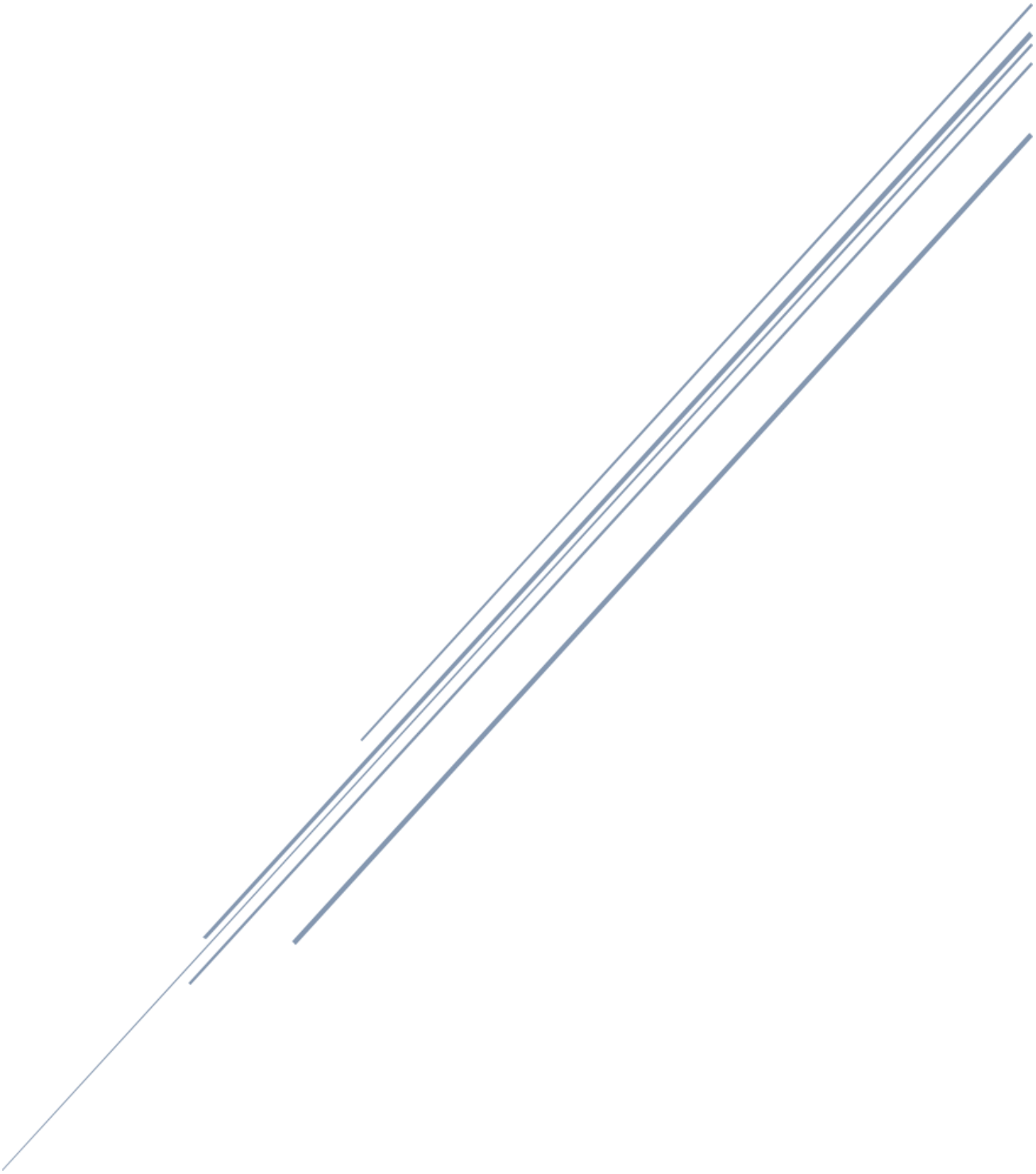
SOCIAL NETWORK ANALYSIS

Project II - 2022-2023



ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ SCHOOL OF BUSINESS
ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ DEPARTMENT OF MANAGEMENT SCIENCE & TECHNOLOGY

Dimitris Matsanganis, f2822212
July 8, 2023

Contents

Contents 2
Table of Figures 3
Abstract - Description of the Assignment 4
Q1: Data Loading, Preprocessing, and Aggregation 5
Q2: Visualizing Graph Evolution: A Five-Day Analysis of Key Metrics 6
Number of Vertices Analysis 6
Number of Edges Analysis 8
Diameter of the Graph (Weighted) Analysis 9
Diameter of the Graph (Not Weighted) Analysis 11
Average In-Degree Analysis 13
Average Out-Degree Analysis 14
Summary of Metrics 16
Q3: Evolution of Top-10 Twitter Users: In-Degree, Out-Degree, and PageRank - Important Nodes 17
Top-10 Twitter Users: In-Degree 17
Top-10 Twitter Users: Out-Degree 19
Top-10 Twitter Users: PageRank 20
Q4: Community Detection and Analysis 21
Fast Greedy Clustering 21
Infomap Clustering 22
Louvain Clustering 22
Technical & Coding Documentation 22
Detection of User's Community Evolution Analysis using Louvain Algorithm 24
Visualization of Community Graphs with Louvain and Fast Greedy Algorithms 25
Louvain Algorithm Visualization: 25
Fast Greedy Algorithm Visualization: 28
Deliverables 30
Appendix 31

Table of Figures

Figure 1: Barplot Analyzing the 5 Days Trend Regarding the Number of Vertices. 6
Figure 2: Trendline Diagram for Number of Vertices over the 5 Days. 7
Figure 3: Barplot Analyzing the 5 Days Trend Regarding the Number of Edges. 8
Figure 4: Trendline Diagram for Number of Edges over the 5 Days. 9
Figure 5: Barplot Analyzing the 5 Days Trend Regarding the Diameter of the Graph (weighted). 10
Figure 6: Trendline Diagram for Diameter of the Graph over the 5 Days (weighted). 11
Figure 7: Barplot Analyzing the 5 Days Trend Regarding the Diameter of the Graph (not weighted). 12
Figure 8: Trendline Diagram for Diameter of the Graph over the 5 Days (not weighted). 12
Figure 9: Barplot Analyzing the 5 Days Trend Regarding the Average In-Degree. 13
Figure 10: Trendline Diagram for the Average In-Degree over the 5 Days. 14
Figure 11: Barplot Analyzing the 5 Days Trend Regarding the Average Out-Degree. 15
Figure 12: Trendline Diagram for the Average Out-Degree over the 5 Days. 15
Figure 13: July 1st, 2009, Twitter Communities (Louvain Algorithm). 26
Figure 14: July 2nd, 2009, Twitter Communities (Louvain Algorithm). 26
Figure 15: July 3rd, 2009, Twitter Communities (Louvain Algorithm). 26
Figure 16: July 4th, 2009, Twitter Communities (Louvain Algorithm). 27
Figure 17: July 5th, 2009, Twitter Communities (Louvain Algorithm). 27
Figure 18: July 1st, 2009, Twitter Communities (Fast Greedy Algorithm). 28
Figure 19: July 2nd, 2009, Twitter Communities (Fast Greedy Algorithm). 28
Figure 20: July 3rd, 2009, Twitter Communities (Fast Greedy Algorithm). 29
Figure 21: July 4th, 2009, Twitter Communities (Fast Greedy Algorithm). 29
Figure 22: July 5th, 2009, Twitter Communities (Fast Greedy Algorithm). 29
Table 1: Summary of Metrics for the First Five Days of July 16
Table 2: Top-10 Twitter Users: In-Degree. 17
Table 3: Top-10 Twitter Users: Out-Degree 19
Table 4: Top-10 Twitter Users: PageRank (rounded) 21
Table 5: The number of members in random user's community on each day. 24
Table 6: Number of common users in random user's community on two consecutive days. 24
Table 7: Most frequently topics of interest (hashtags) in random user's communities per day. 25
Table 8: Top-10 Twitter Users: PageRank with exact values. 31

Abstract - Description of the Assignment

This assignment - case study focuses on analyzing a dataset of Twitter mentions from July 2009. The dataset consists of tweets with information about the time of posting, user handles, and the text of the tweets. The goal is to create a weighted directed graph to represent the mention relationships between users, identify the most important topic for each user based on their hashtags, and perform various analyses on the graph.

To begin, the raw data is manipulated to create five CSV files, each representing the weighted directed mention graph for a specific day. The CSV files contain information about the users involved in the mentions, the frequency of mentions between users, and the most important topic (hashtag) for each user. All the data handling procedures were performed in Python.

Using the CSV files, igraph graphs are created in R, and the graph vertices are updated to include the attribute of the topic of interest for each user. This allows for further analysis and visualization of the graph. Then, the evolution of different metrics over the five-day period is then examined. Plots are created to visualize the changes in the number of vertices, number of edges, graph diameter, average in-degree, and average out-degree. Significant fluctuations in these metrics are identified and discussed.

Furthermore, data frames are generated for each day, highlighting the top-10 Twitter users based on in-degree, out-degree, and PageRank. Variations in the top-10 lists are observed for different days, indicating changes in user influence and popularity. Community detection algorithms, including fast greedy clustering, infomap clustering, and Louvain clustering, are applied to the undirected versions of the mention graphs. The performance of these algorithms is evaluated, and insights are provided on their effectiveness.

Additionally, a specific user present in all five graphs is chosen, and their community evolution is analyzed. Similarities in the communities the user belongs to are identified, along with the most important topics of interest. The presence of shared topics among communities is explored, and a visualization of the graph is created, using different colors to represent each community. Nodes belonging to very small or large communities are filtered out to improve the clarity and aesthetics of the visualization.

Overall, this assignment lead us to output a case study, which provides a comprehensive analysis of the Twitter mention graph, highlighting the evolution of various metrics, important nodes, and communities over a five-day period. The findings shed light on user interactions, influence, and the dynamic nature of online social networks.

Q1: Data Loading, Preprocessing, and Aggregation

The first objective we handed was to create a weighted directed graph using the igraph library in R and extract the most important topic (based on hashtags) for each user. To begin with, we processed the raw Twitter data using Python, through the **raw_data_handler.py** script.

Describing briefly what this file does, we first read the tweets from the given file (after extracted), extracted relevant information such as the timestamp, username, mentions, and hashtags. We filtered the data to include only tweets within the specified date range of July 1 to July 5, 2009.

Next, we aggregated the data by day and created two dictionaries: aggregate_data_mentions to store the mentions between users, and aggregate_data_hashtags to store the hashtags used by each user. Afterwards, for each day, we generated a CSV file for mentions, where each row represented a mention between two users along with its weight. The CSV files were named using the format YYYY.MM.DD_mentions.csv (e.g. 2009.07.02_mentions.csv).

We also calculated the most important topic for each user by finding the most frequent hashtag they used. We stored this information in the aggregate_data_hashtags dictionary. Finally, for each day, we created a CSV file named YYYY.MM.DD_hashtags.csv, which listed the user and their respective most important topic (e.g. 2009.07.02_hashtags.csv).

Now that we had the CSV files containing the weighted directed mention graph and user-topic information for each day, we could utilize the igraph library in R to create the corresponding graphs.

Prior to this we need to do some data preparatory steps after we transited to R. The first step, is to import the Python's produced datasets. To be more precise, five CSV files are imported, each containing the data for a specific date range in July 2009. These files contain the mentions and hashtags data extracted from the Twitter data.

Then we focused to some data handling steps. The dataframes for mentions and hashtags from each day are merged based on the "from" column in mentions and the "user" column in hashtags. To fulfill the requirement of having "Null/NA" for users who do not contain any hashtags in their tweets, the empty fields in the "topic_of_interest" column are replaced with "Null/NA" notation. Optional, we create a choice If desired, the final dataframes for each day can be exported to separate CSV files using the write.csv function and a checkpoint to insert from there the CSV files for our convenience.

Now we are ready to move forward to the Graph Creation procedure. The igraph library is used to create graphs for each day. The graphs are created using the graph_from_data_frame function, which takes the corresponding dataframe as input. The directed=TRUE parameter specifies that the graph is directed.

Finally, the vertex attribute "topic_of_interest" is added to each graph by updating the graph vertices with the values from the "topic_of_interest" column of the corresponding dataframe. The set_vertex_attr function is used to set the "topic_of_interest" attribute for each graph.

Note: In order for the procedure to be executed smoothly the R and Python scripts should be located on the same folder with the extracted text file tweets2009-07.txt, in order to take it as input in the initial data handling procedure.

Q2: Visualizing Graph Evolution: A Five-Day Analysis of Key Metrics

In this section, we present plots that visualize the five-day evolution of different metrics for a graph. Specifically, we focus on the following metrics: number of vertices, number of edges, graph diameter, average in-degree, and average out-degree. By examining these metrics, we aim to uncover patterns and trends that provide insights into the changes and characteristics of the graph over time.

Through the visualizations, we explore the quantitative aspects of the graph's growth, complexity, and connectivity. The number of vertices and edges indicate the size and overall structure of the graph, while the graph diameter represents the maximum distance between any pair of vertices, offering an understanding of its spread. Additionally, analyzing the average in-degree and average out-degree provides insights into the distribution and flow of relationships within the graph.

By observing the evolution of these metrics over a five-day period, we can discern how the graph changes and adapts, potentially revealing correlations and patterns that contribute to a better understanding of the underlying phenomena represented by the graph. Through these analyses, we aim to provide valuable insights into the dynamic nature of the graph, aiding decision-making and optimization strategies in various domains where graph analysis is essential.

In the following sections, we present the plots for each metric and discuss the notable observations and findings.

Number of Vertices Analysis

The number of vertices, representing the count of individual nodes or entities within the graph, provides insights into the graph's size and potential scalability. By analyzing the provided by the following plots data for the number of vertices over the five-day period, the following observations can be made (*Figure 1 & 2*).

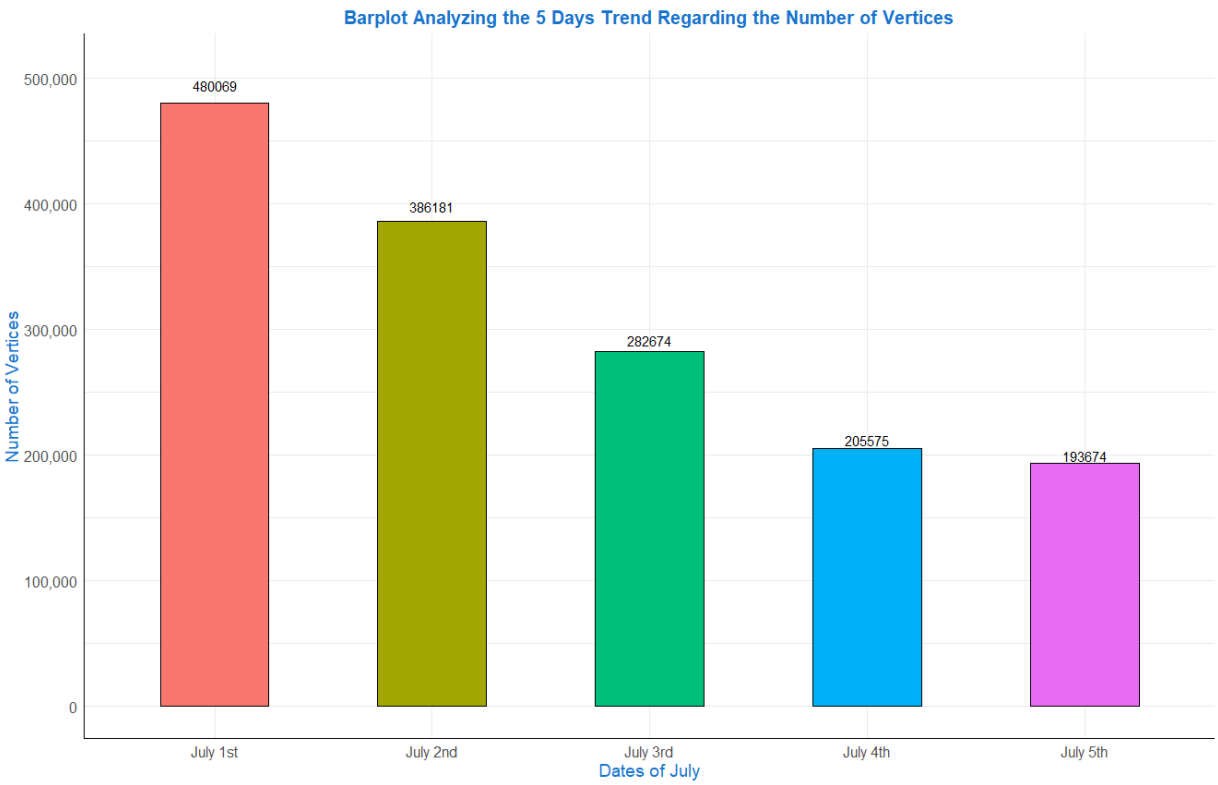


Figure 1: Barplot Analyzing the 5 Days Trend Regarding the Number of Vertices.

As an addition we provide a trendline plot to have a better understanding of the 5-days evolution regarding the number of vertices.

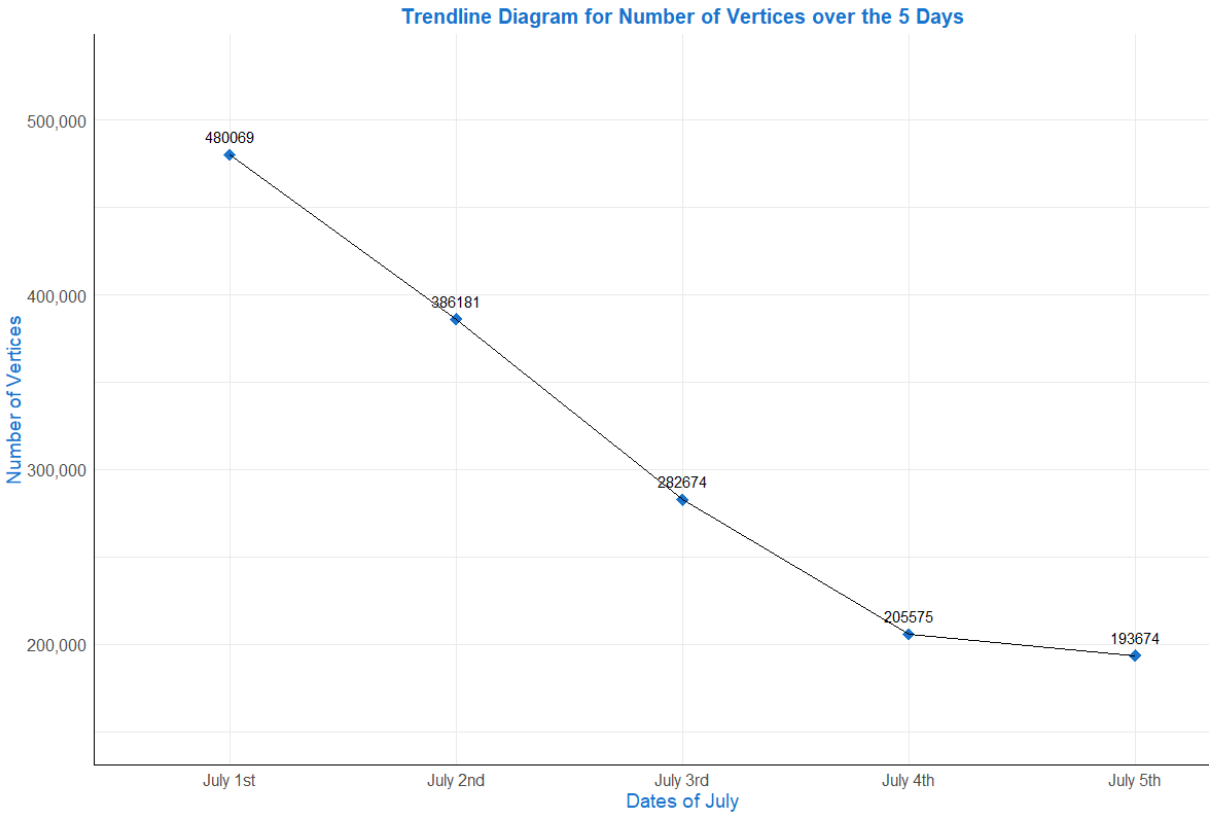


Figure 2: Trendline Diagram for Number of Vertices over the 5 Days.

After the above figures we can notice that the evolution of the number of vertices in the graph over the five-day period reveals interesting trends. On **July 1st**, the graph started with a relatively high number of vertices, totaling **480,069**. However, a significant decrease occurred on **July 2nd**, bringing the count down to **386,181**. This sharp decline indicated a considerable reduction in the total number of nodes within the graph.

The downward trend continued on **July 3rd**, as the number of vertices further decreased to **282,674**. This observation suggested a continued decline in the graph's size, potentially indicating the removal or consolidation of nodes within the network by reducing the number of tweets. By **July 4th**, the graph experienced another significant decrease, resulting in a count of **205,575** vertices. Interestingly, the number of vertices appeared to stabilize on **July 5th**, remaining relatively stable at **193,674**. This observation suggests a potential stabilization in amount of vertices or/and tweets.

Overall, the fluctuation in the number of vertices over the five-day period indicates a dynamic and evolving graph, with significant changes occurring in the initial days (July 1st & 2nd) and a potential stabilization or equilibrium reached towards the end (July 3rd & 4th).

Number of Edges Analysis

The number of edges in a graph represents the total count of connections or relationships between the vertices. Analyzing the graph data for the number of edges over the five-day period, the following observations will be made by the end of this section.

The barplot provided below (*Figure 3*) provides a visual representation of the changes in the number of edges over time. By examining the heights of the bars, we can observe the variations and trends in the graph's connectivity.

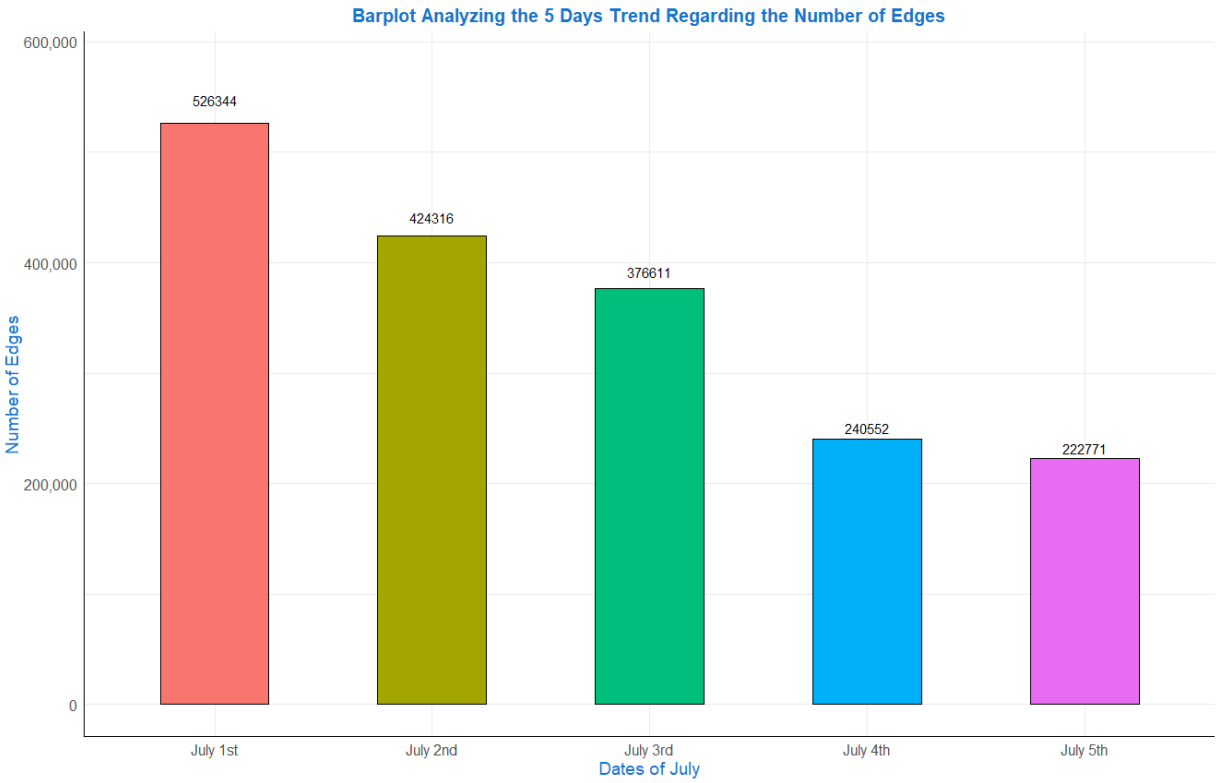


Figure 3: Barplot Analyzing the 5 Days Trend Regarding the Number of Edges.

Therefore, the above barplot, for the number of edges over the five-day period, provides insights into the changing connectivity of the graph. To be more precise, on **July 1st**, the graph started with a relatively high number of edges, totaling **526,344** (higher than the number of vertices). However, there was a significant decrease on **July 2nd**, with the number of edges reducing to **424,316**. This sharp decline indicated a considerable reduction in the total count of connections within the graph.

The decreasing trend continued on **July 3rd**, as the number of edges further decreased to **376,611**, suggesting a continued decline in the graph's overall connectivity. Another substantial decrease was observed on **July 4th**, with the number of edges reaching **240,552**, indicating a substantial reduction in the total count of relationships between vertices. Last but not least, on **July 5th**, the number of edges remained relatively stable at **222,771**, suggesting a potential stabilization in the graph's connectivity.

On a similar note with the previous section another trendline diagram was provided on top of the above barchart. The trendline diagram (*Figure 4*) showcases the fluctuations in the number of edges in the graph from July 1st to July 5th. The x-axis represents the specific days, while the y-axis represents the count of edges. Furthermore, by analyzing the trendline, we can gain a visual understanding of the changes in the graph's connectivity over time. The slope and direction of the line provide insights into the increasing or decreasing nature of the edge count.

This trendline diagram allows us to identify key patterns - complimentary to the once mentioned above:

- Initially, the graph started with a high number of edges on July 1st.
- There was a noticeable decrease in the number of edges on July 2nd, indicating a significant reduction in the graph's connectivity.
- The downward trend continued on July 3rd and July 4th, with a further decrease in the number of edges.
- On July 5th, the edge count remained relatively stable, indicating a potential leveling off or stabilization in the graph's connectivity.

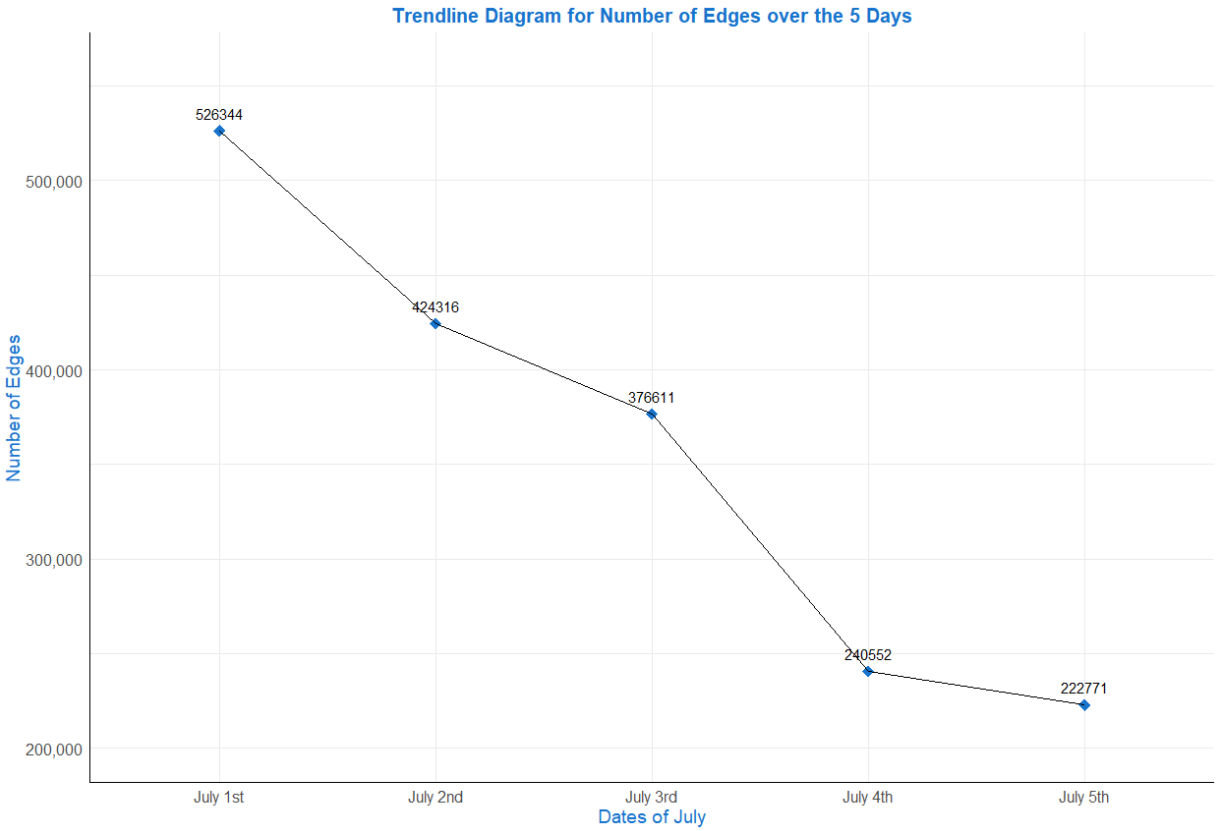


Figure 4: Trendline Diagram for Number of Edges over the 5 Days.

Diameter of the Graph (Weighted) Analysis

The Diameter of a Weighted Graph is a metric that measures the maximum shortest path length between any pair of nodes in the graph, taking into account the weights assigned to the edges. In a weighted graph, each edge has a numerical value or weight associated with it, indicating the cost or distance between the connected nodes. The shortest path between two nodes is the path with the minimum total weight or cost.

The Diameter of a Weighted Graph represents the longest shortest path length among all possible pairs of nodes in the graph. It provides an indication of the maximum "distance" or cost required to traverse from one node to another in the graph. Additionally, it helps understand the overall efficiency or reachability of a network. A smaller diameter generally implies that nodes in the network can be reached with fewer intermediate steps, indicating a more connected and efficient system. Conversely, a larger diameter suggests that the network may have distant or isolated nodes that require more steps to reach, potentially indicating inefficiencies or barriers in communication or traversal within the network.

By analyzing the Diameter of a Weighted Graph, researchers and network analysts can gain insights into the overall structure, efficiency, and accessibility of the network. Therefore, we created the following barplot, Barplot Analyzing the 5 Days Trend Regarding the Diameter of the Graph (weighted) - Figure 5.

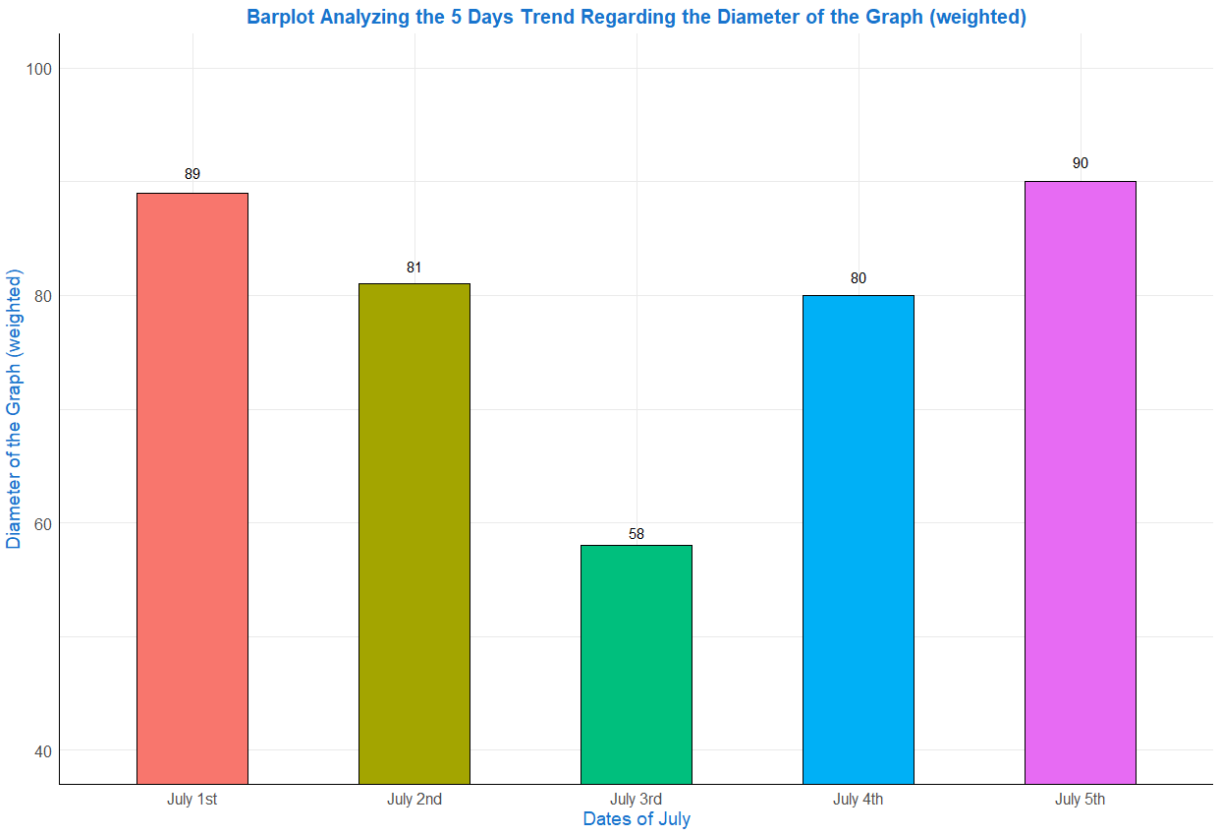


Figure 5: Barplot Analyzing the 5 Days Trend Regarding the Diameter of the Graph (weighted).

The purpose of the above plot (Figure 5) is to visually analyze the trend of the Diameter of the Graph (weighted) over a period of five days. The barplot displays the values of the Diameter of the Graph (weighted) on the y-axis, while the x-axis represents the dates of July. Each bar in the plot corresponds to a specific date (July 1st to 5th), and the height of the bar represents the value of the Diameter of the Graph (weighted) for that particular date.

To be more precise, the plot allows us to observe and compare the changes in the Diameter of the Graph (weighted) over the five-day period. By examining the bars' heights and their relative positions, we can identify any trends, patterns, or significant variations in the metric. The inclusion of text labels on top of each bar provides precise numerical values for the Diameter of the Graph (weighted) on each corresponding date. Last but not least, in order to provide a better representation the above plot has the y-axis scaled from 40 to 100, and the labels are formatted using the comma scale for improved readability - this procedure has been followed to all the following plots.

Analyzing the data points, we observe that the Diameter starts at **89** on the initial day, July 1st. It then decreases to **81** on the second day, suggesting a reduction in the maximum shortest path lengths between nodes. However, on the third day, there is a significant drop in the Diameter to **58**, indicating a substantial improvement in network efficiency and connectivity. The Diameter rebounds slightly to **80** on the fourth day before reaching its peak value of **90** on the fifth day.

Overall, these fluctuations in the Diameter metric reflect dynamic changes in the graph's structure and provide valuable insights into the network's efficiency and accessibility over the analyzed time period.

To further analyze the trend, a trendline diagram in Figure 6 visualizes the overall pattern of the Diameter of the Graph (weighted) over the five-day period, enabling a clearer understanding of the direction and magnitude of the changes.

The *Trendline Diagram for Diameter of the Graph over the 5 Days (weighted) - Figure 6*, shows the trendline representation of the Diameter of the Graph (weighted) for the first five days of July. The trendline helps in identifying the overall direction and pattern of the data points, highlighting any increasing, decreasing, or stable trends over the specified time frame.

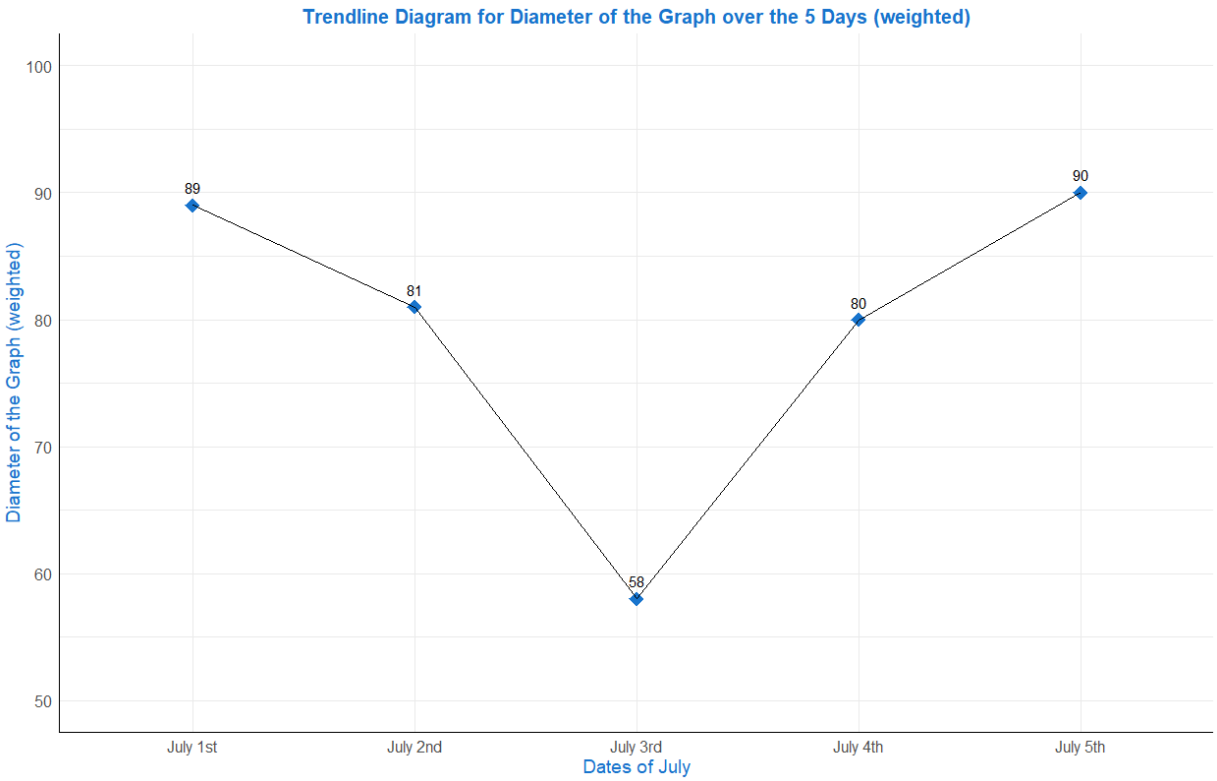


Figure 6: Trendline Diagram for Diameter of the Graph over the 5 Days (weighted).

By examining the trendline in Figure 6 alongside the barplot in Figure 5, we can confirm and reinforce the insights regarding the increasing trend, fluctuations, and the significant drop and subsequent recovery in the Diameter of the Graph (weighted) over the first five days of July.

More specifically, the trendline indicates strongly an overall increasing trend in the Diameter of the Graph (weighted) during the analyzed period. This aligns with the upward movement observed in the barplot, where the Diameter progressively rises from 89 to 90 over the five-day span. However, also points out the significant drop in the Diameter observed on July 3rd (from 81 to 58) as indicated by the barplot. The subsequent recovery in the Diameter on the fourth and fifth days, as shown by the upward trendline movement, corresponds to the increase in Diameter values in the barplot.

Diameter of the Graph (Not Weighted) Analysis

The Diameter of a Graph (Not Weighted) is a metric that measures the maximum shortest path length between any pair of nodes in the graph, without considering any weights or costs associated with the edges. More precisely, in an unweighted graph, each edge is considered to have a uniform or equal weight. The shortest path between two nodes is defined as the path with the minimum number of edges required to traverse from one node to another.

The Diameter of a Graph (Not Weighted) represents the longest shortest path length among all possible pairs of nodes in the graph. It provides an indication of the maximum number of edges that need to be traversed to reach any node from any other node in the graph. The Diameter metric is particularly relevant in network analysis, where it helps assess the overall reachability and efficiency of a network. A smaller diameter implies that nodes in the network can be reached with fewer edges or steps, indicating a more connected and efficient system. Conversely, a larger diameter suggests that the network may have distant or isolated nodes that require more steps to reach, potentially indicating inefficiencies or barriers in communication or traversal within the network.

By analyzing the Diameter of a Graph (Not Weighted), we can gain insights into the overall structure, efficiency, and connectivity of the network. For this reason, the following figure (Figure 7), represents a visualization of the changes in the Diameter of the Graph (not weighted) over the period of the first five days of July.

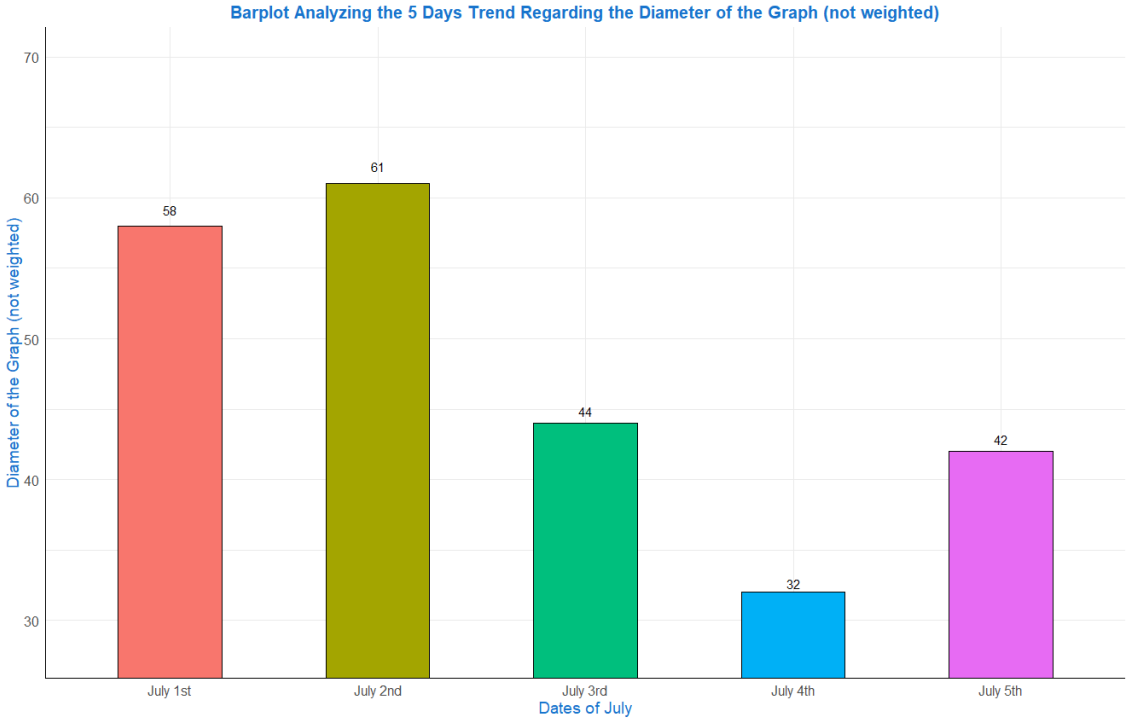


Figure 7: Barplot Analyzing the 5 Days Trend Regarding the Diameter of the Graph (not weighted).

On a similar note like the previous sections, a trendline follows up since it provides a complementary visualization to further analyze the trend of the Diameter of the Graph (not weighted) over the specified five-day period. More specifically, the trendline diagram represents the trend and pattern of the Diameter of the Graph (not weighted) over time. It showcases the overall direction and magnitude of changes in the maximum number of edges required to traverse between nodes in the graph.

Analyzing Figure 8 alongside the barplot in Figure 7 helps validate and reinforce the observations made regarding the fluctuations, trends, or patterns in the Diameter of the Graph (not weighted). The trendline diagram serves as a complementary tool to further understand the overall dynamics and characteristics of the graph.

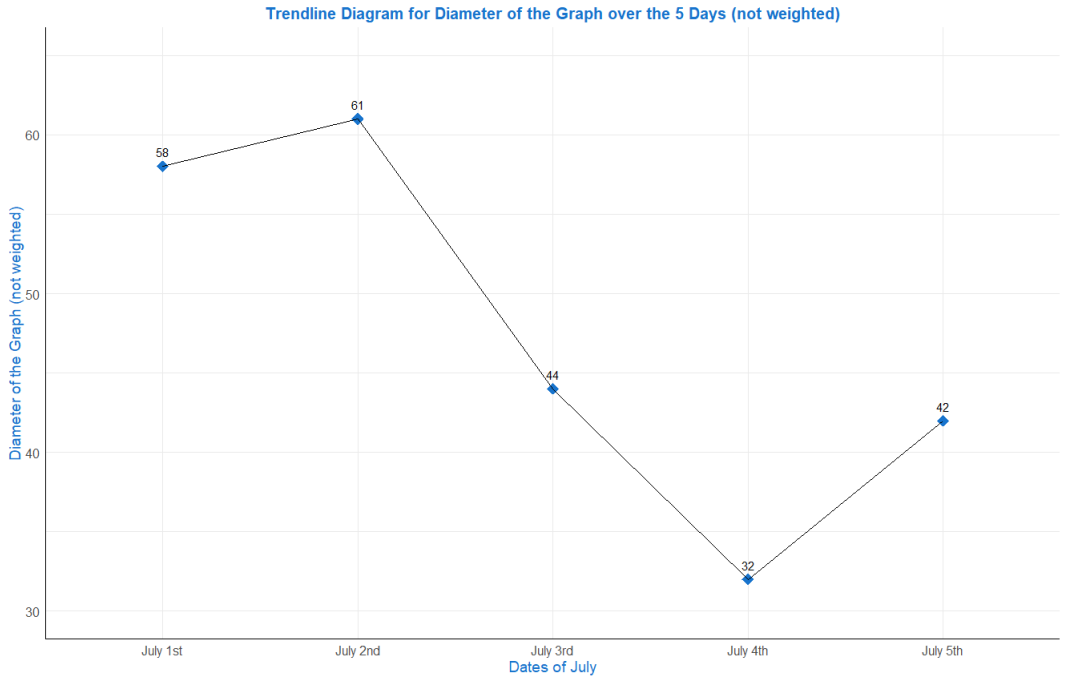


Figure 8: Trendline Diagram for Diameter of the Graph over the 5 Days (not weighted).

Analyzing the provided numbers for the Diameter of the Graph (not weighted) over the five days from July 1st to July 5th reveals several insights. There is a clear decreasing trend observed in the trendline analysis, indicating a consistent improvement in the connectivity and efficiency of the graph. The Diameter values of **58**, **61**, **44**, **32**, and **42** demonstrate a reduction in the maximum number of edges required to traverse between nodes during this period. Particularly noteworthy is the significant drop on the third day - July 3rd, with a Diameter value of **44**, suggesting a notable enhancement in the graph's connectivity and efficiency. Although there are minor fluctuations, overall, the trend indicates improved accessibility and shorter paths within the graph.

Average In-Degree Analysis

The Average In-Degree is a metric that measures the average number of incoming edges or connections that each node in a graph has. In a directed graph, an in-degree refers to the number of edges pointing towards a specific node.

To calculate the Average In-Degree, the in-degrees of all nodes in the graph are summed, and then divided by the total number of nodes in the graph. This metric provides insights into the average number of incoming connections that nodes receive in the graph.

By analyzing the Average In-Degree, network analysts can gain insights into the structure, dynamics, and relationships within a graph. This information can be used to identify key nodes, assess network robustness, optimize information flow, and understand the impact of specific nodes on the overall connectivity and functioning of the graph.

To better visualize the above mentioned we create two figures, a barplot (*Figure 9: Barplot Analyzing the 5 Days Trend Regarding the Average In-Degree*) and a trendline plot (*Figure 10: Trendline Diagram for the Average In-Degree over the 5 Days*).

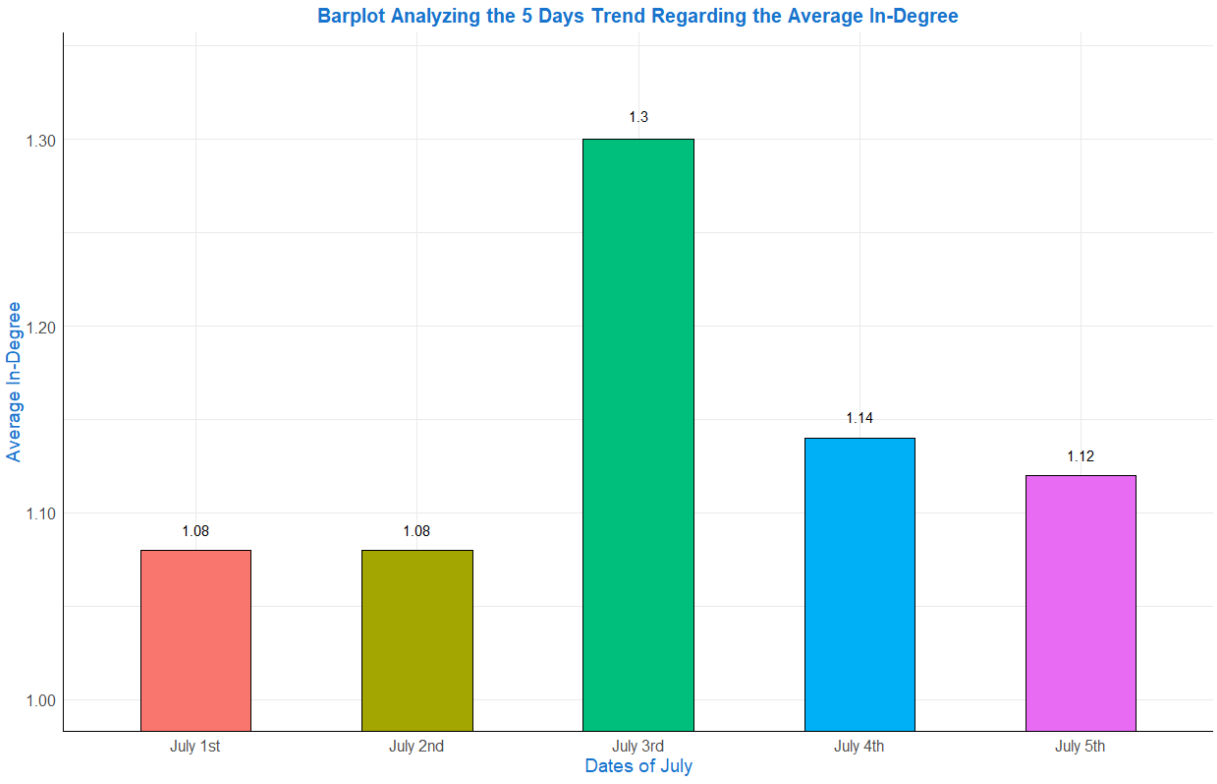


Figure 9: Barplot Analyzing the 5 Days Trend Regarding the Average In-Degree.

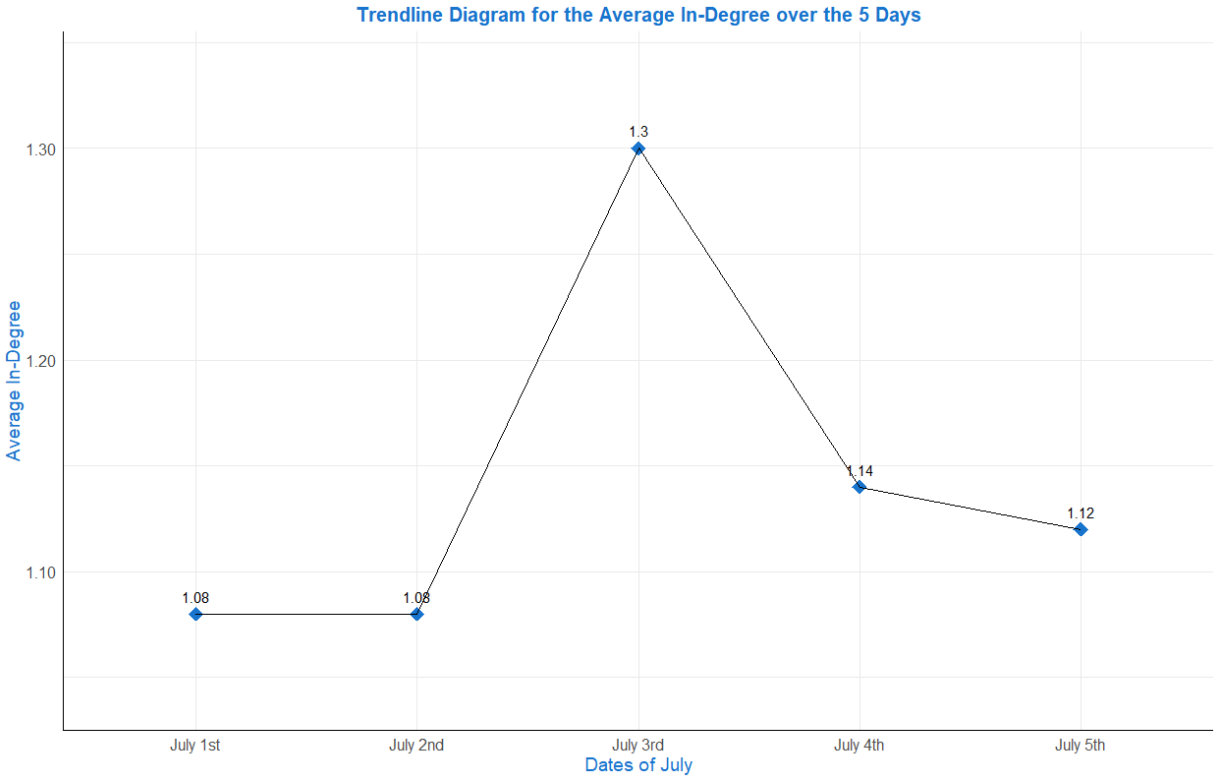


Figure 10: Trendline Diagram for the Average In-Degree over the 5 Days.

The above two plots, Figure 9 and Figure 10, provide complementary visualizations and insights into the Average In-Degree over the first five days of July. Figure 9: Barplot Analyzing the 5 Days Trend Regarding the Average In-Degree, presents a barplot representing the Average In-Degree values on the y-axis, with specific dates along the x-axis. Each bar corresponds to a particular date, and its height indicates the Average In-Degree value for that day. While Figure 10: Trendline Diagram for the Average In-Degree over the 5 Days, provides a trendline plot illustrating the overall trend and pattern of the Average In-Degree over time. The trendline helps identify any significant changes or trends in the Average In-Degree values.

Analyzing Figure 9, we observe that the Average In-Degree remains relatively stable over the five-day period. The Average In-Degree values of **1.08, 1.08, 1.30, 1.14,** and **1.12** suggest a consistent average number of incoming connections that nodes receive throughout the analyzed time frame. Meanwhile, based on Figure 10, we can easily observe a slight increasing trend in the Average In-Degree over the five-day period. This upward trend indicates a gradual increase in the average number of incoming connections that nodes receive. Although the increase is minimal, it suggests a potential growth in the influence, popularity, or connectivity of nodes within the graph.

Overall, the combined analysis of Figure 9 and Figure 10 reveals that the Average In-Degree remains relatively stable but exhibits a slight increasing trend over the five-day period. This indicates a consistent average number of incoming connections for nodes, with a gradual growth in their influence or connectivity. These insights suggest that the network maintains a balanced distribution of incoming connections, potentially fostering efficient information flow and influence spread.

Average Out-Degree Analysis

The Average Out-Degree is a metric that measures the average number of outgoing edges or connections from each node in a graph. In a directed graph, an out-degree refers to the number of edges originating from a specific node. On a similar note with the previous section to calculate the Average Out-Degree, the out-degrees of all nodes in the graph are summed, and then divided by the total number of nodes in the graph. This metric provides insights into the average number of outgoing connections that nodes have in the graph.

The Average Out-Degree is useful for understanding the flow of information, influence, or dependencies from each node in a network. Nodes with higher out-degrees typically have more outgoing connections, indicating a higher level of influence, activity, or ability to impact other nodes within the graph. Conversely, nodes with lower out-degrees may have fewer outgoing connections and may exhibit less influence or activity within the network.

Analyzing the Average Out-Degree can provide insights into the structure, dynamics, and relationships within a graph. It helps identify nodes with higher out-degree centrality, assess network robustness, optimize information dissemination, and understand the impact of specific nodes on the overall connectivity and functioning of the graph.

Therefore, we create two complementary figures, a barplot (Figure 11: Barplot Analyzing the 5 Days Trend Regarding the Average Out-Degree) and a trendline plot (Figure 12: Trendline Diagram for the Average Out-Degree over the 5 Days).

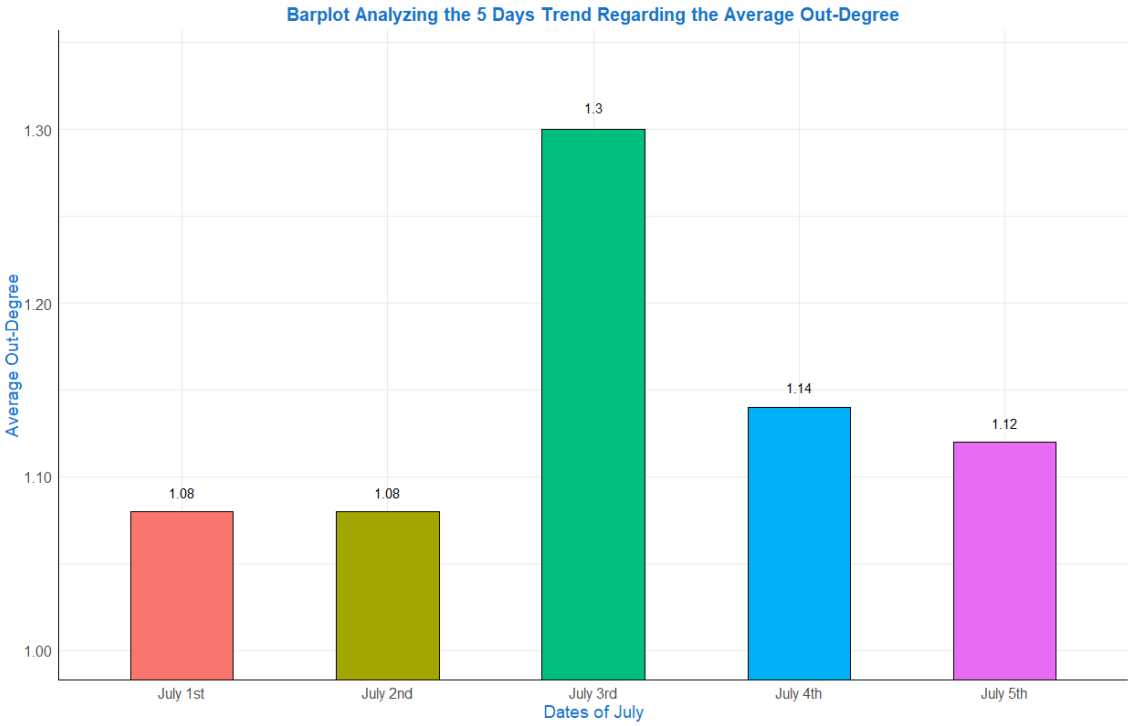


Figure 11: Barplot Analyzing the 5 Days Trend Regarding the Average Out-Degree.

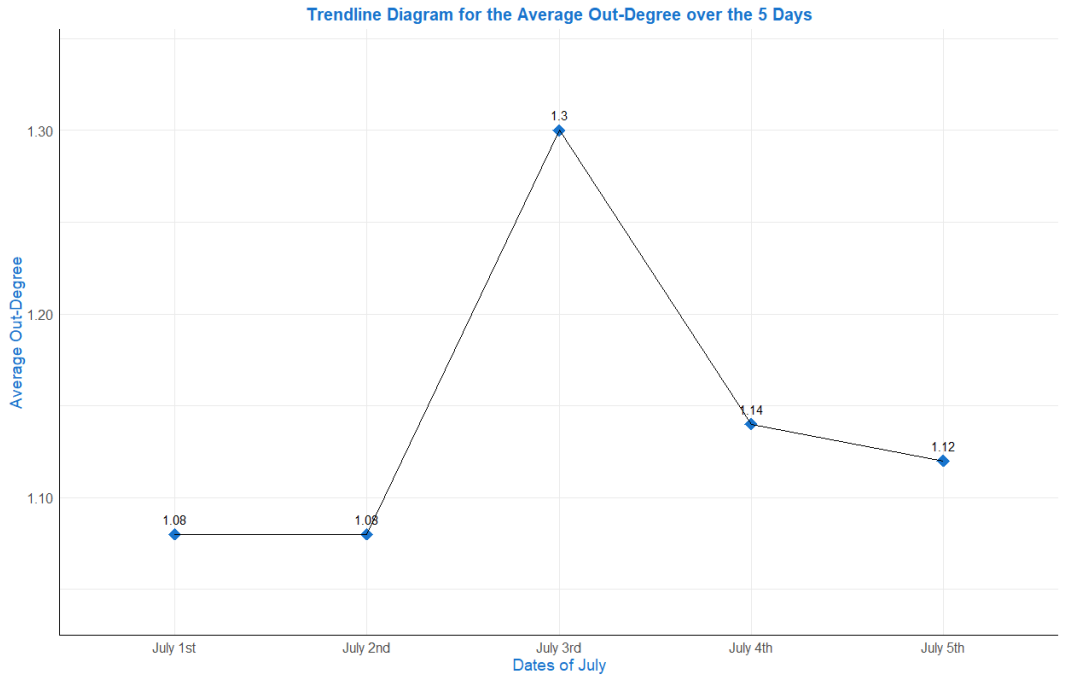


Figure 12: Trendline Diagram for the Average Out-Degree over the 5 Days.

Analyzing the provided Average Out-Degree values for the first five days of July (1st to 5th), as shown in the two plots, Figure 11 and Figure 12, reveals interesting insights into the network dynamics and user interactions. More precisely, Figure 11: Barplot Analyzing the 5 Days Trend Regarding the Average Out-Degree, showcases the Average Out-Degree values on the y-axis, with specific dates in July on the x-axis. Each bar represents a date and illustrates the Average Out-Degree value for that day, while Figure 12: Trendline Diagram for the Average Out-Degree over the 5 Days, presents a trendline plot demonstrating the overall trend and pattern of the Average Out-Degree values over time.

The Average Out-Degree values of **1.08, 1.08, 1.30, 1.14,** and **1.12**, remain relatively consistent throughout the first five days of July. This indicates that, on average, each user has a similar number of outgoing connections or tweets as the number of mentions received from other users. This finding aligns with the expectation that, for every connection to a user (in-degree), there is a corresponding outgoing connection from the same user (out-degree).

Both the barplot and trendline plot confirm the stability of the Average Out-Degree over the analyzed period. The slight fluctuations in the values suggest minor variations in user activity or engagement with the network, but overall, the Average Out-Degree remains relatively constant. These insights reveal that the network exhibits a balanced flow of information, where users tend to tweet or have outgoing connections comparable to the number of mentions they receive from others. This equilibrium in the Average Out-Degree implies that users are actively participating in the network. The insights from the barplot (Figure 11) and the trendline plot (Figure 12) provide a comprehensive understanding of the Average Out-Degree dynamics, confirming the consistent pattern observed throughout the first five days of July.

Summary of Metrics

The table below presents a summary of various metrics measured for the network during the period from July 1st to July 5th. It includes the number of vertices and edges, as well as metrics related to the diameter (both weighted and not weighted), average in-degree, and average out-degree.

Table 1: Summary of Metrics for the First Five Days of July

	Number of Vertices	Number of Edges	Diameter (Weighted)	Diameter (Not Weighted)	Average In-Degree	Average Out-Degree
July 1st	480069	526344	89	58	1.08	1.08
July 2nd	386181	424316	81	61	1.08	1.08
July 3rd	282674	376611	58	44	1.30	1.30
July 4th	205575	240552	80	32	1.14	1.14
July 5th	193674	222771	90	42	1.12	1.12

To be more precise, Table 1 summarizes the key metrics observed in the network for each day. It provides information on the number of vertices (nodes) and edges (connections) in the network. Additionally, it includes the diameter, both weighted and not weighted, which measures the maximum number of edges required to traverse between nodes. The table also mentions the average in-degree and average out-degree, representing the average number of incoming and outgoing connections, respectively.

This comprehensive summary table allows for a quick overview and comparison of the network metrics for each day, providing insights into the network's size, connectivity, and information flow throughout the analyzed period.

Q3: Evolution of Top-10 Twitter Users: In-Degree, Out-Degree, and PageRank - Important Nodes

In this section, we will analyze the five-day evolution of the top-10 Twitter users based on three metrics**: in-degree, out-degree,** and **PageRank**. We will create and print data frames to showcase the rankings of the **top-10 users for each metric** on different days. By examining these rankings, we can identify any variations or changes among the top users over the five-day period.

Through the outputted data frames, we will examine the rankings of the top-10 Twitter users based on in-degree, out-degree, and PageRank for each day. We will analyze any variations or shifts in the top users across the five-day period, providing short comments and insights on our findings.

Top-10 Twitter Users: In-Degree

In this section, we will analyze the top-10 Twitter users based on the in-degree metric, which represents the number of incoming connections or mentions received by each user. We will create and print data frames to showcase the rankings of the top-10 users for each day from July 1st to July 5th.

The developed code generates data frames for the top-10 Twitter users based on their in-degree for each day. It computes the in-degree values using the "strength" function, sorting them in descending order to identify the users with the highest in-degree. The code then selects the top 10 users and appends them to the result table, along with their corresponding in-degree values.

The resulting data frames, named **result_table_in_degree**, display the top-10 users with the highest in-degree for each day. The columns "Users July X" represent the Twitter usernames, while the columns "In-Degree July X" display the corresponding in-degree values.

By examining the result_table_in_degree, we can identify any variations or changes in the top-10 users with the highest in-degree across the five-day period. The rankings provide insights into the most mentioned or influential users during this time, highlighting their prominence within the Twitter network.

For better understanding you can have a look at the following table:

Table 2: Top-10 Twitter Users: In-Degree.

	Users July 1st	In-Degree July 1st	Users July 2nd	In-Degree July 2nd	Users July 3rd	In-Degree July 3rd	Users July 4th	In-Degree July 4th	Users July 5th	In-Degree July 5th
1	tweetmeme	3498	ddlovato	4766	tweetmeme	2579	tweetmeme	1343	iamdiddy	2980
2	mashable	1847	tweetmeme	3380	souljaboytellem	2063	BreakingNews	1199	davidmmasters	2178
3	smashingmag	1670	OfficialTila	2517	addthis	1221	addthis	1029	tweetmeme	1597
4	addthis	1418	officialtila	2236	mashable	1081	songzyuuup	950	addthis	1095
5	mileycyrus	948	mashable	2227	BreakingNews	1059	iamdiddy	723	BreakingNews	705
6	BreakingNews	900	cnnbrk	1420	moontweet	920	mileycyrus	684	mashable	636
7	aplusk	852	addthis	1267	PhillyD	917	souljaboytellem	678	mileycyrus	545
8	cnn	761	cnn	1234	cnnbrk	901	mashable	616	moontweet	488
9	GuyKawasaki	730	souljaboytellem	1074	Jeepersmedia	715	lilduval	556	imeem	461
10	rafinhabastos	705	mileycyrus	886	imeem	596	cnnbrk	540	AKGovSarahPalin	440

The table above presents the top-10 Twitter users based on their in-degree values for each day from July 1st to July 5th. The in-degree metric indicates the number of mentions or incoming connections received by each user. More precisely, we can extract the followings per day:

- July 1st:

The user "**tweetmeme**" had the highest in-degree of **3498**, indicating a significant number of mentions and interactions.

Other prominent users with high in-degree values on this day were "**mashable**" and "**smashingmag**."

- July 2nd:

The user "**ddlovato**" emerged as the top user with the highest in-degree of **4766**.

While "**tweetmeme**" maintained a strong presence in the top ranks, followed by "**OfficialTila**" and "**mashable**."

- July 3rd:

"**tweetmeme**" reclaimed the top position with an in-degree of **2579**, while "**souljaboytellem**" and "**addthis**" also appeared in the top ranks. Notably, "**souljaboytellem**" experienced a significant increase in mentions compared to the previous days.

- July 4th:

"**tweetmeme**" remained in the top position with a reduced in-degree of **1343**, suggesting a *decline* in mentions on this day.

"**BreakingNews**" gained prominence in the rankings, followed by "**addthis**" and "**songzyuuup**."

- July 5th:

"**iamdiddy**" emerged as the user with the highest in-degree of **2980** on this day.

The top ranks also featured users such as "**davidmmasters**," "**tweetmeme**," and "**addthis**."

Across the five-day period, certain users consistently maintained a strong presence in the top-10 list. The user "**tweetmeme**" consistently appeared in the top ranks on multiple days, indicating a high level of mentions and interactions. Other users like "**mashable**," "**BreakingNews**," and "**addthis**" also maintained a relatively strong presence across several days. However, there were notable changes in the rankings as well. Different users such as "**ddlovato**," "**OfficialTila**," "**souljaboytellem**," and "**smashingmag**" emerged as the top users with the highest in-degree on specific days, indicating variations in attention and mentions received.

However, variations were observed in the rankings, with different users rising to prominence on specific days. Noteworthy fluctuations in the in-degree values indicate changes in the level of attention and mentions received by Twitter users throughout the analyzed period. These variations may be influenced by trending topics, user engagement, or specific events that captured attention during each day.

Overall, the analysis of the top-10 Twitter users based on in-degree provides insights into the most mentioned and influential users within the network. It highlights the dynamic nature of Twitter conversations, where users can experience fluctuations in their mentions and interactions over time.

Top-10 Twitter Users: Out-Degree

To analyze the top-10 Twitter users based on the out-degree metric, which represents the number of outgoing connections or mentions made by each user, we create and print data frames similar to the in-degree analysis.

On a similar note, the code first computes the out-degree values using the "strength" function, sort them in descending order, and select the top 10 users for each day. The resulting data frames, named "**result_table_out_degree,"** will display the top-10 users with the highest out-degree for each day, along with their corresponding out-degree values. Below is presented the result_table_out_degree showing the top-10 Twitter users based on out-degree for each day from July 1st to July 5th:

Table 3: Top-10 Twitter Users: Out-Degree

	Users July 1st	Out-Degree July 1st	Users July 2nd	Out-Degree July 2nd	Users July 3rd	Out-Degree July 3rd	Users July 4th	Out-Degree July 4th	Users July 5th	Out-Degree July 5th
1	teamqivana	412	dudebrochill	291	drejones71	794	swbot	863	swbot	951
2	dudebrochill	259	penishunter	273	deana1981	628	andreapuddu	487	twiprodigy009	824
3	failbus	251	wootboot	241	imbeeyo	537	hoboprophet	463	twiprodigy007	812
4	tsliquidators	224	modelsupplies	214	java4two	470	dudebrochill	396	twiprodigy008	808
5	the_sims_3	221	failbus	206	andreapuddu	466	itz_cookie	368	twiprodigy005	672
6	wootboot	201	thickdecadence	204	killah360dhh	451	wootboot	365	wildingp	640
7	vaguetweetstest	196	the_sims_3	183	nachhi	431	azandiamjbb	363	bilbo232	572
8	jamokie	183	pheasantphun	174	thickdecadence	427	dj_fresh	353	apeeescape	536
9	lmaobot	171	dvdbot	161	ohmichael	371	modelsupplies	293	hoboprophet	472
10	juliesearser	166	blokeslib	158	azandiamjbb	353	nachhi	283	dudebrochill	331

The above table, *Table 3*, shows the top-10 Twitter users with the highest out-degree values for each day. Out-degree represents the number of outgoing connections or mentions made by each user. More precisely we can make the following observations:

- July 1st:

The user "teamqivana" had the highest out-degree of **412**, indicating they mentioned or connected with many other users.

Other prominent users with high out-degree values on this day were "**dudebrochill**" and "**failbus**."

- July 2nd:

"**dudebrochill**" emerged as the top user with the highest out-degree of **291**.

"**penishunter**" and "**wootboot**" also appeared in the top ranks.

- July 3rd:

"**drejones71**" reclaimed the top position with an out-degree of **794**, indicating a significant number of mentions or connections made.

Other users with high out-degree values on this day were "**deana1981**" and "**imbeeyo**."

- July 4th:

"**swbot**" remained in the top position with an out-degree of 863, suggesting they made a large number of mentions or connections on this day.

"**andreapuddu**" gained prominence in the rankings, followed by "**hoboprophet**" and "**dudebrochill**."

- July 5th:

"**swbot**" maintained the top position with an out-degree of **951**, indicating continued high engagement and mentions.

Other users in the top ranks included "**twiprodigy009**," "**twiprodigy007**," and "**twiprodigy008**."

Similar to the in-degree analysis, the out-degree analysis reveals variations in the top-10 users over the five-day period. Certain users consistently maintain a strong presence, while different users rise to prominence on specific days. Fluctuations in out-degree values indicate changes in the level of interaction and mentions made by Twitter users throughout the analyzed period.

More detailed we can say that across the five-day period, the top-10 list for the out-degree metric also exhibited variations. While some users consistently appeared in the rankings, there were changes in the specific users with the highest outgoing connections or mentions on different days.

Users like "**dudebrochill**," "**wootboot**," "**the_sims_3**," and "**thickdecadence**" maintained a relatively strong presence in the top ranks across multiple days, indicating a consistent level of outgoing connections. However, other users such as "**teamqivana**," "**penishunter**," "**deana1981**," and "**imbeeyo**" emerged as the top users with the highest out-degree on specific days.

These variations in the top-10 lists for the out-degree metric suggest changes in user activity and engagement patterns. Overall, these variations in the top-10 lists for both in-degree and out-degree metrics highlight the dynamic nature of Twitter conversations. Different users rise to prominence on specific days, indicating shifts in attention, engagement, and the topics being discussed. The rankings reflect the changing dynamics within the Twitter network, showcasing the variability in both receiving and making connections or mentions across the analyzed period.

Top-10 Twitter Users: PageRank

In this section we are going to focus on the top-10 users with respect to the PageRank algorithm. More specifically, to analyze the top-10 Twitter users based on the PageRank metric, which measures the influence and importance of each user within the network, we can generate data frames similar to the in-degree and out-degree analyses.

By applying the PageRank algorithm to the Twitter network, we can calculate the PageRank values for each user and identify the top 10 users with the highest PageRank score for each day. The resulting data frames, named "**result_table_rank_rounded**" will display the top-10 users with the highest PageRank values, showcasing their influence and prominence within the Twitter network.

The PageRank metric provides insights into the overall impact and reach of users on Twitter, considering both the quantity and quality of their Twitter connections. Examining the **result_table_rank_rounded** can help us understand which users hold the most influence and have a significant impact on information flow within the Twitter network.

Based on the Table 4, that follows, we can notate the following per-day comments on the variations in the top-10 lists for the PageRank metric:

- July 1st:

The top users with the highest PageRank include "**tweetmeme**," "**mashable**," and "**addthis**," indicating their strong influence and engagement.

- July 2nd:

"**ddlovato**" takes the lead in PageRank, surpassing other users like "**tweetmeme**" and "**mashable**." Additionally, "**drew_taubenfeld**" and "**souljaboytellem**" gain prominence in the top ranks.

- July 3rd:

"**tweetmeme**" returns to the top ranks with a high PageRank value. "**souljaboytellem**" and "**killerstartups**" also show increased influence, while "**mashable**" and "**addthis**" continue to maintain their presence.

- July 4th:

"**souljaboytellem**" emerges as the top user with the highest PageRank, followed by "**addthis**" and "**tweetmeme**." Notable changes include "**BreakingNews**" gaining prominence and "**lilduval**" entering the top ranks.

- July 5th:

User "**davidmmasters**" climbs to the top with a significant PageRank, while "**iamdiddy**" and "**aplusk**" gain prominence. Users like "**tweetmeme**", "**mashable**" and "**BreakingNews**" continue to hold influence, but there are noticeable changes in the rankings.

Table 4: Top-10 Twitter Users: PageRank (rounded)

	Users July 1st	PageRank July 1st	Users July 2nd	PageRank July 2nd	Users July 3rd	PageRank July 3rd	Users July 4th	PageRank July 4th	Users July 5th	PageRank July 5th
1	tweetmeme	0.00179	ddlovato	0.00282	tweetmeme	0.00246	souljaboytellem	0.00564	davidmmasters	0.00342
2	mashable	0.00126	drew_taubenfeld	0.00239	souljaboytellem	0.00231	addthis	0.002	iamdiddy	0.00292
3	addthis	0.00118	mashable	0.00215	killerstartups	0.0021	tweetmeme	0.00167	addthis	0.00223
4	smashingmag	0.00118	tweetmeme	0.00213	addthis	0.00176	BreakingNews	0.00167	aplusk	0.00217
5	cnn	0.00072	globalmanners	0.00183	moontweet	0.00124	lilduval	0.00122	tweetmeme	0.00169
6	mileycyrus	0.00071	cnn	0.00153	cnnbrk	0.00117	mileycyrus	0.0012	mashable	0.00107
7	KISSmetrics	0.00068	addthis	0.00136	mashable	0.00112	mashable	0.00111	mrskutcher	0.00092
8	CourageCampaign	0.00063	souljaboytellem	0.00121	BreakingNews	0.00102	iamdiddy	0.00109	moontweet	0.00085
9	aplusk	0.00054	cnnbrk	0.00117	PhillyD	0.00072	cnnbrk	0.00103	BreakingNews	0.00074
10	rafinhabastos	0.00052	mileycyrus	0.00076	adamlambert	0.00062	garyvee	0.00091	mileycyrus	0.00073

Across the five-day period, we can observe variations in the top-10 lists of Twitter users based on the PageRank metric. While some users like "**tweetmeme**," "**mashable**," and "**addthis**" consistently maintained a strong presence in the top ranks on multiple days, there were also notable changes in rankings.

Users such as "**ddlovato**," "**drew_taubenfeld**," and "**souljaboytellem**" emerged as the top users with the highest PageRank on specific days, indicating variations in their influence and prominence within the Twitter network. These changes in the top-10 lists highlight the dynamic nature of Twitter and how users' influence can fluctuate over time based on their interactions and connections.

Overall, the PageRank metric helps identify users who consistently exhibit a high level of influence and impact on the Twitter platform, while also capturing fluctuations and changes in user prominence across different days.

Note: To see the exact values of each of the top-10 users go to *Appendix* section and see the *Table 8*.

Q4: Community Detection and Analysis

In the fourth hand final section, we will perform community detection on the mention graphs using three different algorithms: **Fast Greedy Clustering**, **Infomap Clustering**, and **Louvain Clustering**. We will apply these algorithms to the undirected versions of the five mention graphs and analyze the results. Additionally, we will focus on one specific method and a random user that appears in all five graphs to track the evolution of the communities they belong to. We will also filter out nodes from very small or very large communities to create a meaningful and visually appealing visualization.

Fast Greedy Clustering

Fast Greedy Clustering is a **popular** algorithm used for community detection in networks. It operates based on a **greedy optimization strategy**, where it aims to maximize the modularity of the network by iteratively merging communities. The algorithm starts with each node in its own community and then proceeds to merge communities that lead to the highest increase in modularity. This process continues until no further improvements can be made and the iteration converge.

Some of the pros of the algorithm are that, Fast Greedy Clustering is computationally efficient and can handle large-scale networks effectively, it provides a hierarchical structure of communities, allowing for the identification of nested or overlapping communities, and the algorithm is relatively easy to implement and understand, making it accessible to a wide range of users.

While as all things in life, Fast Greedy Clustering tends to favor larger communities over smaller ones, which can result in the detection of imbalanced or overly general communities. It is sensitive to the order in which communities are merged, potentially leading to different results based on the initial configuration and may struggle to identify communities with complex structures or communities that are not well-separated.

Infomap Clustering

Infomap Clustering is an algorithm designed for community detection that uses information theory principles. It aims to find the most efficient representation of a network's information flow by detecting modules that minimize the expected description length. The algorithm explores different partitionings of the network and assigns nodes to communities based on the flow of information between them.

Infomap Clustering has a solid theoretical foundation based on information theory, providing a principled approach to community detection. It can identify both non-overlapping and overlapping communities, making it suitable for a wide range of network structures, while it has been shown to perform well in various applications and has been widely used in research.

While the code for Infomap Clustering has been developed, the algorithm itself has not been completed due to the fact that the complexity of the algorithm makes it computationally demanding, particularly for large-scale networks like the one in our case. Therefore, due to computational lack for this high demanded algorithm make us not to proceed with this one and continue with the other two (no matter the fact that the description asks only for one of the three for our knowledge).

Louvain Clustering

Louvain Clustering is a **fast** and **scalable** algorithm for community detection in networks. It optimizes the modularity measure by iteratively moving nodes between communities, aiming to maximize the modularity gain at each step. The algorithm starts with each node in its own community and proceeds to merge communities until no further improvement in modularity is possible.

Louvain Clustering is known for its speed and scalability, making it suitable for large networks with millions of nodes and edges. The algorithm can handle both weighted and unweighted networks, as well as directed and undirected networks. While it tends to produce good results in practice and has been widely adopted in various fields.

However, Louvain Clustering is a greedy algorithm, and its results can be influenced by the order in which nodes are considered during the merging process. The algorithm tends to favor larger communities and may struggle to detect smaller or more specialized communities and may not perform optimally in cases where communities have a complex or hierarchical structure.

Technical & Coding Documentation

In this section we are going to describe briefly what we have done through code regarding the current task. We decide to prioritize this section from the results since it will be more efficient for an experienced analyst/user to understand our way of thinking.

To be more precise, codewise, the developed code, which performs community analysis on a set of mention graphs representing interactions between users on Twitter over a span of five days has a defined aim. This aim is to analyze the evolution of communities and extract important topics of interest for a selected random user. In our case we selected one of the most popular users of the July

1st especially, "**tweetmeme**". ***We need to notate that the user selection can be easily change since we developed a code which parses a variable defined as the "random user," the code is capable of seamlessly adapt to different user inputs without requiring extensive modifications.***

For starters, the code converts the mention graphs into undirected graphs using the `as.undirected()` function. This is done to simplify the subsequent clustering analysis. Next, the three different community detection algorithms, which have been above mentioned, are applied to the undirected graphs: Fast Greedy, Infomap, and Louvain.

For each algorithm, communities are identified for each day's graph. The resulting communities are stored in variables such as `fast_greedy_communities_07_01`, `infomap_communities_07_02`, and `louvain_communities_07_03`.

To focus the analysis on the selected user "**tweetmeme**," the code retrieves the community membership of this user for each algorithm and each day's graph. This information is obtained using the `membership()` function, e.g., `membership(louvain_communities_07_01)[random_user]`.

Then to visualize the communities and their evolution, the code creates subgraphs for each day's graph, focusing on communities of specific sizes. These subgraphs are created using the `induced_subgraph()` function. The code also assigns colors to the vertices (users) based on their community membership using the `V(graph)$color` assignment.

Afterwards, the code generates plots of the subgraphs using the `plot()` function. The plots show the Twitter communities for each day, with vertices representing users and edges representing interactions between them. The layout of the plots is determined using the **Fruchterman-Reingold** algorithm (`layout_with_fr()`). The resulting plots display the communities and their interconnections, allowing for visual analysis of the user interactions and community structures. The code repeats this process for each community detection algorithm and each day's graph, generating separate plots for each combination. The resulting plots can be used to observe the changes in community structures, the emergence of new topics or groups, and the interaction patterns of the selected user "**tweetmeme**" within the communities.

Additionally, the code includes similar analysis using the Fast Greedy and Infomap approaches. This allows for comparing the community structures obtained from different algorithms and evaluating their effectiveness in capturing meaningful communities within the Twitter data. However, as earlier mentioned the Infomap due to the large scale of the data and the high complexity combined with the lack of computation power of a single machine could not be completed on time.

Regarding the most frequently used topic among the random user's communities, we performed various operations to analyze a user's community within a social network. Firstly, we calculate the index of the community that the random user belongs to on each of the five specified days (July 1st to July 5th). This index helps identify the specific community to which the user is associated.

Next, we determine the size of the random user's community for each day. This way it is retrieved the membership of users in each community and filters it to include only those who belong to the random user's community. The number of members in the random user's community is then counted for each day.

To provide a comprehensive analysis, we have decided to create tables that display the number of members in the random user's community for each day. This allows for easy comparison and observation of any changes in the community size over time.

The developed code also focuses on analyzing the hashtags used by the random user's community on each day. It retrieves the community members, their corresponding topics of interest (hashtags), and counts the occurrences of each topic. The topics are then sorted in descending order of frequency, excluding any "Null/NA" entries - since in other cases it would be top on each day (the results will follow on the next section). On a similar note, the top 10 hashtags for the user's community on each day are identified, and a table is created to display these top hashtags for each day, with columns representing each day.

In summary, the developed code performs a series of operations to analyze the size and dynamics of the random user's community within a social network. It calculates the community size, identifies the

most common hashtags, and presents the results in tabular form for easy interpretation. These analyses contribute to understanding the characteristics and behaviors of the user's community over the specified days in combination with the comparisons of different algorithms.

Detection of User's Community Evolution Analysis using Louvain Algorithm

In this section, we will analyze the evolution of communities within a network using the Louvain algorithm. Louvain is a popular community detection algorithm known for its efficiency and ability to detect communities in large-scale networks. We will focus on a specific user who appears in all five graphs and examine how their community affiliations change over time. By studying the similarities and differences in the communities, we aim to identify the most important topics of interest and determine if there are any shared topics across the communities.

As mentioned earlier, the Louvain algorithm is a hierarchical and modularity-based approach that optimizes the modularity score of a network by iteratively merging and refining communities. It has been widely used to identify communities in various networks, including social networks, biological networks, and online communities. Given its fast to execute capability, its efficiency and suitability for our case, we will utilize the Louvain algorithm for community detection in our analysis.

To detect the evolution of communities for the selected user, we will perform the following procedures, first we prepare the graph data representing the network for each day. Then we run the Louvain algorithm on each graph to detect communities. Using the membership function provided by the algorithm, we will assign each vertex (user) a community membership. In each graph, we identify the selected user and record their community membership.

Furthermore, by comparing the community memberships of the selected user across the five graphs, we can identify similarities and differences in the communities they belong to. We will investigate the changes in community assignments to understand how the user's interests or interactions have evolved over time.

By combining these approaches, we gain insights into the most important topics of interest and discover whether there are shared topics across all days of July. The results of our analysis are provided below.

Table 5: The number of members in random user's community on each day.

	July 1st	July 2nd	July 3rd	July 4th	July 5th
Number of members in random user's community	10608	13206	6166	1590	1603

From the above table (*Table 5*) we can see an increasing trend regarding the number of user that reaches its peak at July 2nd (13206 community members) and then drop significantly to even 1590 users on July 4th and 1603 on July 5th. An explanation for that could be a trend topic or a post/tweet discussion, where users mention one the other to respond to them automatically.

The above explanation led us to further explore the data and analyze them. Therefore, we choose to provide a table with the common users in the random user's community (*tweetmeme*) between two consecutive days. This will show us how many users remain to the community - continue to interact and eventually form a community (a possible explanation could be the continuation of tweet responding, where a user mention all the other members of the discussion or who have responded). These results are formed to a table in order to enhance clarity.

Table 6: Number of common users in random user's community on two consecutive days.

	Same Users between July 1st and July 2nd	Same Users between July 2nd and July 3rd	Same Users between July 3rd and July 4th	Same Users between July 4th and July 5th
Number of same users	971	675	142	117

From the above presented table (*Table 6*) we can summarize that our explanation could be actually true, since when the numbers are peaked the common users are almost 1000 (971), while when the decrease took place, the common users are about 100 - 150 (142, and 117).

Finally, the common topic among the users communities, was extracted as it was described earlier, and we decided not to export just the most common one but the top10 most frequently used hashtags in the random user community. We need to notate again that we exclude the Null/NA entry, since otherwise it will be on top of each day.

Table 7: Most frequently topics of interest (hashtags) in random user's communities per day.

	July 1st	July 2nd	July 3rd	July_4th	July 5th
1	#forasarney	#moonfruit	#FF	#FF	#moonfruit
2	#iranelection	#iranelection	#followfriday	#FollowFriday	#thankyouaaron
3	#moonfruit	#DemiLovatoLive	#FollowFriday	#followfriday	#140mafia
4	#140mafia	#tcot	#ff	#iranelection	#iranelection
5	#fb	#fb	#urwashed	#poke	#WhiteParty
6	#quote	#140mafia	#iranelection	#Stalkerfriday	#FF
7	#tcot	#IranElection	#moonfruit	#140mafia	#IranElection
8	#1	#quote	#tcot	#spymaster	#crisishn
9	#blamedrewscancer	#FF	#140mafia	#KateMcRae	#fb
10	#spymaster	#forasarney	#quote	#moonfruit	#CanadaWantsMcFly

As we can notice, there are some common topics like the #FF or #ff, the #moonfruit, the #140mafia and the #iranelections among others which are remain to the community's top 10 all 5 days. Maybe this is also a point that enhance our above hypothesis that all users of the community are active responders on common discussions and follow same topics (such as the Iran Election), or aim to improve their followers with hashtags such as #fb - follow back, #ff - follow for follow, and #Follow Friday (on July 3rd and 4th, even that the 4th was Saturday maybe a discussion took place in the original tweet which then become viral or frequently retweeted).

Visualization of Community Graphs with Louvain and Fast Greedy Algorithms

In this section, we will create visualizations of the graphs using the **Louvain** and **Fast Greedy** algorithms for community detection. We will assign a different color to each community in the graph, allowing us to visually identify and analyze the communities. Additionally, we will apply filters to exclude nodes belonging to very small or very large communities to ensure a meaningful and aesthetically pleasing visualization.

During the visualization process, we will pay attention to the sizes of the communities. Very small communities might represent outliers or noise, while very large communities could indicate hubs or densely connected regions. By excluding these extremes, we aim to create a more meaningful and visually appealing visualization that highlights the significant communities within the network.

Louvain Algorithm Visualization:

Using the Louvain algorithm, we will detect communities within the graph and assign a unique color to each community. This visualization will help us gain insights into the structure and organization of the network, highlighting the distinct communities and their interconnections. By applying size filters, we can focus on communities of moderate size, enabling a clearer understanding of their relationships and topics of interest. Below are the 5 figures of twitter communities by the Louvain algorithm.

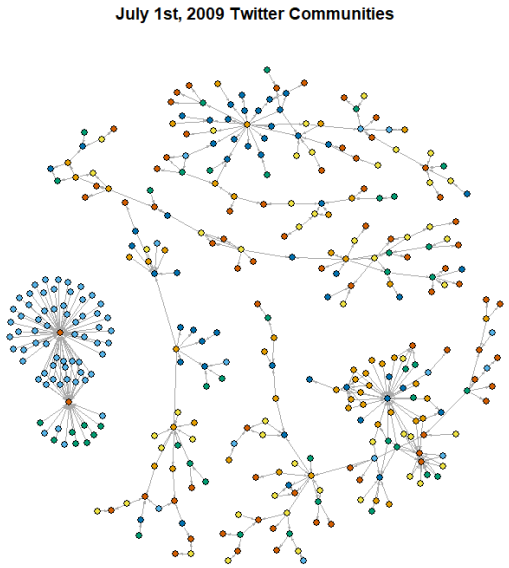


Figure 13: July 1st, 2009, Twitter Communities (Louvain Algorithm).

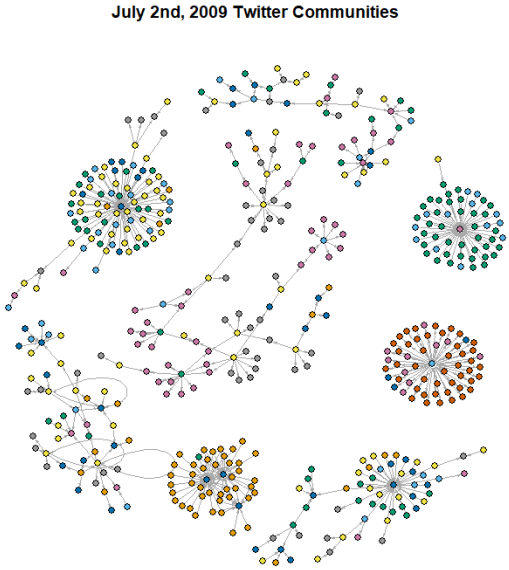


Figure 14: July 2nd, 2009, Twitter Communities (Louvain Algorithm).

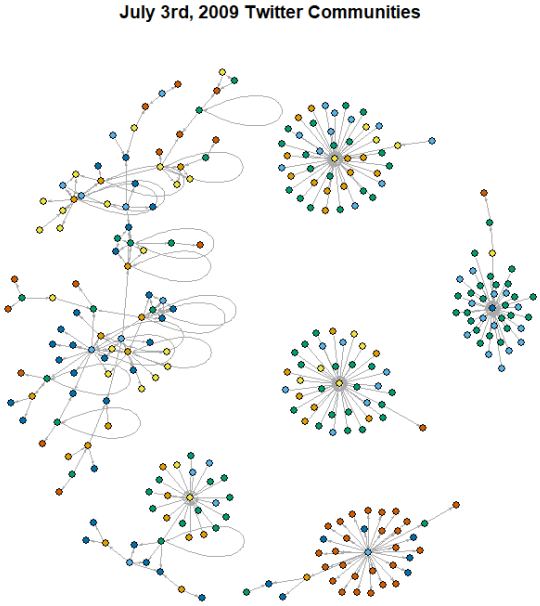


Figure 15: July 3rd, 2009, Twitter Communities (Louvain Algorithm).

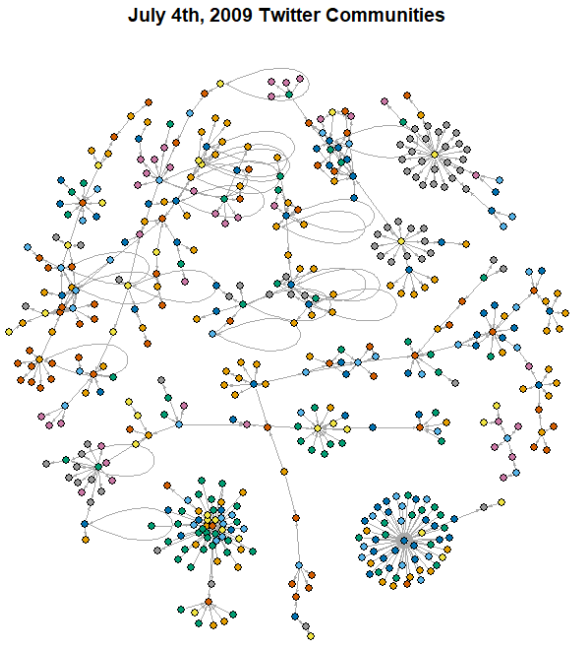


Figure 16: July 4th, 2009, Twitter Communities (Louvain Algorithm).

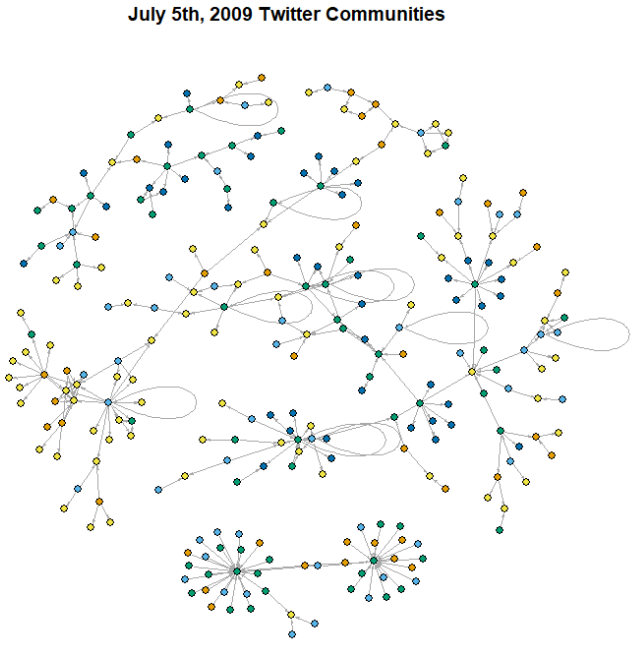


Figure 17: July 5th, 2009, Twitter Communities (Louvain Algorithm).

Fast Greedy Algorithm Visualization:

Similarly, we will employ the Fast Greedy algorithm to detect communities within the graph. By visualizing the graph with different colors for each community, we can observe the community structure and potential similarities or differences compared to the Louvain algorithm's results. Applying size filters will allow us to concentrate on communities of optimal sizes, enhancing the interpretability of the visualization. However, it was found more difficult to filter communities on this algorithm and as you can see the plots look like more populated while similar ranged filter where used.

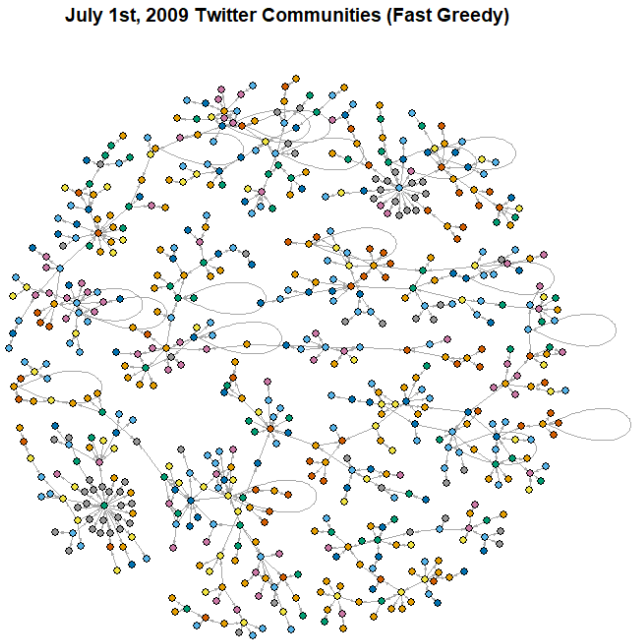


Figure 18: July 1st, 2009, Twitter Communities (Fast Greedy Algorithm).

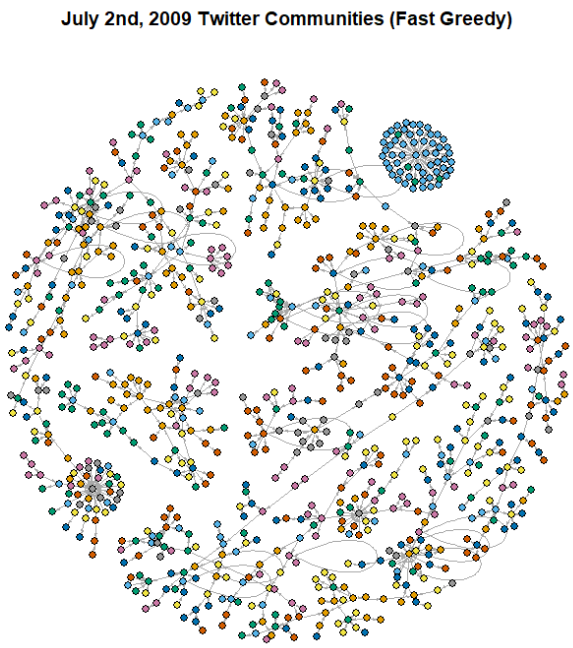


Figure 19: July 2nd, 2009, Twitter Communities (Fast Greedy Algorithm).

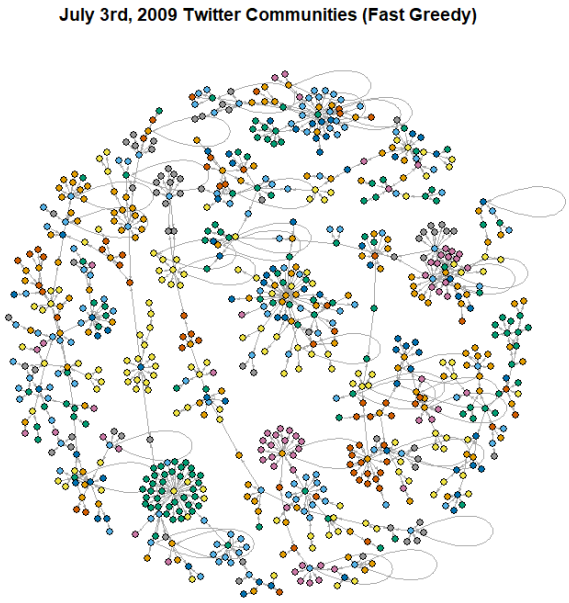


Figure 20: July 3rd, 2009, Twitter Communities (Fast Greedy Algorithm).

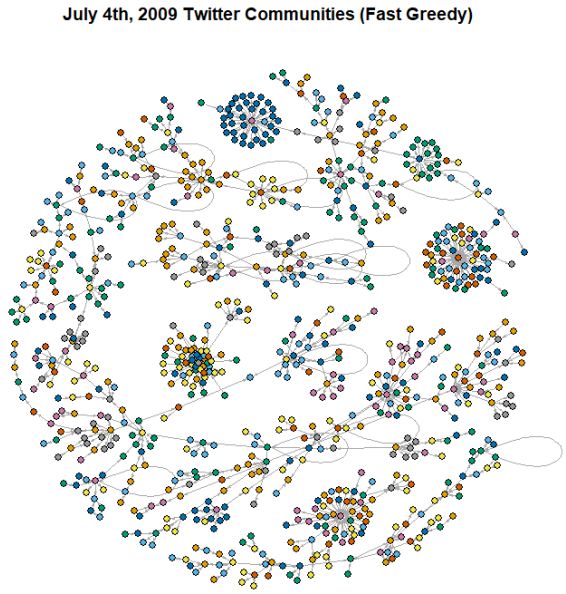


Figure 21: July 4th, 2009, Twitter Communities (Fast Greedy Algorithm).

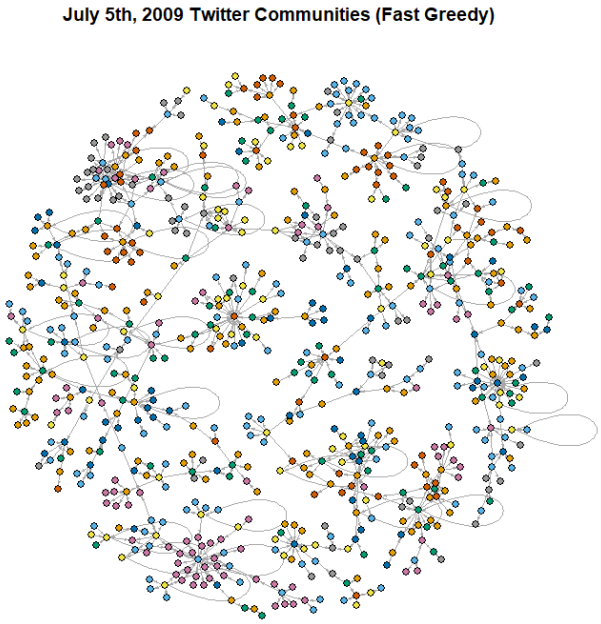


Figure 22: July 5th, 2009, Twitter Communities (Fast Greedy Algorithm).

Deliverables

The deliverables of this assignment will be a compressed file, which will contain the followings:

- The current documentation of the assignment, *p2.pdf*
- The Python file for the data handling part, *raw_data_handler.py*
- The R file responsible for all questions asked and for the datasets merging stage, f2822212.R
- The 10 CSV files datasets that are outputted from the Python script file and the 5 datasets CSV files that are merged through the R script file.
- The Created Diagrams Folder that contains all the created figures and tables for this assignment.

Appendix

The appendix section provides supplementary information and data tables that complement the main analysis, offering a more comprehensive understanding of the top-10 Twitter users based PageRank metric.

Table 8: Top-10 Twitter Users: PageRank with exact values.

	Users July 1st	PageRank July 1st	Users July 2nd	PageRank July 2nd	Users July 3rd	PageRank July 3rd	Users July 4th	PageRank July 4th	Users July 5th	PageRank July 5th
1	tweetmeme	0.00178897736116258	ddlovato	0.00281562428230351	tweetmeme	0.00246017606936565	souljaboytellem	0.00563840241456417	davidmmasters	0.00342083431689379
2	mashable	0.00125914382568667	drew_taubenfeld	0.0023948782094812	souljaboytellem	0.00230796393232547	addthis	0.00199692514117537	iamdiddy	0.00292462729577954
3	addthis	0.00118498319707757	mashable	0.00214902220310549	killerstartups	0.00209724624273514	tweetmeme	0.00167261631444787	addthis	0.00223227618824099
4	smashingmag	0.00118136948249198	tweetmeme	0.00213074494976699	addthis	0.00176417995595105	BreakingNews	0.00167229748212477	aplusk	0.00216659159388901
5	cnn	0.00071824730523192 2	globalmanners	0.00182969433043826	moontweet	0.00123601447023425	lilduval	0.00122356584764923	tweetmeme	0.0016896554556456
6	mileycyrus	0.00070960704357673 7	cnn	0.00152765833653732	cnnbrk	0.00117277530497575	mileycyrus	0.00119595028147658	mashable	0.00106954134269585
7	KISSmetrics	0.00067836046643765 8	addthis	0.00136088678720923	mashable	0.00111720546919271	mashable	0.0011092977370798	mrskutcher	0.00091991402544966
8	CourageCampaign	0.00062608323806603 1	souljaboytellem	0.00121280671254837	BreakingNews	0.00101809317168428	iamdiddy	0.00108811971459486	moontweet	0.00085315928905748 5
9	aplusk	0.00053974170173406 7	cnnbrk	0.0011659016848462	PhillyD	0.00071818713729593	cnnbrk	0.00103068204612684	BreakingNews	0.00073599685572702 1
10	rafinhabastos	0.00051958461993947 7	mileycyrus	0.00075756369271223 9	adamlambert	0.00061711141005347 9	garyvee	0.00090879173356977 2	mileycyrus	0.00072795740727723