# Project mid-term report - Road Graphs

**Carraro Eddie - 2121248**
**Gava Mattia - 2130893**
**Giacomin Marco - 2119285**

**Title**: Exploring the correlation between a country's history/geography and its road network graph

## Intended experiments:

- We will use the python language together with the graph analysis library networkx (networkx.org) which implements many common graph operations
- Machine for experiments: We plan to interchangeably use the personal computers of the members of this group (a linux pc with 8gb RAM, a windows pc with 8gb RAM, a windows pc with 12 gb RAM. None posess a GPU). Should they prove insufficiently powerful for this task we will make use of Google Colab
- Experiments:
- ○ For each graph: compute average closeness centrality and betweenness centrality, clustering coefficient and number of motifs up to a certain number of nodes depending on computational time
- ○ Save each computed statistic as a feature in a multi-dimensional space describing each graph
- ○ Generate some random graphs based on the actual road networks and compute the same statistics for them
- ○ Employ a clustering algorithm to group together road networks that share similar characteristics
- ○ Verify if states with similar histories and geographies are grouped together in the same cluster and if they are distinguished from randomly generated networks

## What changed from the proposal

For what concerns the **features to find**, doing some trials of execution in our machines, we noticed that the time of execution when finding the features we wanted to find first (closeness, betweenness and clustering coefficient) was very high when using the *networkx* package. So, we tried using *igraph* which is written in C (and so it should be faster than *networkx* for sure) and we noticed an improvement in the execution time, but it was not enough for the bigger graphs we have to deal with (which have a number of nodes and edges of the order of millions). To solve this issue, we are both trying to use the academic cluster **Blade** in order to try the execution using a more powerful machine, and we are also trying to find some other relevant features to accomplish our initial target.

For **random graphs**, the choice of model is crucial to accurately simulate the characteristics of the graphs we are analyzing. Specifically, we need to focus on random graphs that are connected. In fact, we verified that even the road network of Italy is connected, which suggests that the major parts of the country are connected with the islands. This connectivity might be achieved through existing marine routes, such as those in the Messina Strait.

Additionally, the nodes in these graphs should only be connected if their real-world distance is within a reasonable range.

After conducting some research, we identified a few random graph models that may be useful for this task:

1. **The Watts-Strogatz Model**: This model generates graphs with small-world properties, which are particularly useful for representing networks where nodes are grouped closely together and connected to one another.
2. **The Random Geometric Graph**: In this model, N nodes are placed randomly in a metric space, following a specified probability distribution. Two nodes are connected by an edge if their distance is within a certain range, such as a specified neighborhood radius, r. The only problem with this approach is that it connects every pair of nodes under this radius, so the randomness factor is lost.

Obviously, we need to use a Monte-Carlo approach in order to compute the properties we need to compare the random graphs with the initial ones.

*References:*
https://chih-ling-hsu.github.io/2020/05/15/Graph-Models#watts-strogatz-model
https://en.wikipedia.org/wiki/Random_geometric_graph