PROJECT I
ALGORITHM DESIGN

Group 24

Akhilesh Kumar – 1208566706

Jaideeep Singh Gour –

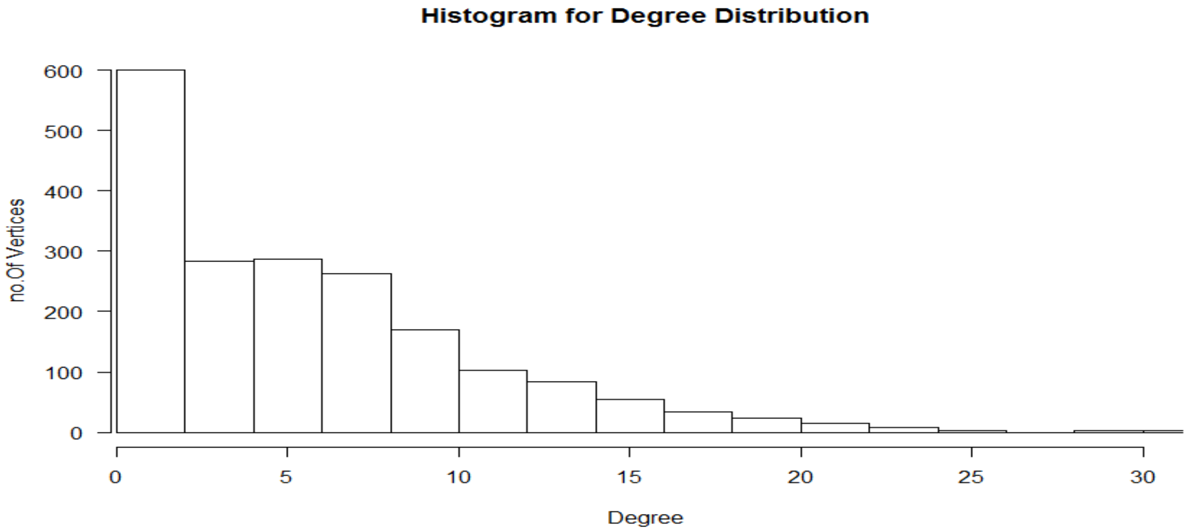
Vivek Singh - 1209521349

We have run the tasks for both the big data and small data.

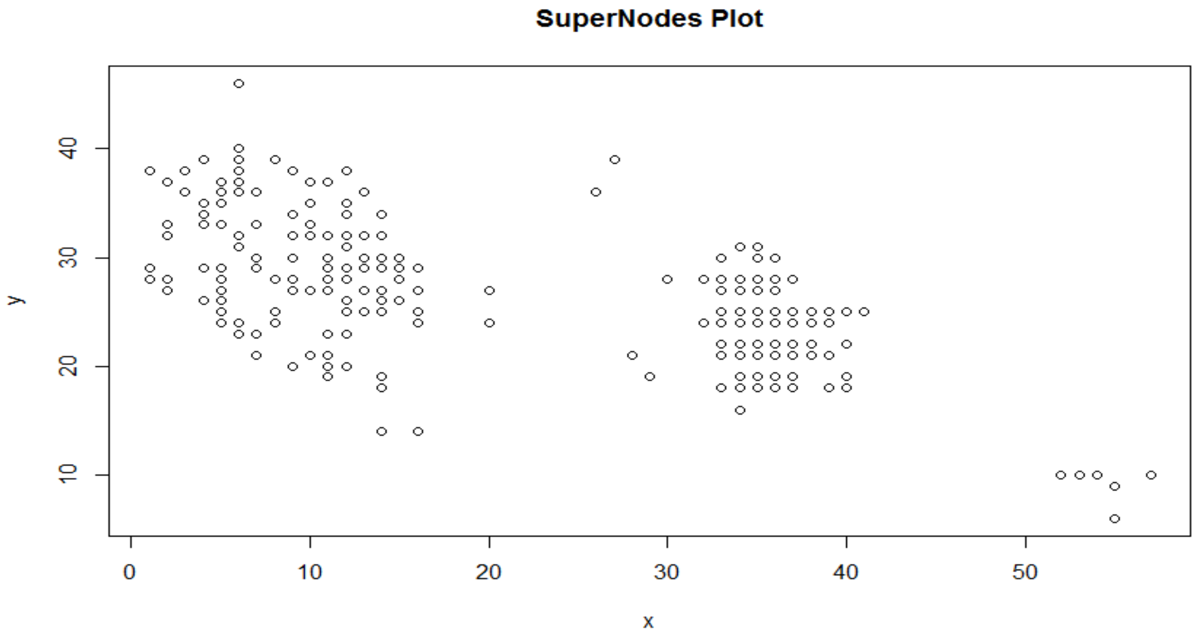
For Small Data

TASK 1

(a) Histogram for degree distribution



Projecting Super Nodes on Map of earth for Visualization



(b) Clustering coefficient for graph G_r $\gamma(G_r)$ is computed as **0.06630997**

Average Path Length for graph G_r $L(G_r)$ is computed as **14.51995**

For computing the above two parameters an efficient library for graph "Igraph " used by plotting the graph in real time and then calculating the value of its characteristics

(c)

Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

Characteristics/Graph	$\mathbf{G_r}$	$\mathbf{G_{random}}$
y(G)	0.06630997	0.0007764889
L(G)	14.51995	7.26031

Clustering Coefficient gives us measure of which nodes of graph are clustered together. Greater the value of clustering coefficient greater is the clustering among the nodes of the graph.

On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of original graph is much greater than the random graph.

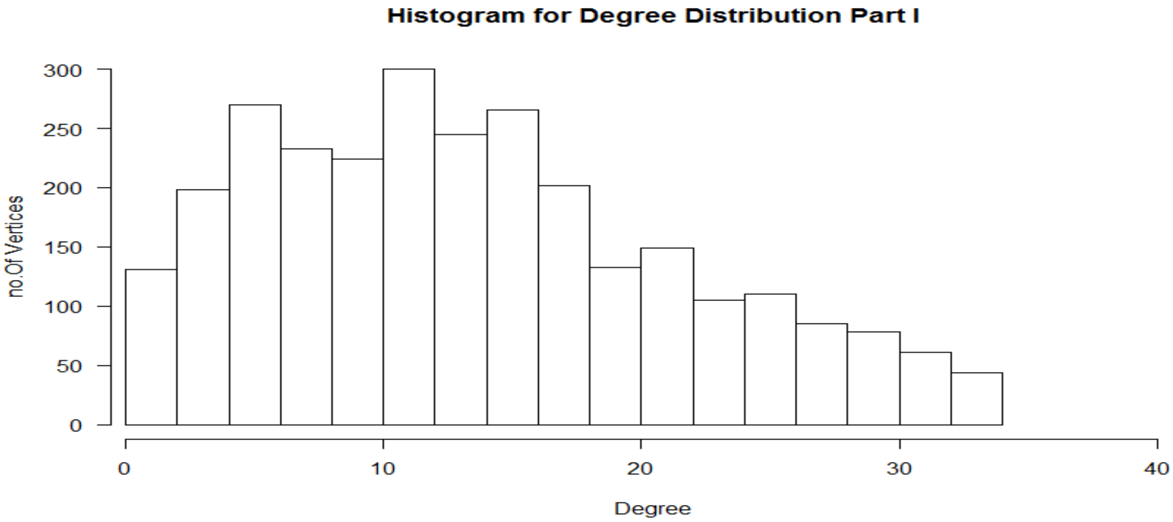
Thus we can say that the graph G_r has much greater clustering of nodes compared to a random graph G_{random} of comparable size. We can say that real and theoretical figures of graph differ by four orders of magnitude.

Comparing the Average path length of original graph with random graph we see that the average path length of random graph is nearly half the original graph

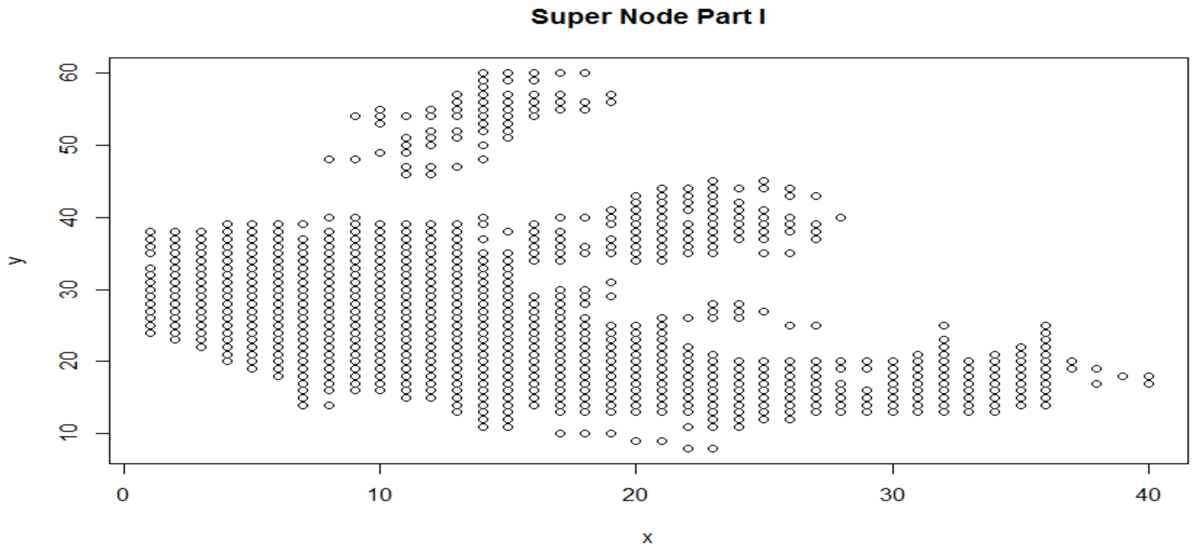
TASK II

For this the data set is divided into 3 equal datasets of n/3 years each and then repeating the Task I for each graph.

DATA SET I



Projecting Super Nodes on Map of earth for Visualization

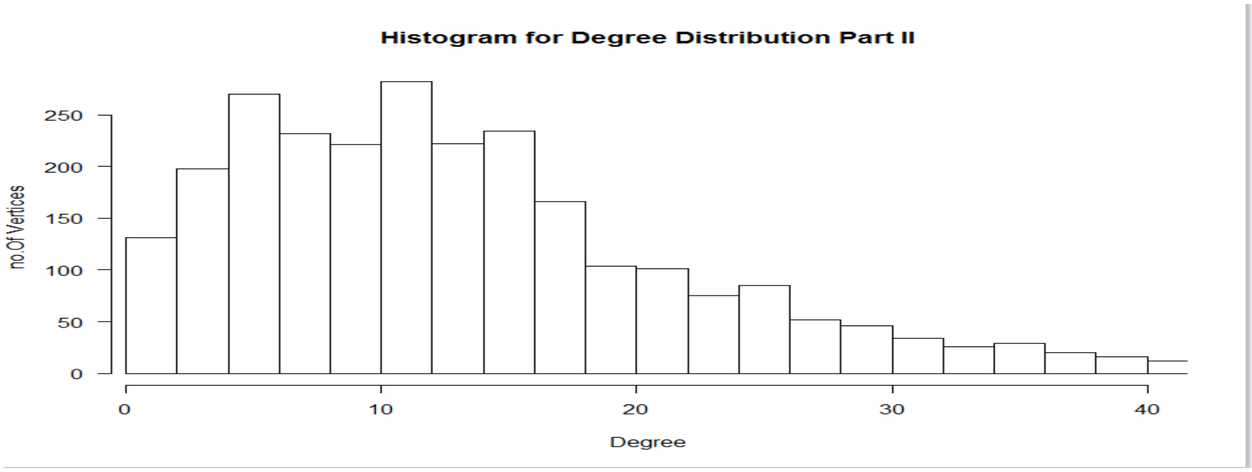


Clustering coefficient for graph G_r $\gamma(G_r)$ is computed as **0.1605248**
Average Path Length for graph G_r $L(G_r)$ is computed as **13.20132**
Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

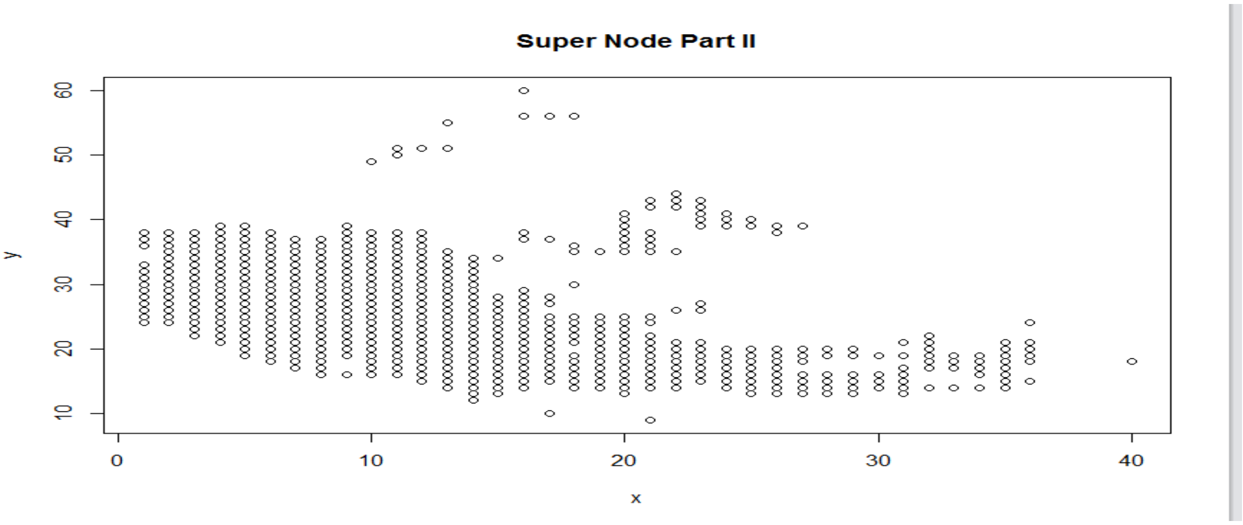
Characteristics/Graph	$\mathbf{G_r}$	$\mathbf{G_{random}}$
y(G)	0.1605248	0.003446209
L(G)	13.20132	3.16779

On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of normal graph is much greater than the random graph. Thus we can say that the graph G_r has much greater clustering of nodes compared to a random graph G_{random} of comparable size similar to task I.

DATA SET II



Projecting Super Nodes on Map of earth for Visualization



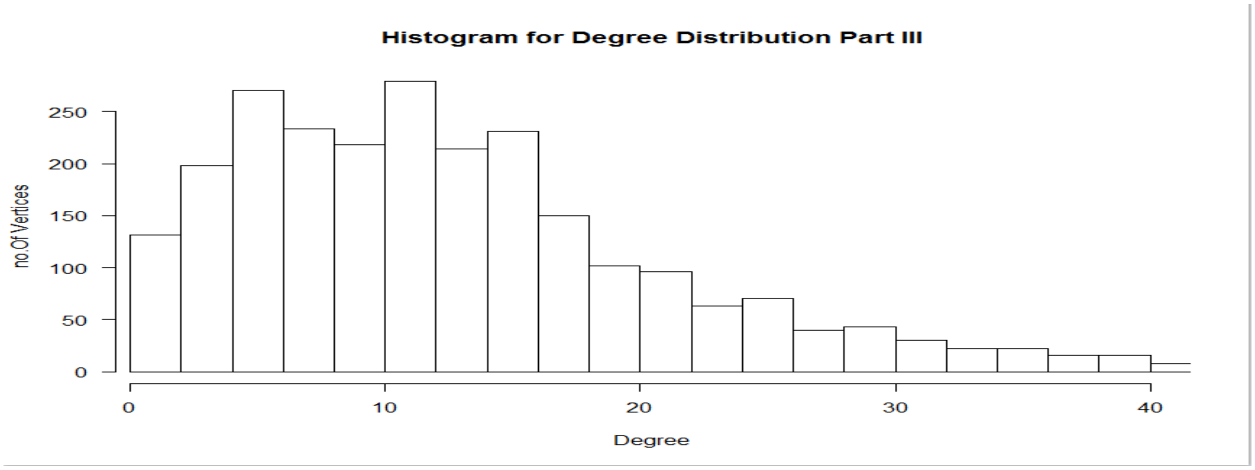
Clustering coefficient for graph Gr γ (G_r) is computed as **0.1642992**
Average Path Length for graph Gr L (G_r) is computed as **11.44235**

Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

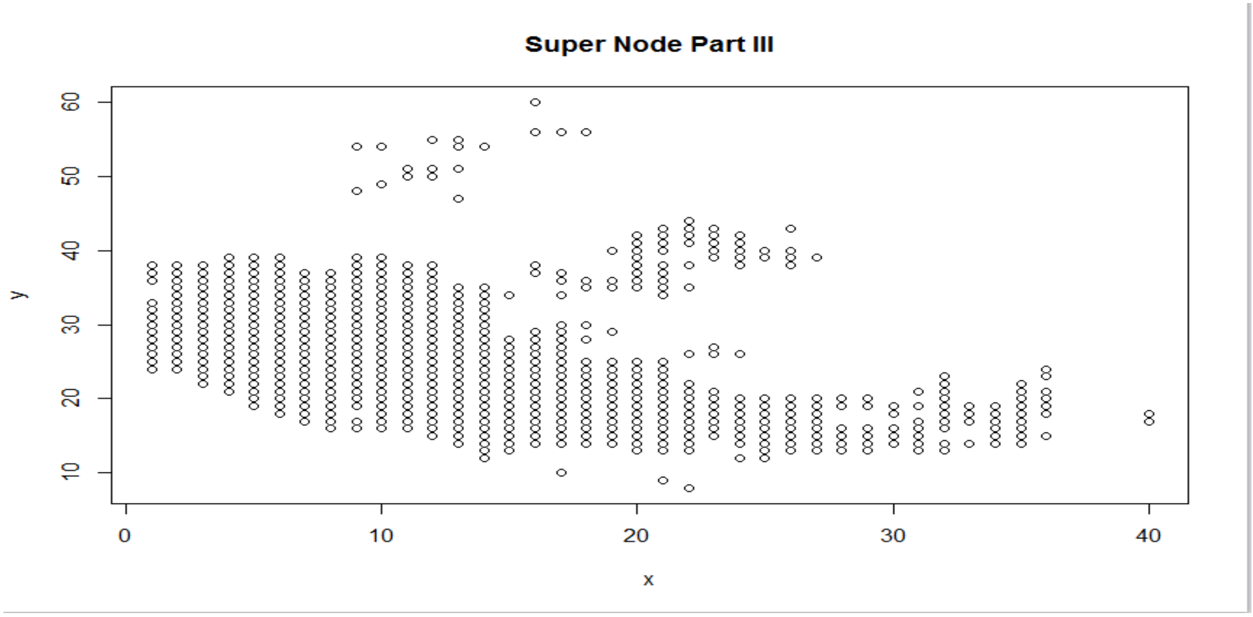
Characteristics/Graph	G_r	G_{random}
y(G)	0.1642992	0.00416445
L(G)	11.44235	2.953978

On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of normal graph is much greater than the random graph. Thus we can say that the graph G_r has much greater clustering of nodes compared to a random graph G_{random} of comparable size similar to task I.

DATA SET III



Projecting Super Nodes on Map of earth for Visualization



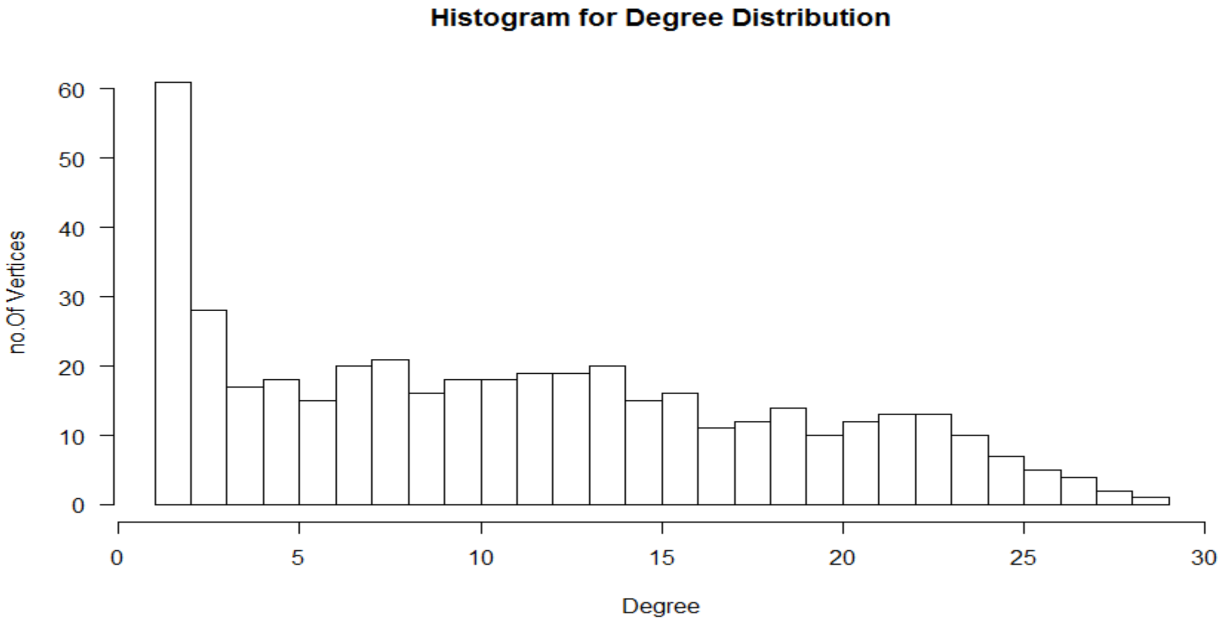
Clustering coefficient for graph G_r $\gamma(G_r)$ is computed as **0.155804**
Average Path Length for graph G_r $L(G_r)$ is computed as **12.2205**
Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

Characteristics/Graph	**G_r**	**G_{random}**
y(G)	0.155804	0.000313211
L(G)	12.2205	2.953978

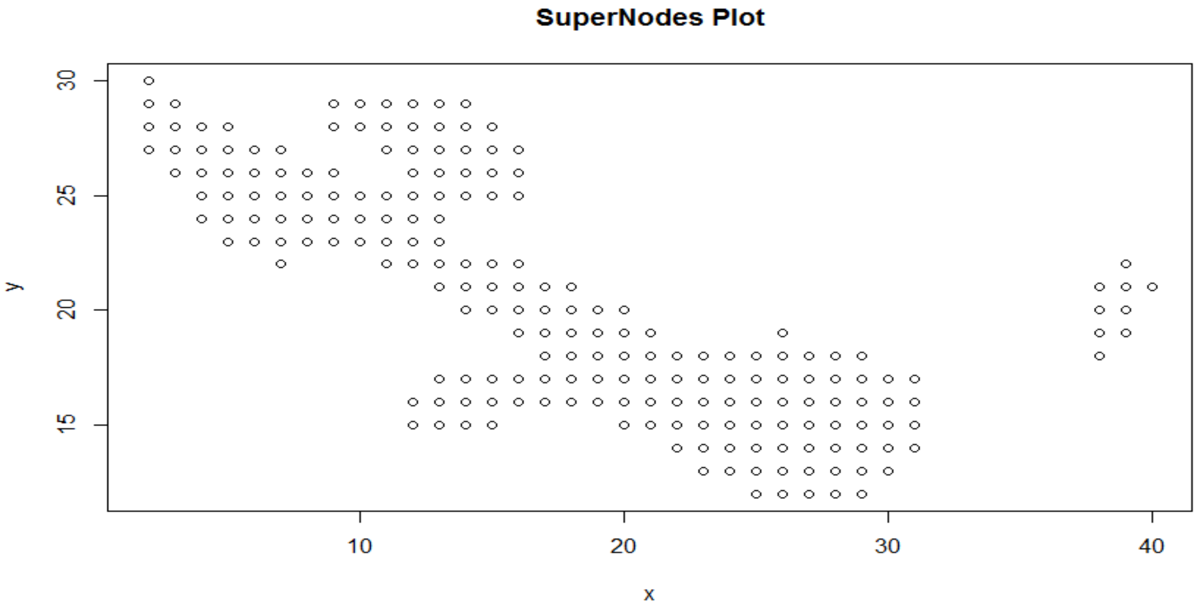
On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of normal graph is much greater than the random graph. Thus we can say that the graph G_r has much greater clustering of nodes compared to a random graph G_{random} of comparable size similar to task I. Here also we can see that clustering coefficient of original graph is much less that random graph Thus nodes of original graph are much more clustered than compared to random graph.

TASK 3

Lag = 1



Super Nodes plot for visualization



Clustering coefficient for graph G_r $\gamma(G_r)$ is computed as **0.155804**
Average Path Length for graph G_r $L(G_r)$ is computed as **0.04238**
Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

Characteristics/Graph	**G_r**	**G_{random}**
y(G)	0.1642992	0.004317284
L(G)	11.44235	2.916529

On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of normal graph is much greater than the random graph. Thus we can say that the graph G_r has much greater clustering of nodes compared to a random graph G_{random} of comparable size Here also we can see that clustering coefficient of original graph is much less that random graph Thus nodes of original graph are much more clustered than compared to random graph. When the lag is increased from 1 to 2, 3,4 then the correlation coefficient becomes less than the given threshold. Thus no edges are added to graph leaving graph **G_r** completely disconnected.

Comparing data for all the the tasks:

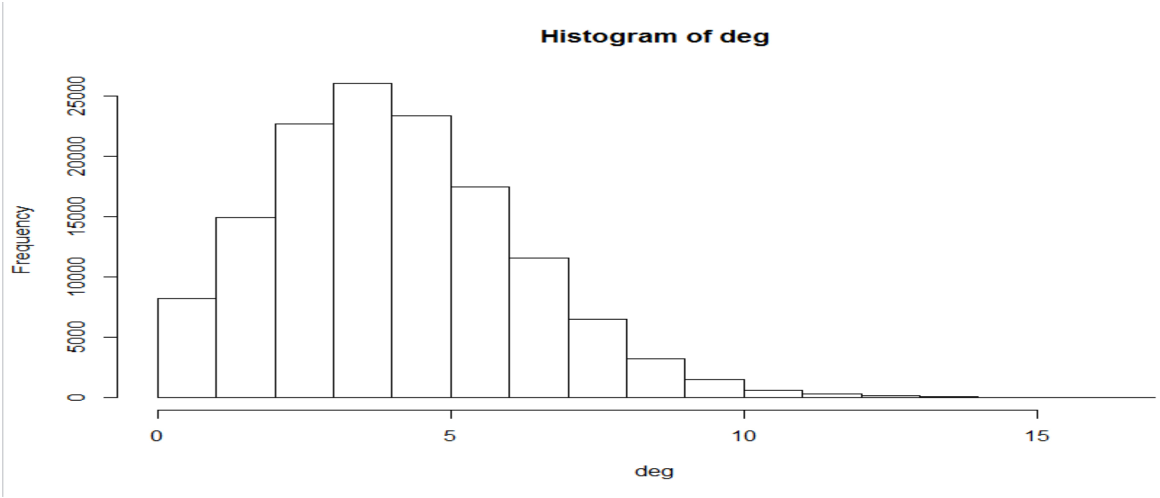
Characteristics/Years	All Years	I N/3 years	II N/3 years	III N/3 years	Lag = 1
y(G)	0.06630997	0.1605248	0.1605248	0.1642992	0.155804
L(G)	14.51995	13.20132	13.20132	11.44235	12.2205

Looking at the table we see that when we divided the time series data into n.3 years the value of clustering coefficient decreased and also the value of average path length decreased. This means that the clustering of vertices have increased as divided the data into equal parts and also in the case of lag=1, but if we increase the lag to a greater value then the graph becomes completely disconnected.

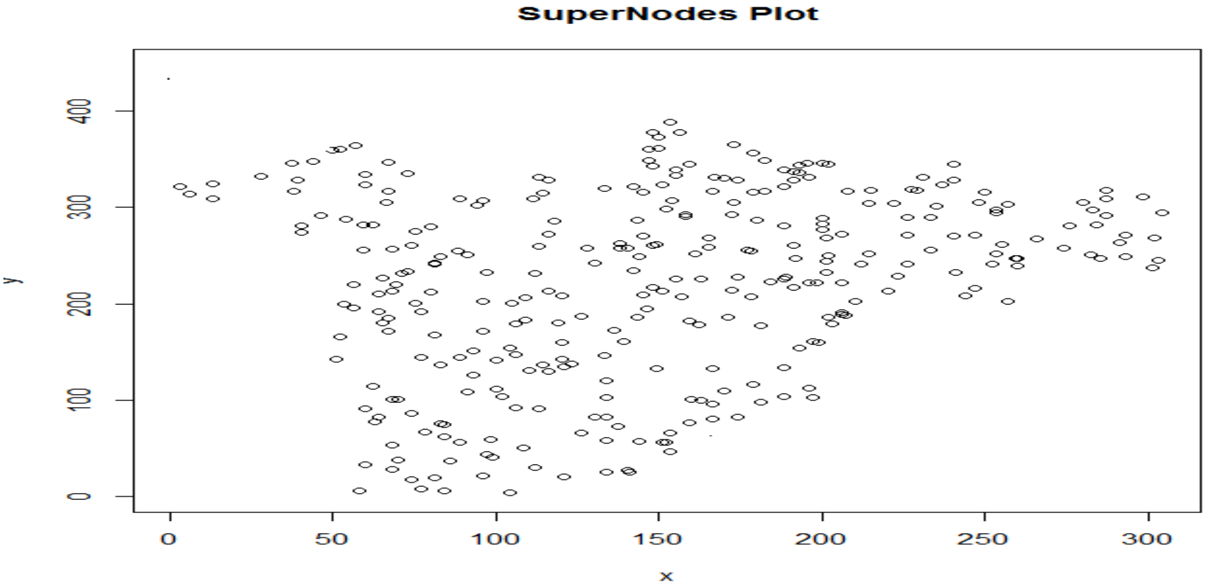
FOR BIG DATA

TASK 1

(a) Histogram for degree distribution



Projecting Super Nodes on Map of earth for Visualization



(b)

Clustering coefficient for graph G_r $\gamma(G_r)$ is computed as **4.458559e-03**

Average Path Length for graph G_r $L(G_r)$ is computed as **13.51992**

For computing the above two parameters an efficient library for graph "Igraph " used by plotting the graph in real time and then calculating the value of its characteristics

(C)

Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

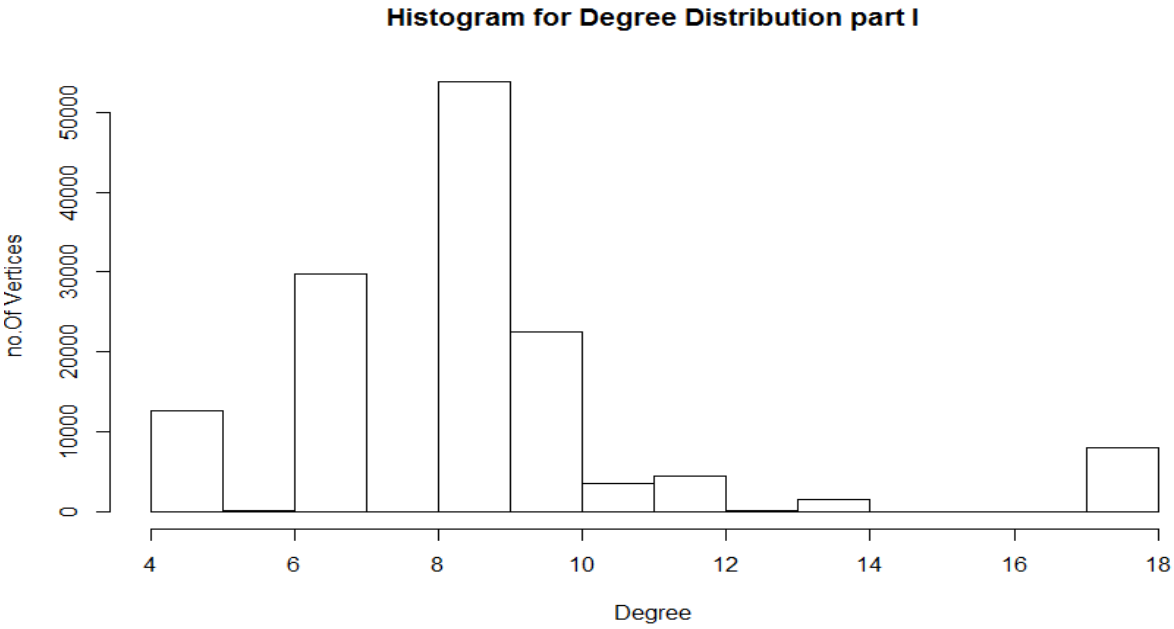
Characteristics/Graph	G_r	G_{random}
y(G)	4.458559e-03	3.336086e-05
L(G)	13.51992	7.809915

On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of original graph is much greater than the random graph. Thus we can say that the graph G_r has much greater clustering of nodes compared to a random graph G_{random} of comparable size. Comparing the Average path length of original graph with random graph we see that the average path length of random graph is smaller than the original graph.

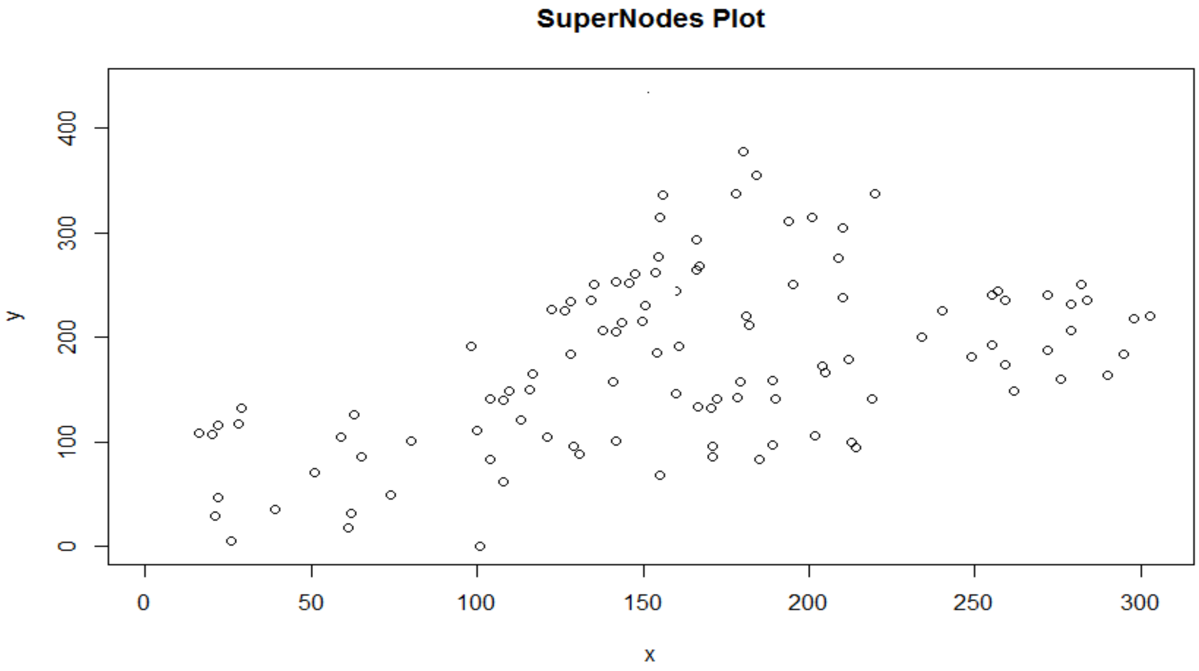
TASK II

For this the data set is divided into 3 equal datasets of 9years each and then repeating the Task I for each graph.

DATA SET I



Projecting Super Nodes on Map of earth for Visualization



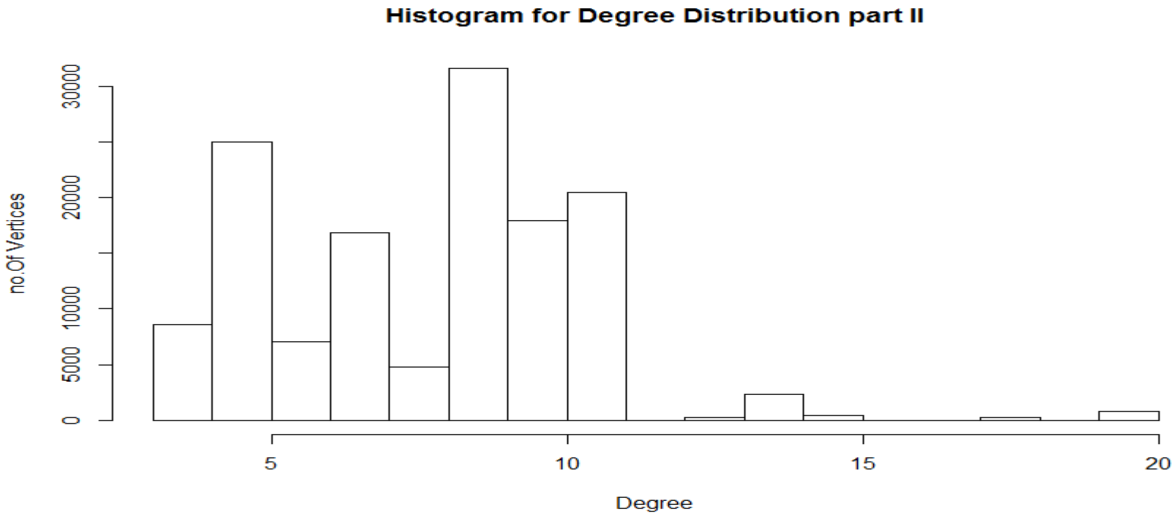
Clustering coefficient for graph G_r $\gamma(G_r)$ is computed as **7.33103e-04**
Average Path Length for graph G_r $L(G_r)$ is computed as **14.5123**

Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

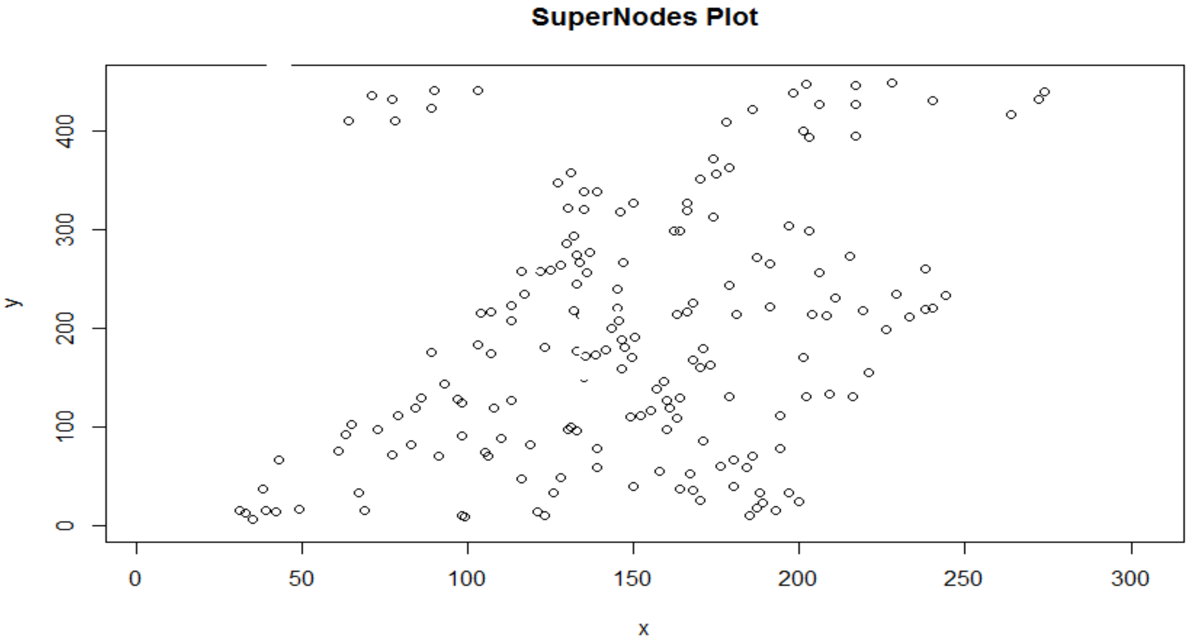
Characteristics/Graph	**G_r**	**G_{random}**
y(G)	7.33103e-04	6.664172e-05
L(G)	14.5123	5.359811

On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of we see that now the difference between them has reduced as compared to earlier results. Thus we can say that the graph G_r has comparable clustering of nodes compared to a random graph G_{random} of comparable size similar to task I.

DATA SET II



Projecting Super Nodes on Map of earth for Visualization

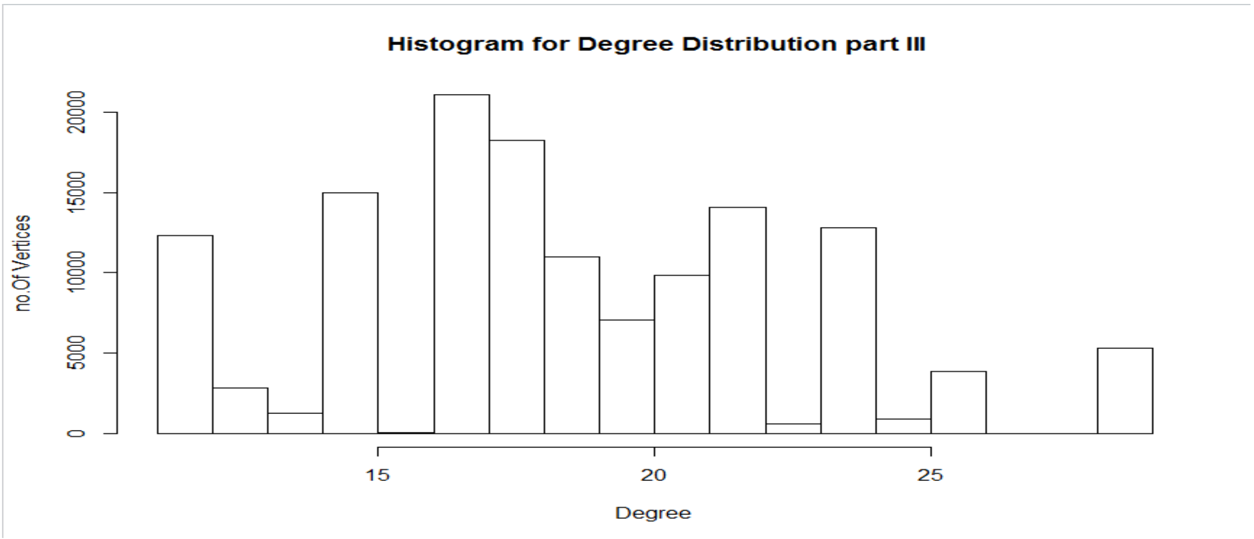


Clustering coefficient for graph G_r $\gamma(G_r)$ is computed as **7.466824e-03**
Average Path Length for graph G_r $L(G_r)$ is computed as **14.8163**
Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

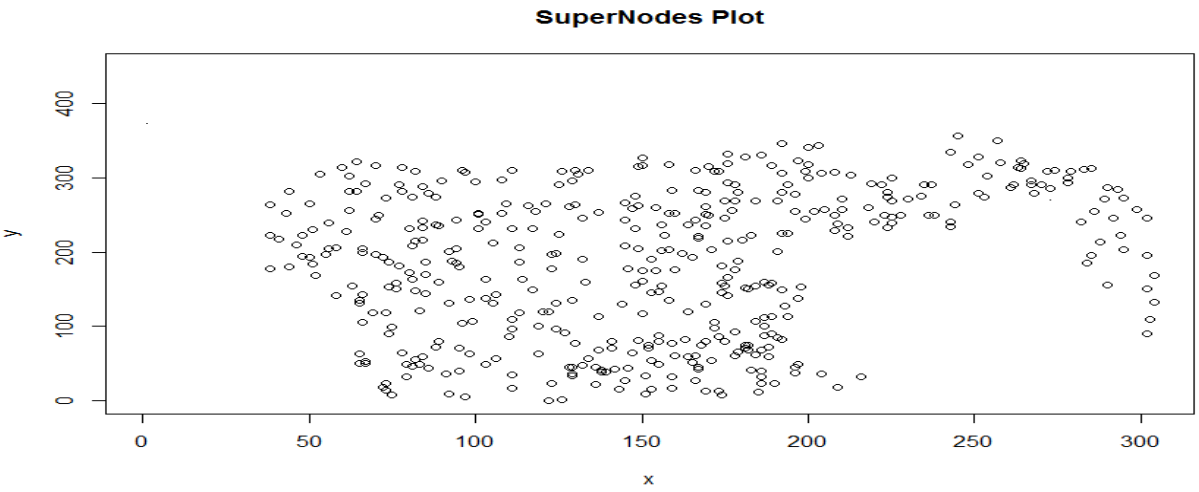
Characteristics/Graph	**G_r**	**G_{random}**
y(G)	7.466824e-03	7.694738e-05
L(G)	14.8163	5.031778

On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of normal graph is much greater than the random graph. Thus we can say that the graph G_r has much greater clustering of nodes compared to a random graph G_{random}.

DATA SET III



Projecting Super Nodes on Map of earth for Visualization



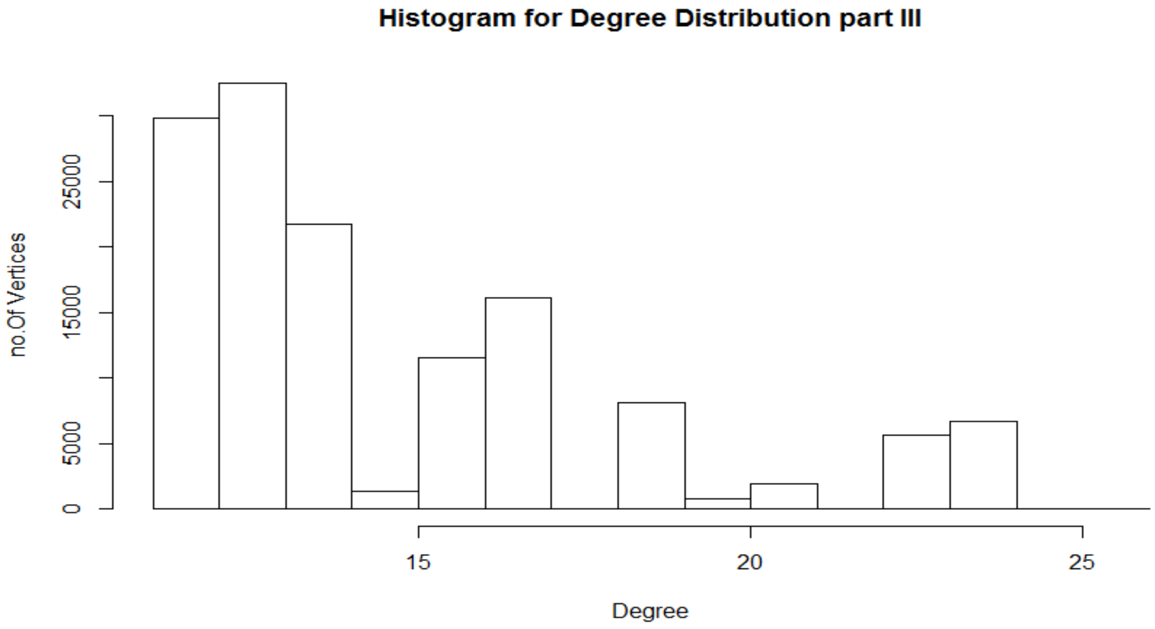
Clustering coefficient for graph G_r $\gamma(G_r)$ is computed as **0.001080996**
Average Path Length for graph G_r $L(G_r)$ is computed as **16.1215**
Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

Characteristics/Graph	**G_r**	**G_{random}**
y(G)	0.001080996	0.0001112978
L(G)	16.1215	4.348625

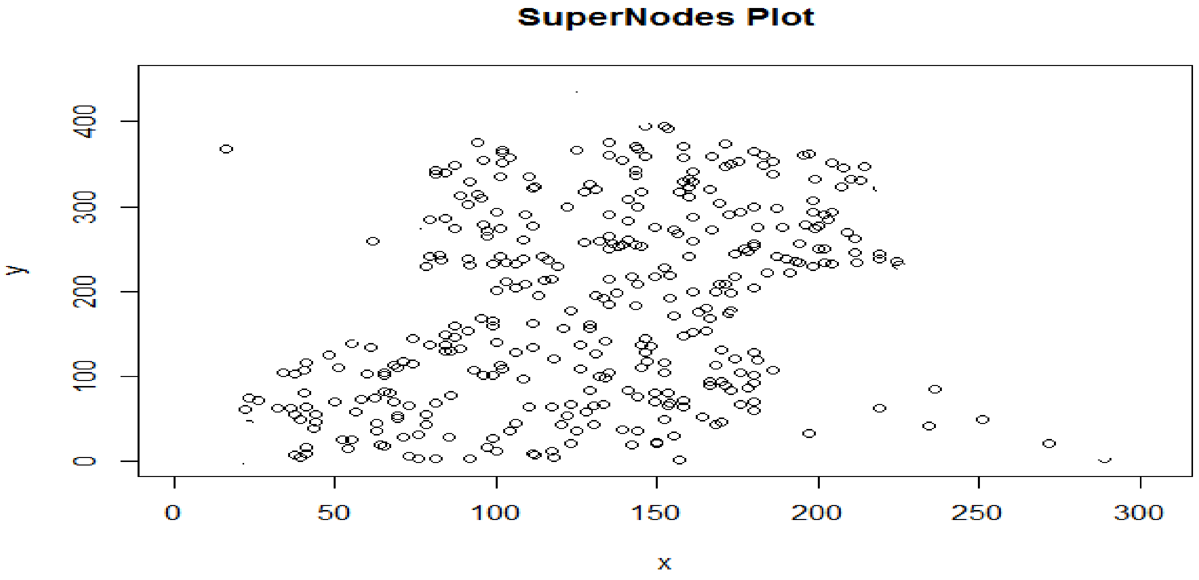
On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of normal graph is much greater than the random graph. Thus we can say that the graph G_r has much greater clustering of nodes compared to a random graph G_{random} of comparable size similar to task I. Here also we can see that clustering coefficient of original graph is much less that random graph Thus nodes of original graph are much more clustered than compared to random graph.

TASK 3

Lag = 1



Super Nodes plot for visualization



Clustering coefficient for graph G_r $\gamma(G_r)$ is computed as **0.155804**
Average Path Length for graph G_r $L(G_r)$ is computed as **0.04238**
Following are the Clustering Coefficient and Average path length of a random Graph G_{random} of comparable size

Characteristics/Graph	G_r	G_{random}
y(G)	0.1642992	0.004317284
L(G)	11.44235	2.916529

On comparing the clustering coefficient of G_r and G_{random} we see that the clustering coefficient of normal graph is much greater than the random graph.

Thus we can say that the graph G_r has much greater clustering of nodes compared to a random graph G_{random} of comparable size Here also we can see that clustering coefficient of original graph is much less that random graph. Thus nodes of original graph are much more clustered than compared to random graph. Similar to small data set when the lag is increased from 1 to 2, 3, 4 then the correlation coefficient becomes less than the given threshold. Thus no edges are added to graph leaving graph $\mathbf{G_r}$ completely disconnected.

Comparing data for all the tasks:

Characteristics/Years	All Years	I N/3 years	II N/3 years	III N/3 years	Lag = 1
y(G)	4.45855e-03	7.33103e-04	7.466824e-03	0.001080996	0.1642992
L(G)	13.51992	14.5123	14.8163	16.1215	11.44235

Looking at the table we see that when we divided the time series data into n.3 years the value of clustering coefficient decreased and also the value of average path length decreased. Similar observation is seen as in small data for the big data set.

Optimization and Complexity Analysis

We have used R as it is one of the most powerful tool for mathematical computation and statistical analysis. The code written in R is compact and easy to debug. There are many optimizations that we have done in the R and code written to boost the performance. Below is the list of techniques used to speed up the processing.

1. R is designed to run on single core and hence we have used Revolution R. Revolution R is a form of R and it has the capability to utilize four cores of the system. Revolution R makes use of MKL library which stands for Intel Math Kernel Library. This library provides the BLAS and LAPACK package which are essential for Linear Algebra calculations. These libraries speed up the system. Since in our project, most of the time is taken in calculating the coefficient matrix which involves linear algebra, so the speed improves. MKL is library provided by Intel and so it is already designed to make use of efficient architecture. Ideally, MKL runs same number of parallel threads as cores.

2. Revolution R, makes use of 4 cores so the R script is multithreaded and computation is done parallel on 4 cores.

3. We have made use of some of the inbuilt function of R such as one for calculating the correlation. R being an interpreter takes time when it hops from one line to another. This time becomes large when there are many loops. Making use of inbuilt function ensures that computation is done as a compiler and not interpreter. The inbuilt function of R is mostly written in C and FORTRAN and highly optimized for speed.

4. R being an interpreter is slow compared to any other compiler language as it takes time to hop from one line to another. It takes large time when there are many nested loops. We tried to address this issue by using Rcpp package. It allows to create C++ binding or functions. Data Object can be passed from R to C++ where they are run as native C++ code. This significantly reduces the time as there is no time involved in going from line to line in nested loop. However, since our code in R is small and we are using inbuilt function of R we did not see any significant improvement for our project. So we removed the Rcpp bindings in the final code.

5. We have used igraph library in R to plot real time graph. Earlier, we tried using covariance matrix to plot the graph but we encountered following issue. Using covariance matrix works fine with small data but with big data the memory requirements are very large and practically unfeasible. Considering 448 rows and 304 columns the total number of cells required for covariance matrix is 448 * 304 * 448 * 304. Assuming 1 byte for each cell for storing 0 and 1, the total size of covariance matrix works out to be (448 * 304 * 448 * 304)/ (1024 * 1024 * 1024) = 17.6 GB. By using the igraph library covariance matrix was not required and graph was plotted in real time.

Time and Space complexity: There are various operation that govern the time and space complexity. The main operation that dominates the time complexity is finding the correlation between every element in the matrix which is O(n^2) where n is the number of vertices (for big data number of vertices is 448 * 304). **Space complexity** is governed by space required to store the data. Here we have 3 dimensional matrix that stores the data. So the space complexity is (n * h) where n is the number of vertices and h is the number of weeks. For big data n is 304 * 448.Also the space complexity can be governed by space used by igraph library to plot the graph