**FINAL PROJECT REPORT**

**SOCIAL NETWORK ANALYSIS OF FACEBOOK PAGES**

*submitted by*

**Anamitra Musib (16BCE0664)**

*In partial fulfilment for the award of the degree of***BACHELOR OF TECHNOLOGY**

in

# COMPUTER SCIENCE AND ENGINEERING

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

November, 2018

**ACKNOWLEDGEMENT**

I would like to deeply thank my Social and Information Networks (CSE3021) professor Mrs. Vijayasherly V for being an amazing and patient teacher. It’s hard to find professors who teach in such a way so as to maintain, if not boost, one’s interest in a subject, and I’m lucky to have studied this course under her, which resulted in my undertaking this project and studying the material and the topic of social network analysis in general, with renewed vigour.   
Thank you, maam.

Anamitra

**Abstract**

Social media is increasingly becoming a popular haven for network analysis projects as the huge amount of real-time data generated every day, proves to be a rich resource for those interested in finding more about the particularities of how networks in social media sites like Facebook and Twitter function, what are some useful insights we can generate from such data, and how this data can be used by corporate & conglomerates alike to improve their business models and enhance cash inflow through revenue.

By using social media analysis techniques on a real life social media network, we can gain an excellent understanding of how a social network functions, and better study the interactions between actors and how they are affected by actions of one another. Learning from randomly generated social networks, as are used in many textbooks and using a real life project to acquire knowledge of the workings of a social network are very different, and the latter will prove helpful to the individual(s) who have worked on the project and also to the reader, as they will gain understanding of how a theoretical social network works, using a practical example.

Social networks like Facebook, Twitter, and Google+ are most visited domains on the Internet. They contain huge data about the users and the relationships among them. To analyze and mine useful information from these huge social network data, special graph based mining tools are required that can easily model the structure of the social networks. A number of such analysis tools are available with their own features and benefits. Choosing an appropriate tool for a particular task is difficult to decide, but as I’m comfortable with the R programming language, I decided to use it to do social network analysis on Facebook pages.

**1. Introduction & Problem Statement**

In this project, I am going to use a graph dataset containing various data about 582 Facebook pages of organizations directly / indirectly related to football. This information includes   
 **a)** fan\_count (no. of likes of the page has received)  
 **b)** category (such as ‘Non-Profit Organization’, ‘Athlete’, ‘Company’, etc.  
 **c)** username (assigned to distinguish every page)  
 **d)** users\_can\_post (can any Facebook user post to that page)  
 **e)** link (URL for the corresponding Facebook page)  
 **f)** post\_activity  
 **g)** talking\_about\_count (how many people have been talking about this page recently)

From the data aforementioned, I shall be generating different statistics and graph statistical measures using the network of these pages as a graph structure. These statistics include -   
 **1)** aggregating pages based on their category  
 **2)** top pages based on their fan count (likes)  
 **3)** top pages based on total people talking about them  
 **4)** top pages based on page posting activity  
 **5)** correlation between fans & talking about for pages  
 **6)** graph properties such as diameter, edge density, transitivity, coreness, centrality measures (Betweenness, Closeness, EigenVector)  
 **7)** Pagerank  
 **8)** Kleinberg’s HITS score  
 **9)** finding communities / clusters in the network  
**10)** modularity score, etc.

During the course of generating these graph statistics, I shall also be producing visualizations which will greatly aid the reader in understanding the subtle nuances of the things happening behind the scenes.

At the end of this project, I will have analyzed 582 pages for different measure and will have gained a greater understanding of graph theory and social media analysis than I do now.

**2 Literature Survey**

**2.1 Survey of existing models / works**

I did the literature survey on some research papers I found on the internet, all related to the topic of social network analysis (SNA). I read papers from websites like ScienceDirect and IEEEXplore. The main focus of the selected articles were SNA and its applications.  
Below given are some of the keywords I used to filter research papers -   
 a) Social Network Analysis  
 b) Social Network Analysis, Techniques  
 c) Social Network Analysis, Knowledge Network  
 d) Social Network Analysis, Supply Chain

I reviewed the works by identifying their contexts and objectives, research methodology, modeling approaches, pros and cons, etc.

***SNA in supply chains***

Borgatti and Li (2009) have done some groundbreaking work in their overview of social network concepts in order to be applied on a supply chain context.

Basole and Bellamy (2013) identified a growing recognition of the noteworthy benefits a network analytic lens provides to understand, design and control supply chain systems.

***SNA techniques***

Romijn and Caniels (2008) use SNA to support Strategic Niche Management (SNM) – an analytical technique based on learning and convergence expectations and networking, to initiate new sustainable technologies through societal experiments. The authors reviewed SNA contributions and delineate a case study on the emerging biofuels sector in Tanzania.

Liebowitz (2005) proposed an integrative approach between SNA for knowledge mapping in organizations & the analytic hierarchy process.

***SNA applications in health organizations***

Zhang et al. (2012) combine SNA techniques and co-word analysis to analyze research literature on patient adherence in the health sector and to demonstrate their knowledge structure and evolution over time.

In the public health context, Malin et al. (2011) use open-source software to bolster and reproduce their investigation on the study of electronic health record access logs for deducing a social network of the users and learning relational policies from these access logs, which would be a proposal to automatically collect data and study the social network.

Westbrook and Dunn (2011) put forward a validation technique for standardized comparison of small networks in healthcare organizations.

In the works of Martinez-Lopez et al. (2009), the basic concepts needed to understand SNA and graph theory are collected; hence it’s a good source for one looking to learn the principles behind SNA. The authors also made a review of recent applications of SNA in preventive veterinary medicine.

***Collaborative platforms and social network sites***

Candi and Roberts (2014) refine an empirical research through a web-based questionnaire to 351 European companies in order to investigate the application of social network sites in the context of business performance in terms of growth, profitability and innovativeness & new product development.

Haenlein and Kaplan (2010) and Kietzmann et al. (2011) define social network sites as virtual platforms on which people can synchronously / asynchronously create, share, modify or react to diverse forms of electronic media and content.

Gṻrkan et al. (2010) developed Deliberatium, a large scale online argumentation platform based on collaborative technologies (Common Lisp & IBIS applications).

**2.2 Summary / Gaps / Limitations / Future Work identified in the survey**

The objectives and application context of several papers related to SNA topics, along with their authors and year of publication have been shown in a tabular format as follows –

|  |  |  |  |
| --- | --- | --- | --- |
| **Year** | **Author(s)** | **Objectives** | **Application context** |
| 2003 | Borgatti & Foster | Literature review and analysis of organizational network research | General organizations |
| 2005 | Liebowitz | Knowledge mapping in organizations | General organizations |
| 2007 | Giulani | Exploring the structural properties of knowledge networks in 3 wine clusters | Wine industry |
| 2008 | Caniels and Romijn | Application of SNM (Strategic Niche Management) and SNA approach to a case study about the emerging biofuels sector in Tanzania | Biofuels |
| 2008 | Giulani and Bell | To study the evolution of a cluster knowledge network | Wine industry |
| 2009 | Borgatti and Li | Literature review of SNA and extension to supply chains | Supply chains |
| 2009 | Martinez-Lopez et al. | To review and apply SNA & graph theory for preventive veterinary medicine | Veterinary medicine |
| 2010 | Gṻrkan et al. | To develop and validate a large scale online argumentation platform | General organizations |
| 2012 | Zhang et al. | To analyze research literature on patient adherence and show their knowledge structure over time | Health |
| 2013 | Natter et al. | To design a self-scaling registry technology for collaborative data sharing on disease registries | Health |
| 2014 | Candi and Roberts | Survey of the use of social network sites in new product development | New product development |

**Summary**

The applications of SNA are as numerous as are its benefits.

Liebowitz (2005) points out the usefulness of SNA combined with the analytic hierarchic process for knowledge mapping in organizations.

Caniels and Romijn (2008) show that SNA also allows a good insight into the morphology of the network and its importance for innovation.

The works by Giulani and Bell (2005, 2008) & Giulani (2007) provide a firm ground to a methodological framework for the data collection and application of SNA techniques for analyzing knowledge networks by focusing on structural properties.

Borgatti and Li (2009) related SNA concepts to the supply chain context – selecting nodes and ties, structural holes, hubs and authorities, node centrality, whole network properties and bipartite graphs.

Zhang et al. (2012) show the use of co-word analysis and SNA techniques which can diminish the reliance on subjective judgement when doing literature reviews.

Gṻrkan et al. (2010) have provided an empirical application of an online large scale argumentation tool that supports the construction of shared knowledge maps.

Malin et al. (2011) infer the social network through the analysis of electronic health record access logs.

**Limitations and critical points**

**A) Large size social networks**

Liebowitz (2005) demonstrated that the combination of SNA and analytic hierarchic process for knowledge mapping would have limitations in large social network maps due to the analytic hierarchy process becoming tedious.

The computational power of available software can be a limitation for SNA applications (Martinez et al. 2009).

**B) Data reliability and results**

Martinez et al (2009) concluded that information could be available for some nodes and there could be a potential risk for bias in the conclusions of the study associated to the procedure used to select the nodes.

Malin et al. (2011) underline the limitations related to the reality of data re-use and even to the time frame in which such data are collected, the authors also suggested improving their proposal through the incorporation of certain semantics and by restructuring the access transactions.

Morrison (2008) noted that results from SNA case studies on cluster knowledge networks can’t be generalized to other clusters.

**C) Metrics appropriateness**

Romijn and Caniels (2008) highlight the necessity of consolidating insights from the 2 main SNA perspectives – connectionism and structuralism. The former per se provides few network indicators for a systematic analysis of a network of values tied, while the latter allows a more elaborate empirical analysis of the structural network properties but give theoretically few relevant insights on the interactions between the network processes.

**D) Supply Chain**

Although Borgatti and Li (2009) defined some criteria to define nodes and ties & also to collect supply network data, till data, there is not a validated tool or methodology for doing it.

**E) Platform Developments**

In the milieu of collaborative platforms, the results of using Deliberatium suggest that – significant design efforts are needed to lower moderation costs in use of collaborative platforms; it’s necessary for improvement of the knowledge representation & a disproportionate use of extrinsic incentives were given (Gṻrkan et al. 2010).

**Future Work**

There’s a much called for requirement to explore SNA applications to more industrial sectors, when health is the most studied of all sectors currently.

Many research papers don’t indicate the data collection methods, nor do they use the process of automatic data collection.

The use of an online platform, such as website or web application, to generate some of the data versus the traditional approaches adopted would be profitable in order to provide more reliability to SNA results.

A methodology for automatic data collection and choosing metrics would be highly appreciated.

Rabellotti and Morrison (2009) showed that the dynamics of networks is only little explored, thus an effort in collecting longitudinal data on inter-firm collaboration within clusters is much needed.

Finally, the scrutiny of knowledge properties may influence the process of inter-firm transfer of managerial, technological and market knowledge (Biaggiero and Sammarra 2008).

**3 Overview of the Proposed System**

**3.1 Introduction**

In the proposed system, I have worked on a dataset containing data on 582 organizations / institutions which are directly / indirectly related to football clubs. Each of the 582 organizations have 6 columns of data to them, namely id, name, category, fans, talking\_about and post\_activity, but the following data was inferred from the summary of the original dataset -   
 a) fan\_count (no. of likes of the page has received)  
 b) category (such as ‘Non-Profit Organization’, ‘Athlete’, ‘Company’, etc.  
 c) username (assigned to distinguish every page)  
 d) users\_can\_post (can any Facebook user post to that page)  
 e) link (URL for the corresponding Facebook page)  
 f) post\_activity  
 g) talking\_about\_count (how many people have been talking about this page recently)

From the data aforementioned, I shall be generating different statistics and graph statistical measures using the network of these pages as a graph structure. These statistics include -   
 1) aggregating pages based on their category  
 2) top pages based on their fan count (likes)  
 3) top pages based on total people talking about them  
 4) top pages based on page posting activity  
 5) correlation between fans & talking about for pages  
 6) graph properties such as diameter, edge density, transitivity, coreness, centrality measures (Betweenness, Closeness, EigenVector)  
 7) Pagerank  
 8) Kleinberg’s HITS score  
 9) finding communities / clusters in the network  
10) modularity score, etc.

During the course of generating these graph statistics, I shall also be producing visualizations which will greatly aid the reader in understanding the subtle nuances of the things happening behind the scenes.

I have used the R programming language for conducting the analysis, specifically using the R Notebook (.rmd) format, where we can write code in different chunks and execute each chunk separately, and also create and display beautiful visualizations and graphs inline (along with the code), which enhances aesthetic appeal.

**3.2 Framework, Architecture or Module of the Proposed System**

The architecture of the proposed system consists of code chunks, each containing some code which can run independently of other chunks. I have outlined the modules and given their brief description below –

1. Importing the required libraries – gridExtra, igraph and ggplot2.
2. Reading in the graph and viewing its brief summary
3. Inspecting the page graph object
4. Aggregating pages based on their category
5. Top pages based on their fan count (likes)
6. Top pages based on total people talking about them
7. Top pages based on page posting activity
8. Checking correlation between fans and talking\_about for pages :
9. Correlation is a statistical technique that can show whether and how strongly pairs of variables are related. For example, height and weight are related; taller people tend to be heavier than shorter people. The relationship isn't perfect. People of the same height vary in weight, and you can easily think of two people you know where the shorter one is heavier than the taller one. Nonetheless, the average weight of people 5'5'' is less than the average weight of people 5'6'', and their average weight is less than that of people 5'7'', etc. Correlation can tell you just how much of the variation in peoples' weights is related to their heights.
10. Although this correlation is fairly obvious your data may contain unsuspected correlations. You may also suspect there are correlations, but don't know which are the strongest. An intelligent correlation analysis can lead to a greater understanding of your data.
11. Plotting page network using degree filter
12. Getting filtered graph
13. The filtered\_graph class template is an adaptor that creates a filtered view of a graph. The predicate function objects determine which edges and vertices of the original graph will show up in the filtered graph. If the edge predicate returns true for an edge then it shows up in the filtered graph, and if the predicate returns false then the edge does not appear in the filtered graph. Likewise for vertices.
14. The filtered\_graph class does not create a copy of the original graph, but uses a reference to the original graph. The lifetime of the original graph must extend past any use of the filtered graph. The filtered graph does not change the structure of the original graph, though vertex and edge properties of the original graph can be changed through property maps of the filtered graph. Vertex and edge descriptors of the filtered graph are the same as, and interchangeable with, the vertex and edge descriptors of the original graph.
15. Plotting the graph
16. Diameter (length of longest path) of the network
17. Getting the longest path of the network
18. Mean distance between 2 nodes in the network
19. Distance between various important pages (nodes)
20. no. of edges / no. of all possible edges
21. transitivity clustering coefficient
22. In graph theory, a clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. Evidence suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit groups characterized by a relatively high density of ties; this likelihood tends to be greater than the average probability of a tie randomly established between two nodes.
23. The k-core of a graph is a maximal subgraph in which each vertex has at least degree k.
24. The coreness of a vertex is k if it belongs to the k-core but not to the (k+1)-core.
25. Max coreness
26. Viewing the core of the network
27. Viewing the periphery of the network
28. We study a model of network formation where the benefits from connections exhibit decreasing returns and decay with network distance. We show that the unique equilibrium network is a periphery-sponsored star, where one player, the center, maintains no links and earns a high payoff, while all other players maintain a single link to the center and earn lower payoffs.
29. Both the star architecture and payoff inequality are preserved in an extension of the model where agents can make transfers and bargain over the formation of links, under the condition that the surplus of connections increases in the size of agents’ neighborhoods. Our model thus generates two common features of social and economic networks: (1) a core-periphery structure; (2) positive correlation between network centrality and payoffs.
30. Degree centrality

Degree centrality assigns an importance score based purely on the number of links held by each node.

1. Closeness centrality

In a connected graph, the normalized closeness centrality (or closeness) of a node is the average length of the shortest path between the node and all other nodes in the graph. Thus the more central a node is, the closer it is to all other nodes.

1. Betweenness centrality

One of those hidden centralities is the betweenness centrality, where we take a look at how often a node lies on the shortest path between two other nodes. The more a node appears on one of those shortest paths, the higher its betweenness centrality. A node with high betweenness is also called a broker as it fulfills a brokerage position in the network, which means that information needs to pass through that entity to be shared by the other nodes. This also means that these nodes are often the vulnerable points of a network: by cutting them out, chances are the network will fall apart into unconnected components.

1. Viewing top pages based on above measures
2. Correlation plots

a) Graph 1 - degree vs. closeness

b) Graph 2- degree vs. betweenness

1. Eigenvector Centrality

Like degree centrality, EigenCentrality measures a node’s influence based on the number of links it has to other nodes within the network. EigenCentrality then goes a step further by also taking into account how well connected a node is, and how many links their connections have, and so on through the network.

1. PageRank

PageRank is a variant of EigenCentrality, also assigning nodes a score based on their connections, and their connections’ connections. The difference is that PageRank also takes link direction and weight into account – so links can only pass influence in one direction, and pass different amounts of influence.

1. Kleinberg's HITS score
2. Example of finding neighbours of page vertices
3. Get communities / clusters
4. filtering graph to get important nodes based on degree
5. Fast greedy clustering
6. The graph package implements a variety of network clustering methods, most of which are based on Newman-Girvan modularity. To see them all, refer to the communities documentation.
7. The simplest such algorithm is the “fast greedy” method, which starts with nodes in separate clusters, and then merges clusters together in a greedy fashion.
8. get total pages in each cluster
9. get page names in each cluster
10. get modularity score
11. Modularity is one measure of the structure of networks or graphs. It was designed to measure the strength of division of a network into modules (also called groups, clusters or communities). Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules.
12. Modularity is often used in optimization methods for detecting community structure in networks. However, it has been shown that modularity suffers a resolution limit and, therefore, it is unable to detect small communities. Biological networks, including animal brains, exhibit a high degree of modularity.
13. edge betweenness clustering

**3.3 Proposed System Model (Mathematical Modeling)**

A mathematical model is an abstract model that uses mathematical language to describe the behavior of a system.

There are several modules in my system which are direct programming implementations of mathematical concepts, listed as follows –

**A)** Aggregating pages based on their category –   
this module uses sorting the pages based on their category, and displaying the top 10 pages

**B)** Top pages based on their fan count (likes), total people talking about them, and page posting activity -   
this module sorts the pages based on the respective attribute and displays the top 10 pages

**C)** Correlation between fans and talking\_about for pages-  
this chunk uses the concept of correlation, which is basically how closely 2 things are related. The R-squared measure for correlation b/w fans and talking\_about attributes is 0.659 which denotes that they are positively correlated.

**D)** Diameter (length of longest path) of the network -   
this module simply finds the diameter (longest path) of the network

**E)** Mean distance between 2 nodes in the network -   
this module finds the mean distance b/w 2 nodes in the network

**F)** Edge density of the graph -   
this module uses the edge\_density function in igraph, which finds the no. of edges divided by no. of all possible edges in the graph

**G)** Transitivity clustering coefficient -   
this chunk finds the clustering coefficient of the graph

**H)** k-core of the graph

**I)** Degree centrality, Closeness centrality, Betweenness centrality

**J)** Correlation plots – degree vs. closeness & degree vs. betweenness

**K)** Eigenvector Centrality

**L)** Pagerank

**M)** Kleinberg’s HITS Score

**N)** Finding communities / clusters in the graph

**O)** Fast greedy clustering

**P)** Getting page names in each cluster

**Q)** Getting the modularity score

**R)** Edge betweenness clustering

**4. Proposed System Analysis and Design**

As mentioned earlier, the main objective of this system is to compute and find different attributes and features related to a graph, by which one can get a deeper understanding of how a social network functions in general.

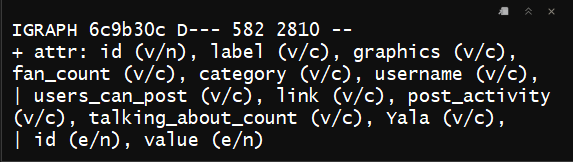
This system is designed in the form of independently executable code chunks, the visualizations produced as a result are also mostly displayed inline (along with the code).

A description of what each chunk contains exactly can be found in the section *3.2 Framework, Architecture or Module of the Proposed System*.

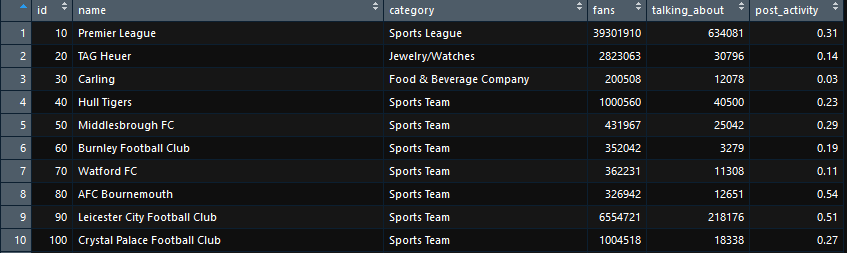
The system contains all major attributes and features of a social network graph including computations of PageRank, Kleinberg’s HITS Score, Modularity score, Edge betweenness clustering, Fast greedy clustering, correlation plots, degree / betweenness / closeness centralities, etc. Thus, this system will serve as a good source for someone who wishes to gain a deep understanding of these concepts via a practical approach.

**5. Implementation**

This section will contain pictures of the respective outputs for the different graph features which have been calculated in the proposed system.  
  
**I) Reading in the graph and viewing its brief summary**



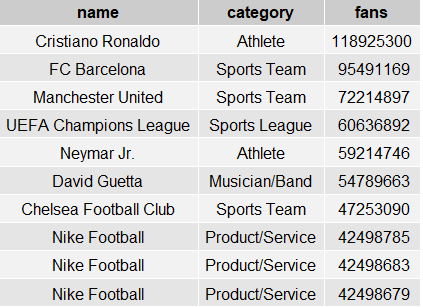
**II) Inspecting the page graph object**

[Showing 10 pages out of 582 total pages]

**III) Aggregating pages based on their category**



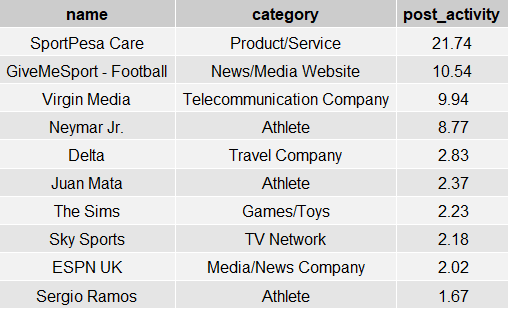
**IV) Top pages based on their fan count (likes)**



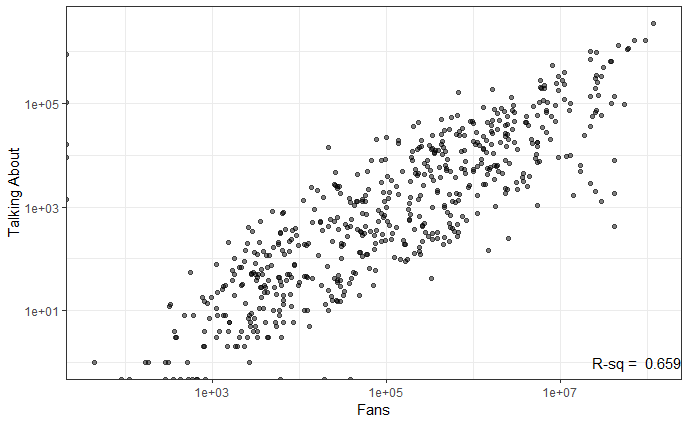
**V) Top pages based on total people talking about them**



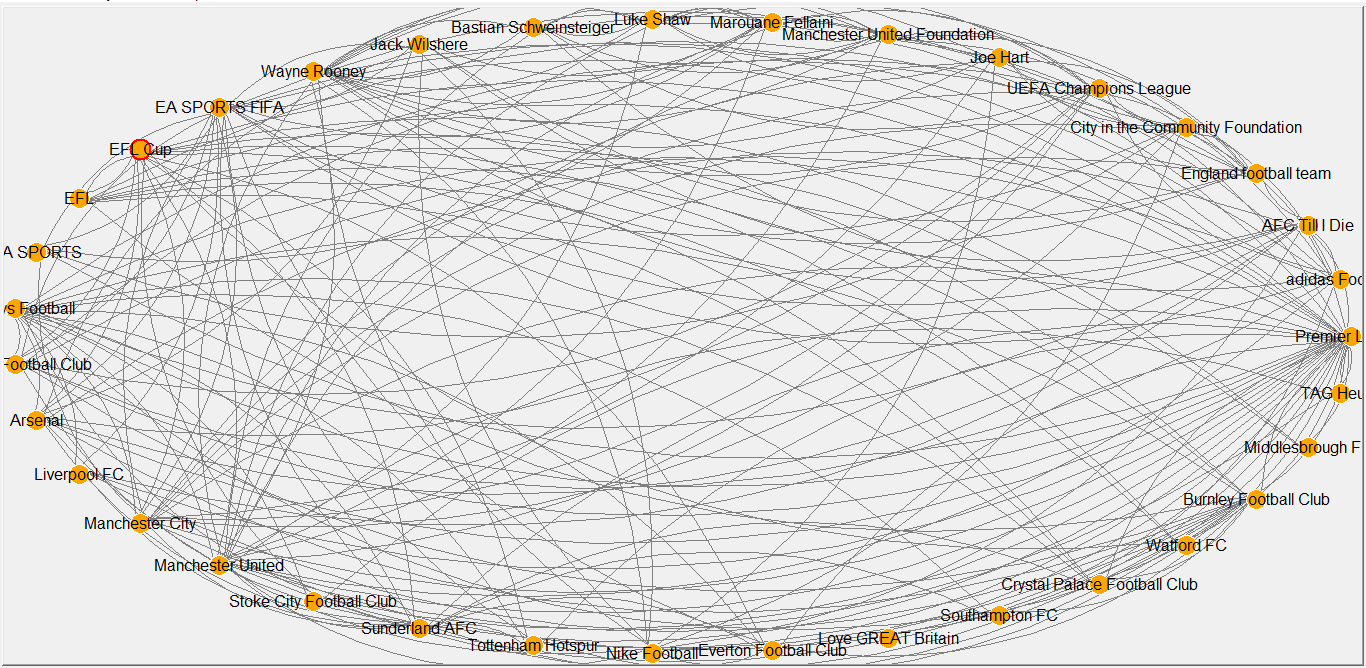
**VI) Top pages based on page posting activity**



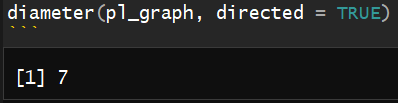
**VII) Checking correlation b/w fans and talking\_about for pages**



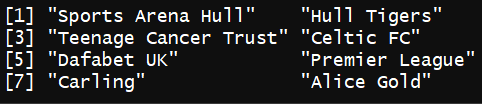
**VIII) Plotting page network using degree filter**



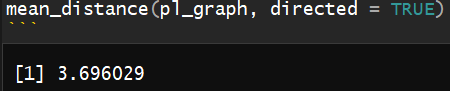
**IX) Diameter (length of the longest path) of the network**



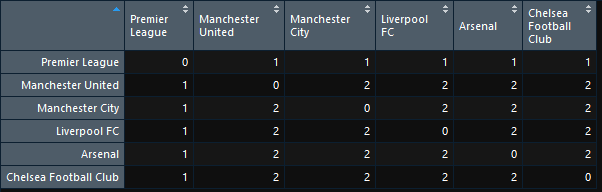
**X) Getting the longest path of the network**



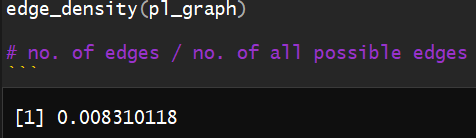
**XI) Mean distance b/w 2 nodes in the network**



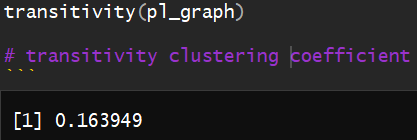
**XII) Distance b/w various important pages (nodes) [an example]**



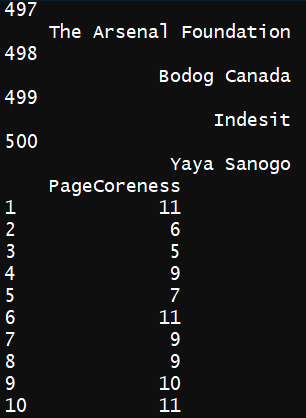
**XIII) Edge density of the graph**



**XIV) Transitivity clustering coefficient**

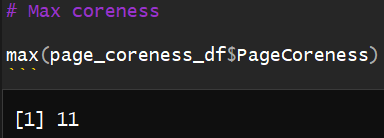


**XV) Page coreness (k-core, coreness)**



[A random output snippet]

**XVI) Max coreness**



**XVII) Viewing the core of the network**



**XVIII) Viewing the periphery of the network**



**XIX) Degree centrality**



[Showing top 10 entries]

**XX) Closeness centrality**



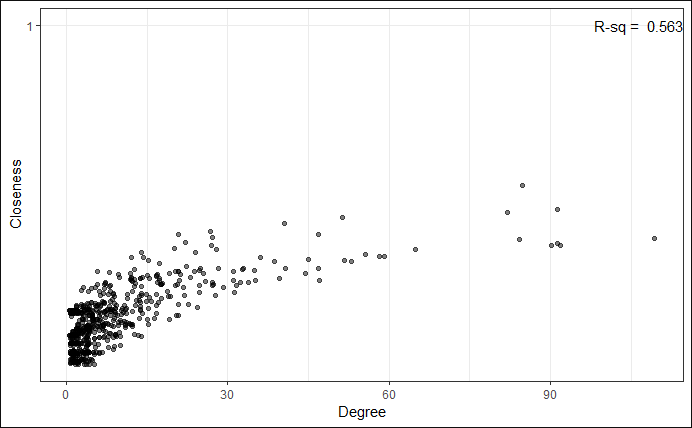
[Showing top 10 entries]

**XXI) Betweenness centrality**

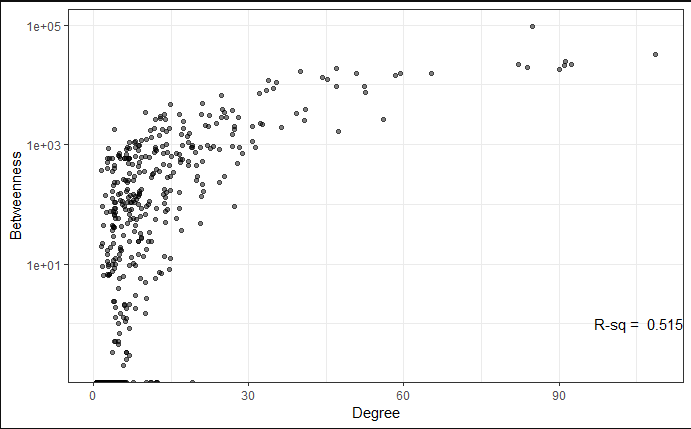


[Showing top 10 entries]

**XXII) Correlation plot – degree centrality vs. closeness centrality**



**XXIII) Correlation plot – degree centrality vs. betweenness centrality**



**XXIV) Eigenvector centrality**



[Showing top 10 pages]

**XXV) PageRank**



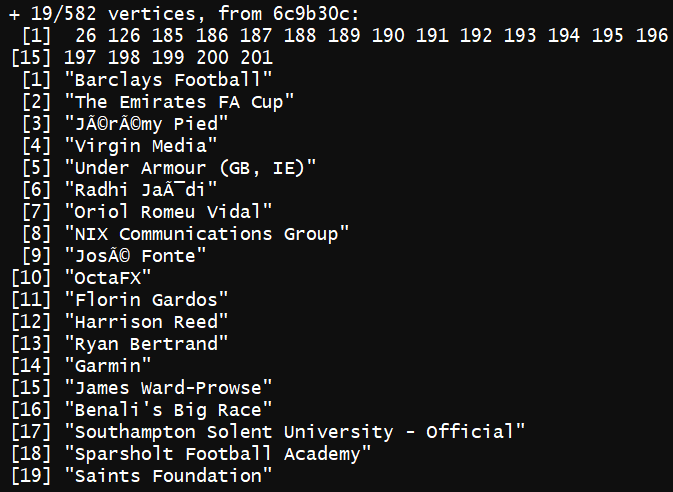
[Showing top 10 pages]

**XXVI) Kleinberg’s HITS Score**

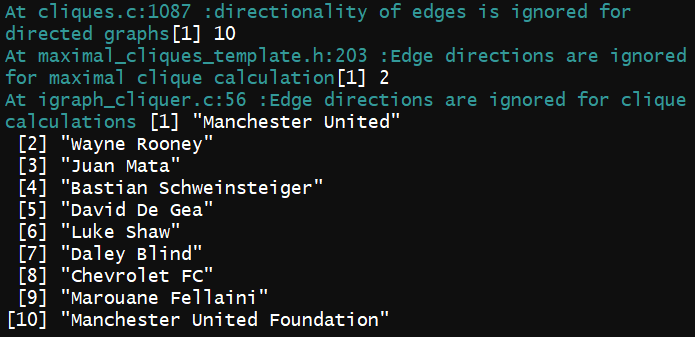


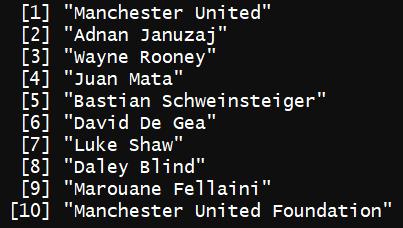
[Showing top 10 pages]

**XXVII) Example of finding neighbors of page vertices**

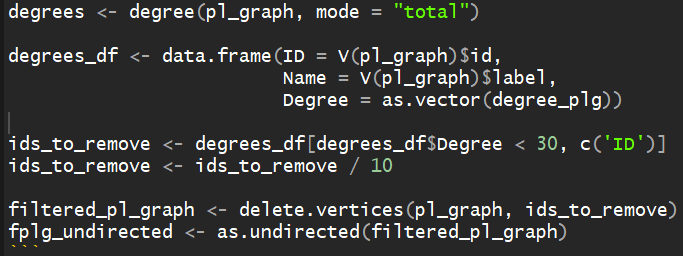


**XXVIII) Getting communities / clusters**

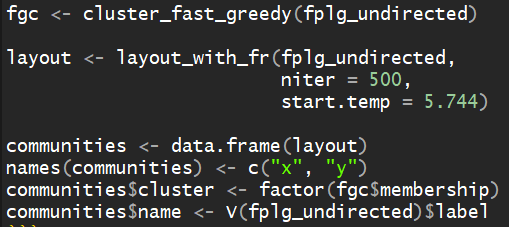




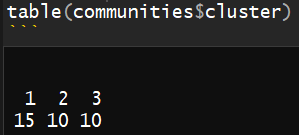
**XXIX) Filtering graph to get important nodes based on degree**



**XXX) Fast greedy clustering**



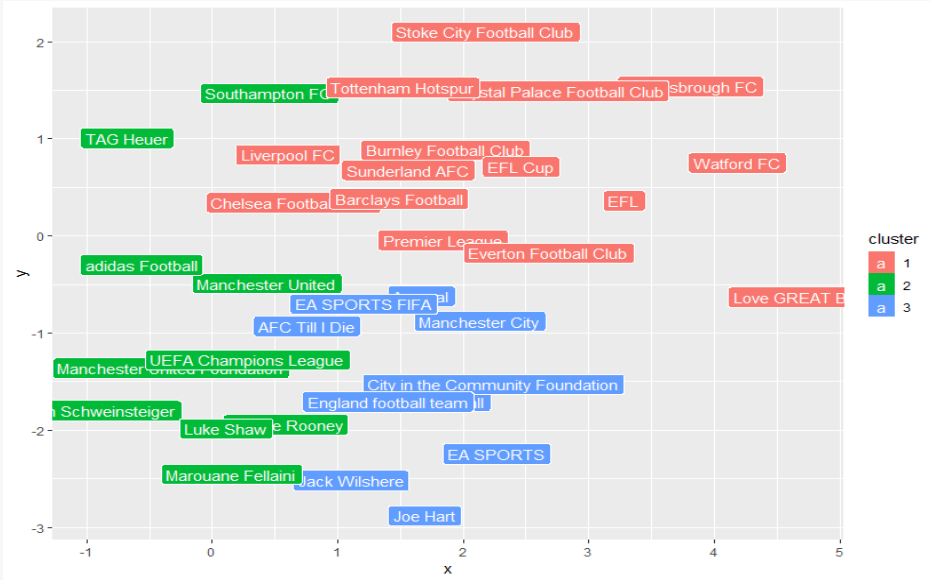
**XXXI) Fast greedy clustering 🡪 total pages in each cluster**

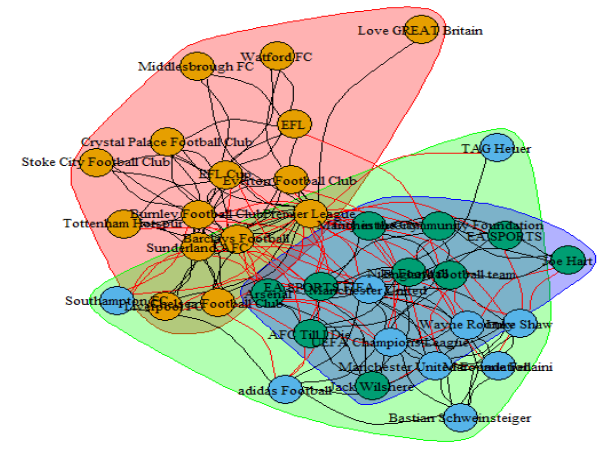


**XXXII) Get page names in each cluster**



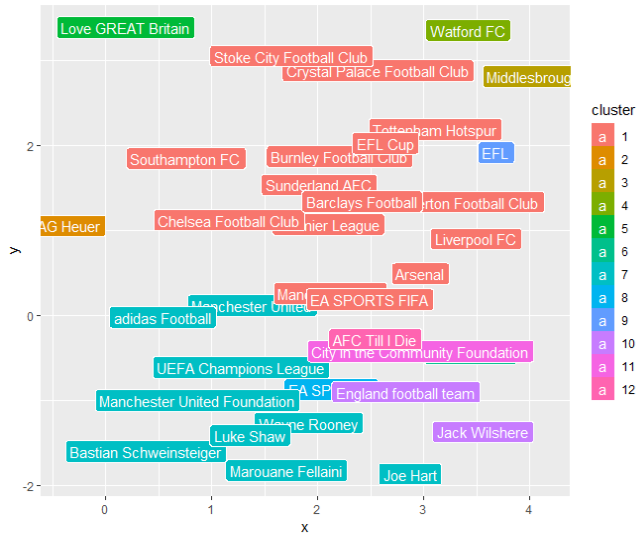
**XXXIII) Modularity score**

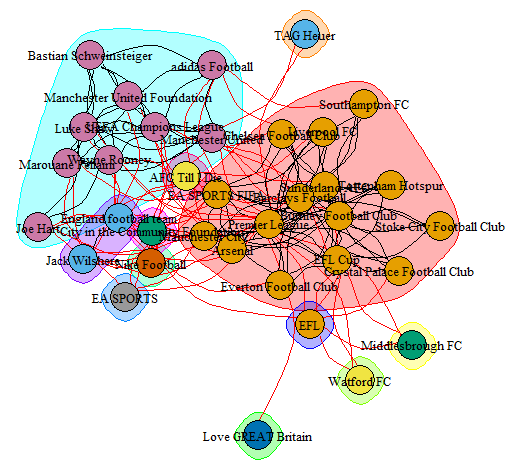




**XXXIV) Edge betweenness clustering**







**6. Results and Discussions**

**I)** By reading in the graph and viewing its summary, we see that the graph contains 582 rows (pages) and 6 columns – id, name, category, fans, talking\_about, post\_activity

**II)** By aggregating pages based on their category, we see that athletes account for the highest no. of pages among all of them, being 151 in number, followed second by ‘Sports Team’ having 77 pages, and so on.

**III)** When we look at the top pages based on their fan count (likes), we find that Cristiano Ronaldo from the Athlete category has the highest no. of fans (~119 million), followed by FC Barcelona (95.5 million) and Manchester United (72 million) from the Sports Team category.

**IV)** When we look at the top pages based on people talking about them, we find that Cristiano Ronaldo from the Athlete category is on the top with 3.5 million people talking about him, followed by FC Barcelona (1.63 million) and Manchester United (1.62 million) from the Sports Team category.

**V)** When we look at the top pages based on page posting activity, we see that SportsPesa Care from the Product / Service category is on the top with 21.74 posting activity, followed by GiveMeSport – Football (10.54) from the News / Media website category and Virgin Media (9.94) from the Telecommunication Company category.

**VI)** The correlation plot b/w talking\_about and fans shows a positive correlation with the R-squared measure being 0.659 indicating that the 2 features are closely related.

**VII)** The diameter of the network is 7, i.e. the maximum no. of nodes which exist b/w any 2 nodes in the network is 7.

**VIII)** The maximum coreness of this network is 11 i.e. the highest value of k for k-core is 11.

**IX)** Pages like ‘Premier League’, ‘Burnley Football Club’ and ‘Crystal Palace Football Club’ have 11 as their coreness and thus form the core of the network. Pages like these are important to the network.

**X)** Pages like ‘Henrik Lundqvist’, ‘Cara Delevingne’ and ‘Patrick Dempsey’ have the least coreness of 1 and thus form the periphery of the network. Pages like these are not important to the network and they generally exist as a outlier in the network diagram.

**XI)** Manchester City has the highest degree centrality of 109, thus it has the most no. of connections in the network (it’s connected to max. no. of other pages)

**XII)** Premier League has the highest closeness centrality of 0.534, thus it tends to have the least average distance from all other nodes.

**XIII)** Premier League also has the highest betweenness centrality of 96082, thus it has the highest tendency to come in the path connecting most other nodes.

**XIV)** The correlation between degree centrality and closeness centrality & degree centrality and betweenness centrality is positive, and the R-squared measures of these relations are 0.563 and 0.515 respectively. The latter relation, having a lower R-squared measure proves to be a better correlation i.e. degree centrality and betweenness centrality are better correlated than the former relation.

**XV)** Premier League has the highest Eigenvector centrality (1.0), Nike Football has the highest Pagerank (0.0272) and Manchester United has the highest HITS Authority score (1.0).  
All the 3 pages above are thus highly important to the graph, as they seem to have significant connections. Manchester United which has the highest HITS authority score, thus seems to be a page which has in-links from many other pages, making it an important authority figure in the network.

**XVI)** The maximal clique calculation process yielded 2 cliques having 10 pages each.

**XVII)** The fast greedy clustering process yielded 3 clusters, with the 1st cluster having 15 pages, and the rest of the 2 having 10 pages each.

**XVIII)** The edge betweenness clustering process yielded 12 communities.

**7. Conclusion, Limitations and Scope for future work**

During the course of this project, I have learnt quite a few new things about social networks and social network analysis in general. I have added to my skillset, the skills of exploring and cleaning a dataset for analysis, choosing which visualizations would be apt when and where, how to select from a vast variety of graph features and attributes and when to apply them, and how to analyze the outputs and outcomes from the computations performed.

The analyses performed in this project are possible due to a dataset freely accessible online. Though I have worked with real-life data, we know that social network data is highly variable and doesn’t stay static, even for a second. I won’t say that the results obtained from the project would hold invalid a few months or years down the line, but as data constantly changes, so do the results, so it’ll be wise not to take the results of this project hard-and-fast.

I have covered important topics related to social network analysis, such as clustering, community analysis, centrality measures, graph features such as diameter, mean distance, coreness, etc. to the best of my knowledge. I have used the R programming language as I feel very comfortable in using it, and because R is the preferred standard for data analysis operations. But R has some limitations in that, we have to import a lot of different packages to do different operations, there hardly exist standalone packages in R which can do multiple tasks related to a topic, e.g. there exists no single package in R which has all the functions and algorithms related to social network analysis – there exist many different packages covering different topics such as the ‘igraph’ package for the plotting and drawing of tables and graphs, or the ‘ggplot2’ package for creating easy and fluid visualizations.

This limitation can be overcome by switching to Python for performing tasks such as social network analysis, where exist packages which are complete in themselves to perform a huge number of operations under the umbrella of a single package e.g. the ‘matplotlib’ package for making huge kinds of visualizations or the standalone ‘sklearn’ (scikit-learn) package for performing machine learning tasks.

Another limitation which exists is that, social network data, such as data generated by social media giants such as Facebook and Instagram are humongous in magnitude – they comprise of billions of rows, and tens of thousands of columns, and thus it’s not possible to analyze such huge data on software like R or Python, without reserving a considerable amount of time constraint which is highly valuable these days. It’s thus recommended to use software which are built to handle high-scale data such as Hadoop and Apache Spark which can withstand data having high magnitudes by parallelizing the entire process i.e the work is divided between clusters or different systems.

I’m still in the learning process when it comes to social network analysis, but I’ve learnt a lot due to this project. I now know have a good idea how complex algorithms like Pagerank, which form the backbone of search engines like Google work. I also know how to find out values of measures such as eigenvector centrality and HITS score of networks on the R language, a task which can take considerable amounts of time if done manually. But there’s still a lot to learn.  
My future plan is to learn how to implement complex concepts such as trust transitivity analysis on R, and also to get fluent in the famous ‘NetworkX’ library in Python, which deals with network analysis. The future seems exciting.

**8. References**

1) “Knowledge Management Operations”, Department of the Army, Washington, DC, Knowledge Management, 2012, available at Army Knowledge Online, <https://armypubs.us.army.mil/doctrine/index.html>

2) <http://thepoliticsofsystems.net/2013/07/scrutinizing-a-network-of-likes-on-facebook-and-some-thoughts-on-network-analysis-and-visualization/>

3) [www.google.com](http://www.google.com) (Google)

4) [www.github.com](http://www.github.com) (Github)

5) [www.wikipedia.org](http://www.wikipedia.org) (Wikipedia)

**Appendix**

**Code:**

[This project was written on R, and was saved in the R Markdown format. Please refer online, if you are not familiar how to write and save code in the R Markdown format.]

```{r}

library(gridExtra)

library(igraph)

library(ggplot2)

```

```{r}

# Reading in the graph, and viewing its brief summary

pl\_graph <- read.graph(file = "pl.gml.txt", format = "gml")

summary(pl\_graph)

```

```{r}

# Inspecting the page graph object

pl\_df <- data.frame(id = V(pl\_graph)$id,

name = V(pl\_graph)$label,

category = V(pl\_graph)$category,

fans = as.numeric(V(pl\_graph)$fan\_count),

talking\_about = as.numeric(V(pl\_graph)$talking\_about\_count),

post\_activity = as.numeric(V(pl\_graph)$post\_activity),

stringsAsFactors = FALSE

)

View(pl\_df)

```

```{r}

# Aggregating pages based on their category

grid.table(as.data.frame(sort(table(pl\_df$category), decreasing = TRUE)[1:10]),

rows = NULL,

cols = c('Category', 'Count'))

```

```{r}

# Top pages based on their fan count (likes)

grid.table(pl\_df[order(pl\_df$fans, decreasing = TRUE),

c('name', 'category', 'fans')][1:10, ],

rows = NULL)

```

```{r}

# Top pages based on total people talking about them

grid.table(pl\_df[order(pl\_df$talking\_about, decreasing = TRUE),

c('name', 'category', 'talking\_about')][1:10, ],

rows = NULL)

```

```{r}

# Top pages based on page posting activity

grid.table(pl\_df[order(pl\_df$post\_activity, decreasing = TRUE),

c('name', 'category', 'post\_activity')][1:10, ],

rows = NULL)

```

```{r}

# Checking correlation between fans and talking about for pages

clean\_pl\_df <- pl\_df[complete.cases(pl\_df), ]

rsq <- format(cor(clean\_pl\_df$fans, clean\_pl\_df$talking\_about) ^ 2, digits = 3)

corr\_plot <- ggplot(pl\_df, aes(x = fans, y = talking\_about)) + theme\_bw() +

geom\_jitter(alpha = 1/2) +

scale\_x\_log10() +

scale\_y\_log10() +

labs(x = "Fans", y = "Talking About") +

annotate("text", label = paste("R-sq = ", rsq), x = +Inf, y = 1, hjust = 1)

corr\_plot

```

```{r}

# Plotting page network using degree filter

degrees <- degree(pl\_graph, mode = "total")

degrees\_df <- data.frame(ID = V(pl\_graph)$id,

Name = V(pl\_graph)$label,

Degree = as.vector(degrees))

ids\_to\_remove <- degrees\_df[degrees\_df$Degree < 30, c('ID')]

ids\_to\_remove <- ids\_to\_remove / 10

# Getting filtered graph

filtered\_pl\_graph <- delete.vertices(pl\_graph, ids\_to\_remove)

# Plotting the graph

tkplot(filtered\_pl\_graph,

vertex.size = 10,

vertex.color = "orange",

vertex.frame.color = "white",

vertex.label.color = "black",

vertex.label.family = "sans",

edge.width = 0.2,

edge.arrow.size = 0,

edge.color = "grey",

edge.curved = TRUE,

layout = layout.fruchterman.reingold)

```

```{r}

# Diameter (length of longest path) of the network

diameter(pl\_graph, directed = TRUE)

```

```{r}

# Getting the longest path of the network

get\_diameter(pl\_graph, directed = TRUE)$label

```

```{r}

# Mean distance between 2 nodes in the network

mean\_distance(pl\_graph, directed = TRUE)

```

```{r}

# Distance between various important pages (nodes)

node\_dists <- distances(pl\_graph, weights = NA)

labels <- c("Premier League", pl\_df[c(21, 22, 23, 24, 25), 'name'])

filtered\_dists <- node\_dists[c(1, 21, 22, 23, 24, 25), c(1, 21, 22, 23, 24, 25)]

colnames(filtered\_dists) <- labels

rownames(filtered\_dists) <- labels

grid.table(filtered\_dists)

View(filtered\_dists)

```

```{r}

edge\_density(pl\_graph)

# no. of edges / no. of all possible edges

```

```{r}

2801 / (582 \* 581)

```

```{r}

transitivity(pl\_graph)

# transitivity clustering coefficient

```

```{r}

page\_names <- V(pl\_graph)$label

page\_coreness <- coreness(pl\_graph)

page\_coreness\_df = data.frame(Page = page\_names,

PageCoreness = page\_coreness)

page\_coreness\_df

# The k-core of a graph is a maximal subgraph in which each vertex has at least degree k.

# The coreness of a vertex is k if it belongs to the k-core but not to the (k+1)-core.

```

```{r}

# Max coreness

max(page\_coreness\_df$PageCoreness)

```

```{r}

View(head(page\_coreness\_df[

page\_coreness\_df$PageCoreness == max(page\_coreness\_df$PageCoreness),], 20))

# Viewing the core of the network

```

```{r}

View(head(page\_coreness\_df[

page\_coreness\_df$PageCoreness == min(page\_coreness\_df$PageCoreness),], 20))

# Viewing the periphery of the network

```

```{r}

degree\_plg <- degree(pl\_graph, mode = "total")

degree\_plg\_df <- data.frame(Name = V(pl\_graph)$label,

Degree = as.vector(degree\_plg))

degree\_plg\_df <- degree\_plg\_df[order(degree\_plg\_df$Degree, decreasing = TRUE), ]

View(degree\_plg\_df)

# Degree Centrality

```

```{r}

closeness\_plg <- closeness(pl\_graph, mode = "all", normalized = TRUE)

closeness\_plg\_df <- data.frame(Name = V(pl\_graph)$label,

Closeness = as.vector(closeness\_plg))

closeness\_plg\_df <- closeness\_plg\_df[order(closeness\_plg\_df$Closeness, decreasing = TRUE), ]

View(closeness\_plg\_df)

# Closeness Centrality

```

```{r}

betweenness\_plg <- betweenness(pl\_graph)

betweenness\_plg\_df <- data.frame(Name = V(pl\_graph)$label,

Betweenness = as.vector(betweenness\_plg))

betweenness\_plg\_df <- betweenness\_plg\_df[order(betweenness\_plg\_df$Betweenness, decreasing = TRUE), ]

View(betweenness\_plg\_df)

# Betweenness Centrality

```

```{r}

View(head(degree\_plg\_df, 10))

View(head(closeness\_plg\_df, 10))

View(head(betweenness\_plg\_df, 10))

# Viewing top pages based on above measures

```

```{r}

# Correlation plots

plg\_df <- data.frame(degree\_plg, closeness\_plg, betweenness\_plg)

# Graph 1 - degree vs closeness

rsq <- format(cor(degree\_plg, closeness\_plg) ^ 2, digits = 3)

corr\_plot <- ggplot(plg\_df, aes(x = degree\_plg, y = closeness\_plg)) +

theme\_bw() +

geom\_jitter(alpha = 1/2) +

scale\_y\_log10() +

labs(x = "Degree", y = "Closeness") +

annotate("text", label = paste("R-sq = ", rsq), x = +Inf, y = 1, hjust = 1)

corr\_plot

```

```{r}

# Graph 2- degree vs betweenness

rsq <- format(cor(degree\_plg, betweenness\_plg) ^ 2, digits = 3)

corr\_plot <- ggplot(plg\_df, aes(x = degree\_plg, y = betweenness\_plg)) +

theme\_bw() +

geom\_jitter(alpha = 1/2) +

scale\_y\_log10() +

labs(x = "Degree", y = "Betweenness") +

annotate("text", label = paste("R-sq = ", rsq), x = +Inf, y = 1, hjust = 1)

corr\_plot

```

```{r}

evcentrality\_plg <- eigen\_centrality(pl\_graph)$vector

evcentrality\_plg\_df <- data.frame(Name = V(pl\_graph)$label,

EVcentrality = as.vector(evcentrality\_plg))

evcentrality\_plg\_df <- evcentrality\_plg\_df[order(evcentrality\_plg\_df$EVcentrality, decreasing = TRUE), ]

View(head(evcentrality\_plg\_df, 10))

# Eigenvector Centrality

```

```{r}

pagerank\_plg <- page\_rank(pl\_graph)$vector

pagerank\_plg\_df <- data.frame(Name = V(pl\_graph)$label,

PageRank = as.vector(pagerank\_plg))

pagerank\_plg\_df <- pagerank\_plg\_df[order(pagerank\_plg\_df$PageRank, decreasing = TRUE), ]

View(head(pagerank\_plg\_df, 10))

# PageRank

```

```{r}

hits\_plg <- authority\_score(pl\_graph)$vector

hits\_plg\_df <- data.frame(Name = V(pl\_graph)$label,

AuthScore = as.vector(hits\_plg))

hits\_plg\_df <- hits\_plg\_df[order(hits\_plg\_df$AuthScore, decreasing = TRUE), ]

View(head(hits\_plg\_df, 10))

# Kleinberg's HITS score

```

```{r}

pl\_neighbours <- neighbors(pl\_graph, v = which(V(pl\_graph)$label == "Southampton FC"))

pl\_neighbours

pl\_neighbours$label

# Example of finding neighbours of page vertices

```

```{r}

## Get communities / clusters

# cliques use different sizes here and experiment

clique\_num(pl\_graph)

count\_max\_cliques(pl\_graph, min = 10, max = 10)

clique\_list <- cliques(pl\_graph, min = 10, max = 10)

for (clique in clique\_list) {

print(clique$label)

cat('\n\n')

}

```

```{r}

# filtering graph to get important nodes based on degree

degrees <- degree(pl\_graph, mode = "total")

degrees\_df <- data.frame(ID = V(pl\_graph)$id,

Name = V(pl\_graph)$label,

Degree = as.vector(degree\_plg))

ids\_to\_remove <- degrees\_df[degrees\_df$Degree < 30, c('ID')]

ids\_to\_remove <- ids\_to\_remove / 10

filtered\_pl\_graph <- delete.vertices(pl\_graph, ids\_to\_remove)

fplg\_undirected <- as.undirected(filtered\_pl\_graph)

```

```{r}

# Fast greedy clustering

fgc <- cluster\_fast\_greedy(fplg\_undirected)

layout <- layout\_with\_fr(fplg\_undirected,

niter = 500,

start.temp = 5.744)

communities <- data.frame(layout)

names(communities) <- c("x", "y")

communities$cluster <- factor(fgc$membership)

communities$name <- V(fplg\_undirected)$label

```

```{r}

# get total pages in each cluster

table(communities$cluster)

```

```{r}

# get page names in each cluster

community\_groups <- unlist(lapply(groups(fgc),

function(item) {

pages <- communities$name[item]

i <- 1;

lim <- 4;

s <- ""

while(i <= length(pages)) {

start = i

end = min((i + lim - 1), length(pages))

s <- paste(s, paste(pages[start:end], collapse = ", "))

s <- paste(s, "\n")

i = i + lim

}

return(substr(s, 1, (nchar(s) - 2)))

}))

grid.table(community\_groups)

```

```{r}

# get modularity score

modularity(fgc)

comm\_plot <- ggplot(communities, aes(x = x, y = y, color = cluster, label = name))

comm\_plot <- comm\_plot + geom\_label(aes(fill = cluster), colour = "white")

comm\_plot

plot(fgc, fplg\_undirected,

vertex.size = 15,

vertex.label.cex = 0.8,

vertex.label = fgc$names,

edge.arrow.size = 0,

edge.curved = TRUE,

vertex.label.color = "black",

layout = layout.fruchterman.reingold)

```

```{r}

# edge betweenness clustering

ebc <- cluster\_edge\_betweenness(fplg\_undirected)

layout <- layout\_with\_fr(fplg\_undirected,

niter = 500, start.temp = 5.744)

communities <- data.frame(layout)

names(communities) <- c("x", "y")

communities$cluster <- factor(ebc$membership)

communities$name <- V(fplg\_undirected)$label

table(communities$cluster)

community\_groups <- unlist(lapply(groups(ebc),

function(item) {

pages <- communities$name[item]

i <- 1;

lim <- 4;

s <- ""

while(i <= length(pages)) {

start = i

end = min((i + lim - 1), length(pages))

s <- paste(s, paste(pages[start:end], collapse = ", "))

s <- paste(s, "\n")

i= i + lim

}

return(substr(s, 1, (nchar(s) - 2)))

}))

grid.table(community\_groups)

modularity(ebc)

comm\_plot <- ggplot(communities, aes(x = x, y = y, color = cluster, label = name))

comm\_plot <- comm\_plot + geom\_label(aes(fill = cluster), colour = "white")

comm\_plot

plot(ebc, fplg\_undirected,

vertex.size = 15,

vertex.label.cex = 0.8,

vertex.label = ebc$names,

edge.arrow.size = 0,

edge.curved = TRUE,

vertex.label.color = "black",

layout = layout.fruchterman.reingold)

```