NETWORK ANALYSIS IN PYTHON I

# Introduction to networks

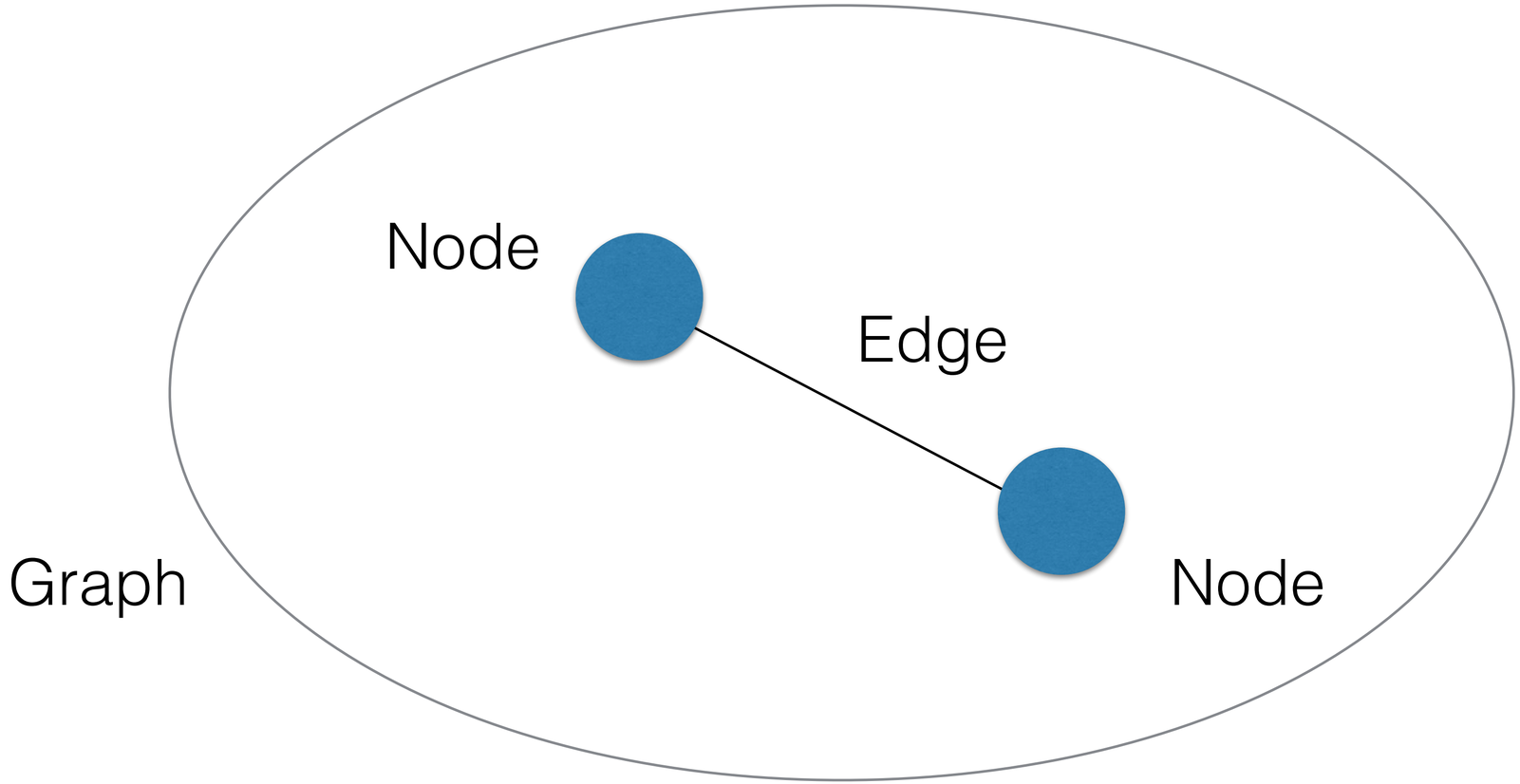
# Networks!

- Examples:
  - Social
  - Transportation
- Model **relationships** between **entities**

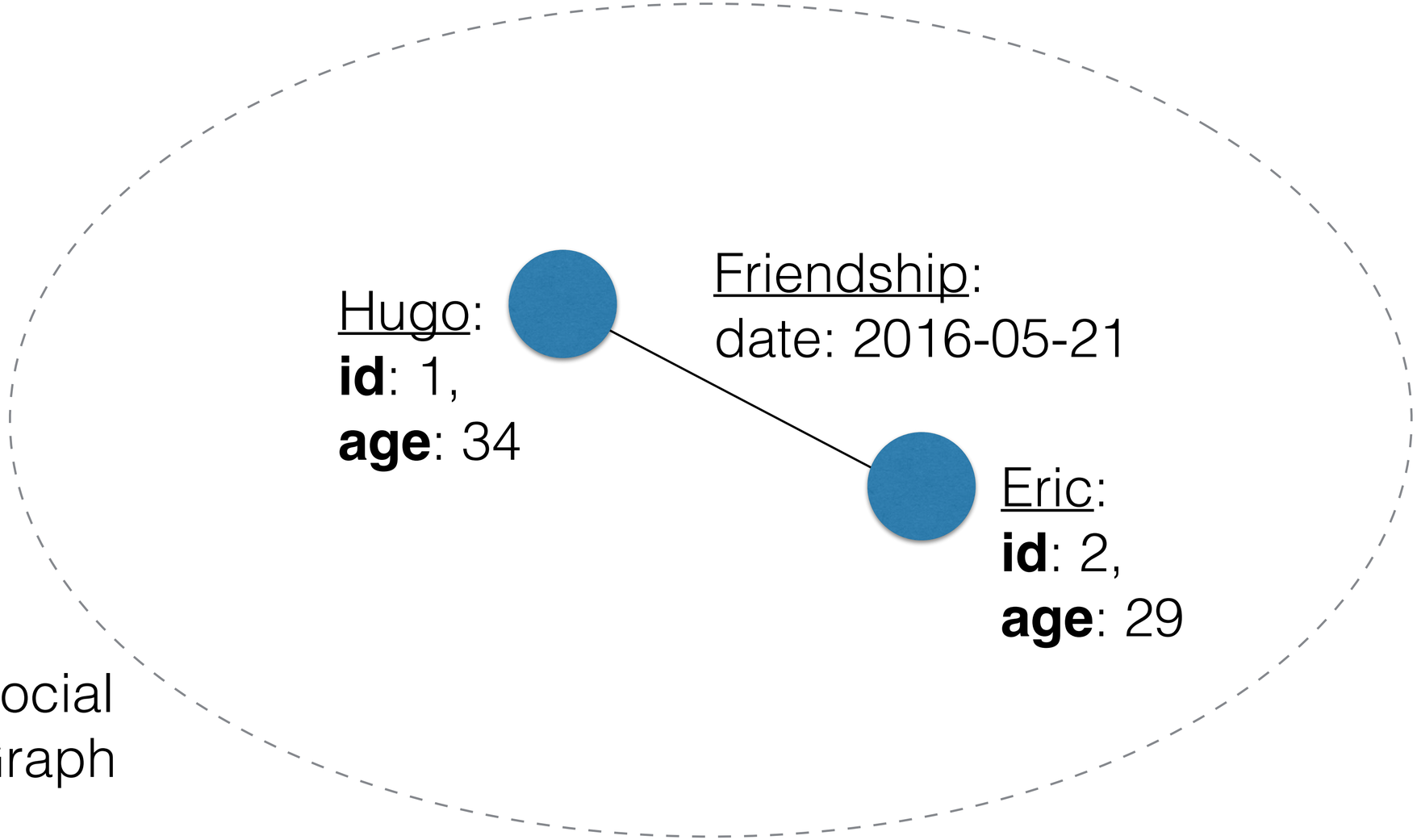
# Networks!

- Insights:
  - Important entities: influencers in social network
  - Pathfinding: most efficient transport path
  - Clustering: finding communities

# Network structure



# Network structure



# NetworkX API basics

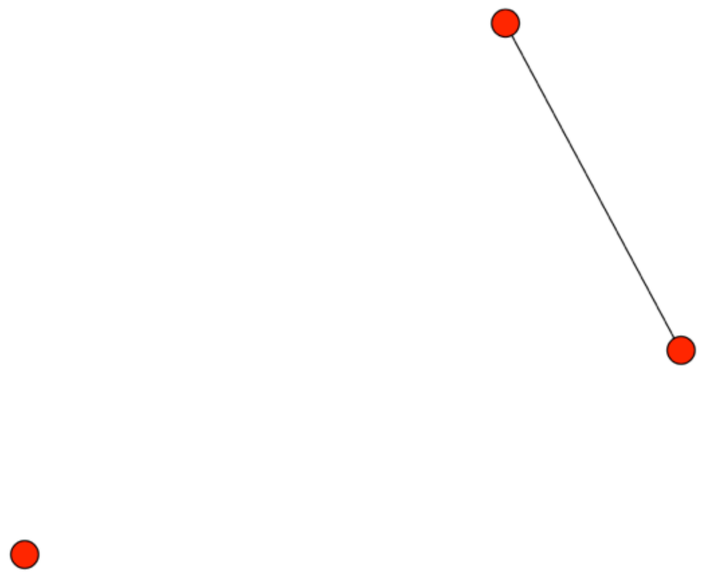
```
In [1]: import networkx as nx

In [2]: G = nx.Graph()

In [4]: G.add_nodes_from([1, 2, 3])

In [5]: G.nodes()
Out[5]: [1, 2, 3]

In [6]: G.add_edge(1, 2)

In [7]: G.edges()
Out[7]: [(1, 2)]
```

# NetworkX API basics

```
In [8]: G.node[1]['label'] = 'blue'

In [9]: G.nodes(data=True)
Out[9]: [(1, {'label': 'blue'}), (2, {}), (3, {})]
```

# NetworkX API basics

```
In [10]: nx.draw(G)

In [11]: import matplotlib.pyplot as plt

In [12]: plt.show()
```



NETWORK ANALYSIS IN PYTHON I

# Let’s practice!

NETWORK ANALYSIS IN PYTHON

# Types of graphs

# Undirected graphs

- Facebook social graph

# Undirected graphs

```
In [1]: import networkx as nx

In [2]: G = nx.Graph()

In [3]: type(G)
Out[3]: networkx.classes.graph.Graph
```

# Directed graphs

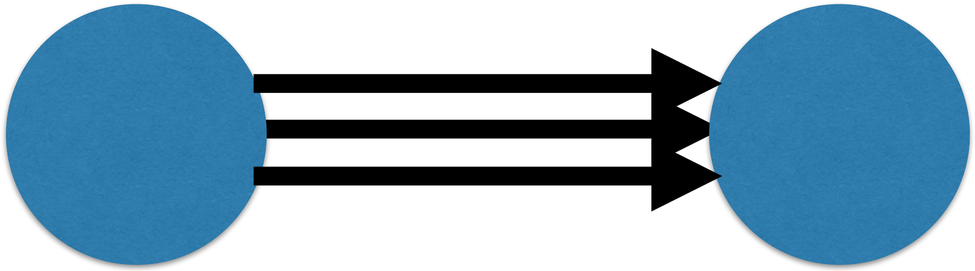
- Directed: Twitter social graph

# Directed graphs

```
In [4]: D = nx.DiGraph()

In [5]: type(D)
Out[5]: networkx.classes.digraph.DiGraph
```

# Types of graphs

- Multi(Di)Graph: Trip records between bike sharing stations

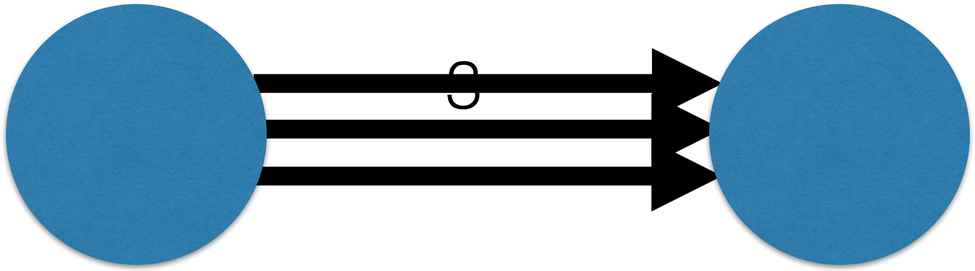


# Multi-edge (Directed) graphs

```
In [6]: M = nx.MultiGraph()

In [7]: type(M)
Out[7]: networkx.classes.multigraph.MultiGraph

In [8]: MD = nx.MultiDiGraph()

In [9]: type(MD)
Out[9]: networkx.classes.multidigraph.MultiDiGraph
```

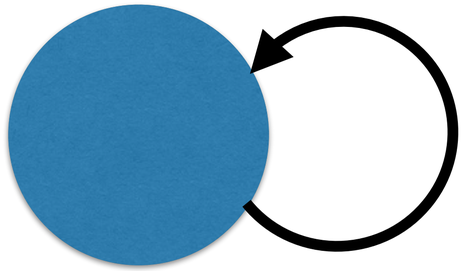
# Weights on graphs

- Edges can contain weights



# Self-loops

- Nodes that are connected to themselves



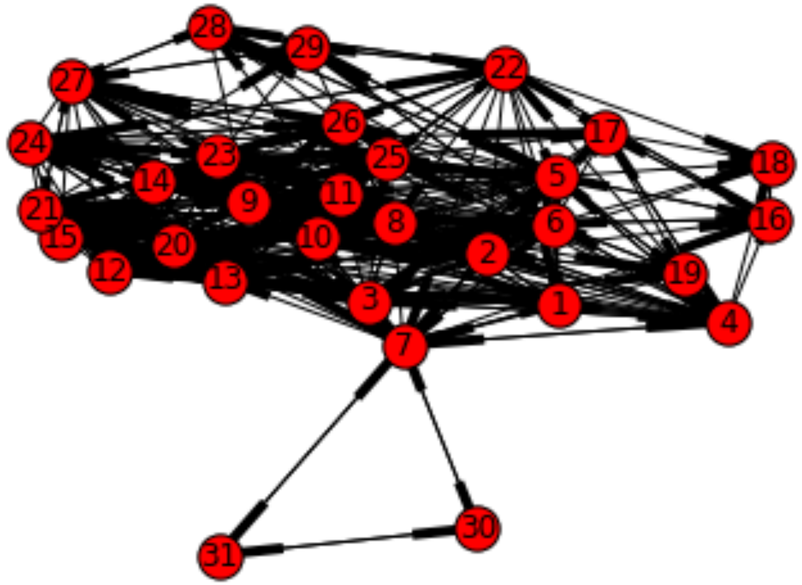
NETWORK ANALYSIS IN PYTHON I

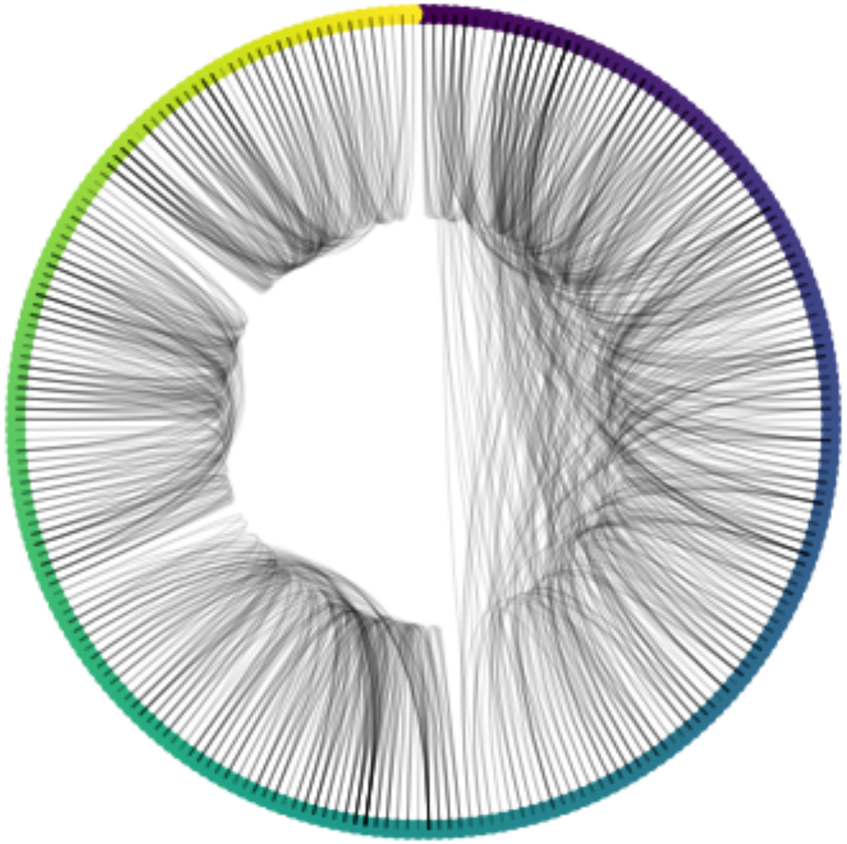
# Let’s practice!

NETWORK ANALYSIS IN PYTHON I

# Network visualization

# Irrational vs. Rational visualizations



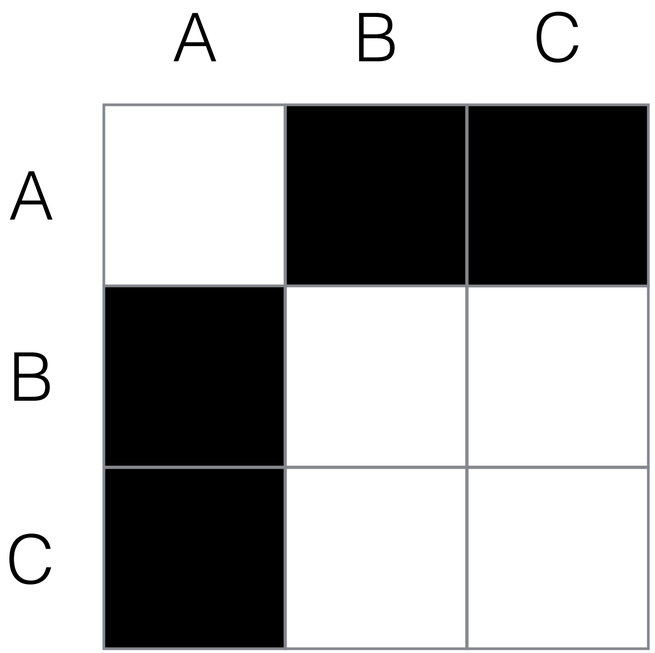
# Visualizing networks

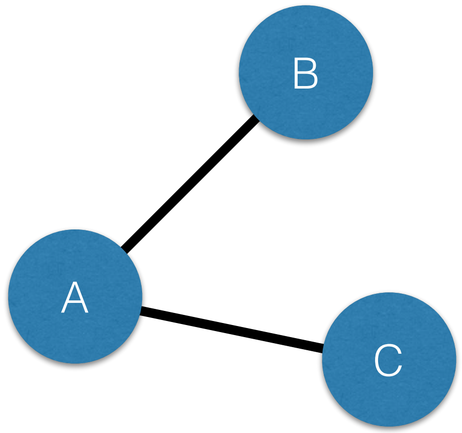
- Matrix plots
- Arc plots
- Circos plots

# Visualizing networks

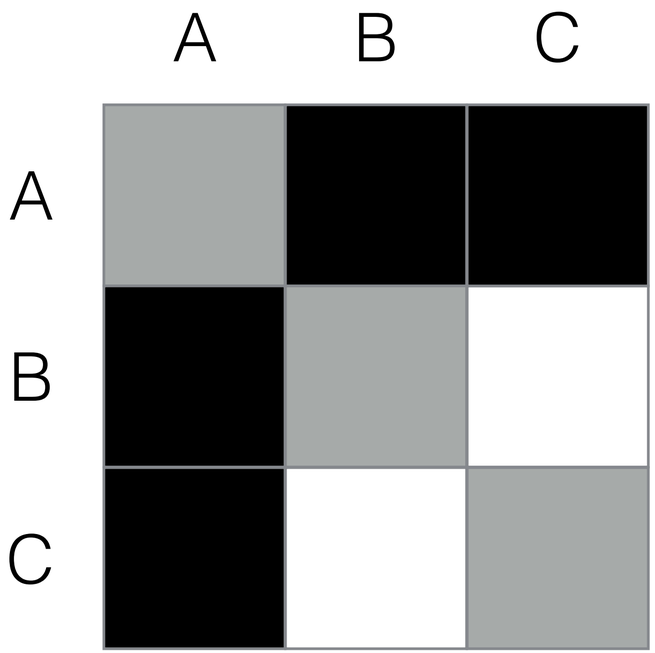
- Matrix plots
- Arc plots
- Circos plots

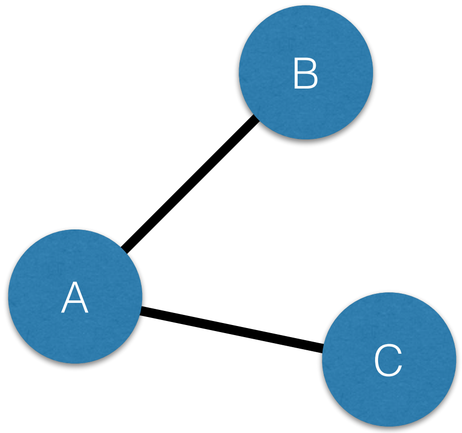
# Matrix plot



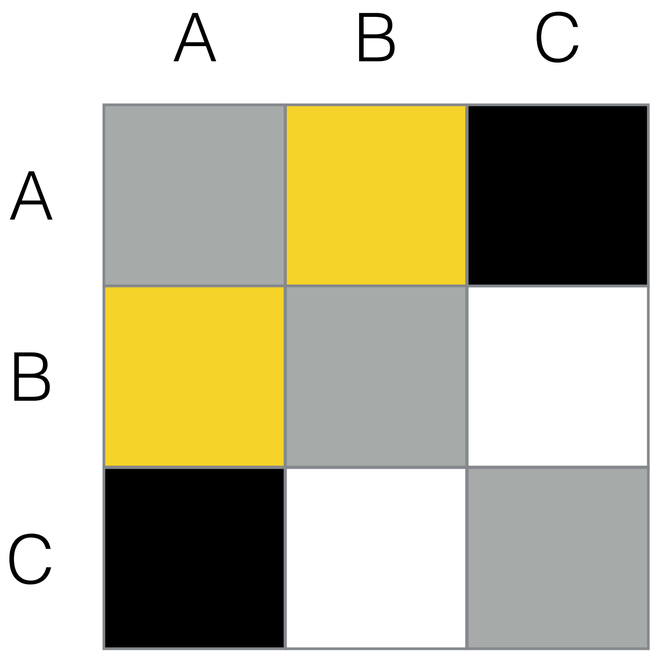



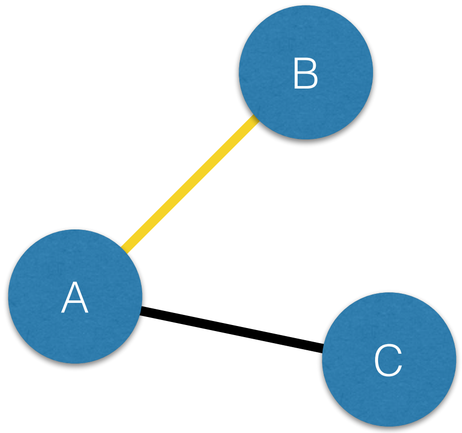
# Matrix plot



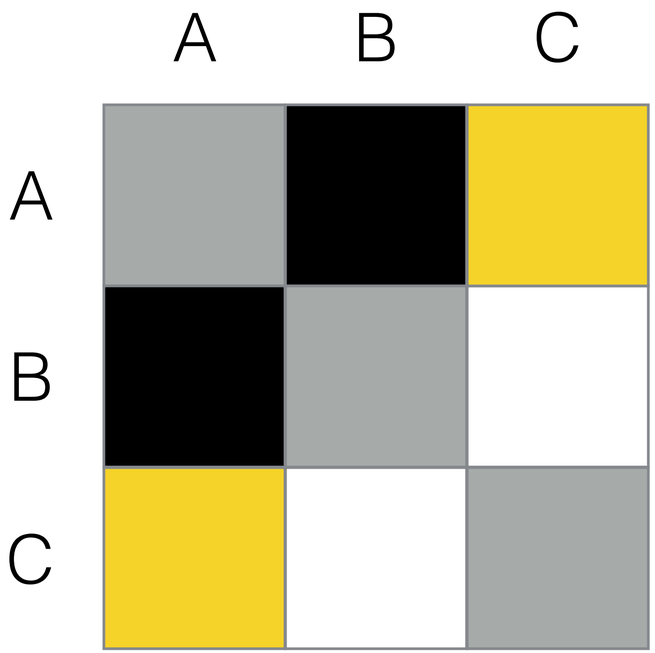



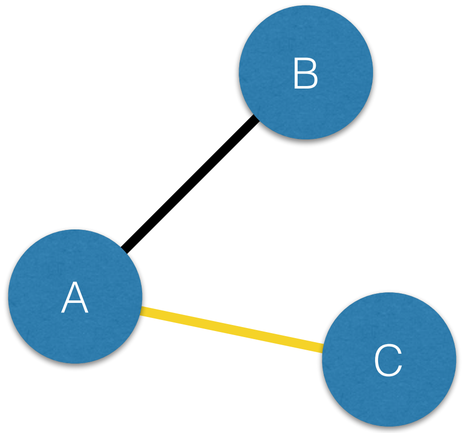
# Matrix plot



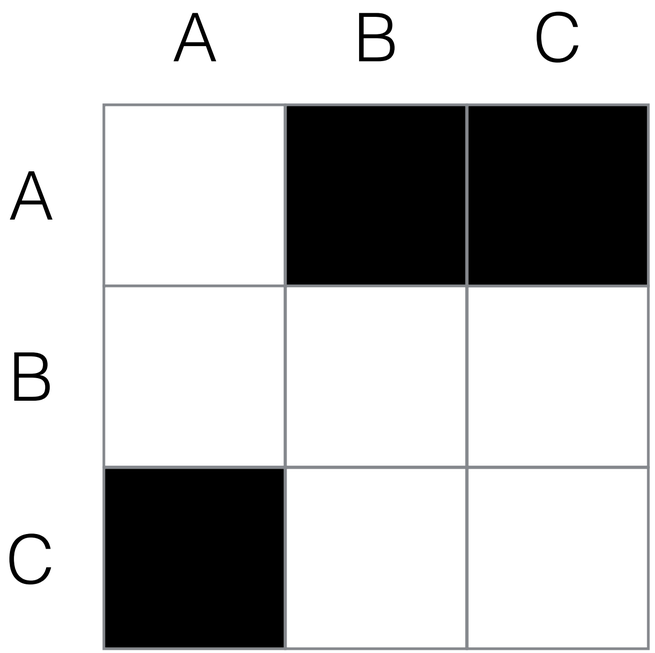



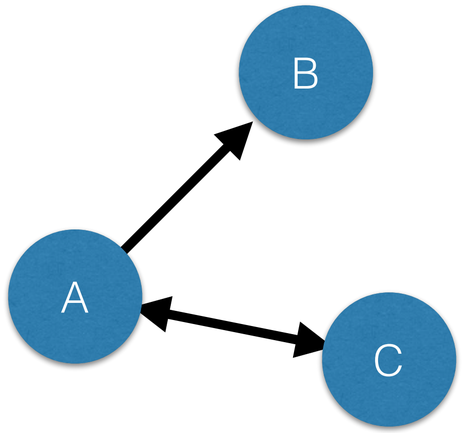
# Matrix plot



# Directed matrices

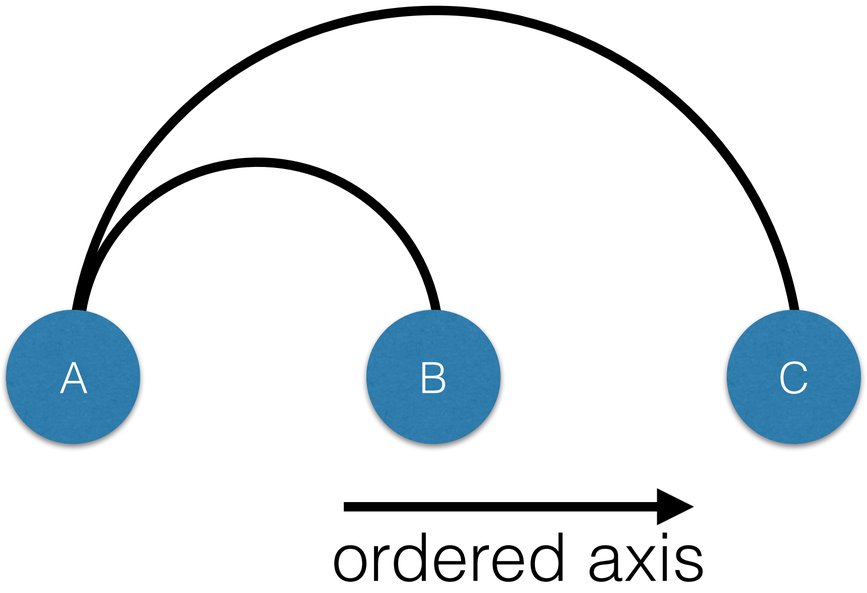


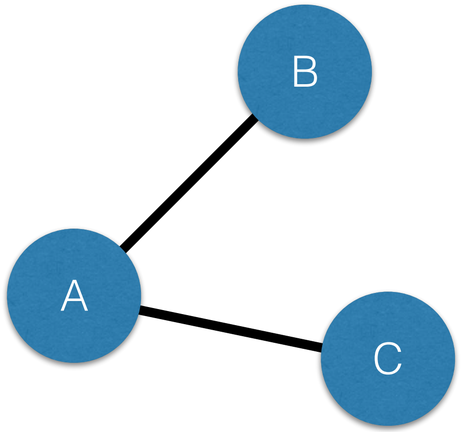



# Visualizing networks

- Matrix Plots
- Arc Plots
- Circos Plots

# Arc plot

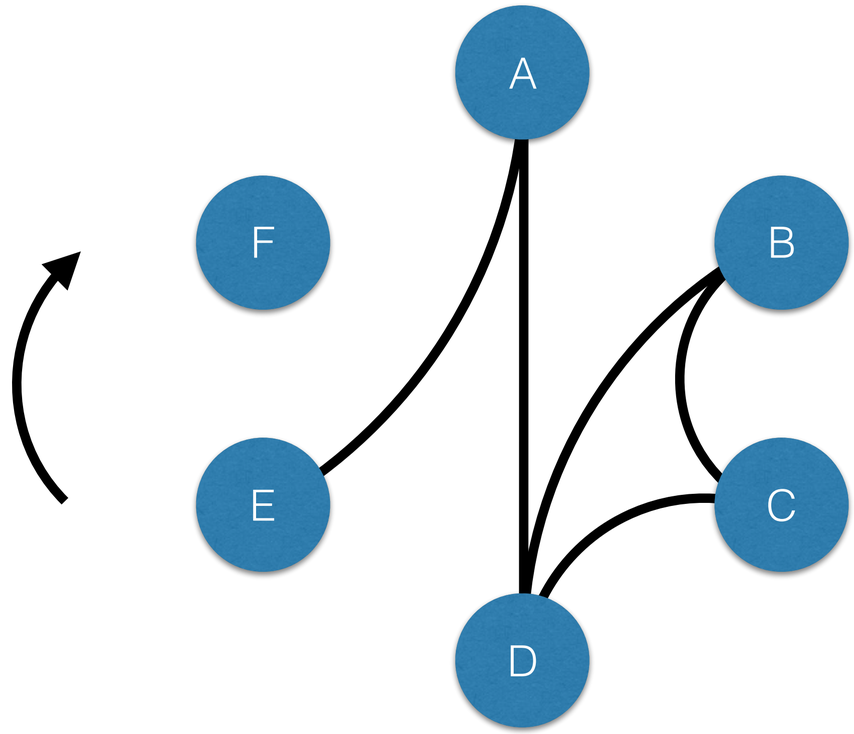


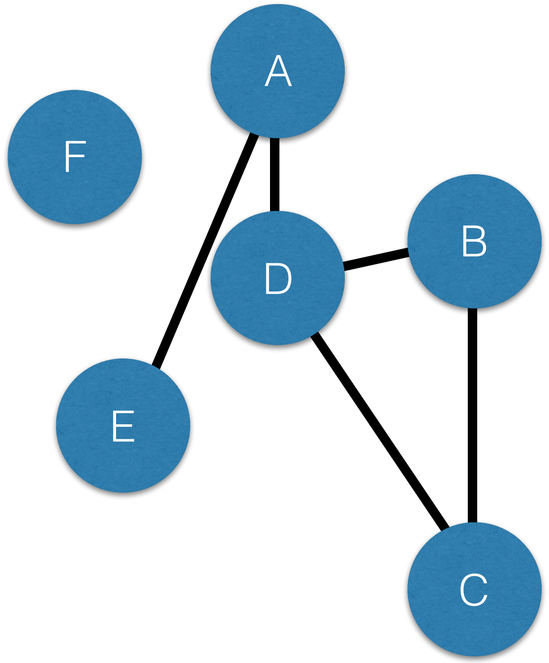



# Visualizing networks

- Matrix Plots
- Arc Plots
- Circos Plots

# Circos plot



# nxviz API

```
In [1]: import nxviz as nv

In [2]: import matplotlib.pyplot as plt

In [3]: ap = nv.ArcPlot(G)

In [4]: ap.draw()

In [5]: plt.show()
```

NETWORK ANALYSIS IN PYTHON I

# Let’s practice!