hmw-1

Key Network Properties

1-

a)

There are 3 147 airports.
There are 66 679 flights, to be a different flight it must only have one different aspect.

b)

I recreated this network on Gephi and just used the native functionality.

- diameter: 3
- avg path length: 1.87

c)

- Node 0 (A): C = 0.0
- Node 1 (B): C = 0.0
- Node 2 (C) : C = 0.0
- Node 3 (D) : C = 0.3333333432674408
- Node 4 (E) : C = 1.0
- Node 5 (F) : C = 1.0
- Average C = 0.3888888905445735

d)

(normalized) betweenness centrality:

- Node 0 (A): 0
- Node 1 (B): 0.7
- Node 2 (C) : 0
- Node 3 (D) : 0.6
- Node 4 (E) : 0
- Node 5 (F) : 0

(normalized) closeness centrality:

- Node 0 (A): 0.46
- Node 1 (B): 0.71

- Node 2 (C) : 0.46
- Node 3 (D) : 0.71
- Node 4 (E) : 0.5
- Node 5 (F) : 0.5

2)

I assume that the transitivity gives more weight to high degree nodes because in those situations the average clustering coefficient is high so will be the transitivity but not the other way around, because a click has the maximum number of triangles but one can be in a high number of triangles (expecially if the degree is high) but none of them help to form a triangle.

Using Gephi

a)

By looking at the "Context" tab one can see that there are 3 147 nodes (airports) and 66679 edges (different flights). Different flights are any connections between different airports that have any different property, e.g. airline.

b)

- outgoing flights:
  Is the out-degree of each node, without merging edges.
  The out degree is just:

$$average\ out\ degree = \frac{number\ of\ edges}{number\ of\ nodes} = 21.19$$

- to different airports:
  Is the out-degree of each node, not accepting parallel edges.
  The out degree is just:

$$average\ out\ degree = \frac{number\ of\ edges}{number\ of\ nodes} = 11.70$$

c)

Using the built in Gephi functionality on the "Statistics" tab, one can easily check that the diameter is 13 and the average path length is 3.97.

d)

Finding the pair of airports with more flights between each other is the same as finding the edge with the highest weight in the undirected version of the graph summing weights on import.
The result are the Hartsfield Jackson Atlanta International Airport and the Chicago O'Hare International airports.

e)

I found that the airports with the highest number of different outgoing flights are

1. Hartsfield Jackson Atlanta International Airport
2. Chicago O'Hare International Airport
3. Beijing Capital International Airport

f)

The betweenness centrality can be calculated with the "Statistics" tab on the "Overview" pane, then ordered in "Data Laboratory":

1. Los Angeles International Airport
2. Charles de Gaulle International Airport
3. London Heathrow Airport
   for this I used the not-merged version of the graph so that airports would not be penalized for having more of what we want to measure with this measure

g)

Ted Stevens Anchorage International Airport has the 8^{th} highest betweenness centrality and in 273^{th} place for the out-degree measure. For beetwenness I just looked at the ordered table, for the out-links I filtered for those with the same number of out-links or higher and looked at "Context".

Winnipeg / James Armstrong Richardson International Airport is another example of an airport with high betweenness centrality and low number of out-links.

This will happen everytime a airport with low number of flights is connects two or more tightly innerconnected communities that are very separate from each other.

h)

By partitioning the nodes by country we get that the countries with more airports are:

1. USA
2. Canada
3. China

i)

By partitioning the edges, we get that the airlines with the highest number of flights are

1. FR - Ryanair
2. AA - American Airlines
3. UA - United Airlines

j)

By partitioning again we get that the number of domestic US flights is 10487.

k)

Filtering we get that 42 chinese airports have at least 50 outgoing different flights. Using a subfilter.

l)

Partitioning "Intra edges (country)" we get that there are 24 different flights between Portugal and Brazil.

m)

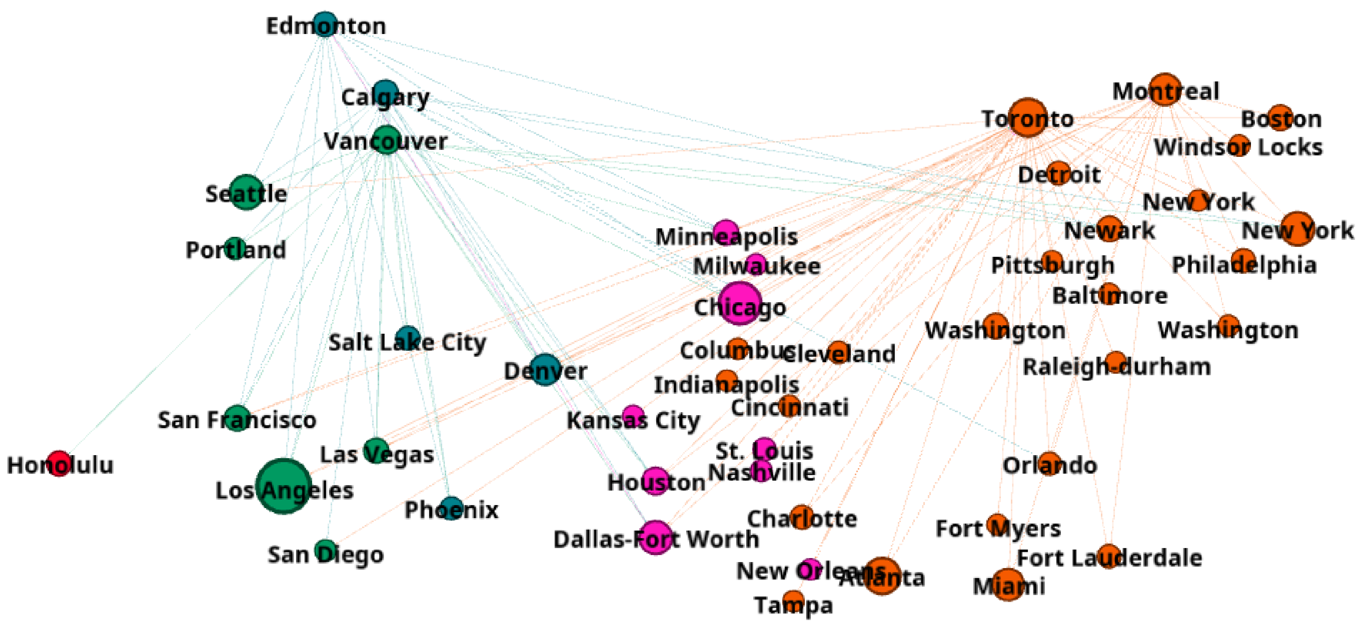
By filtering by airline FR and topology giant component one gets 176 airports with 1484 flights.

The airport with highest closeness centrality of all of those is Leonardo da Vinci–Fiumicino Airport.

n)

Using the functionality "ego network", 61 airports are at distance 1 from Francisco de Sá Carneiro Airport, 755 at distance 2 or less, and 3147 at distance 3 or less.

o)



Erdos-Renyi Model

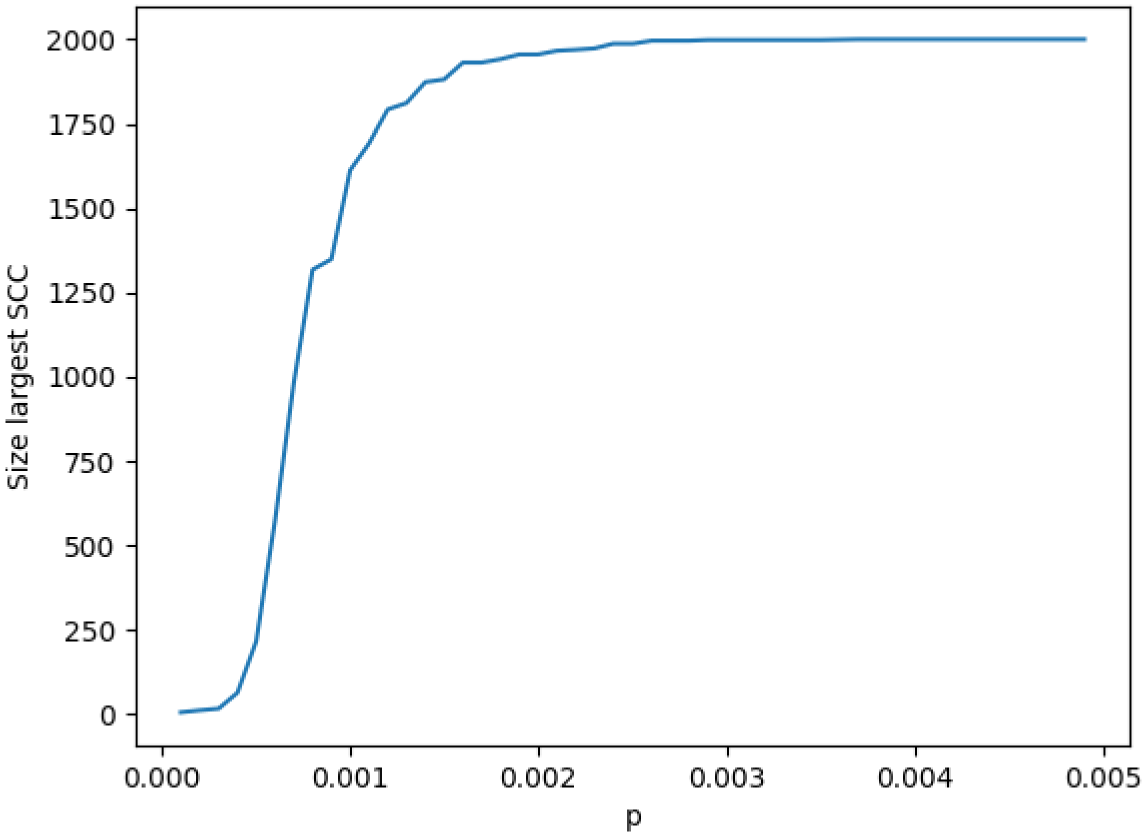
4)

The code is in the file tarjan-undirected.py
binomial-erdos.py contains the same utilities to print those graphs but the graphs themselves are directed.

5)

The code is in the file tarjan-undirected.py
The function tarjan(), inside the class Graph automatically divides the the nodes in strongly connected components then the function getSizeGiantComponent() returns the size of the largest scc.

6)



Based on the literature, the behaviour seen is exatly what was expected:

- If np < 1, then a graph in G(n, p) will almost surely have no connected components of size larger than O(log(n)).
- If np = 1, then a graph in G(n, p) will almost surely have a largest component whose size is of order n2/3.
- If np → c > 1, where c is a constant, then a graph in G(n, p) will almost surely have a unique giant component containing a positive fraction of the vertices. No other component will contain more than O(log(n)) vertices.

The reader can see this behaviour if he knows that the number of nodes was always 2 000.

Barabási-Albert Model

7)

The file that generates those networks is barabasi_albert.py

8)

For this exercise I used a logarithm for both the x an y data then used linear regression then exponentiated the x and y data from the regression line.
I don't like this method nor do I think the results are that trustworthy so I added an extra function: getAlpha(), that uses another method presented in the same slides and the result is closer to 3.