G.add_edge(1,'x',weight=3.142)

# Add edges from a list of tuples.
edgelist=[('a','b',5.0),('b','c',3.0),('a','c',1.0),('c','d',7.3)]
G.add_weighted_edges_from(edgelist)

# Visualize the graph structure.
nx.draw_networkx(G, with_labels=True, node_color='green')

# Plot the graph structure.
plt.axis('off')
plt.show()
```

Visualizing the previous example



Irrational vs Rational Visualization

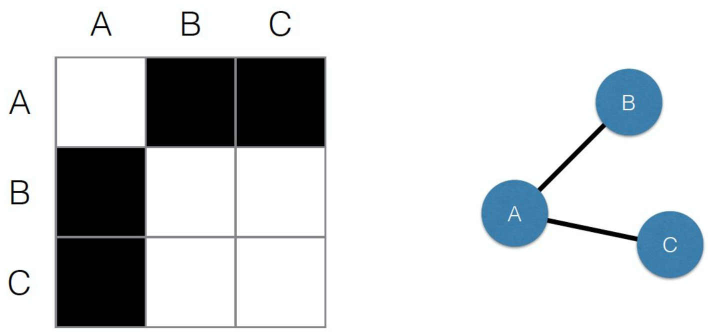




Visualizing networks

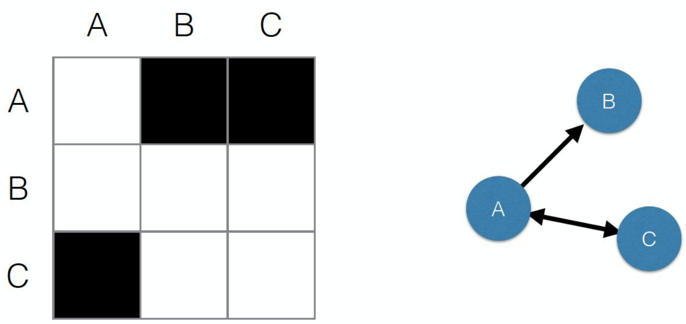
- Matrix plots
- Arc plots
- Circos plots

Matrix plots



	A	B	C
A		■	■
B	■		
C	■		

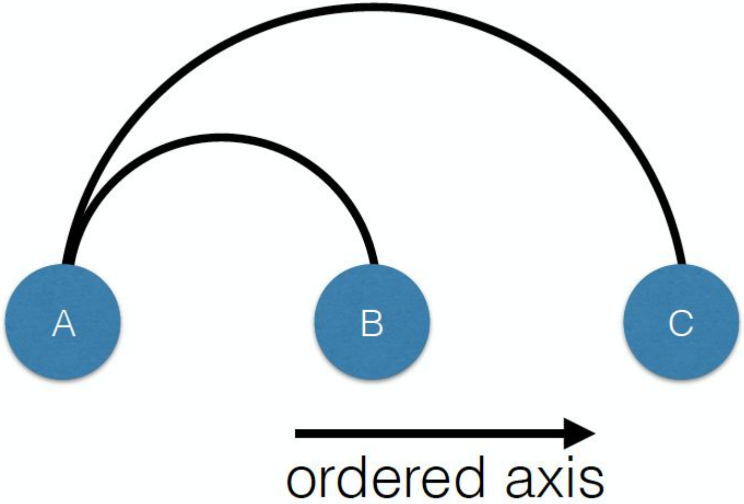
Undirected Graph

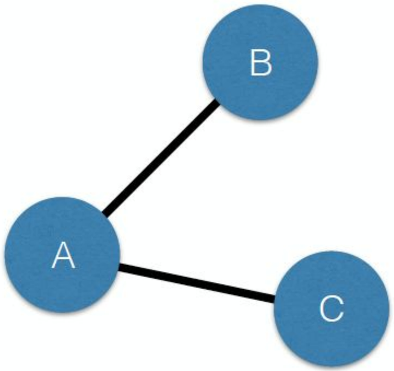


	A	B	C
A		■	■
B			
C	■		

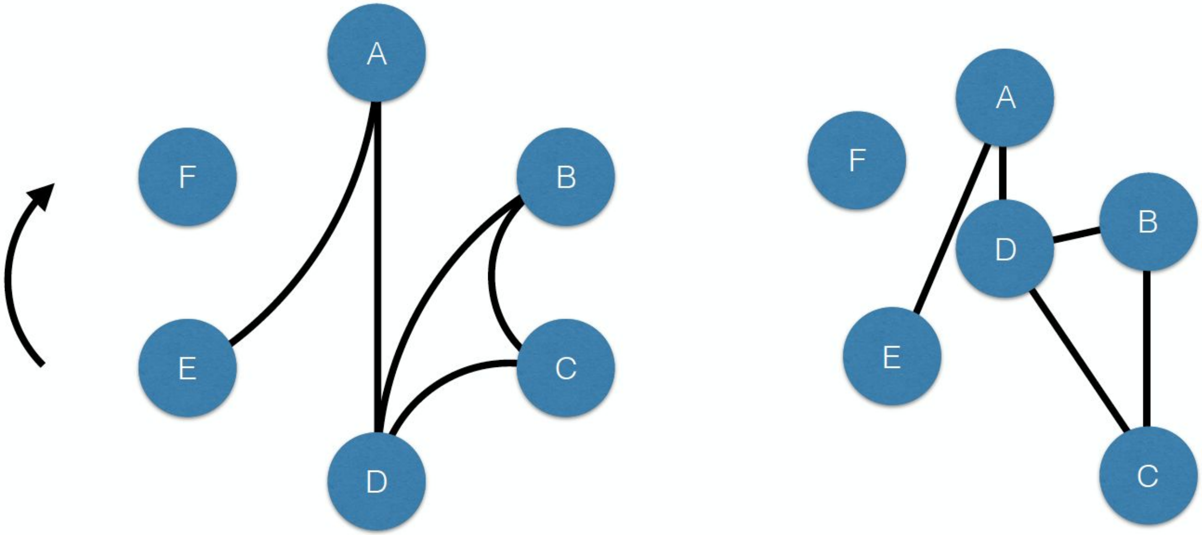
Directed Graph

Arc Plot

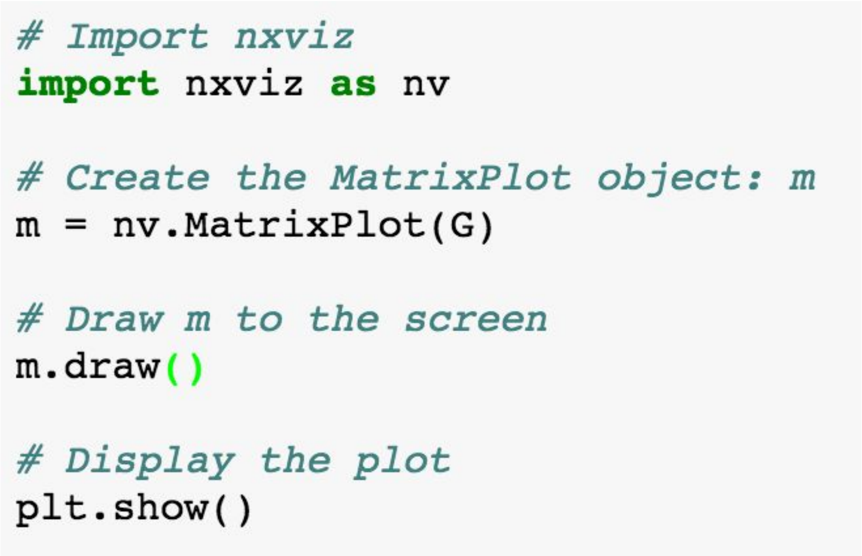




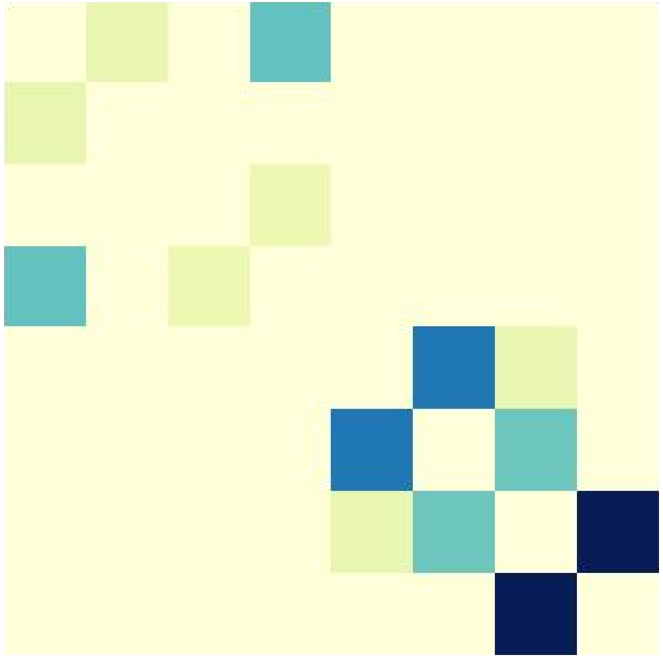
Circos Plot



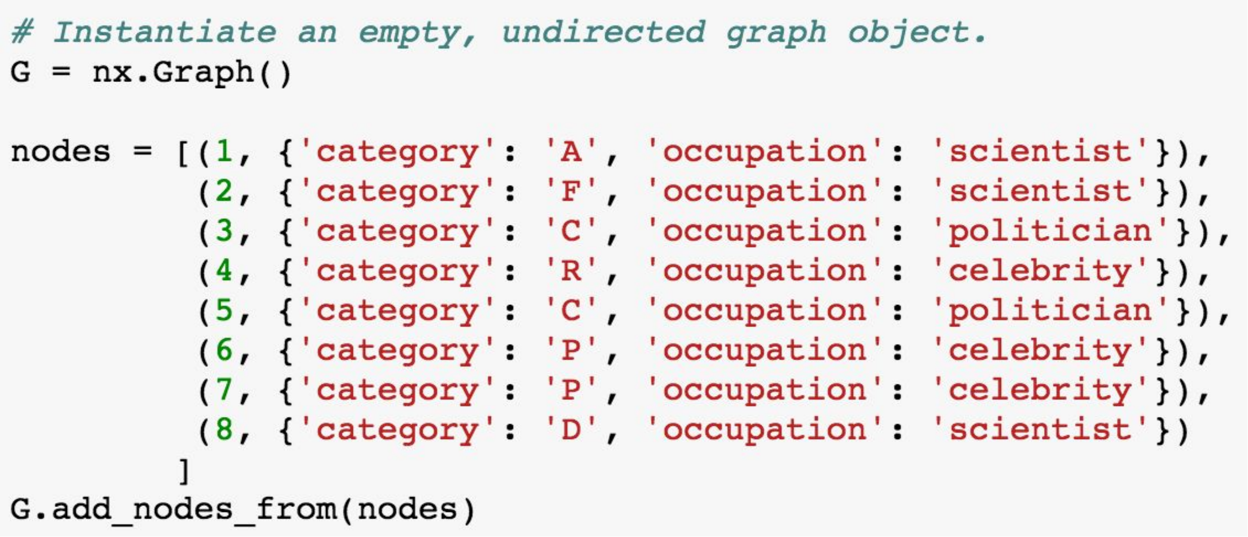
Visualizing using Matrix plot



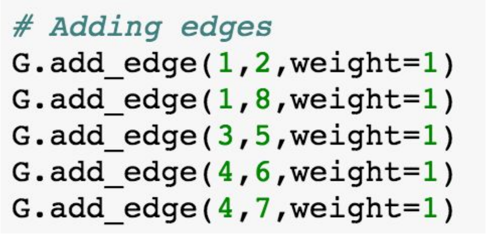
```
# Import nxviz
import nxviz as nv

# Create the MatrixPlot object: m
m = nv.MatrixPlot(G)

# Draw m to the screen
m.draw()

# Display the plot
plt.show()
```



Visualizing using Circos plot: step #1

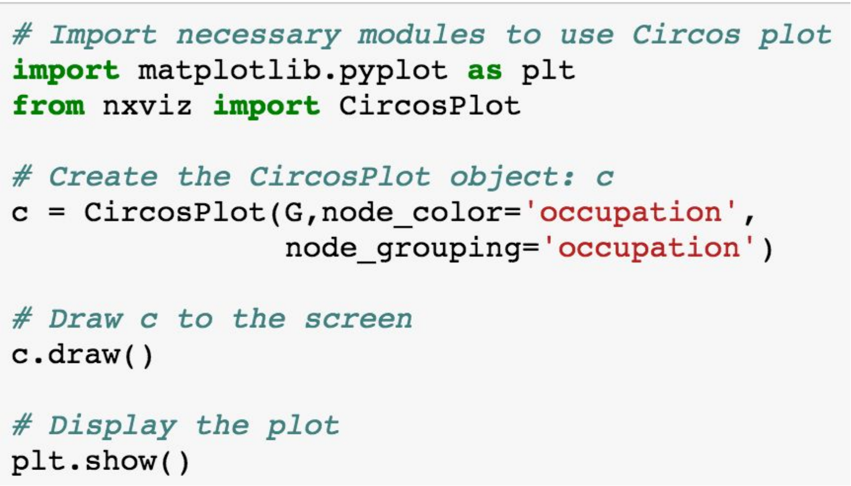


```
# Instantiate an empty, undirected graph object.
G = nx.Graph()

nodes = [(1, {'category': 'A', 'occupation': 'scientist'}),
         (2, {'category': 'F', 'occupation': 'scientist'}),
         (3, {'category': 'C', 'occupation': 'politician'}),
         (4, {'category': 'R', 'occupation': 'celebrity'}),
         (5, {'category': 'C', 'occupation': 'politician'}),
         (6, {'category': 'P', 'occupation': 'celebrity'}),
         (7, {'category': 'P', 'occupation': 'celebrity'}),
         (8, {'category': 'D', 'occupation': 'scientist'})
        ]
G.add_nodes_from(nodes)
```

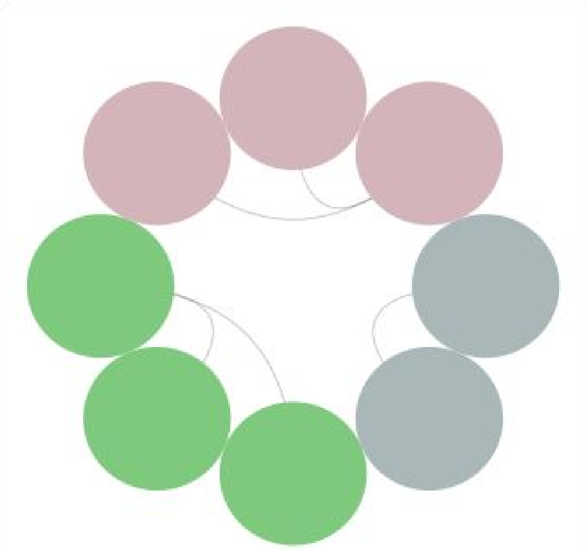


```
# Adding edges
G.add_edge(1,2,weight=1)
G.add_edge(1,8,weight=1)
G.add_edge(3,5,weight=1)
G.add_edge(4,6,weight=1)
G.add_edge(4,7,weight=1)
```

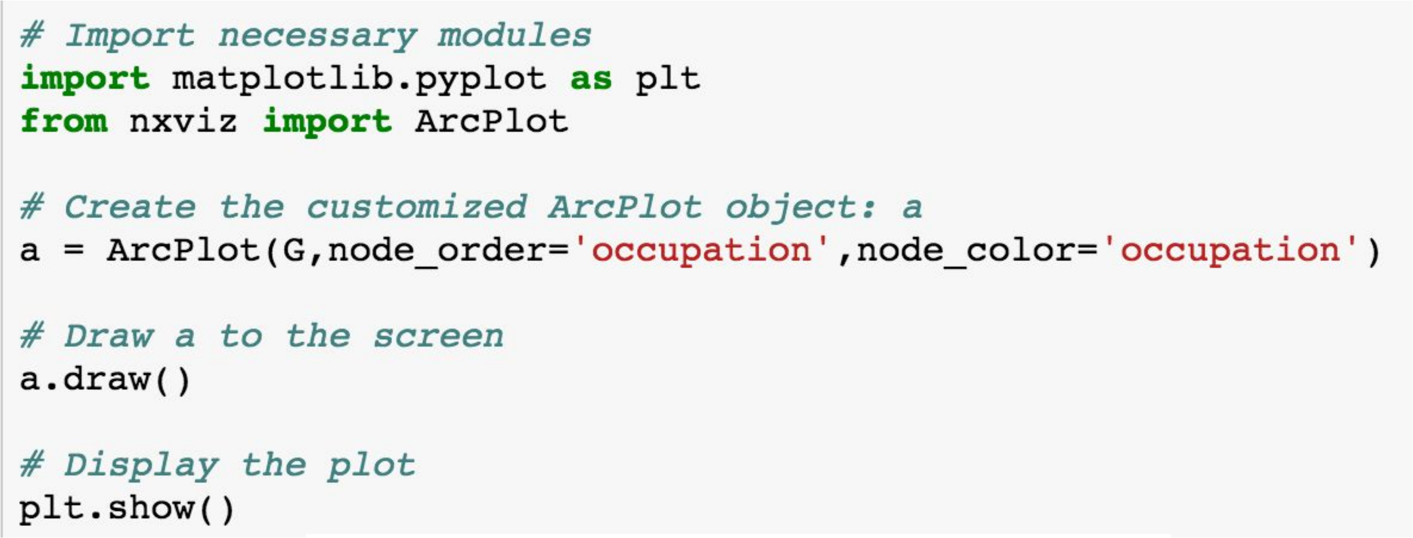
Visualizing using Circos plot: step #2



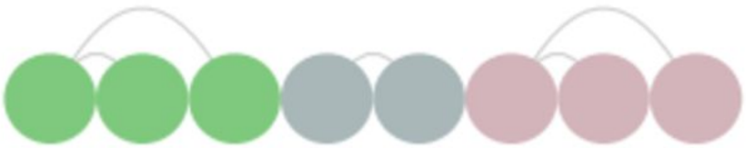
```
# Import necessary modules to use Circos plot
import matplotlib.pyplot as plt
from nxviz import CircosPlot

# Create the CircosPlot object: c
c = CircosPlot(G,node_color='occupation',
               node_grouping='occupation')

# Draw c to the screen
c.draw()

# Display the plot
plt.show()
```



Visualizing Arc plots



```
# Import necessary modules
import matplotlib.pyplot as plt
from nxviz import ArcPlot

# Create the customized ArcPlot object: a
a = ArcPlot(G,node_order='occupation',node_color='occupation')

# Draw a to the screen
a.draw()

# Display the plot
plt.show()
```



Requesting network structure

- Getting neighbor information
- Removing nodes and edges
- Graph generators

Notebook - https://goo.gl/DeQJVv

Reference

- https://media.readthedocs.org/pdf/networkx/stable/networkx.pdf
- https://github.com/sandrofsousa/awesome-network-analysis
- https://www.researchgate.net/publication/304946197_Estudo_das_propriedades_e_robustez_da_rede_de_transporte_publico_de_Sao_Paulo
- http://www.hiveplot.com/
- https://github.com/ericmjl
- https://dev.to/vaidehijoshi/a-gentle-introduction-to-graph-theory