1. Characterising a simple network
   (a) Using networkx, load the airport dataset (graphml)
   (b) Compute the assortativity of the country attribute, using attribute_assortativity_coefficient function. What does it mean?
   (c) Compute the degree assortativity using the degree_assortativity_coefficient function. What does it mean?
   (d) Compare the degree assortativity with a randomized version of the graph, to check if it is similar or significantly different
   (e) Compute the average degrees of neighbors using average_degree_connectivity. You can plot it, for instance as a scatter plot.
   (f) Do the same analysis of degree assortativity and correlation on other networks, for instance those included in networkx such as nx.karate.club_graph() or nx.les.miserables.graph()
   (g) In a previous experiment, you computed communities on the airport graph. You can save those communities as a node attribute in the graph. Get them as a dictionary with partition=louvain.com.to_node_community_map(), and insert them using set_node_attributes. Be careful, CDlib clusterings allow overlap, so in the dictionary, communities are in lists. So you need to do something like \{k:v[0] for k,v in partition.items()\}. Compute the assortativity coefficient of the community structure(s), and compare it with the one obtained for countries. What do you think of the results?

2. Going further : Network→Assortativity→Network
   (a) Using the attribute_mixing_dict function, you can compute the number of edges between nodes of a given attribute. Use this to create a network in which nodes correspond to countries. (tip: the Graph() constructor function of networkx accept dictionaries.)
   (b) Try to plot and characterize quickly this network.
   (c) To obtain a more interesting network, we should keep only links between countries which are not expected by chance, given the degree of countries. Using formula of the assortativity coefficient as an inspiration, propose a way to create such a network. Beware negative values...
   (d) Plot this network and compare it with the one obtained with the naive approach.