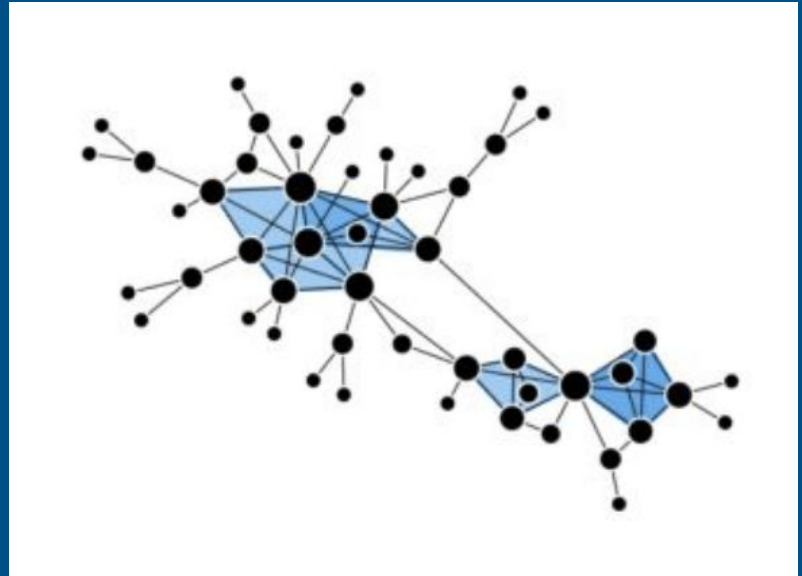


Network visualisation

in Python



1. **From data to visualization**
2. **Static visualizations**
3. **Interactive visualizations**
4. **Hands-on exercise**

1. **From data to visualization**
2. Static visualizations
3. Interactive visualizations
4. Hands-on exercise

From data to visualization

- deciding what network you want to observe (social network, semantic network, biological network, ...)
- getting the data (existing datasets online, scraping/collecting)
- extracting the **nodes** and finding connections between them to create **edges**
- creating network representation (**edgelist**, **adjacency matrix**, **adjacency list**, **dataframe...**)
- analyzing network properties
- visualizing the network

From data to visualization

- deciding what network you want to observe (social network, semantic network, biological network, ...)
- getting the data (existing datasets online, scraping/collecting)
- extracting the nodes and finding connections between them to create edges
- creating network representation (edgelist, adjacency matrix, adjacency list, dataframe...)
- **analyzing network properties**
- **visualizing the network**

Overview of tools

- NetworkX and iGraph (beginner friendly)
- SNAP and NetworKit (for large-scale networks)
- Gephi and Pyvis (focused on visualization)

Overview of tools

- **NetworkX** and **iGraph** (beginner friendly)
- SNAP and NetworKit (for large-scale networks)
- **Gephi** and **Pyvis** (focused on visualization)

What can I learn about my network?

- explore all the concepts taught in the class:
 - degree and degree distribution
 - centrality
 - connectedness
 - modularity

What can I show on a graph?

- whatever you want
- should be relevant to what is interesting about your specific network
- creative ways to achieve what you want

What can I show on a graph?

- node properties: size and color
- edge properties: thickness and color
- network properties: size and connectedness
- layout properties: different layouts to highlight relationships between the nodes – gravity based, modularity based, etc.

1. From data to visualization
2. **Static visualizations**
3. Interactive visualizations
4. Hands-on exercise

Static visualization

- let's head on over to Colab notebook to ...
- ... import the data and create a networkx graph
- ... change into iGraph and visualize
- ... play with node and edge properties

Static visualization – key takeaways

- in code visualization seamlessly integrate analysis and visualization – more control, good for experimenting with your data, faster computation
- Networkx most intuitive to use for analysis, relies on matlab for visualization support
- iGraph enables both visualization and slightly faster computing time
- for larger networks explore more advanced libraries (such as SNAP, Graph-tool, NetworKit)
- for easier control over visualization turn to software such as Gephi

1. From data to visualization
2. Static visualizations
3. **Interactive visualizations**
4. Hands-on exercise

1. From data to visualization
2. Static visualizations
3. Interactive visualizations
4. **Hands-on exercise**

Hands on exercise

- create network (from your data, using Chat GPT, from internet sources)
- import the network into Python and create a networkx graph
- change the graph into iGraph
- calculate different network properties
- visualize different network properties
- try visualizing an interactive plot

Questions & comments

lejla.dzanko@st.swps.edu.pl