A hierarchical consensus clustering approach to community detection   
within a regional labour market network

* Labour market data can provide useful insight on the connections between companies and reveals meaningful clusters
* The basic idea is that clusters are formed on the basis of geo-located opportunities, given by customers, local resources, and the presence of skilled employees.
* When a skilled person changes job, they may pursue new opportunities to value their skills within the same geographical area
* This has been explored by [NatureCom] at a global scale, and also on a regional scale by [DeStefano]
* In this paper we use a consensus clustering approach on community detection within a regional labour marked network

1. Intro
   1. What is a labour market network
   2. What has been done before in LMN analysis
   3. The optimal clusters should be
      1. \*interpretable\*: similar in size, not trivial (e.g. 99% of the network, or just 2 companies), and provide a probability of being a member of that cluster (as in classical fuzzy clustering algorithms)
      2. highly connected intra-cluster and as low as possible inter-cluster
      3. independent of random seeds and parameters, stable
2. Methodology
   1. Building the labour market network: Source of data. Key info are
      1. Professions (to be used also for analysis) 3digit level
      2. Duration (weight)
      3. Location
      4. Industry

* 1. Clustering (compare Eigenvector, Betweenness, Leiden, Loivian). They identify communities on different approaches. Louvian and Leiden use “modularity” to optimiza inter and intra cluster distances. Cluster size distribution varies, and Louvian is the most interpretable. But it depends on a random seed
  2. Consensus clustering. We borrow some ideas from random forrest, DBSCAN and Fuzzy clustering. Repeat the algorithm many times, each with a small variation on the data, parameters and random seed. We obtain hundreds of solutions thate are similar (measured by entropy). Then aggregate the results on a single consensus cluster that is stable. We borrow from DBSCAN the idea to control for trivial-small clusters with a “minPoints” and “minWeight” and we aggregate trivially small clusters in a cluster 0 “fringe” (as in “noise” fro DBSCAN). Finally we calculate a probability of each node to be assigned to each cluster (as in Fuzzy clustering).
  3. Hierarcical consensus clustering:
     1. COMPONENTS are “level-0” clusters. There is a giant component, that needs further to be splitted further, there are trivial components of 1.2 or 3 nodes that will be labelend a “fringe”, and there are some communities of 3-50 nodes that are themselves cuslters (indeed very well defined clusters!)
     2. when the components are larger than 50 units, we recursively apply clustering, to identify the size. And we keep all the smaller clusters as well as cluster 0 (fringe)
     3. We show the results on 3 levels: level-1 clusters (components), level-2 clusters (consensus louvian) and individual companies. Clusters are named by the prevalent location and industry.
  4. We check the results with 2 functions
     1. Inter- and intra- clusters connections (matrix, B&W) mixmat() function
     2. Homogeneity of industry and location (industries and locations are aggregated. E.g. 10 geoclusters by modified dbscan and 5 industries aggregating nace codes). In this case we use a bootstrap approach to extract random induced subgraphs of the appropriate size 1000 times and calculate the distribution of parameter, and compare against the measured valute. We use Gini index.

1. Results: apply the meth above to FVG 2014-2021 8 years. Describe communities with Name (by location, industry and profession)