Consensus community detection on regional labour market network

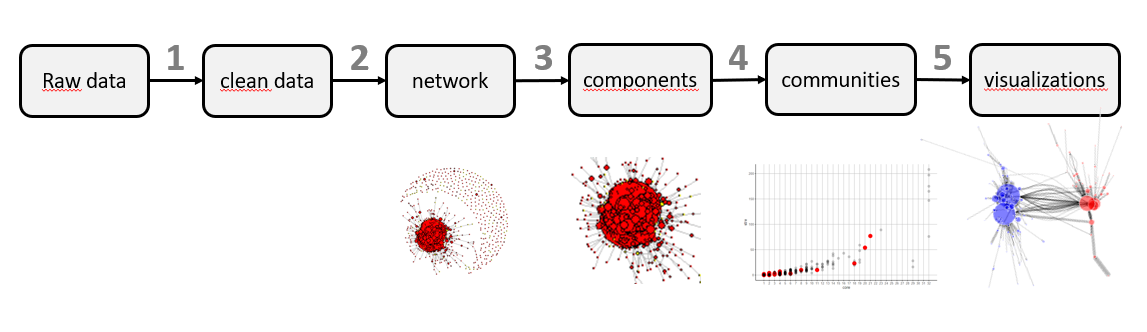
* Labour market data can provide useful insight on the connections between companies and reveals meaningful clusters, by means of community detection algorithms.
* This paper investigates the tasks involved in creating a labour market network, defining an appropriate community detection algorithm and analysing the connections between communities.
* In this paper we use a consensus clustering approach on community detection within a regional labour marked network

# Intro

* 1. What is labour market data and how it has been used with networ kanalysis. 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]
  2. In this paper we investigate how to identify clusters within a labour market network.   
     there are many options to create clusters, that provide different results depending on parameters and random initialization. The main issues are
     1. Trivial clusters that are exceedingly small (composed of a single vertex or a couple of vertices joined by a single edges or) or large (e.g. including over 95% of the vertices),
     2. Results depend on random initialisation
  3. We argue that “good” clusters should be easily interpretable by analysts and policy makers, hence they should have the following properties
     1. A partition (each node belongs to a single cluster); preferably associated with a probability of being a member of that cluster
     2. highly connected intra-cluster and as low as possible inter-cluster (hence modularity is a good option)
     3. informative (avoid trivially small or exceedingly large clusters
     4. independent of random seeds
     5. preferably control size by one or few parameters

# Methodology

The process is illustrated in figure below



Raw data is labour marhet data, that encodes the start and end of eomplyment contracts.

**Step 1**: data preparation, i.e. cleaning, completing missing lines and addressing known issues (such as missing contract terminations). A pseudonymization is carried out at this stage.

Clean data is composed of two tables: contracts and transitions. It is still pseudonymous

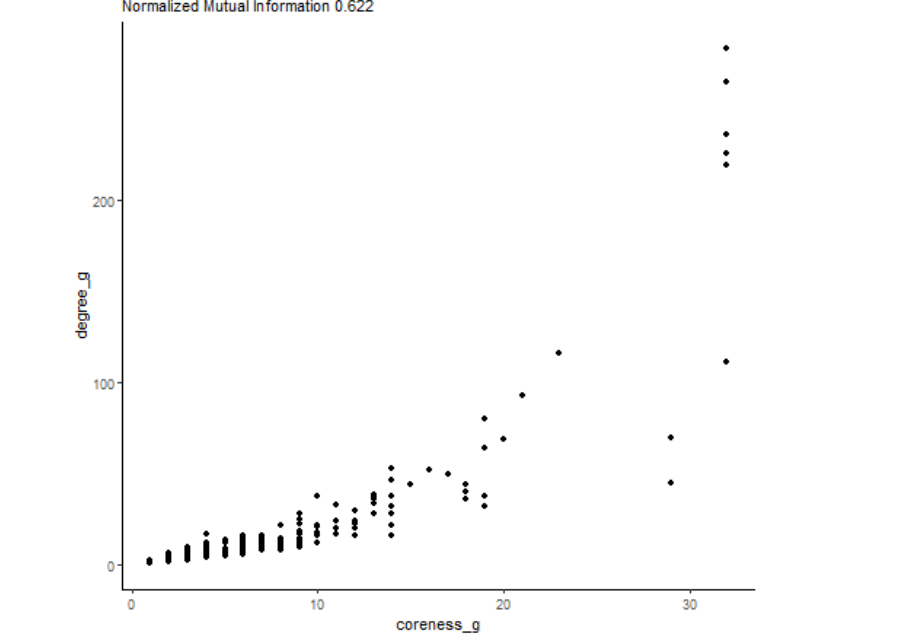
**Step 2**: network construction. The core decisions at this stage are to select relevant transitions for the analysis (e.g. by profession, year, location, …) and to assign a weight to each transition. Relevant information is associated with edges: professional group, year, location, sector.

Weighting strategy influences the results, and should be carefully selected. A basic selection is to give the same weight to all edges (which is simple, but does not exploit the potential of the data)

We calculated the weight of the edge connecting organization A with organization B as the minimum duration of the contracts in A and B (expressed in years). Moreover, all values above a threshold (max\_weight = 1.0) are limited to threshold value. As shown in the figure below, weights are evenly distributed between 0.0 and 0.99, with a sharp peak at 1.0.

|  |  |
| --- | --- |
| Histogram of weight | Histogram of strength |
|  |  |

The resulting distribution of weights within the network is illustrated in the figure below: most of the nodes have a very small strength (sum of weights) and only 5 nodes exceed strength = 50.



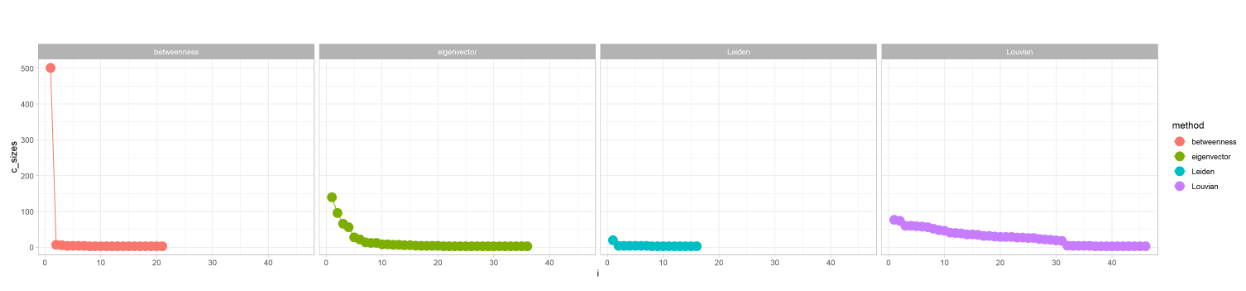
Additional rules can be applied: for example we can assign decreasing weights as a function of time, with respect to a reference year, using a discount factor calculated according to the NPV methodology

The network is composed of nodes (organisations) and edges (transitions). It can be visualized and analyzed as a whole. Centrality measures include degree, strength and coreness. Attributes are categorical variables, that may be aggregated in a small number of groups to improve the analysis.

**Step 3**: component analysis [explain]

Components comply with the most of the requirements of a cluster stated above, but are generally not informative. In our case a giant component includes most of the nodes, and the rest is fragmented in signletons, 2- or 3- cliques. We proceed with the next steps using the giant component (but if the graph contains more relevant components, the same procedure applies for each component).

**Step 4**: community detection. There are many well established techniques for community detection, which provide slightly different results. We compared edge betweenness, eigenvector, Leiden and Louvain. All the algorithms use only the weight associated with edges (other edge attributes are not relevant at this stage)



We argue that Louvain produces the most interesting results (a number of non-trivial clusters). Edge betweenness results in a giant community; all the others produce exceedingly small clusters.

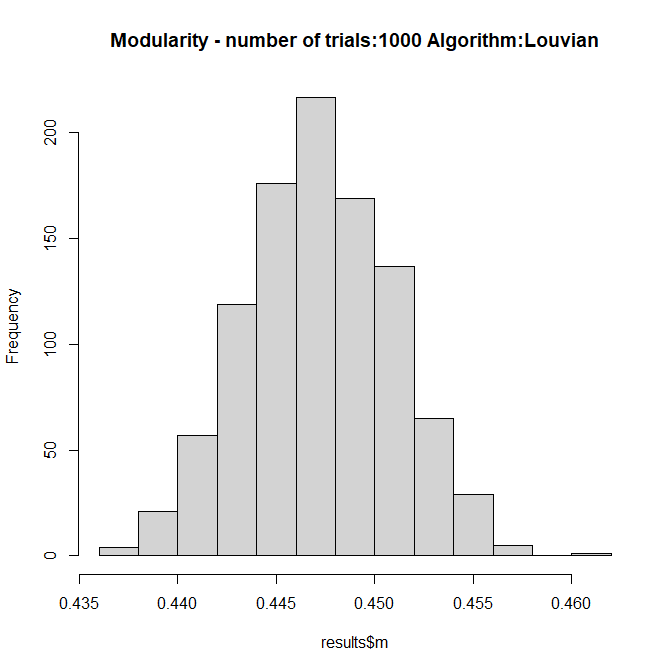
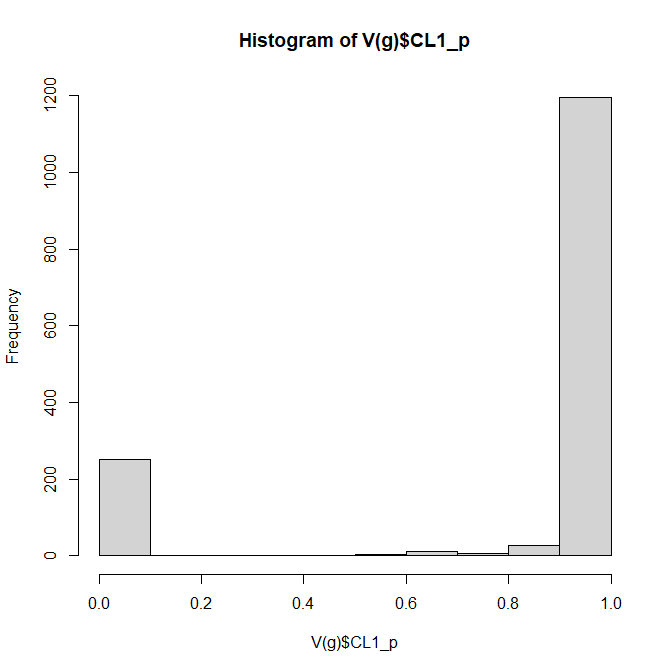
Louvain has an advantage: it minimizes modularity, so it complies with the requirement of optimizing iter- and intra- cluster connections

However, Louvain has a drawback: results depend on a random initialization, thus vary at each execution. This may not be a relevant issue, since it involves marginal elements, but we suggest to address it as in the random forrest algorithm: repeat the clustering algoritm a number of trials, and select an optimal result.

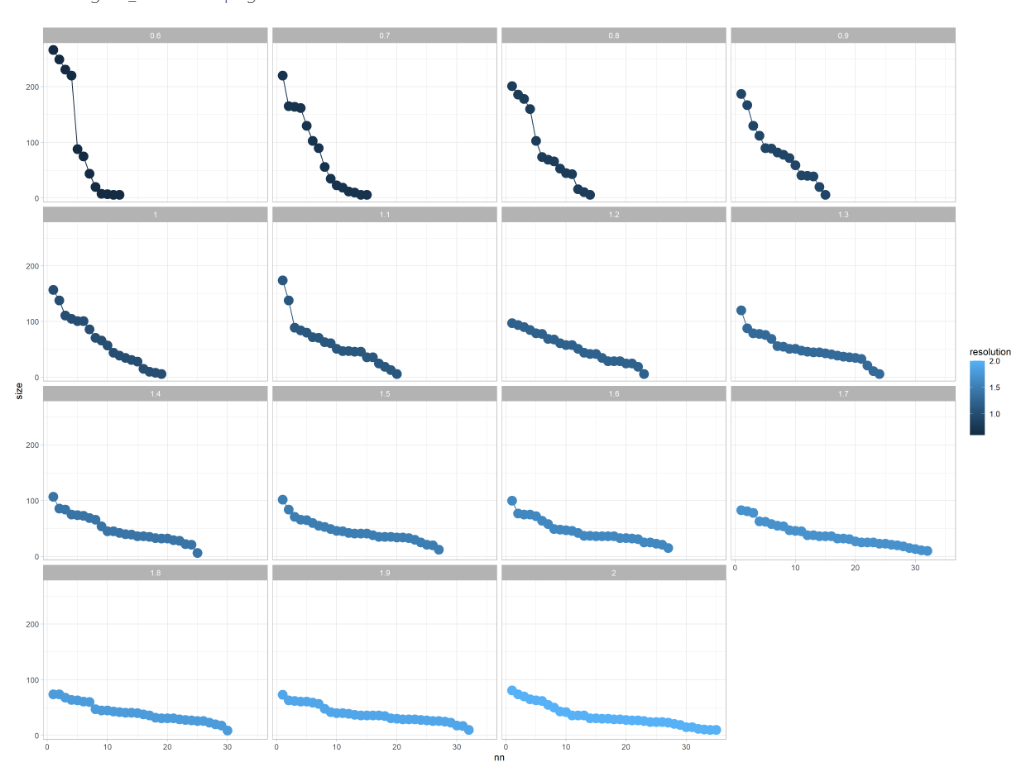
A basic strategy is to select minimum modularity (clear and explainable but still dependent on the random seed)

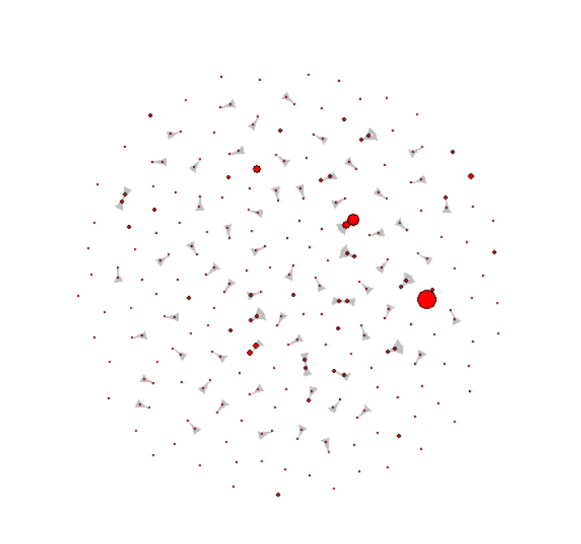
We argue that a better strategy is to apply a consensus clustering to the results of Louvain trials. Moreover, as in the random forrest algorithm, each trial is slightly perturbated by cancelling out a small fraction of the weights. This induces more variability. A key advantage of consensus clustering is that it allows to calculate a probability of membership.

FIGURE: boxplot variability of modularity and probability of membership (depending on alpha = 0.0 to 0.5)

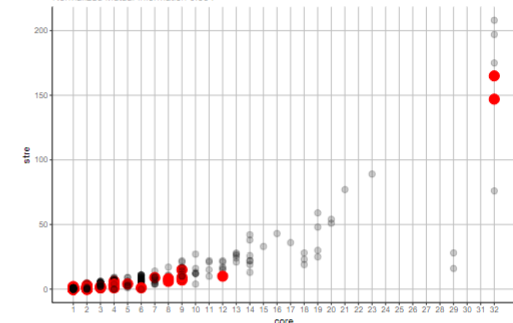
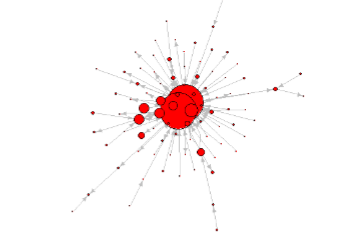
 

Resolution parameter allows to change community size. In our case resolution below 1.0 generates a small number of large communities, while resolutions above 2.0 generates more evenly distributed size.



An additional feature of this procedure is to group weakly linked vertices and “small” communities in a “cluster 0”. A vertex is assigned to cluster 0 if its maximum probability of membership is below a given threshold (min\_prob = 50%), meaning that it has weak links with 3 or more clusters and cannot be clearly assigned to any. Moreover small community are defined as having less than a given number of vertices (in our case set to min\_vids = 4) and a total weight below a given threshold (min\_weight = 0.1% of total network weight. This allows the analyst to focus on a number of relevant communities. Setting min\_prob, min\_vids and min\_weight values allows to modulate the size of communities to be taken into account for further analysis).

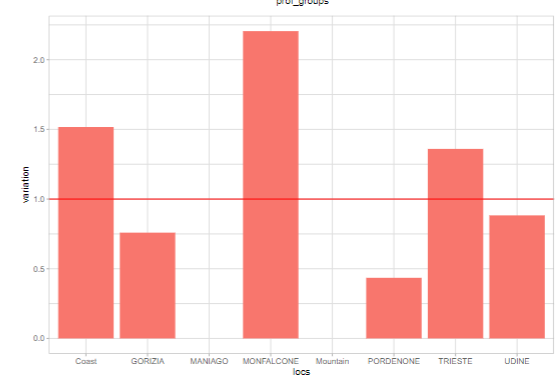
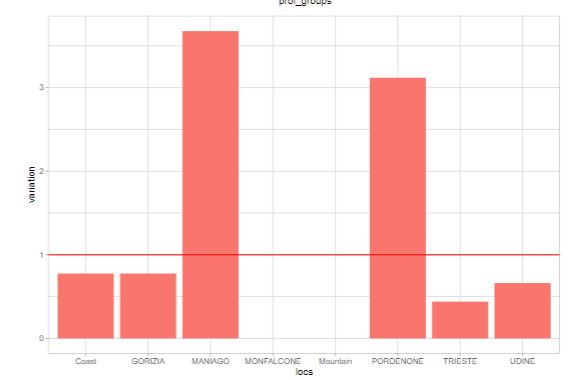
The result at this stage is community assignation. Each vertex (organization) is assigned with a community identifier and a probability of membership. Vertices are further characterized by strength and coreness, which allows to examine the structure of a community and its position within the whole network



**Step 5**: analysis and visualization of communities. Edge attributes can be analyzed for each community/cluster.

We analyzed the relative frequency of professions, locations and sectors. Some clusters are characterized by a prevalence of professions (e.g. cluster 3, mainly research organizations in Trieste) while others are characterized by sector or location.

Figure below shows cluster 8 (jobs located mainly in Maniago and Pordenone) and cluster 13 (organisations located in Monfalcone, Trieste and the coast)



Moreover, the relations between communities can be analyzed.

|  |  |
| --- | --- |
|  |  |
| Graph of clusters | heatmap |

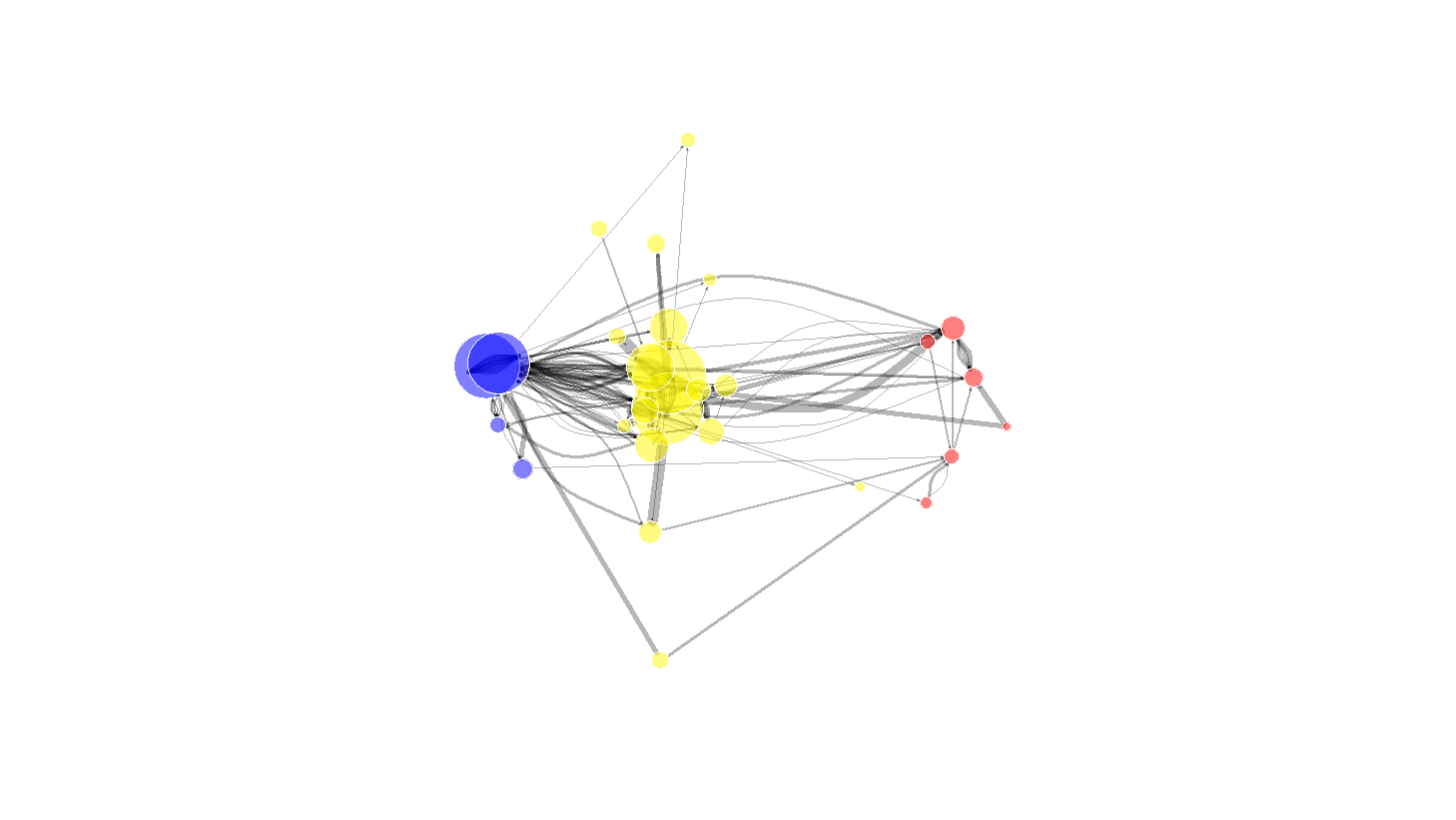
More in detail: an induced subgraph of 2 clusters, expanded along horizontal azis proportional to probability.

|  |  |
| --- | --- |
| Immagine che contiene cielo, esterni  Descrizione generata automaticamente | Immagine che contiene cielo, esterni, linea, parecchi  Descrizione generata automaticamente |
| communities 2 and 7 (strong connections)(research orgs) | communities 8 and 13 (weak connections) |

Finally compare 2 clusters and neighborhood (distanced by coreness)

Communities 2 and 7 (university and research) and common neighborhood order 1

Check how many startups and patenting; spill over of research



* 1. Building the labour market network: Source of data. Key information is
     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)