Information Retrieval
Assignment-3

Kush Jain (MT22107)
Sarthak Jain (MT22064)
Prajeet P. Sahoo(2020393)

Question 1

This code performs an analysis of a dataset containing information about the voting behavior of editors of Wikipedia, where each node represents an editor and an edge represents a vote cast by one editor to another.

Report:

Methodologies:

- The code uses various libraries such as pandas, numpy, and matplotlib.pyplot to load, analyze and visualize data.
- The data is loaded from a text file 'Wiki-Vote.txt' using the numpy.loadtxt() method with parameters dtype, comments, and delimiter to specify the data type, comments format, and delimiter respectively.
- A node_map is created using a for loop to enumerate and store the unique nodes in the data.
- An adjacency matrix is created using a pandas DataFrame with the same size as the node_map and filled with zeros.
- Another for loop is used to iterate over the data and set the corresponding value in the adjacency matrix to 1 to represent an edge between nodes.
- The code then computes the number of nodes, edges, average in-degree, average out-degree, maximum in-degree node, maximum out-degree node, and density of the graph.
- The in-degree and out-degree distributions are plotted using the matplotlib.pyplot.hist() method with parameters bins, alpha, color, edgecolor, linewidth, log, xlabel, ylabel, and title to specify the number of bins, transparency, colors, and labels for the x and y axes and the title.
- The adjacency matrix is then renamed using the node_map.
- The local clustering coefficient is computed for each node using a for loop that iterates over the adjacency matrix and calculates the clustering coefficient using the formula: 2 * E / (k * (k - 1)), where E is the number of edges between the neighbors of the node, and k is the number of neighbors.

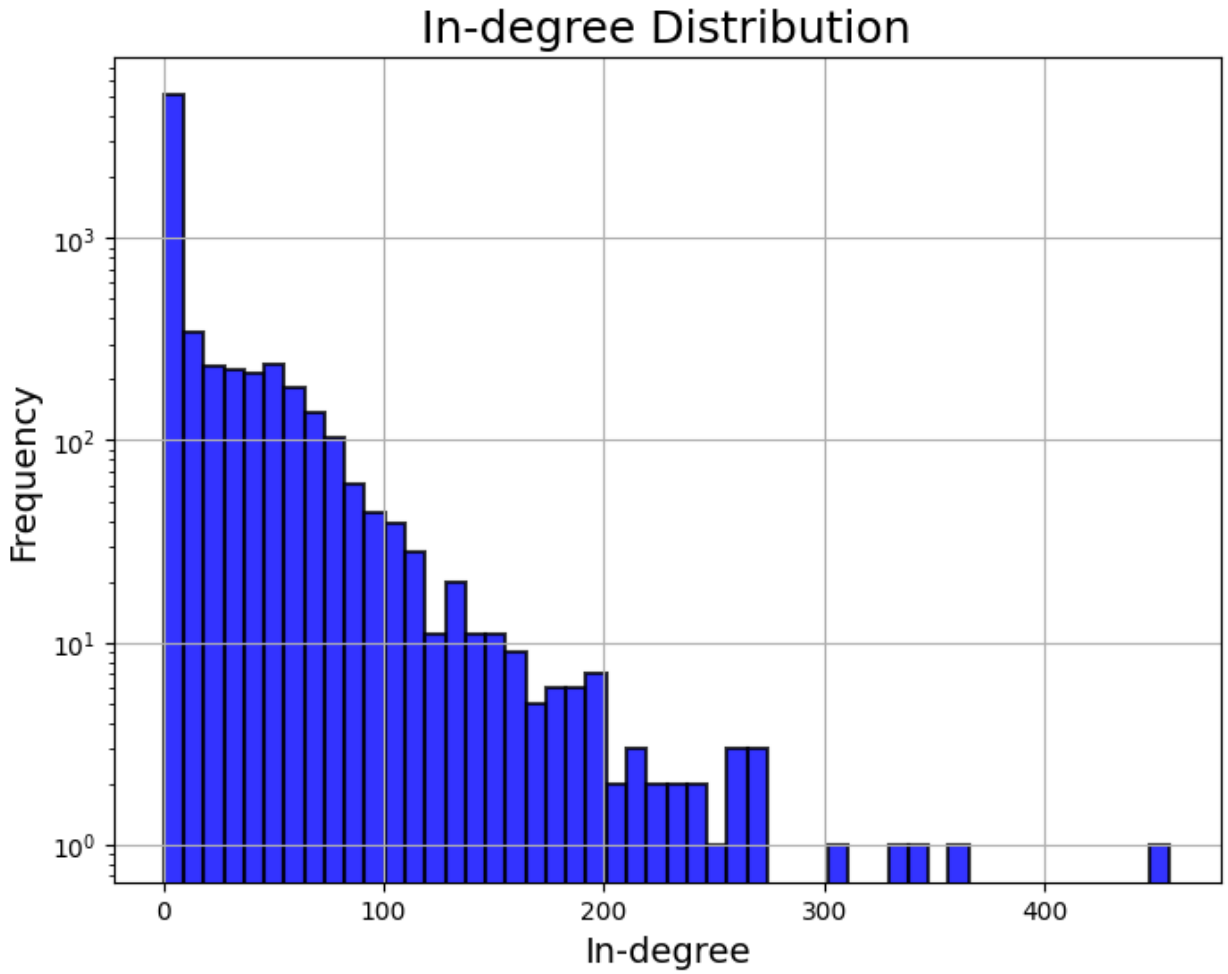
- Finally, the local clustering coefficient distribution is plotted using the matplotlib.pyplot.hist() method.

Assumptions:

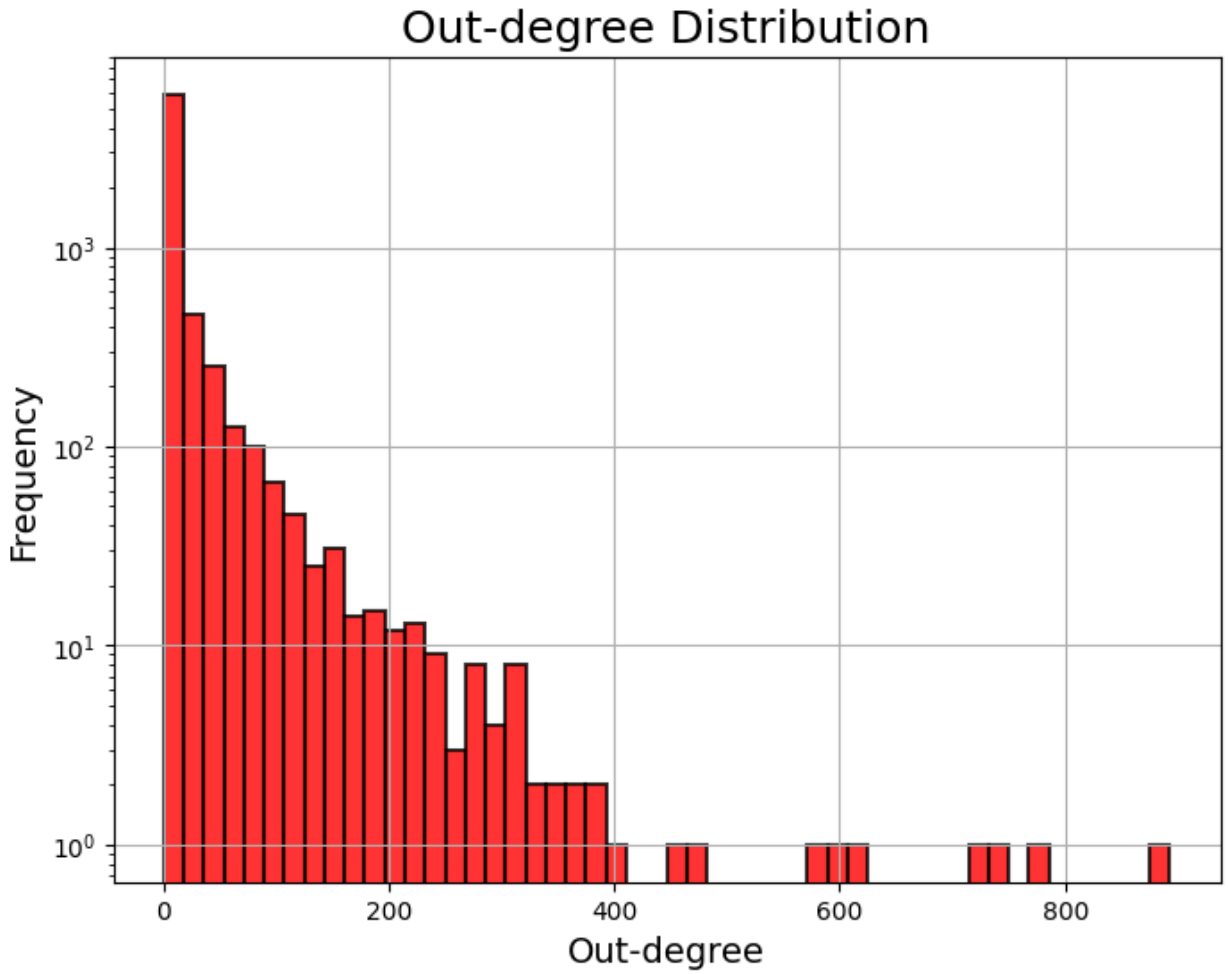
- The code assumes that the data is stored in a text file with the format specified in the parameters of the numpy.loadtxt() method.
- The code assumes that the data represents a directed graph, and edges are represented by pairs of nodes.
- The code assumes that the graph is simple, meaning that it has no self-loops or multiple edges between the same pair of nodes.

Results:

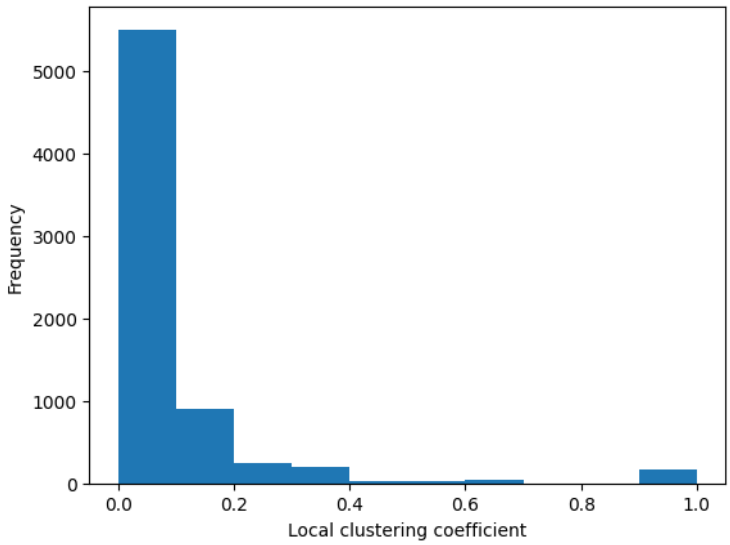
- Plot degree distribution of the network:
  i)For in-degree:



ii)For out-degree:



iii)The local clustering coefficient of each node and plot the clustering-coefficient distribution (lcc vs frequency of lcc) of the network:



Question 2

i)For PageRank Score:

Methodologies:

- The code uses the NetworkX library in Python to read an edgelist file and create a directed graph from it.
- The PageRank algorithm is then applied to the graph to calculate the importance of each node in the network.
- The alpha parameter of the PageRank algorithm is set to 0.85, which represents the probability that a user will continue clicking on links instead of stopping and randomly jumping to another page.

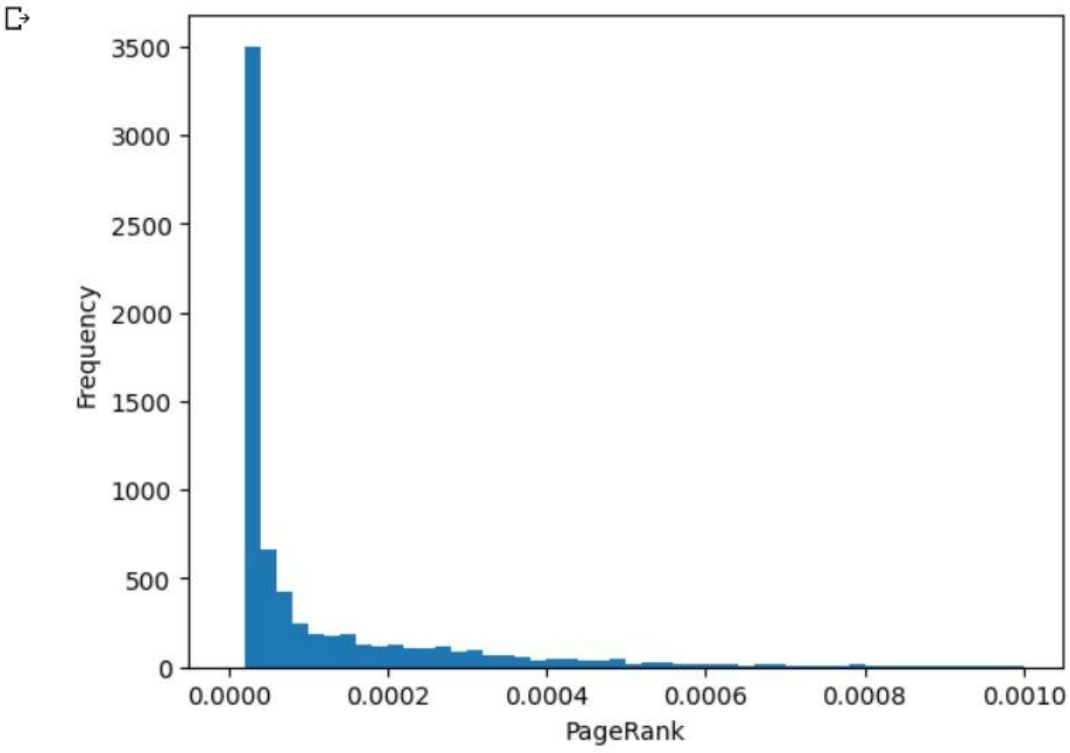
Assumptions:

- The edgelist file is assumed to be in the format expected by NetworkX, where each line contains two integers representing the endpoints of an edge.
- It is assumed that the graph represented by the edgelist is a directed graph, as specified by the creation of a DiGraph object in the code.
- The alpha parameter of the PageRank algorithm is set to 0.85 based on the common default value used in practice, but this value could be adjusted depending on the specific application.

Results:

- The code prints the PageRank scores for each node in the graph, in descending order of importance according to the algorithm.
- The results can be used to identify the most influential nodes in the network, which could be useful for a variety of applications such as identifying key players in a social network or important pages on a website.
- PageRank score:

```
Streaming output truncated to the last 5000 lines.
Node 7371: PageRank score = 0.00012290683500522525
Node 3877: PageRank score = 0.0001229064768266434 8
Node 5815: PageRank score = 0.0001228582946937080 4
Node 5788: PageRank score = 0.000122850931664286
Node 4764: PageRank score = 0.0001227928406000836 7
Node 4603: PageRank score = 0.0001227826728895725 3
Node 5072: PageRank score = 0.0001227775973072643 5
Node 5415: PageRank score = 0.0001227728850426453 1
Node 5437: PageRank score = 0.0001227321011717182 4
Node 2085: PageRank score = 0.0001226822717879800 5
Node 1984: PageRank score = 0.0001226723016210787 2
Node 4934: PageRank score = 0.0001226690959044418 3
Node 1991: PageRank score = 0.0001226688993022286 4
Node 2676: PageRank score = 0.000122638378453237 2
Node 2607: PageRank score = 0.0001226355084755385 6
Node 2073: PageRank score = 0.0001226308280919943 7
Node 5339: PageRank score = 0.0001226180692847979 6
Node 3488: PageRank score = 0.0001226084255020923 4
Node 3849: PageRank score = 0.000122607034720121 7
Node 4717: PageRank score = 0.0001225847986137816 5
Node 3144: PageRank score = 0.0001225465251247789 6
Node 4832: PageRank score = 0.0001225427536216833 8
Node 5757: PageRank score = 0.000122539198344201 6
Node 5830: PageRank score = 0.0001225334982778318 4
Node 1566: PageRank score = 0.0001225177680020308 6
Node 803: PageRank score = 0.000122509631245675
Node 5741: PageRank score = 0.0001224957394637913 9
Node 2591: PageRank score = 0.0001224931685325727 4
Node 2511: PageRank score = 0.000122373657471757 1
```



ii)For Authority and Hub Score:

Methodologies:

- The code uses the NetworkX library in Python to read an edgelist file and create a directed graph from it.
- The HITS (Hyperlink-Induced Topic Search) algorithm is then applied to the graph to calculate the authority and hub scores for each node.
- The algorithm iteratively assigns scores to each node based on the number and quality of incoming and outgoing links.
- The max_iter and tol parameters control the maximum number of iterations and the convergence tolerance of the algorithm, respectively.

Assumptions:

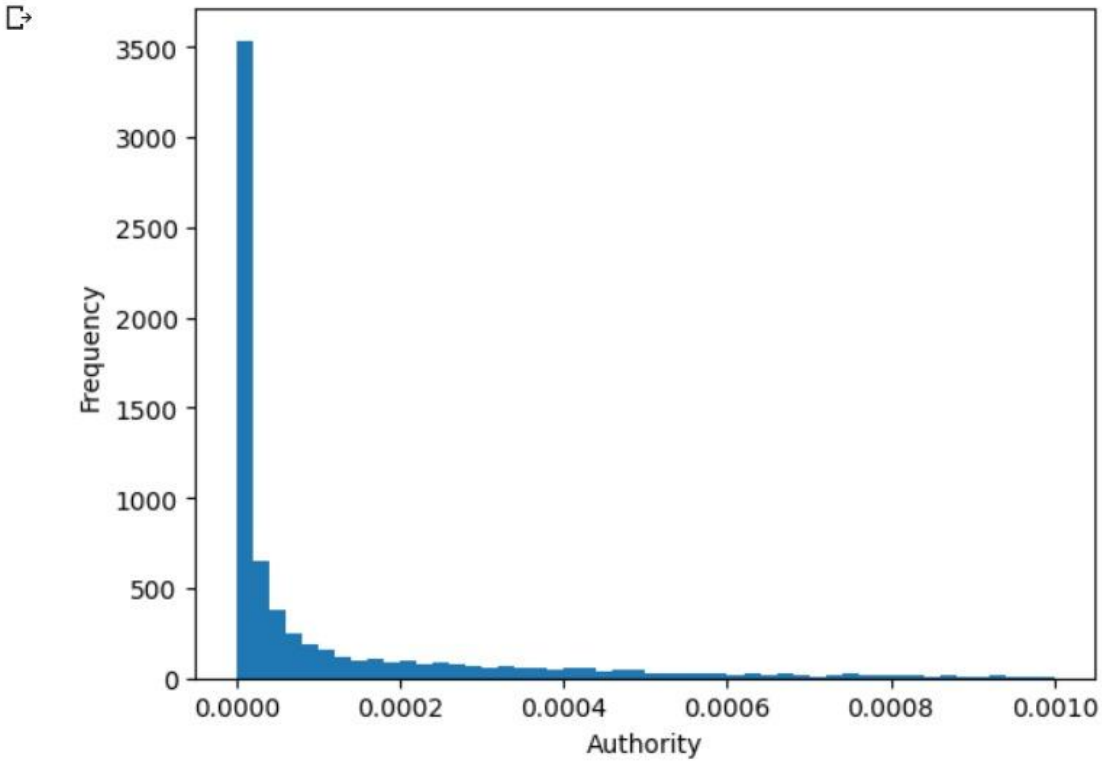
- The edgelist file is assumed to be in the format expected by NetworkX, where each line contains two integers representing the endpoints of an edge.
- It is assumed that the graph represented by the edgelist is a directed graph, as specified by the creation of a DiGraph object in the code.
- The max_iter and tol parameters are set to default values of 100 and 1e-06, respectively, but these values could be adjusted depending on the specific application.

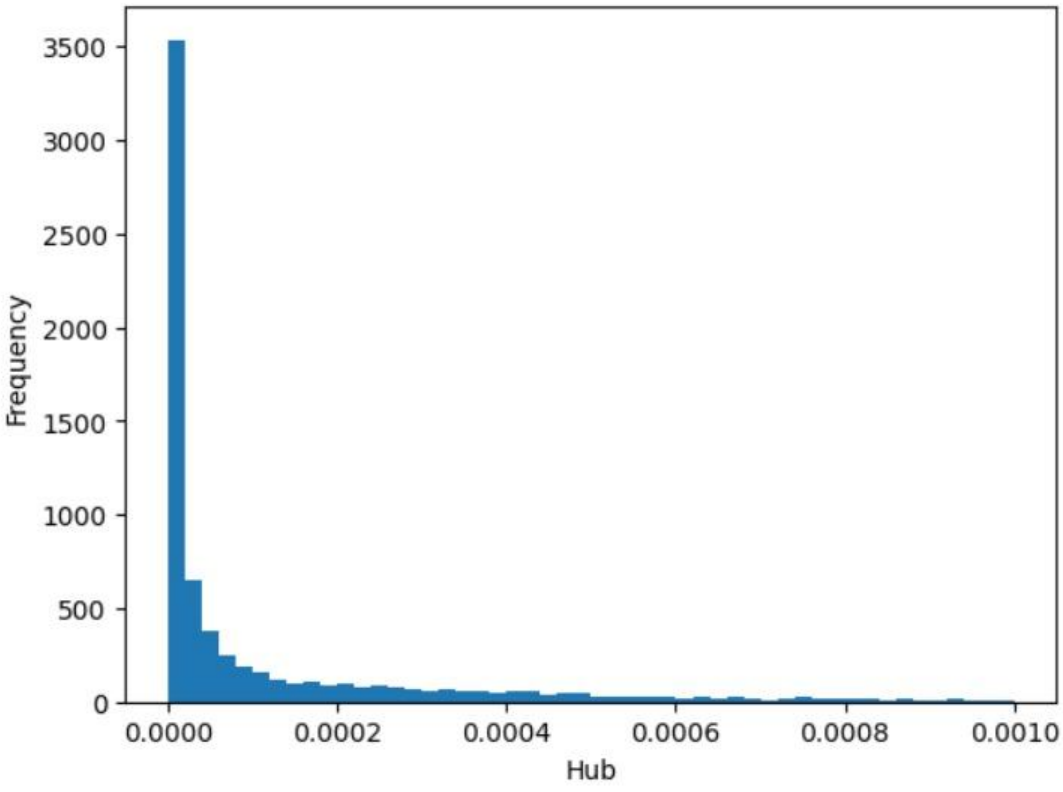
Results:

- The code prints the authority and hub scores for each node in the graph, in descending order of importance according to the algorithm.
- The authority score measures the importance of a node based on the quality and quantity of incoming links it receives, while the hub score measures the importance of a node based on the quality and quantity of outgoing links it has.
- The results can be used to identify nodes that are authoritative sources of information and nodes that are hubs for disseminating information in the network.

- Authority and Hub score:

```
Streaming output truncated to the last 5000 lines.
Node 5150: Authority score = 9.6640358504459l9e-05, Hub score = 9.6640358504459l9e-05
Node 6567: Authority score = 9.663918833771178e-05, Hub score = 9.663918833771178e-05
Node 3977: Authority score = 9.656187636204032e-05, Hub score = 9.656187636204036e-05
Node 1569: Authority score = 9.649275673540667e-05, Hub score = 9.64927567354067e-05
Node 5326: Authority score = 9.642282873947764e-05, Hub score = 9.642282873947767e-05
Node 4746: Authority score = 9.632349688916175e-05, Hub score = 9.632349688916151e-05
Node 2552: Authority score = 9.625905816010291e-05, Hub score = 9.625905816010286e-05
Node 6316: Authority score = 9.624027896849968e-05, Hub score = 9.624027896849953e-05
Node 192: Authority score = 9.616792309784042e-05, Hub score = 9.616792309784045e-05
Node 1861: Authority score = 9.616564115936872e-05, Hub score = 9.616564115936876e-05
Node 4934: Authority score = 9.614564434437708e-05, Hub score = 9.614564434437715e-05
Node 6522: Authority score = 9.584225828569251e-05, Hub score = 9.58422582856924e-05
Node 3481: Authority score = 9.581451986509605e-05, Hub score = 9.581451986509605e-05
Node 1595: Authority score = 9.580318792766734e-05, Hub score = 9.580318792766733e-05
Node 3961: Authority score = 9.572012202651473e-05, Hub score = 9.572012202651454e-05
Node 3127: Authority score = 9.534823180692531e-05, Hub score = 9.534823180692528e-05
Node 1513: Authority score = 9.525149780276734e-05, Hub score = 9.525149780276713e-05
Node 324: Authority score = 9.505651257668888e-05, Hub score = 9.505651257668884e-05
```


3500
3000
2500
2000
1500
1000
500
0
Frequency
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
Authority


3500
3000
2500
2000
1500
1000
500
0
Frequency
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
Hub

- **Average values for PageRank, Authority and Hub respectively:**

```
0.00014054813773717488
0.00014054813773717483
0.00014054813773717488
```