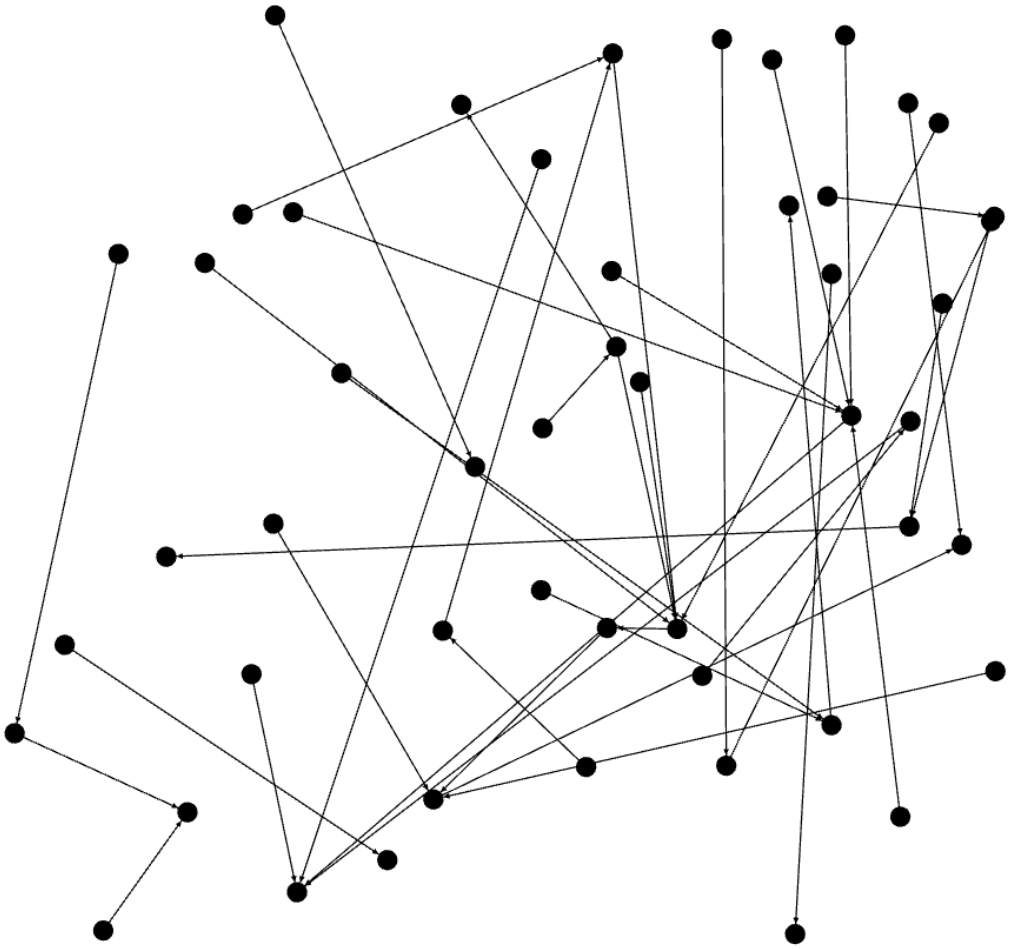
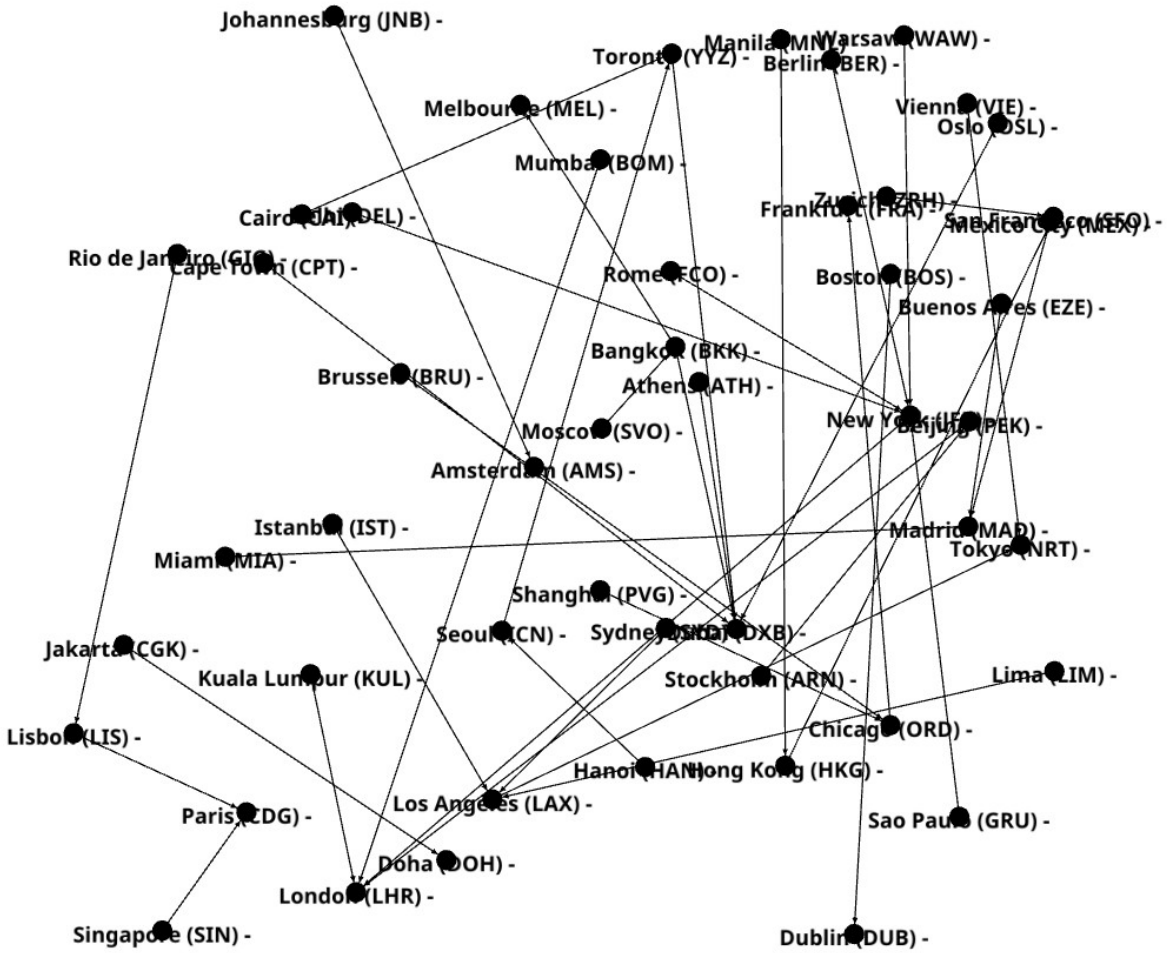
1. Load the Dataset

Open Gephi and import "air_routes_practice_gephi.csv" as a Directed Graph.



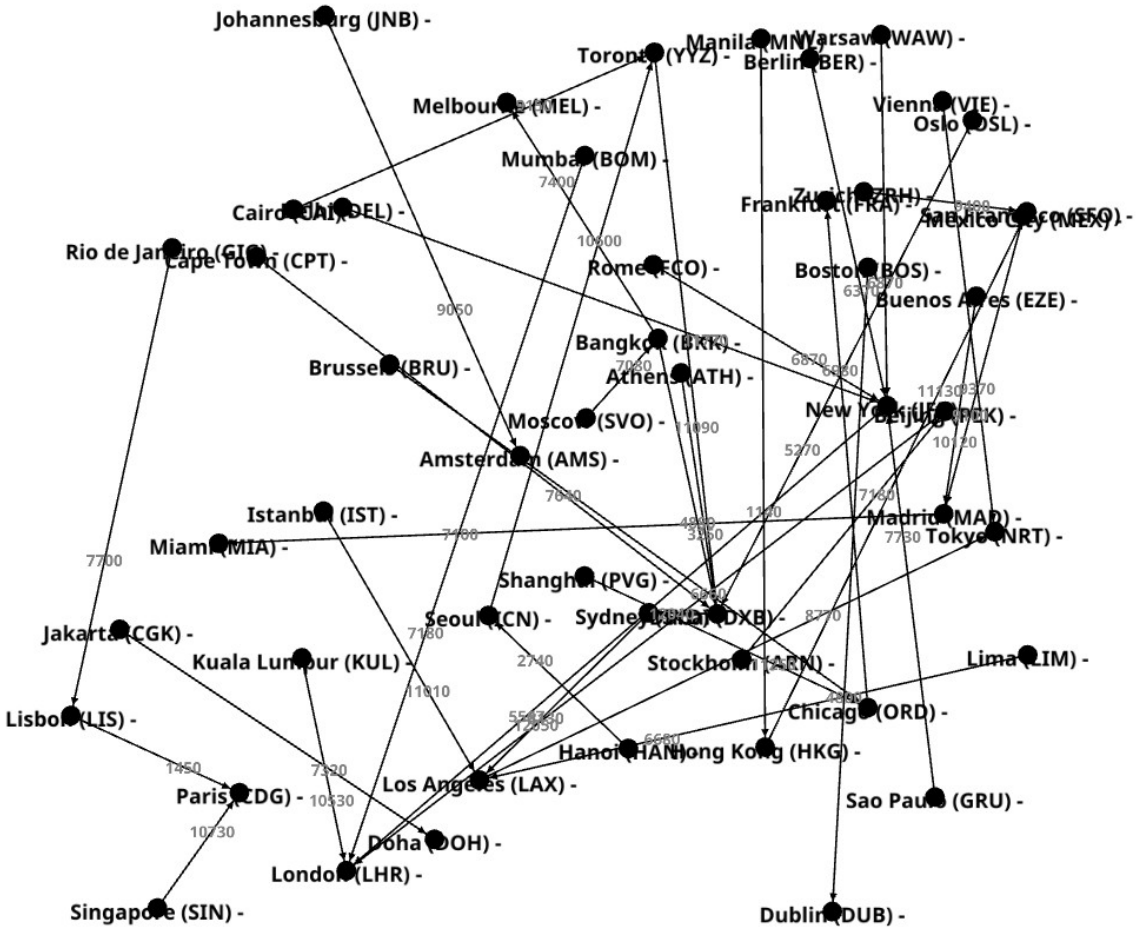
2. Show Node Labels (Take Screenshot)

Display node names (airport names or codes) on the graph.



3. Show Edge Labels (Distance) (Take Screenshot)

Display the distance on each edge in the graph.

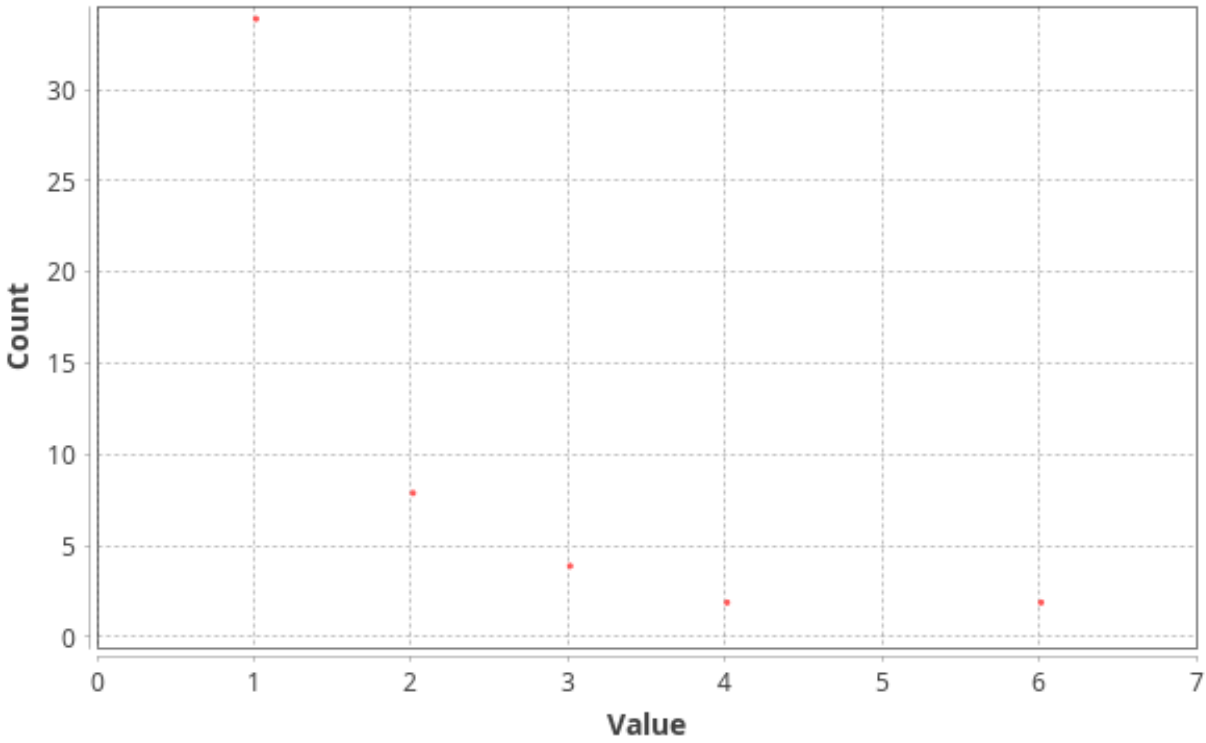


4. Calculate Node Degrees

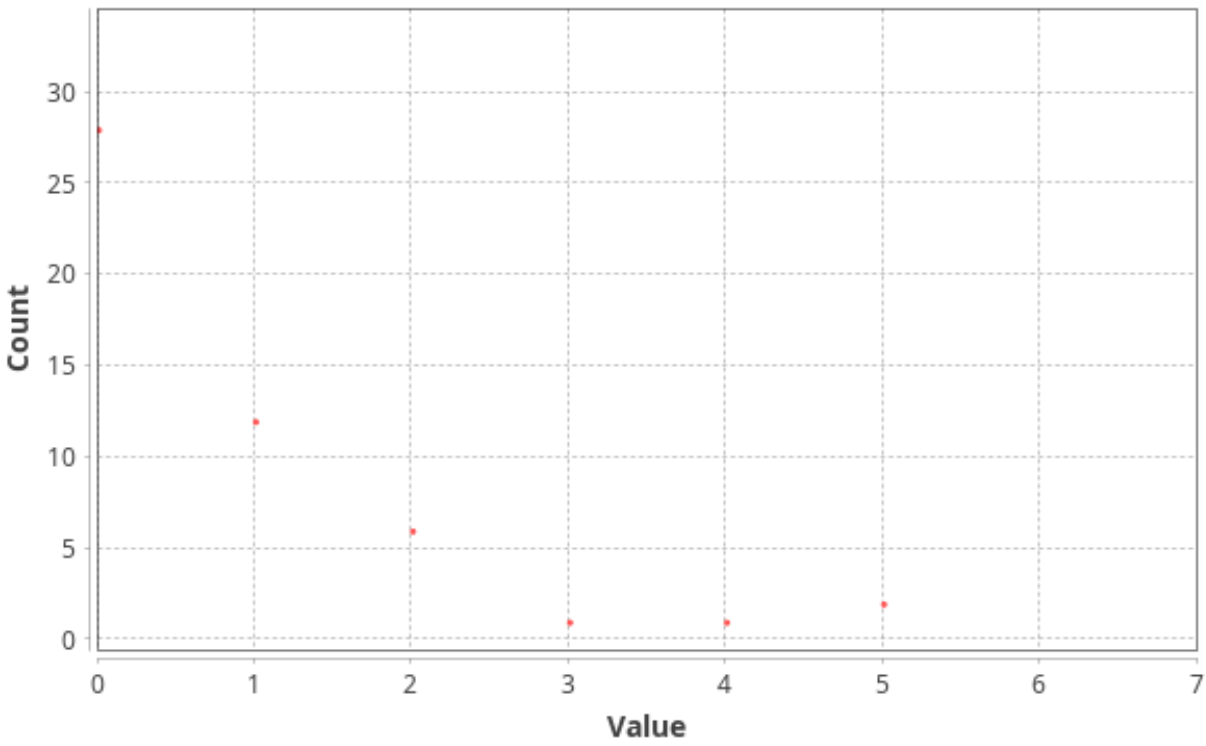
Find Indegree, Outdegree, and Total Degree of each node.

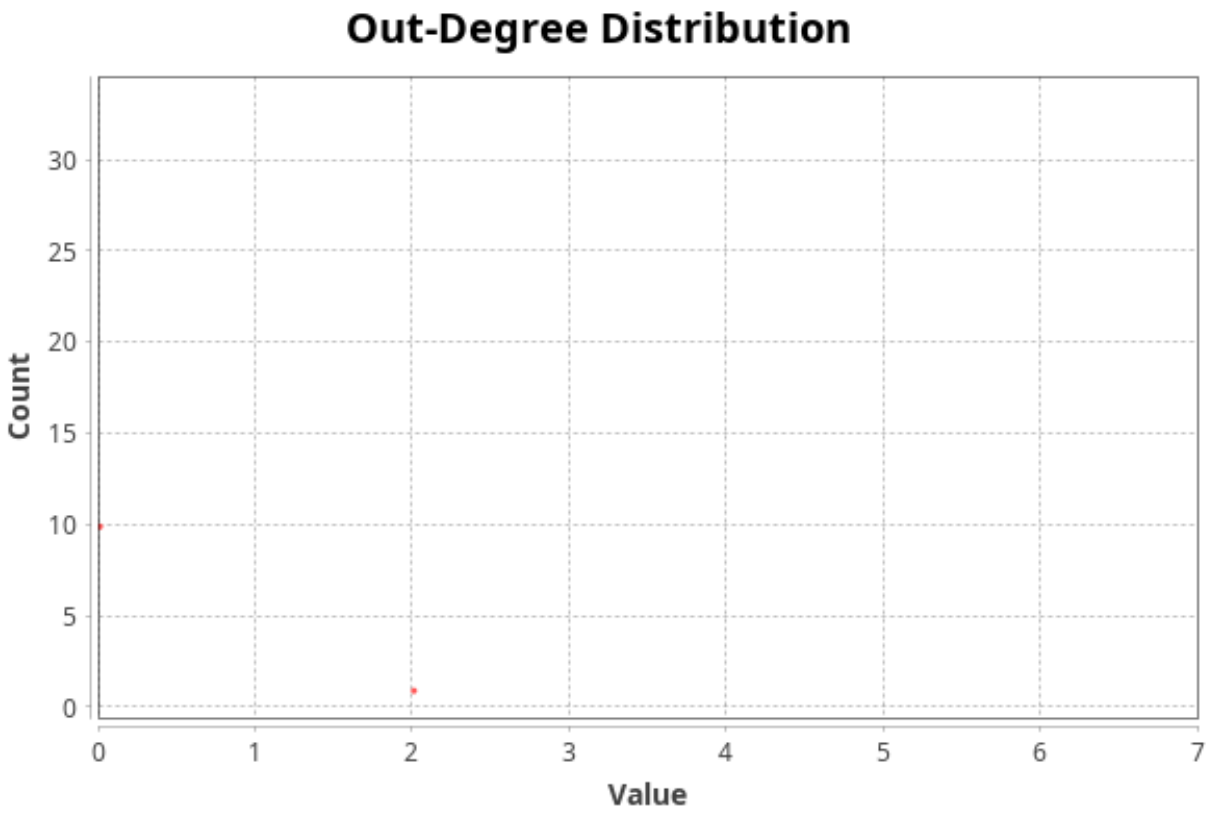
View the values in the Data Laboratory.

Degree Distribution



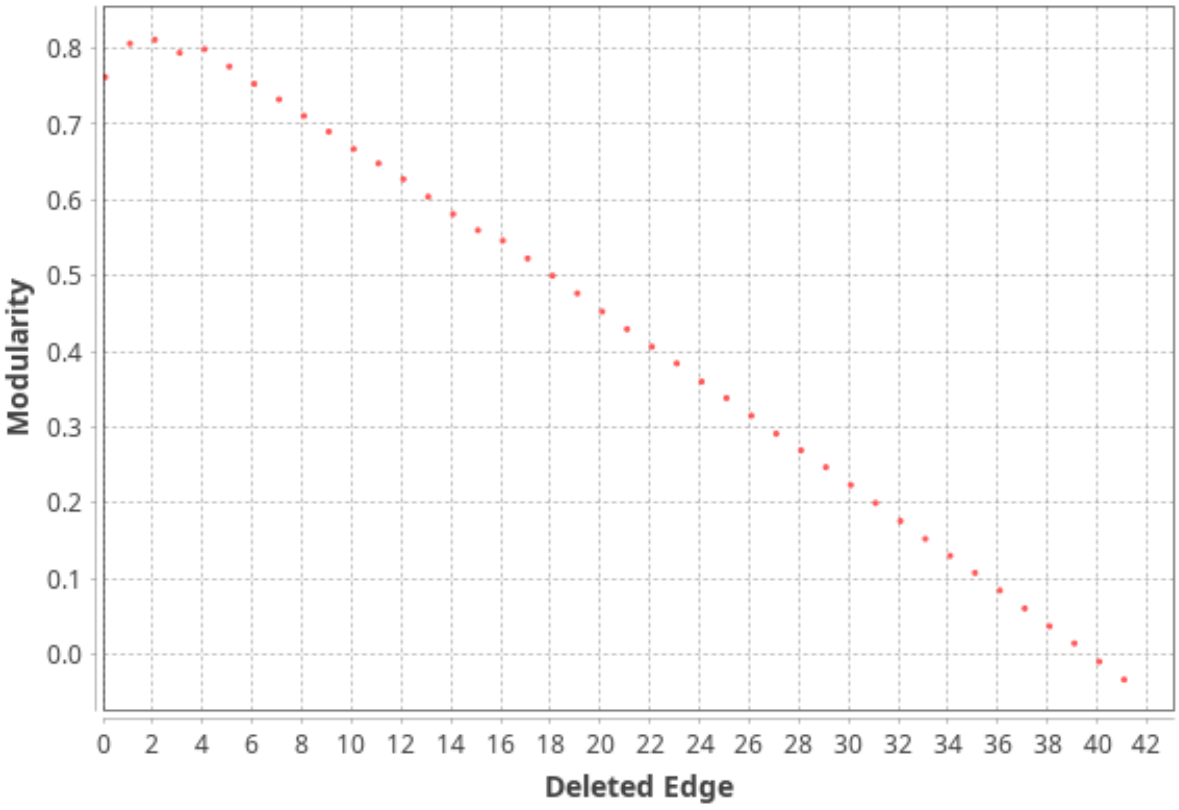
In-Degree Distribution





5. Apply Girvan-Newman Algorithm

Run the Girvan-Newman community detection (available as a plugin).



Parameters: Respect edge type for shortest path betweeness: yes Respect parallel edges for shortest path betweeness: no Respect edge type for modularity computation: no Respect parallel edges for modularity computation: no

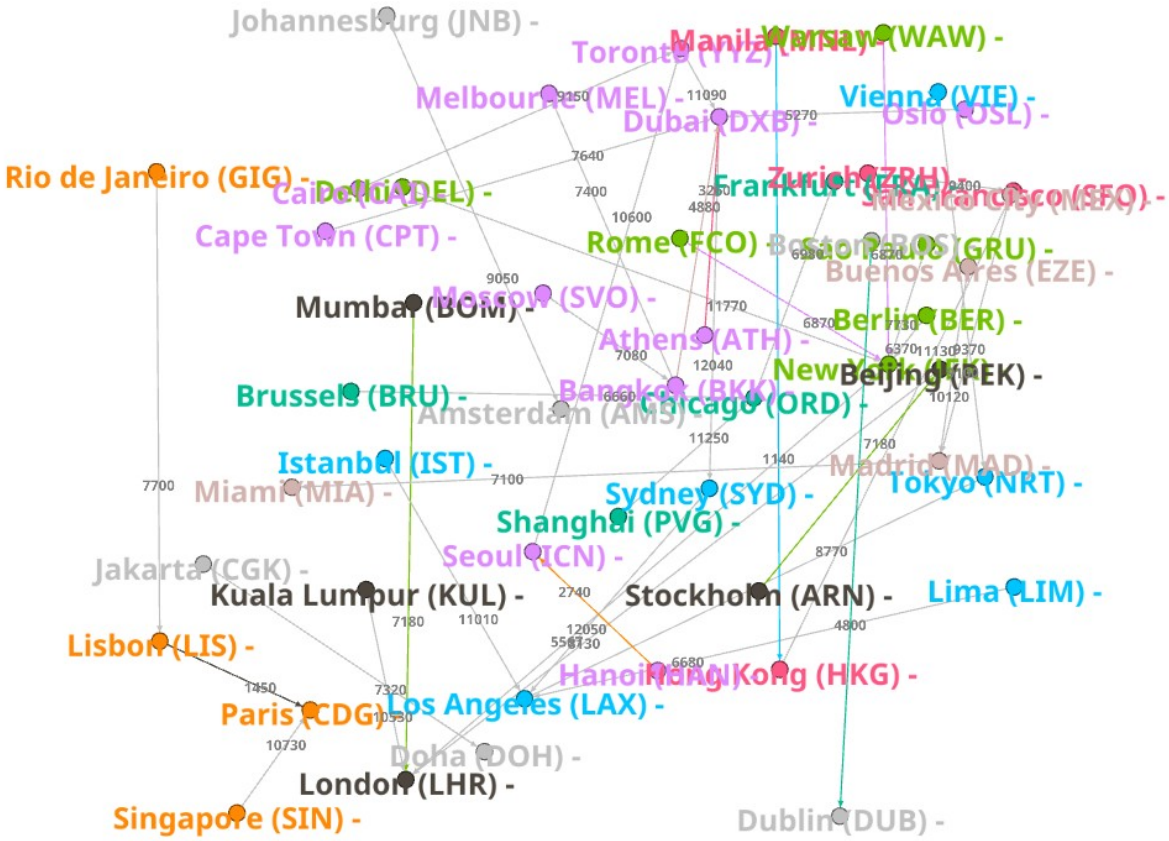
Processed Graph Data Nodes: 50 Edges 41

Processing time: 0.064 sec.

Communities Number of communities: 11 Maximum found modularity: 0.8143962

6. Color Nodes & Edges by Community (Take Screenshot)

Change node and edge colors based on their community ID.



7. List Communities

Identify which nodes (airports) belong to which community. (Text Description)

```
s = """Id,Label,indegree,outdegree,Degree,Cluter-ID
New York (JFK),,5,1,6,0
London (LHR),,4,0,4,10
Los Angeles (LAX),,3,1,4,9
Tokyo (NRT),,2,0,2,9
Dubai (DXB),,5,1,6,1
Sydney (SYD),,1,1,2,9
Singapore (SIN),,0,1,1,2
Paris (CDG),,2,0,2,2
Hong Kong (HKG),,1,1,2,3
San Francisco (SFO),,2,0,2,3
Chicago (ORD),,2,1,3,4
Frankfurt (FRA),,1,0,1,4
Toronto (YYZ),,2,1,3,1
```

```
Mumbai (BOM),,0,1,1,10
Delhi (DEL),,0,1,1,0
Bangkok (BKK),,1,2,3,1
Melbourne (MEL),,1,0,1,1
Istanbul (IST),,0,1,1,9
Shanghai (PVG),,0,1,1,4
Madrid (MAD),,2,1,3,5
Miami (MIA),,1,0,1,5
Seoul (ICN),,1,1,2,1
Johannesburg (JNB),,0,1,1,6
Amsterdam (AMS),,1,0,1,6
Rio de Janeiro (GIG),,0,1,1,2
Lisbon (LIS),,1,1,2,2
Mexico City (MEX),,0,1,1,5
Beijing (PEK),,1,1,2,10
Cape Town (CPT),,0,1,1,1
Jakarta (CGK),,0,1,1,7
Doha (DOH),,1,0,1,7
Berlin (BER),,0,1,1,0
Kuala Lumpur (KUL),,0,1,1,10
Zurich (ZRH),,0,1,1,3
Sao Paulo (GRU),,0,1,1,0
Athens (ATH),,0,1,1,1
Cairo (CAI),,0,1,1,1
Manila (MNL),,0,1,1,3
Hanoi (HAN),,0,1,1,1
Moscow (SVO),,0,1,1,1
Buenos Aires (EZE),,0,1,1,5
Warsaw (WAW),,0,1,1,0
Oslo (OSL),,0,1,1,1
Lima (LIM),,0,1,1,9
Stockholm (ARN),,0,1,1,10
Brussels (BRU),,0,1,1,4
Rome (FCO),,0,1,1,0
Boston (BOS),,0,1,1,8
Dublin (DUB),,1,0,1,8
Vienna (VIE),,0,1,1,9"""
l = s.split("\n")
l = [i.split(',') for i in l]

import pandas as pd
df = pd.DataFrame(l[1:], columns=l[0])

for cluster_id, group in df.groupby("Cluter-ID"):
    print(f"Cluster {cluster_id}: {list(group['Id'])}")

Cluster 0: ['New York (JFK)', 'Delhi (DEL)', 'Berlin (BER)', 'Sao
Paulo (GRU)', 'Warsaw (WAW)', 'Rome (FCO)']
Cluster 1: ['Dubai (DXB)', 'Toronto (YYZ)', 'Bangkok (BKK)',
'Melbourne (MEL)', 'Seoul (ICN)', 'Cape Town (CPT)', 'Athens (ATH)',
```

```
'Cairo (CAI)', 'Hanoi (HAN)', 'Moscow (SVO)', 'Oslo (OSL)']
Cluster 10: ['London (LHR)', 'Mumbai (BOM)', 'Beijing (PEK)', 'Kuala
Lumpur (KUL)', 'Stockholm (ARN)']
Cluster 2: ['Singapore (SIN)', 'Paris (CDG)', 'Rio de Janeiro (GIG)',
'Lisbon (LIS)']
Cluster 3: ['Hong Kong (HKG)', 'San Francisco (SFO)', 'Zurich (ZRH)',
'Manila (MNL)']
Cluster 4: ['Chicago (ORD)', 'Frankfurt (FRA)', 'Shanghai (PVG)',
'Brussels (BRU)']
Cluster 5: ['Madrid (MAD)', 'Miami (MIA)', 'Mexico City (MEX)',
'Buenos Aires (EZE)']
Cluster 6: ['Johannesburg (JNB)', 'Amsterdam (AMS)']
Cluster 7: ['Jakarta (CGK)', 'Doha (DOH)']
Cluster 8: ['Boston (BOS)', 'Dublin (DUB)']
Cluster 9: ['Los Angeles (LAX)', 'Tokyo (NRT)', 'Sydney (SYD)',
'Istanbul (IST)', 'Lima (LIM)', 'Vienna (VIE)']
```