Case study!

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON

Eric Ma

Data Carpentry instructor and author of nxviz package

datacamp

Data

- Github user collaboration network
- Nodes: users
- Edges: collaboration on same GitHub repository
- Goals:
    - Analyze structure
    - Visualize
    - Build simple recommendation system

Graph properties

```
import networkx as nx
G = nx.erdos_renyi_graph(n=20, p=0.2)
len(G.edges())
```

```
29
```

```
len(G.nodes())
```

```
20
```

Graph properties

```
nx.degree_centrality(G)
```

```
{0: 0.15789473684210525,
 1: 0.15789473684210525,
 2: 0.15789473684210525,
 3: 0.10526315789473684,...
```

```
nx.betweenness_centrality(G)
```

```
{0: 0.019493177387914233,
 1: 0.060916179337231965,
 2: 0.12768031189083822,
 3: 0.03313840155945419,...
```

Data

- Number of nodes
- Number of edges
- Degree centrality distribution
- Betweenness centrality distribution

Let's practice!

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON

Case study part II: Visualization

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON

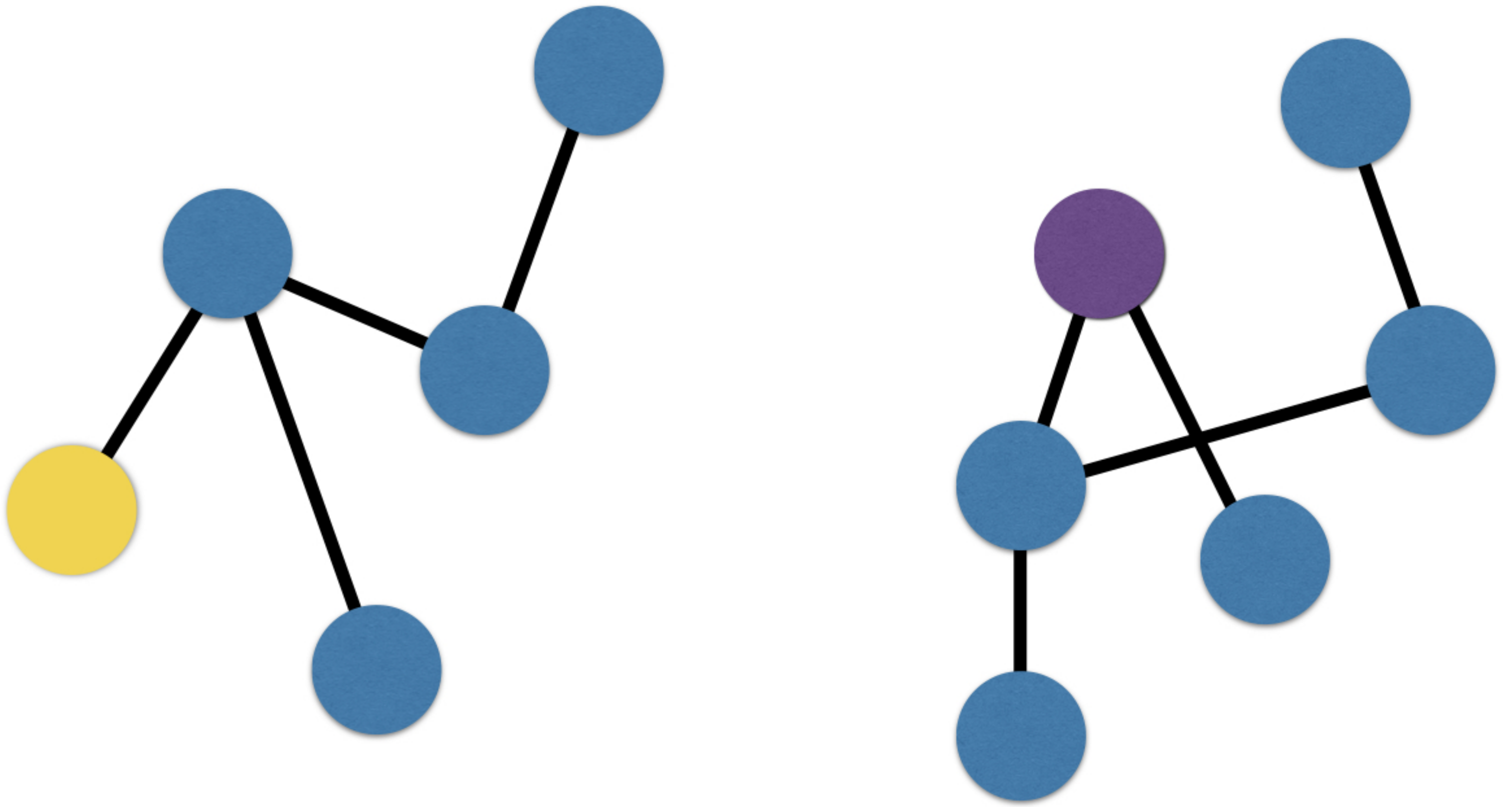
Eric Ma

Data Carpentry instructor and author of nxviz package

nxviz API

```python
import networkx as nx
import nxviz as nv
G = nx.erdos_renyi_graph(n=20, p=0.3)
circ = nv.CircosPlot(G, node_color='key', node_group='key')
circ.draw()
```

Connected component subgraphs



NetworkX API

```python
import networkx as nx
G = nx.erdos_renyi_graph(n=100, p=0.03)
nx.connected_component_subgraphs(G)
```

```
<generator object connected_component_subgraphs at 0x10cb2c990>
```

```python
list(nx.connected_component_subgraphs(G))
```

```
[<networkx.classes.graph.Graph at 0x10ca24588>,
 <networkx.classes.graph.Graph at 0x10ca244e0>]
```

```python
for g in list(nx.connected_component_subgraphs(G)):
    print(len(g.nodes()))
```

```
99
1
```

Let's practice!

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON

Case study part III: Cliques

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON

Eric Ma

Data Carpentry instructor and author of nxviz package

Cliques

- Definition:
  - Groups of nodes
  - Fully connected
- Simplest clique: edge
- Simplest complex clique: triangle

Maximal cliques

- Definition:
  - A clique
  - Cannot be extended by adding a node

Finding cliques

```python
import networkx as nx
G = nx.erdos_renyi_graph(n=100, p=0.15)
nx.find_cliques(G)
```

```
<generator object find_cliques at 0x10ca8bca8>
```

```python
for clique in nx.find_cliques(G):
    print(len(clique))
```

Let's practice!

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON

Case Study Part IV: Final Tasks

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON

Eric Ma

Data Carpentry instructor and author of nxviz package

Final tasks

- Find important users
- Find largest communities of collaborators
- Build a collaboration recommendation system

Final tasks

- **Find important users**
- Find largest communities of collaborators
- Build a collaboration recommendation system

Final tasks

- Find important users
- **Find largest communities of collaborators**
- Build a collaboration recommendation system

Final tasks

- Find important users
- Find largest communities of collaborators
- **Build a collaboration recommendation system**

Let's practice!

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON

Final thoughts

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON

Eric Ma

Data Carpentry instructor and author of nxviz package

What You've Learned

- The basics of networks and network analysis
- How to find important nodes
- How to identify communities of nodes
- How to apply these concepts in case studies
- How to use the NetworkX and nxviz packages
- How to write network algorithms

Let's practice!

INTRODUCTION TO NETWORK ANALYSIS IN PYTHON