We have implemented non-parallel k-means, hierarchical and DBSCAN algorithm in R. Please follow the below steps to run the R code.

* There are two files, script.R and utility.R. We have implemented the algorithm in utility.R and result visualization is done in script.R. So open both .R files in RStudio and place the source data files in your working directory.
* We have commented the code, so first run the section under load dataset comment. Comment or uncomment the below lines to select a particular data file.

dataset = read.csv(file = "iyer.txt", header = FALSE, sep = "\t")

dataset = read.csv(file = "cho.txt", header = FALSE, sep = "\t")

* Once the data is loaded, run the code sections commented as Hierarchical Clutering with Single Linkage Kmeans, Kmeans in MapReduce and DBScan.

We have implemented, mapreduce k-means in JAVA. Please follow the below steps to run the Kmeans.java file.

* Open Kmeans.java and set the below path to the paths to hdfs input and output directories on your machine. Please edit the below mentioned path present in the beginning of Kmeans class

public static String data\_path = "/user/anhduc/kmeans\_data/";

public static String output\_path = "/user/anhduc/kmeans\_output/";

* Place the input files cho.txt and iyer.txt on hdfs file system using the command

Hdfs dfs –put /file\_location\_on\_local\_machine /location\_on\_hdfs\_file\_system\_where\_the\_input\_file\_should\_be\_placed

* Start hadoop on your system and compile Kmeans.java file using

bin/hadoop com.sun.tools.javac.Main Kmeans.java

* After compilation combine the class files into 1 jar file using the below command

jar cf Kmeans.jar Kmeans\*.class

* Run the jar file in hadoop using

bin/hadoop jar Kmeans.jar Kmeans

* Once the program completes execution copy the result into local folder

bin/hdfs dfs -get /Hdfs\_output\_folder\_path /\*