# Exercises

## Network structure

Find 3-4 interesting network datasets on Kaggle, preferably undirected network, some could be weighted some could be unweighted network,

Some may have information on node characteristics: different types of nodes (by gender or location, or something else)

The dataset should not be big, its hard to visualize it nicely, or we can just extract a subgraph of a larger dataset.

### Part 1

Plot the network using different types of plot from NetworkX, there are several layouts: Kamada-Kawai, circular etc.

Do different versions: simple one, some showing link weights indicated by the thickness of the links, some may have colored nodes with colors representing node characteristics. It could also be interesting to vary the size of nodes maybe by their degree (number of neighbors).

### Part 2

Compute the following network measures

* Number of nodes, edges, number and size of components
* Degree, weighted and unweighted, average degree, plot the degree distribution, also degree distribution on a log-log scale
* Clustering by node and for the whole network
* Plot correlation between degree and centrality: for each degree from 0 to N, compute the average clustering coefficient of nodes and plot it as a line graph.
* Centrality measures: betweenness, closeness, eigenvector centrality
* Average distance in the graph, diameter

Create a function that returns the top 10 nodes by network characteristics

### Part 3

Generate different graphs using the random graph packages described in the NetworkX reference manual under section 5.6

* Erdos-Renyi random graph
* Small-world network
* Barabási–Albert preferential attachment

Apply the plotting and network measure calculating functions from above on these graphs, to see how the network changes with the parameters of these models.

### Part 4

Use community detection algorithms and apply them on the datasets from Part 1.

We cover the following: Girvan-Newman method, modularity based methods, e.g. greedy algorithm, Ravasz algorithm

The first two are in NetworkX, the Ravasz algorithm I could not find in the reference, but here it is implemented: <https://notebook.community/eflegara/NetStruc/6.%20Community%20Detection>

These algorithms should return a list indicating which nodes belongs to which community

And plot the graph with node colors indicating community membership.

Generate stochastic block models: see NetworkX reference 5.18.9 and apply the community detection algorithms on them.

Use core-periphery detection algorithms and plot core/periphery membership by colors on the network graph. See: <https://github.com/skojaku/core-periphery-detection>

Install cpnet package

### Part 5

Simulate epidemic models on graphs

The most interesting is the SIS model: two parameters: infection rate, recovery rate

Generate a network: let say random graph as in Part 3

Initial state: Infect 5% of the population at random

Each infected node infects each of her neighbors with probability = infection rate

Each infected node recovers with probability.= recovery rate

Iterate this for T periods, in each iteration we go through all N nodes of the network

After the simulation:

Plot the rate of infections over time from period 1 to T

We would like to see how the final infection rate and the dynamics changes as we change the network.