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

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

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.

Gű̈rkan 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	Gữrkan et al.	To develop and validate a large scale online argumentation platform	General organizations
2012	Zhang et al.	To analyze research	Health

		literature on patient adherence and show their knowledge structure over time	
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.

Gũrkan 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 (Gű̈rkan 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 :

   (a) 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.

   (b) 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.

9. Plotting page network using degree filter

10. Getting filtered graph

(a) 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.

(b) 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.

11.Plotting the graph

12.Diameter (length of longest path) of the network

13.Getting the longest path of the network

14.Mean distance between 2 nodes in the network

15.Distance between various important pages (nodes)

16.no. of edges / no. of all possible edges

17. transitivity clustering coefficient

    (a) 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.

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

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

20. Max coreness

21. Viewing the core of the network

22. Viewing the periphery of the network

    (a) 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.
    (b) 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.

23. Degree centrality

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

24.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.

25.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.

26. Viewing top pages based on above measures

27.Correlation plots

a) Graph 1 - degree vs. closeness

b) Graph 2- degree vs. betweenness

28.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.

29.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.

30.Kleinberg's HITS score

31. Example of finding neighbours of page vertices

32.Get communities / clusters

33.filtering graph to get important nodes based on degree

34.Fast greedy clustering

(a) 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.

(b) 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.

35.get total pages in each cluster

36.get page names in each cluster

37.get modularity score

(a) 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.

(b) 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.

38.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).

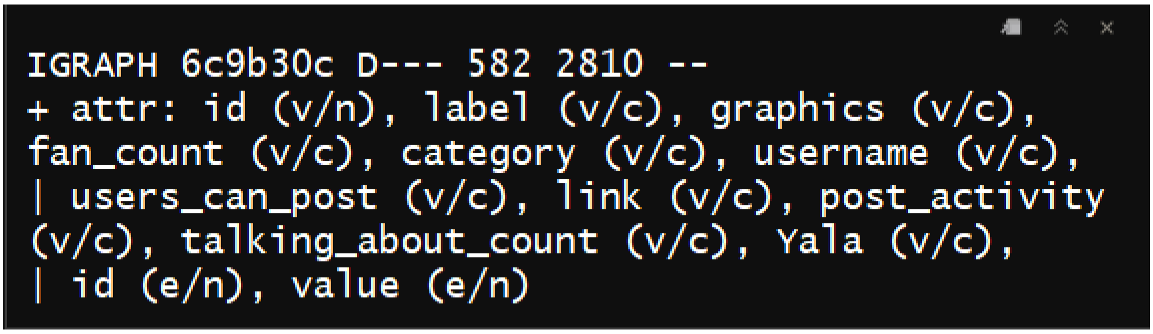
A description of what each chunk contains exactly can be found in the section *3.2 Framework, Architecture or Module of the Proposed System*.

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

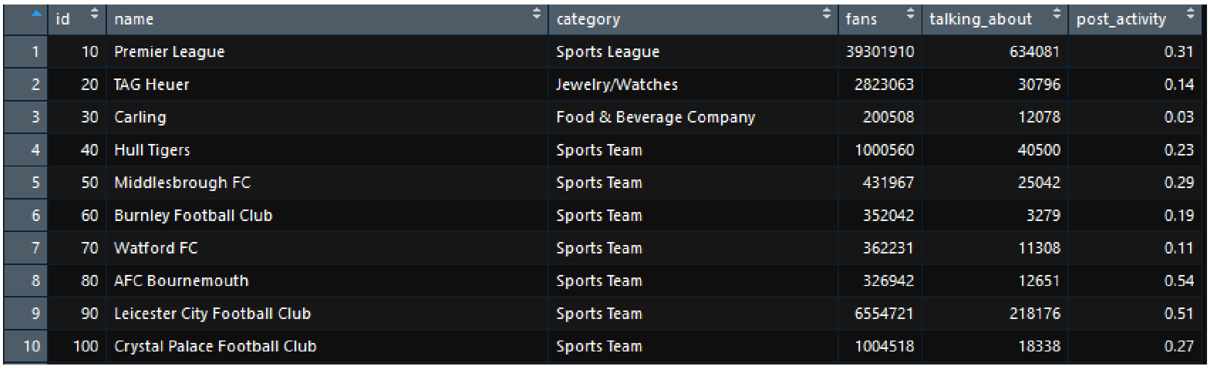
This section will contain pictures of the respective outputs for the different graph features which have been calculated in the proposed system.

I) Reading in the graph and viewing its brief summary



```
IGRAPH 6c9b30c D--- 582 2810 --
+ attr: id (v/n), label (v/c), graphics (v/c),
fan_count (v/c), category (v/c), username (v/c),
| users_can_post (v/c), link (v/c), post_activity
(v/c), talking_about_count (v/c), Yala (v/c),
| id (e/n), value (e/n)
```

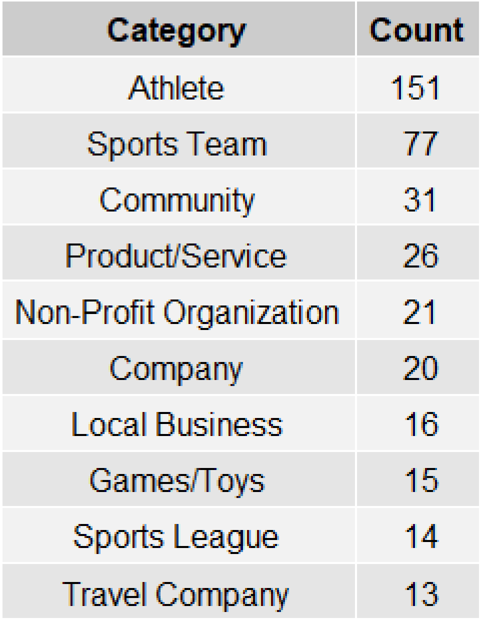
II) Inspecting the page graph object



	id	name	category	fans	talking_about	post_activity
1	10	Premier League	Sports League	39301910	634081	0.31
2	20	TAG Heuer	Jewelry/Watches	2823063	30796	0.14
3	30	Carling	Food & Beverage Company	200508	12078	0.03
4	40	Hull Tigers	Sports Team	1000560	40500	0.23
5	50	Middlesbrough FC	Sports Team	431967	25042	0.29
6	60	Burnley Football Club	Sports Team	352042	3279	0.19
7	70	Watford FC	Sports Team	362231	11308	0.11
8	80	AFC Bournemouth	Sports Team	326942	12651	0.54
9	90	Leicester City Football Club	Sports Team	6554721	218176	0.51
10	100	Crystal Palace Football Club	Sports Team	1004518	18338	0.27

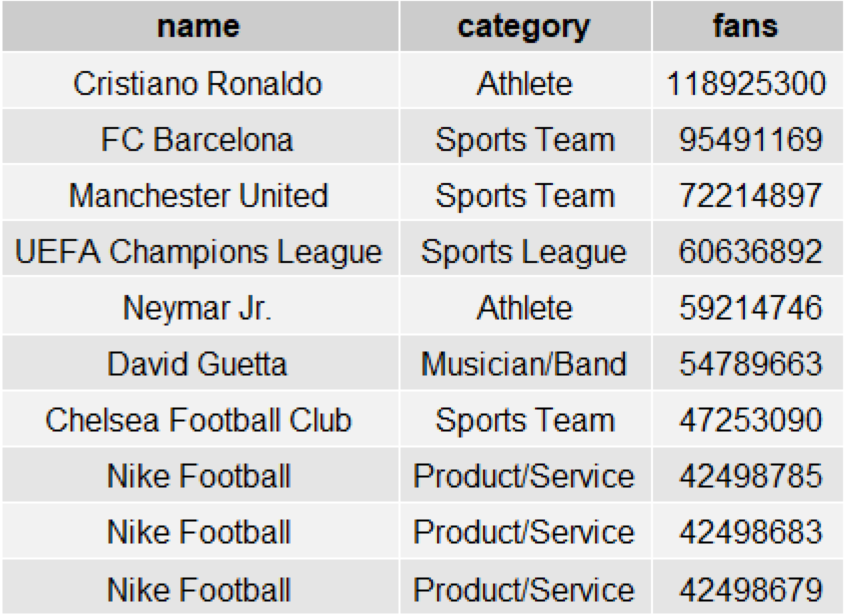
[Showing 10 pages out of 582 total pages]

III) Aggregating pages based on their category



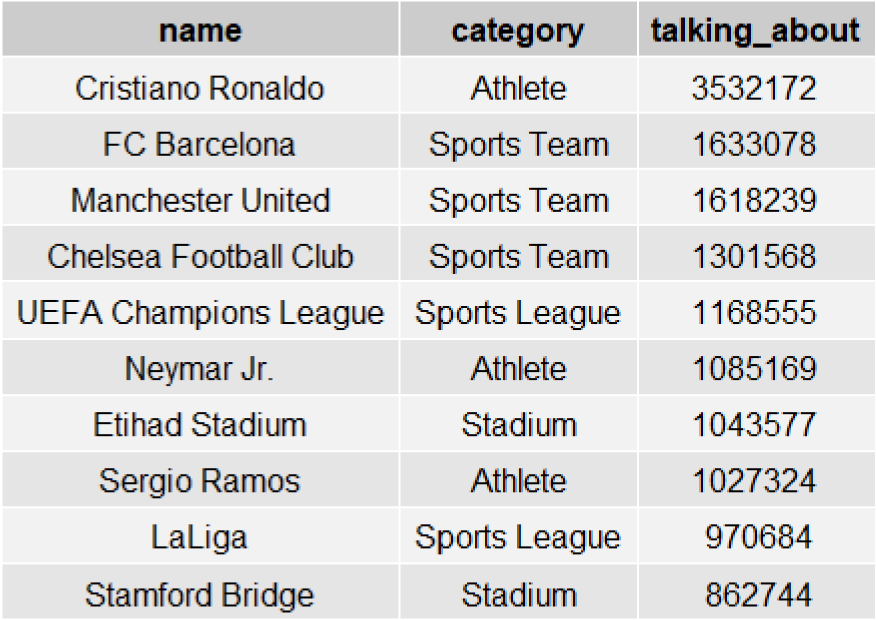
Category	Count
Athlete	151
Sports Team	77
Community	31
Product/Service	26
Non-Profit Organization	21
Company	20
Local Business	16
Games/Toys	15
Sports League	14
Travel Company	13

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



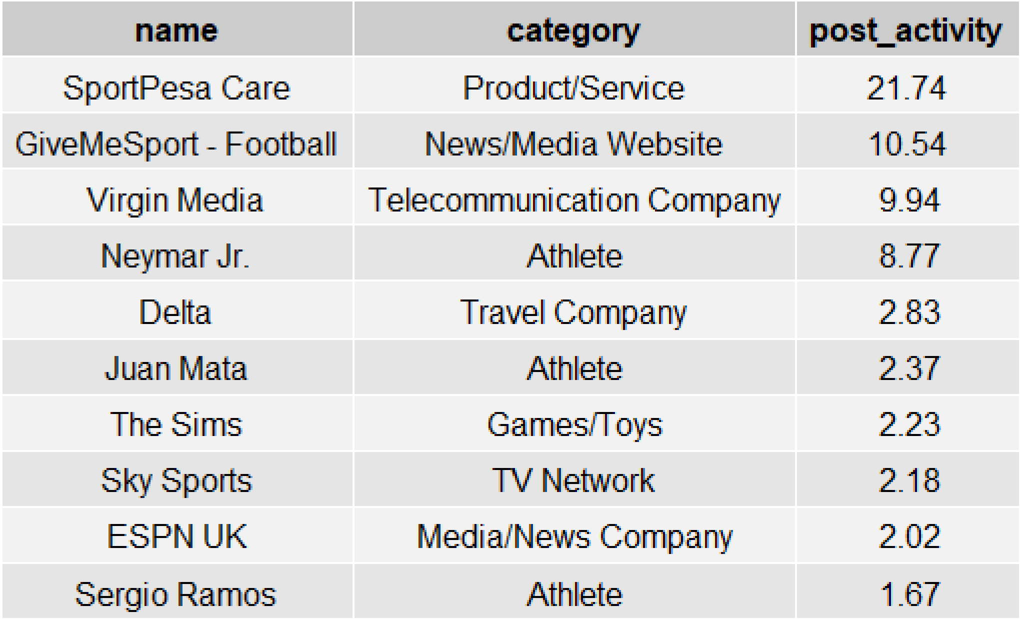
name	category	fans
Cristiano Ronaldo	Athlete	118925300
FC Barcelona	Sports Team	95491169
Manchester United	Sports Team	72214897
UEFA Champions League	Sports League	60636892
Neymar Jr.	Athlete	59214746
David Guetta	Musician/Band	54789663
Chelsea Football Club	Sports Team	47253090
Nike Football	Product/Service	42498785
Nike Football	Product/Service	42498683
Nike Football	Product/Service	42498679

V) Top pages based on total people talking about them



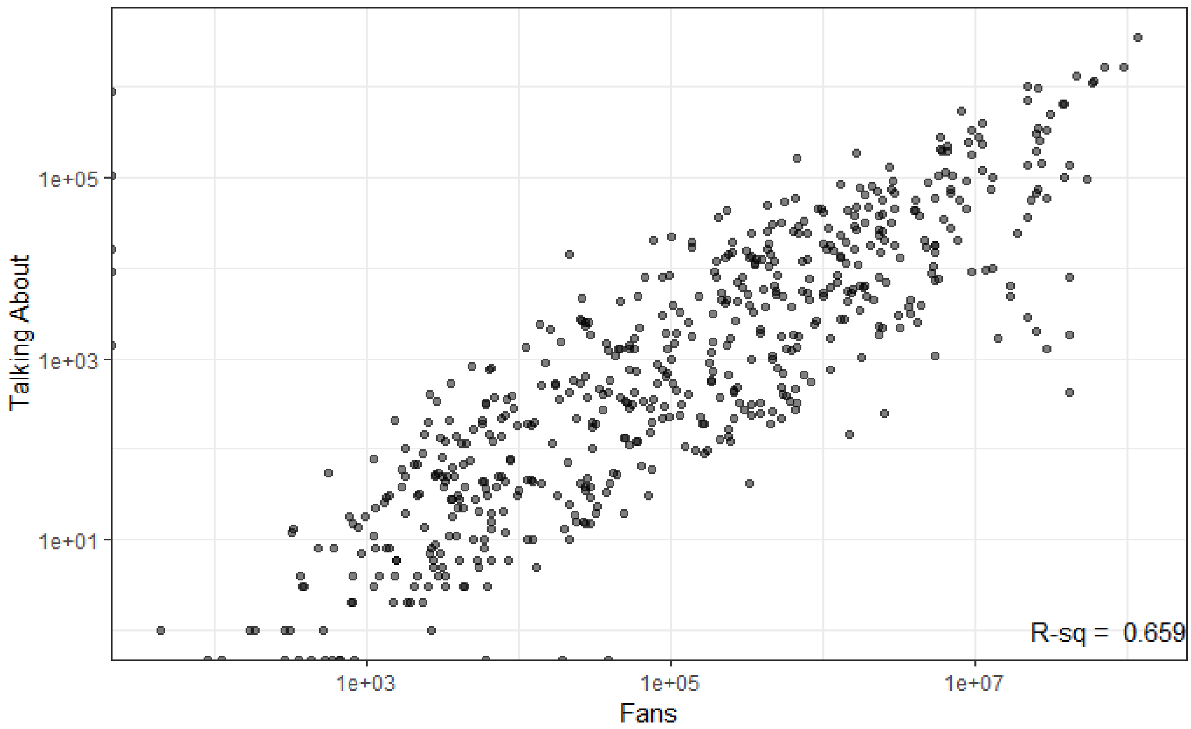
name	category	talking_about
Cristiano Ronaldo	Athlete	3532172
FC Barcelona	Sports Team	1633078
Manchester United	Sports Team	1618239
Chelsea Football Club	Sports Team	1301568
UEFA Champions League	Sports League	1168555
Neymar Jr.	Athlete	1085169
Etihad Stadium	Stadium	1043577
Sergio Ramos	Athlete	1027324
LaLiga	Sports League	970684
Stamford Bridge	Stadium	862744

VI) Top pages based on page posting activity

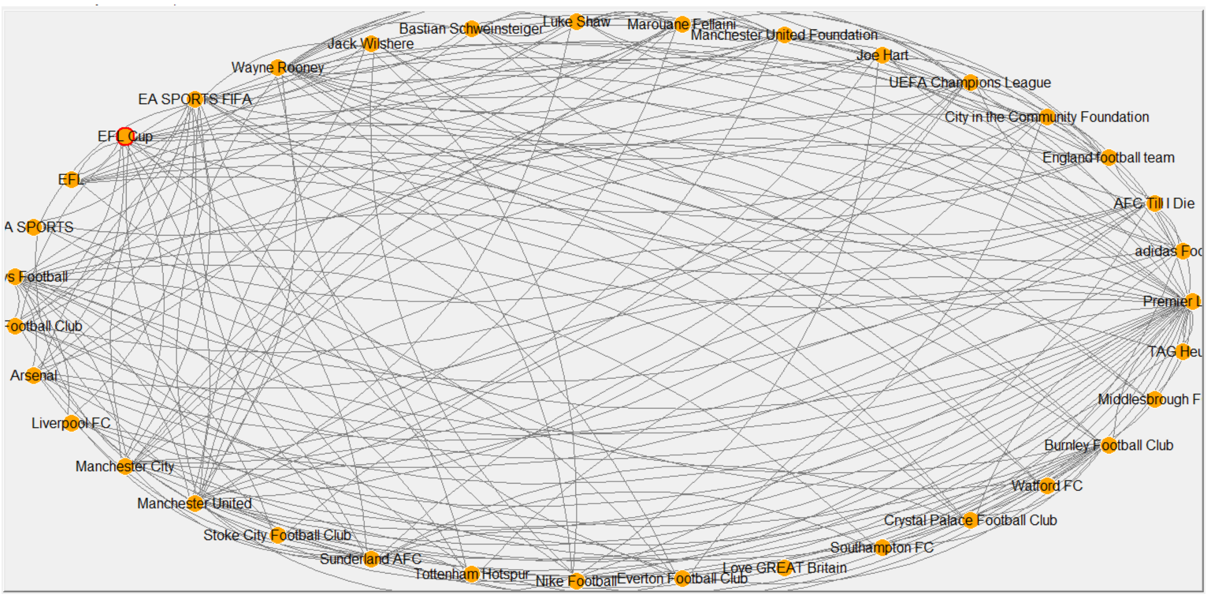


name	category	post_activity
SportPesa Care	Product/Service	21.74
GiveMeSport - Football	News/Media Website	10.54
Virgin Media	Telecommunication Company	9.94
Neymar Jr.	Athlete	8.77
Delta	Travel Company	2.83
Juan Mata	Athlete	2.37
The Sims	Games/Toys	2.23
Sky Sports	TV Network	2.18
ESPN UK	Media/News Company	2.02
Sergio Ramos	Athlete	1.67

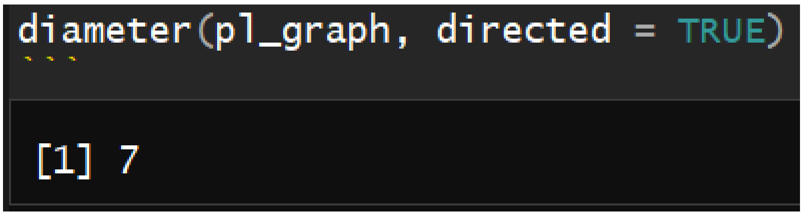
VII) Checking correlation b/w fans and talking_about for pages



VIII) Plotting page network using degree filter



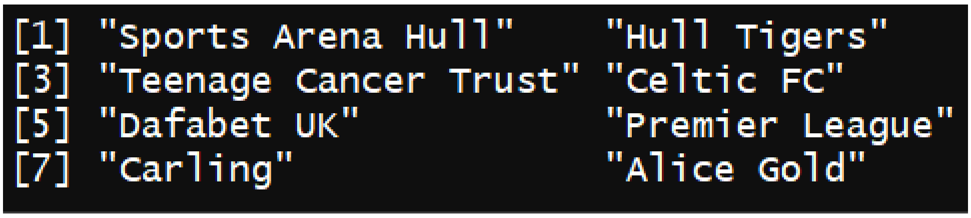
IX) Diameter (length of the longest path) of the network



```
diameter(pl_graph, directed = TRUE)
```

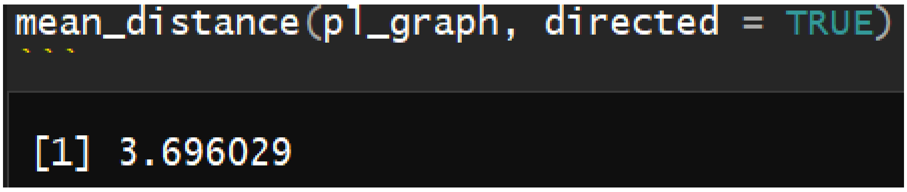
```
[1] 7
```

X) Getting the longest path of the network



```
[1] "Sports Arena Hull"     "Hull Tigers"
[3] "Teenage Cancer Trust" "Celtic FC"
[5] "Dafabet UK"            "Premier League"
[7] "Carling"               "Alice Gold"
```

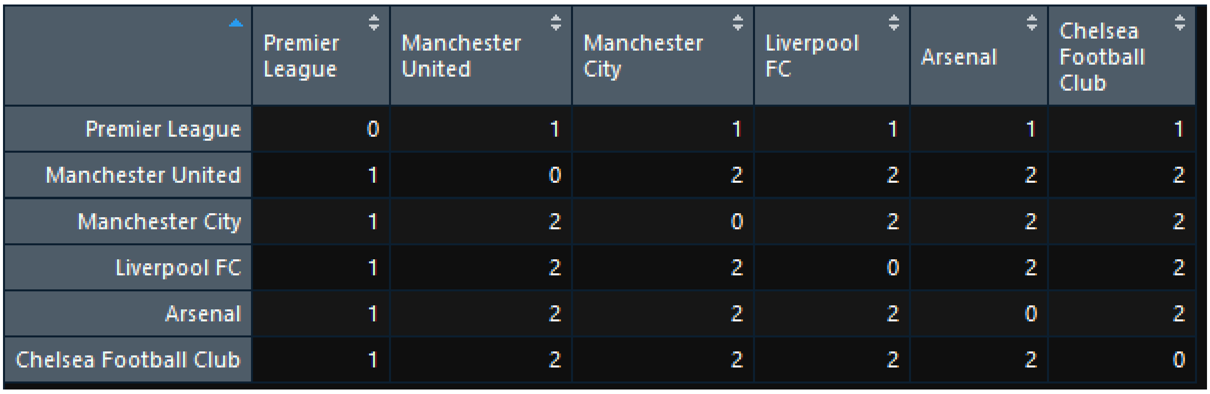
XI) Mean distance b/w 2 nodes in the network



~~~
mean_distance(pl_graph, directed = TRUE)
```

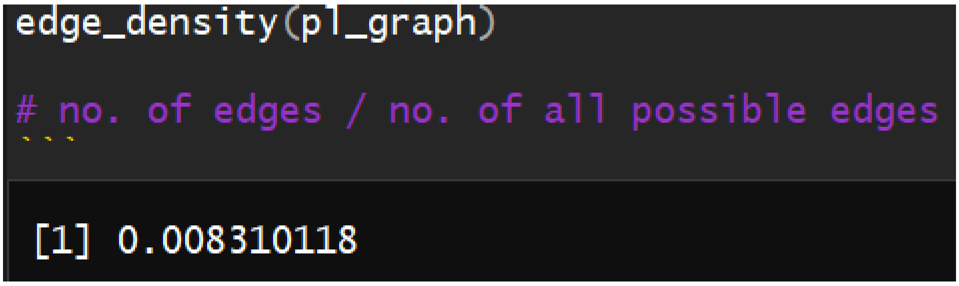
[1] 3.696029
~~~

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



	Premier League	Manchester United	Manchester City	Liverpool FC	Arsenal	Chelsea Football Club
Premier League	0	1	1	1	1	1
Manchester United	1	0	2	2	2	2
Manchester City	1	2	0	2	2	2
Liverpool FC	1	2	2	0	2	2
Arsenal	1	2	2	2	0	2
Chelsea Football Club	1	2	2	2	2	0

XIII) Edge density of the graph

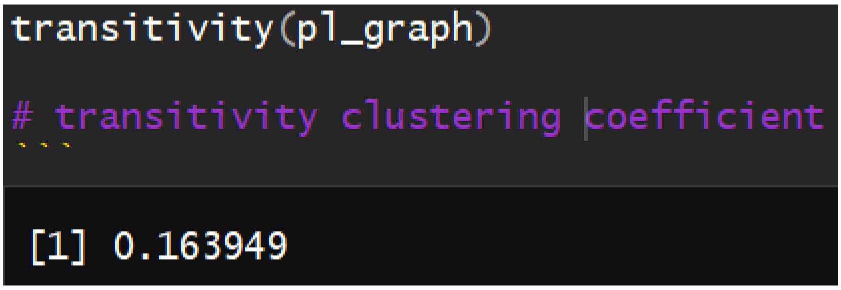


~~~
edge_density(pl_graph)

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

[1] 0.008310118
~~~

XIV) Transitivity clustering coefficient

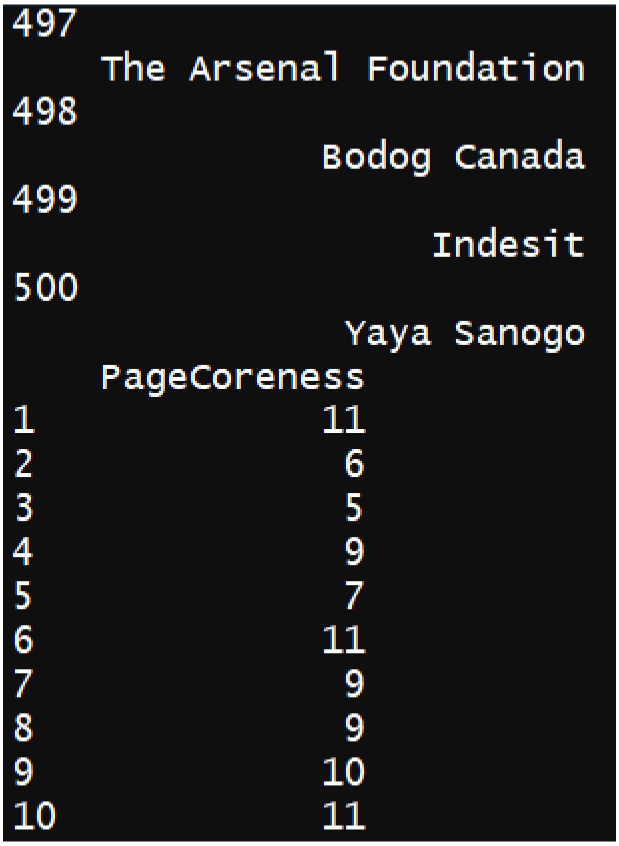


~~~
transitivity(pl_graph)

# transitivity clustering coefficient
```
~~~

```
[1] 0.163949
```

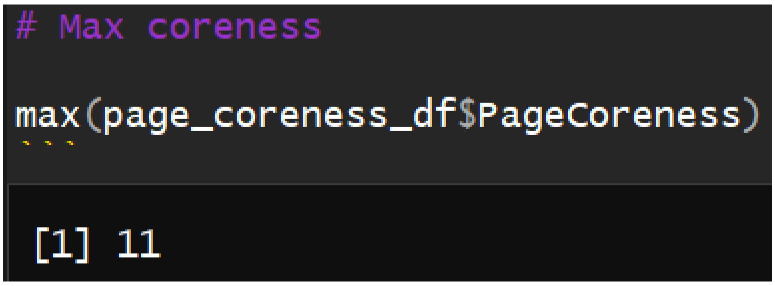
XV) Page coreness (k-core, coreness)



```
497
    The Arsenal Foundation
498
              Bodog Canada
499
                   Indesit
500
               Yaya Sanogo
    PageCoreness
1             11
2              6
3              5
4              9
5              7
6             11
7              9
8              9
9             10
10            11
```

[A random output snippet]

XVI) Max coreness

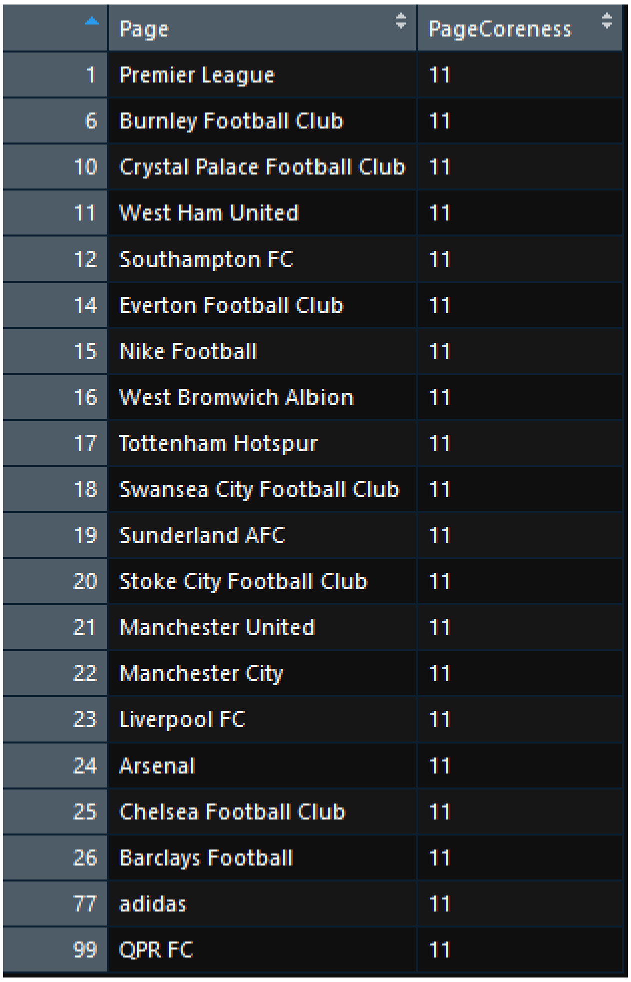


~~~
# Max coreness

max(page_coreness_df$PageCoreness)
```
~~~

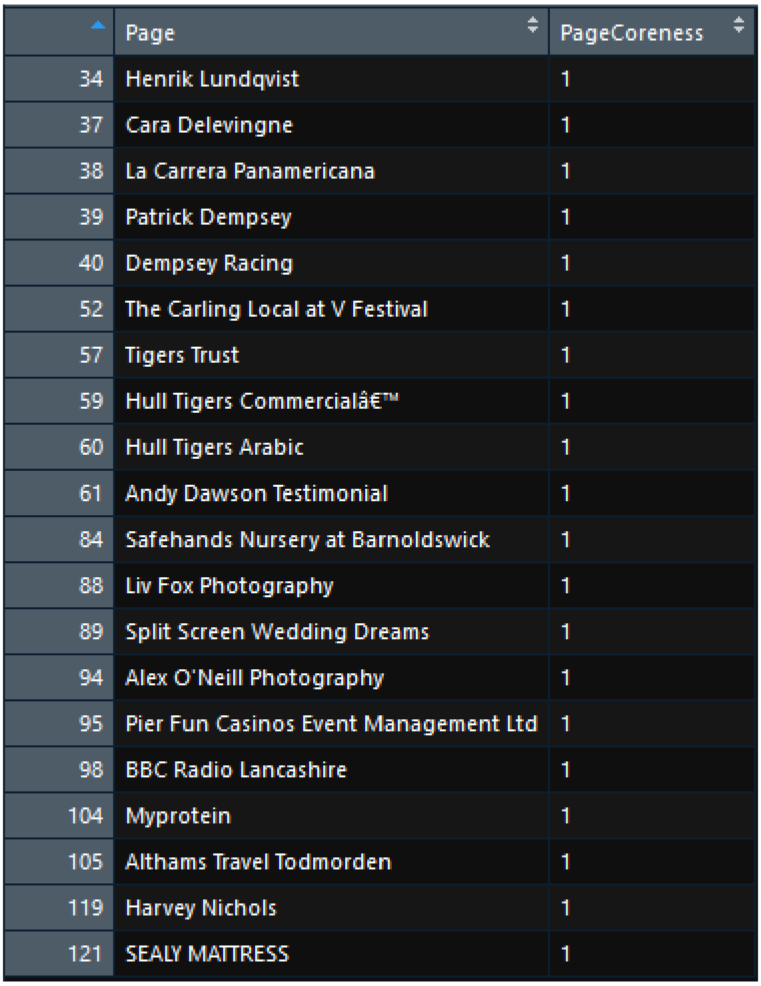
```
[1] 11
```

XVII) Viewing the core of the network



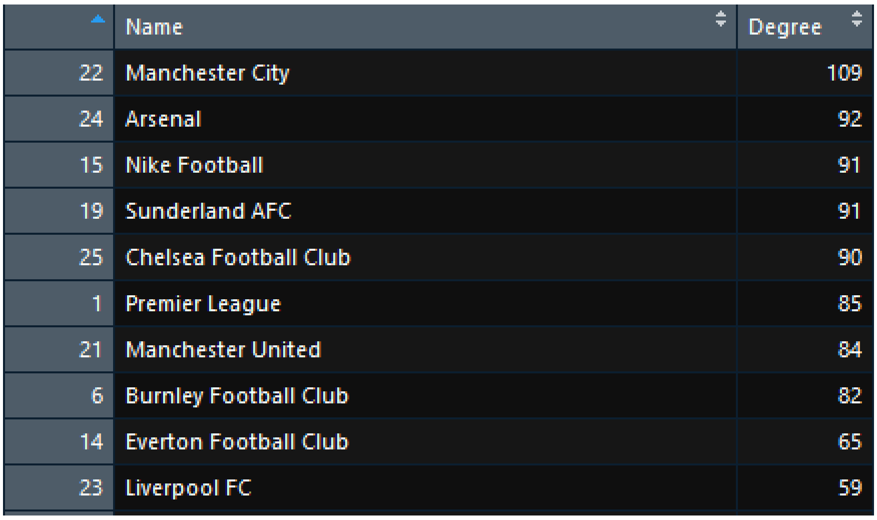
	Page	PageCoreness
1	Premier League	11
6	Burnley Football Club	11
10	Crystal Palace Football Club	11
11	West Ham United	11
12	Southampton FC	11
14	Everton Football Club	11
15	Nike Football	11
16	West Bromwich Albion	11
17	Tottenham Hotspur	11
18	Swansea City Football Club	11
19	Sunderland AFC	11
20	Stoke City Football Club	11
21	Manchester United	11
22	Manchester City	11
23	Liverpool FC	11
24	Arsenal	11
25	Chelsea Football Club	11
26	Barclays Football	11
77	adidas	11
99	QPR FC	11

XVIII) Viewing the periphery of the network



	Page	PageCoreness
34	Henrik Lundqvist	1
37	Cara Delevingne	1
38	La Carrera Panamericana	1
39	Patrick Dempsey	1
40	Dempsey Racing	1
52	The Carling Local at V Festival	1
57	Tigers Trust	1
59	Hull Tigers Commercialâ€™	1
60	Hull Tigers Arabic	1
61	Andy Dawson Testimonial	1
84	Safehands Nursery at Barnoldswick	1
88	Liv Fox Photography	1
89	Split Screen Wedding Dreams	1
94	Alex O'Neill Photography	1
95	Pier Fun Casinos Event Management Ltd	1
98	BBC Radio Lancashire	1
104	Myprotein	1
105	Althams Travel Todmorden	1
119	Harvey Nichols	1
121	SEALY MATTRESS	1

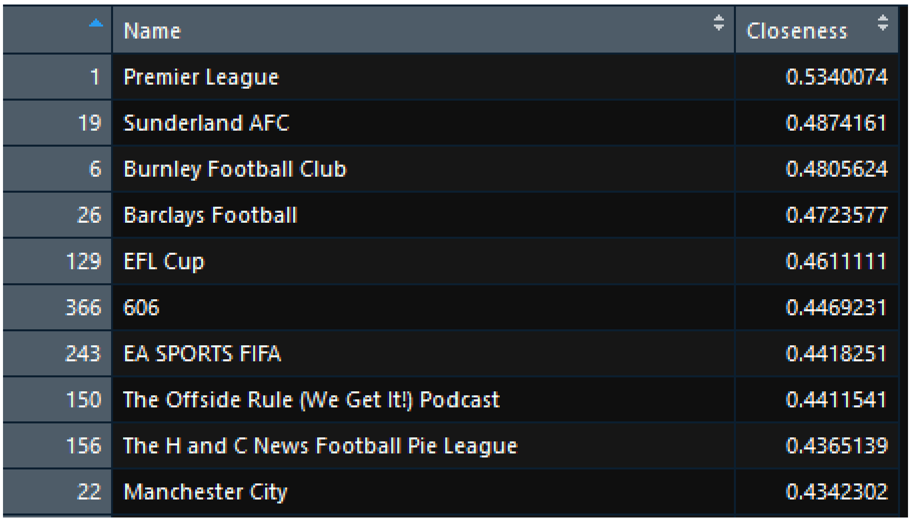
XIX) Degree centrality



	Name	Degree
22	Manchester City	109
24	Arsenal	92
15	Nike Football	91
19	Sunderland AFC	91
25	Chelsea Football Club	90
1	Premier League	85
21	Manchester United	84
6	Burnley Football Club	82
14	Everton Football Club	65
23	Liverpool FC	59

[Showing top 10 entries]

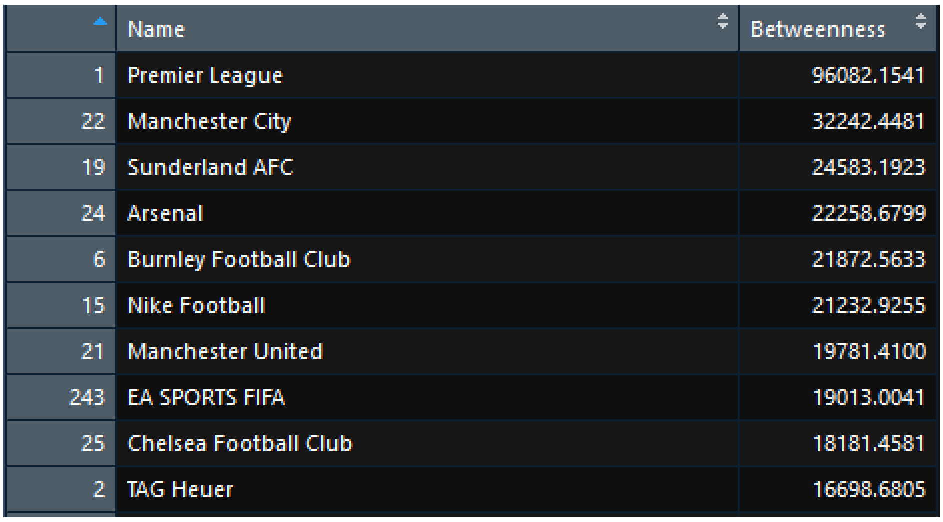
XX) Closeness centrality



	Name	Closeness
1	Premier League	0.5340074
19	Sunderland AFC	0.4874161
6	Burnley Football Club	0.4805624
26	Barclays Football	0.4723577
129	EFL Cup	0.4611111
366	606	0.4469231
243	EA SPORTS FIFA	0.4418251
150	The Offside Rule (We Get It!) Podcast	0.4411541
156	The H and C News Football Pie League	0.4365139
22	Manchester City	0.4342302

[Showing top 10 entries]

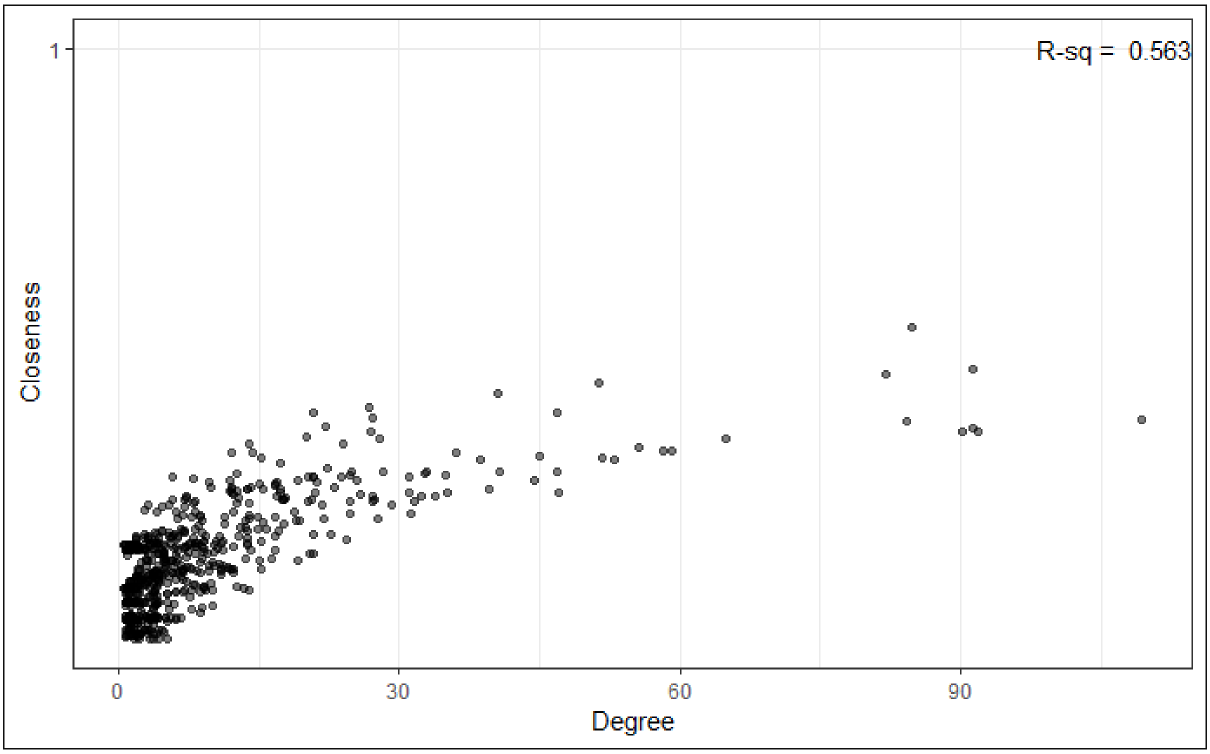
XXI) Betweenness centrality



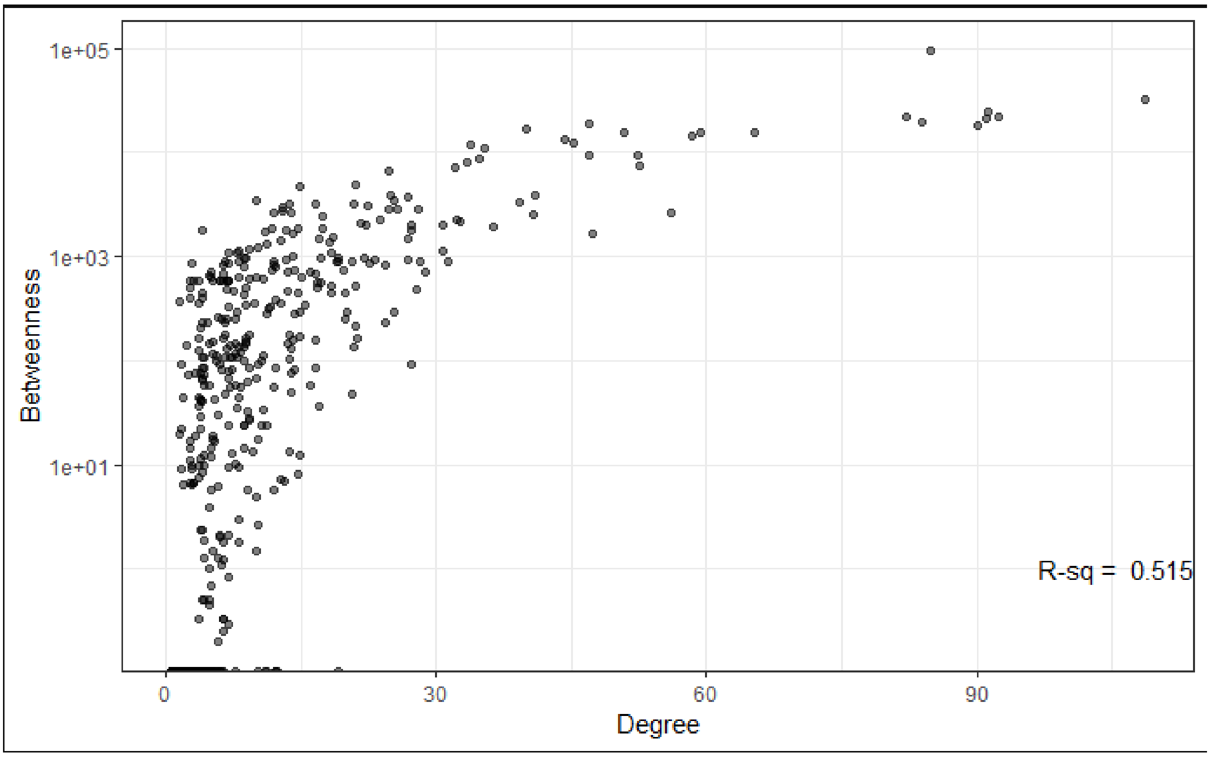
	Name	Betweenness
1	Premier League	96082.1541
22	Manchester City	32242.4481
19	Sunderland AFC	24583.1923
24	Arsenal	22258.6799
6	Burnley Football Club	21872.5633
15	Nike Football	21232.9255
21	Manchester United	19781.4100
243	EA SPORTS FIFA	19013.0041
25	Chelsea Football Club	18181.4581
2	TAG Heuer	16698.6805

[Showing top 10 entries]

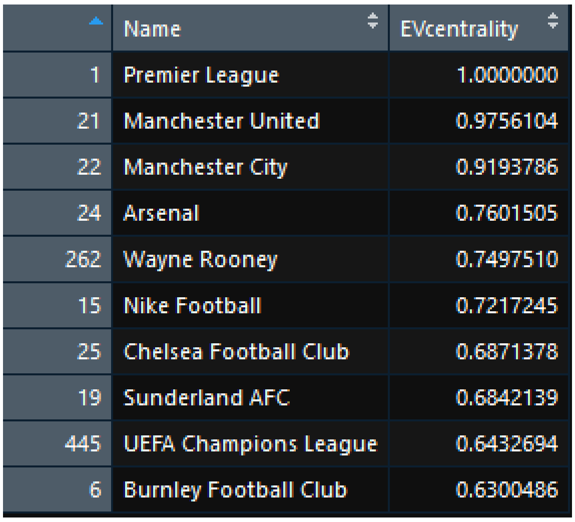
XXII) Correlation plot – degree centrality vs. closeness centrality



XXIII) Correlation plot – degree centrality vs. betweenness centrality



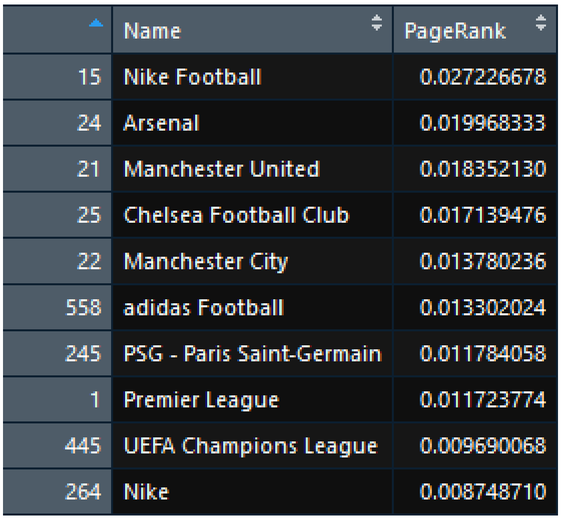
XXIV) Eigenvector centrality



	Name	EVcentrality
1	Premier League	1.0000000
21	Manchester United	0.9756104
22	Manchester City	0.9193786
24	Arsenal	0.7601505
262	Wayne Rooney	0.7497510
15	Nike Football	0.7217245
25	Chelsea Football Club	0.6871378
19	Sunderland AFC	0.6842139
445	UEFA Champions League	0.6432694
6	Burnley Football Club	0.6300486

[Showing top 10 pages]

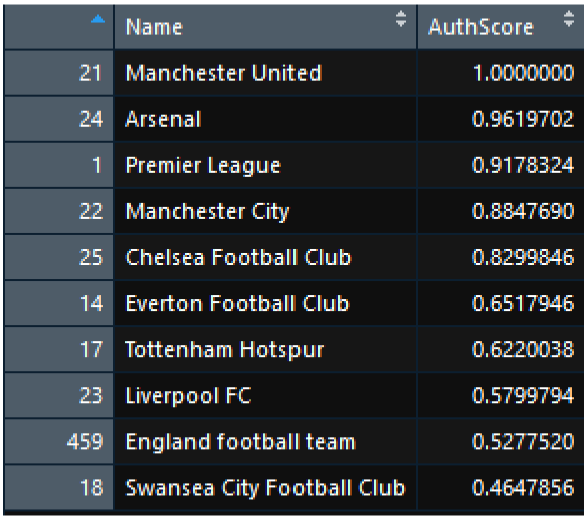
XXV) PageRank



	Name	PageRank
15	Nike Football	0.027226678
24	Arsenal	0.019968333
21	Manchester United	0.018352130
25	Chelsea Football Club	0.017139476
22	Manchester City	0.013780236
558	adidas Football	0.013302024
245	PSG - Paris Saint-Germain	0.011784058
1	Premier League	0.011723774
445	UEFA Champions League	0.009690068
264	Nike	0.008748710

[Showing top 10 pages]

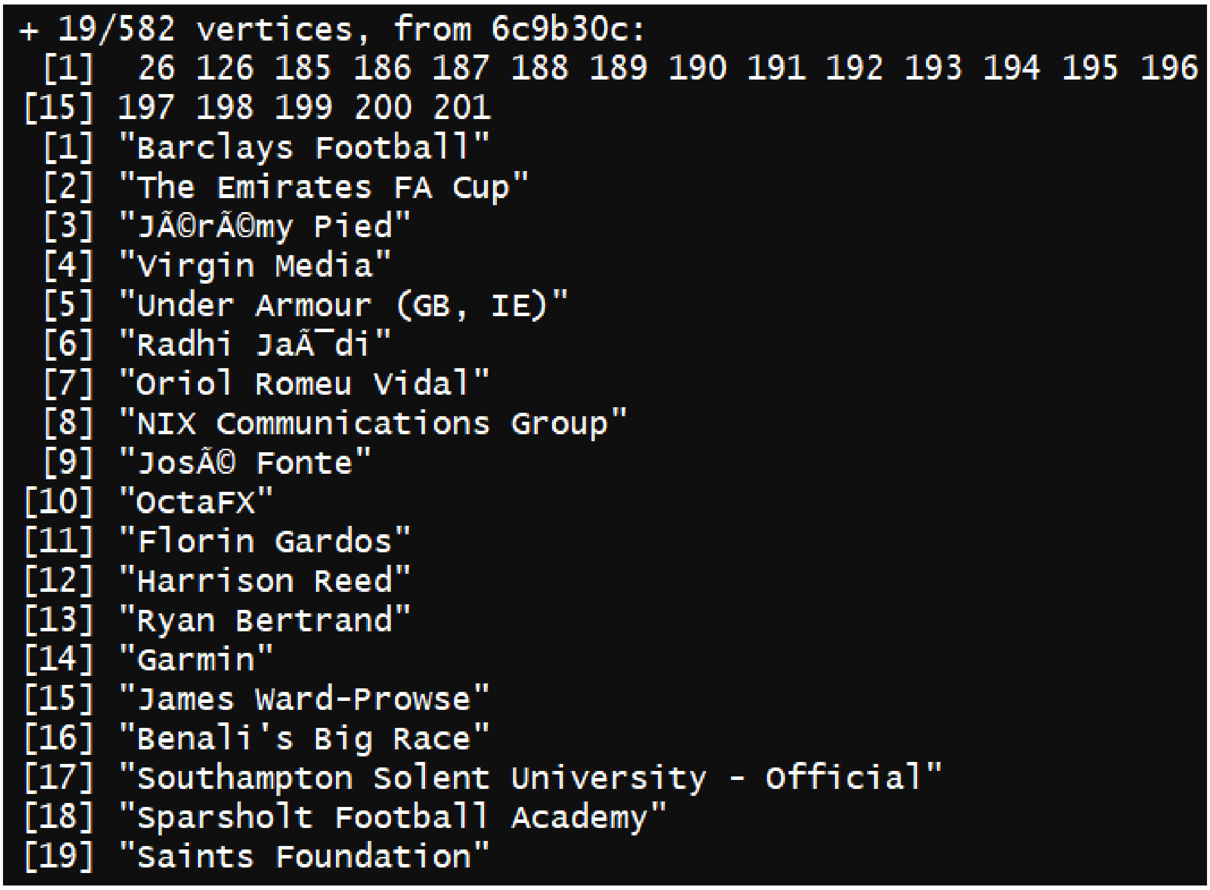
XXVI) Kleinberg’s HITS Score



	Name	AuthScore
21	Manchester United	1.0000000
24	Arsenal	0.9619702
1	Premier League	0.9178324
22	Manchester City	0.8847690
25	Chelsea Football Club	0.8299846
14	Everton Football Club	0.6517946
17	Tottenham Hotspur	0.6220038
23	Liverpool FC	0.5799794
459	England football team	0.5277520
18	Swansea City Football Club	0.4647856

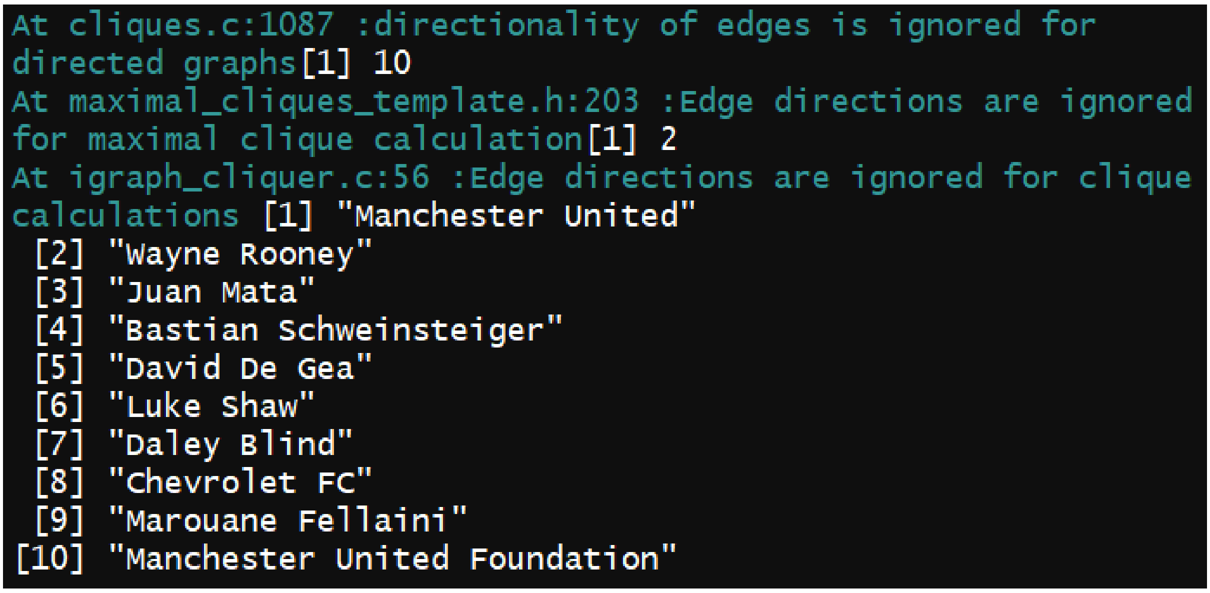
[Showing top 10 pages]

XXVII) Example of finding neighbors of page vertices

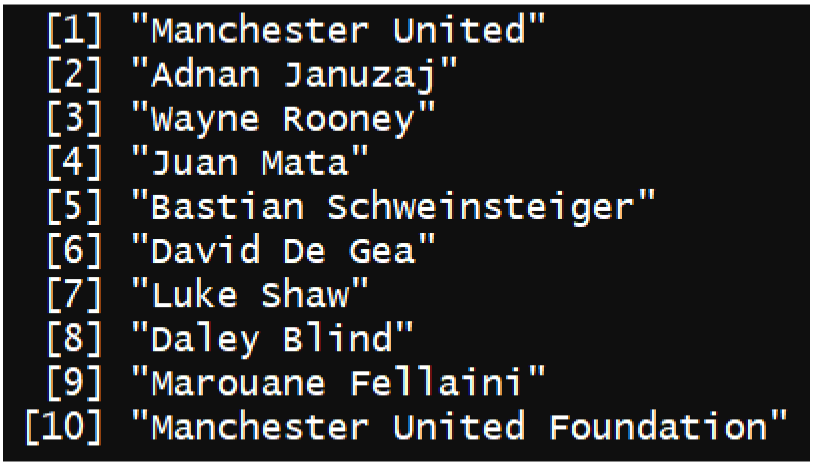


```
+ 19/582 vertices, from 6c9b30c:
  [1]  26 126 185 186 187 188 189 190 191 192 193 194 195 196
 [15] 197 198 199 200 201
  [1] "Barclays Football"
  [2] "The Emirates FA Cup"
  [3] "JÃ©rÃ©my Pied"
  [4] "Virgin Media"
  [5] "Under Armour (GB, IE)"
  [6] "Radhi JaÃ¯di"
  [7] "Oriol Romeu Vidal"
  [8] "NIX Communications Group"
  [9] "JosÃ© Fonte"
 [10] "OctaFX"
 [11] "Florin Gardos"
 [12] "Harrison Reed"
 [13] "Ryan Bertrand"
 [14] "Garmin"
 [15] "James Ward-Prowse"
 [16] "Benali's Big Race"
 [17] "Southampton Solent University - Official"
 [18] "Sparsholt Football Academy"
 [19] "Saints Foundation"
```

XXVIII) Getting communities / clusters

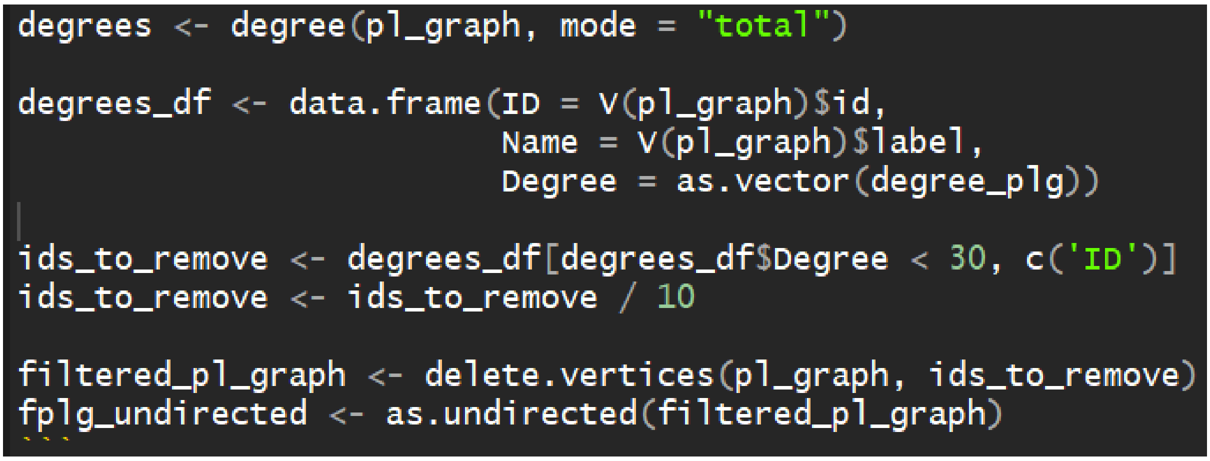


```
At cliques.c:1087 :directionality of edges is ignored for
directed graphs[1] 10
At maximal_cliques_template.h:203 :Edge directions are ignored
for maximal clique calculation[1] 2
At igraph_cliquer.c:56 :Edge directions are ignored for clique
calculations [1] "Manchester United"
 [2] "Wayne Rooney"
 [3] "Juan Mata"
 [4] "Bastian Schweinsteiger"
 [5] "David De Gea"
 [6] "Luke Shaw"
 [7] "Daley Blind"
 [8] "Chevrolet FC"
 [9] "Marouane Fellaini"
[10] "Manchester United Foundation"
```



```
 [1] "Manchester United"
 [2] "Adnan Januzaj"
 [3] "Wayne Rooney"
 [4] "Juan Mata"
 [5] "Bastian Schweinsteiger"
 [6] "David De Gea"
 [7] "Luke Shaw"
 [8] "Daley Blind"
 [9] "Marouane Fellaini"
[10] "Manchester United Foundation"
```

XXIX) Filtering graph to get important nodes based on degree



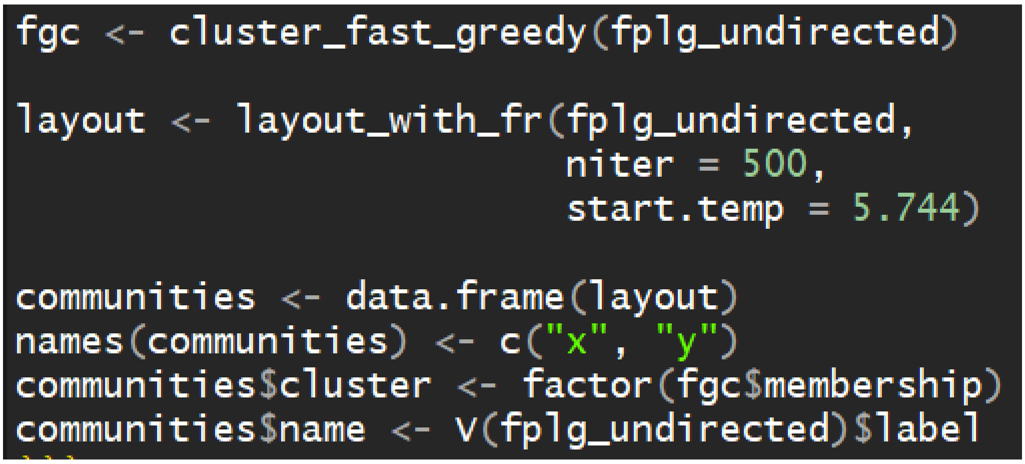
```
degrees <- degree(pl_graph, mode = "total")

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)
```

XXX) Fast greedy clustering

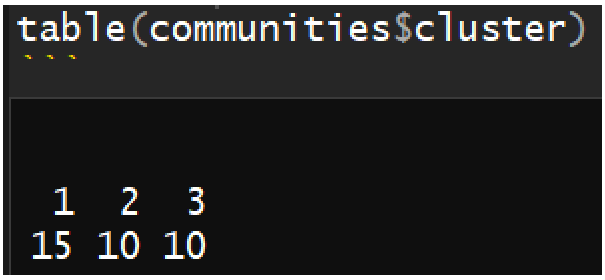


```
fgc <- cluster_fast_greedy(fplg_undirected)

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
```

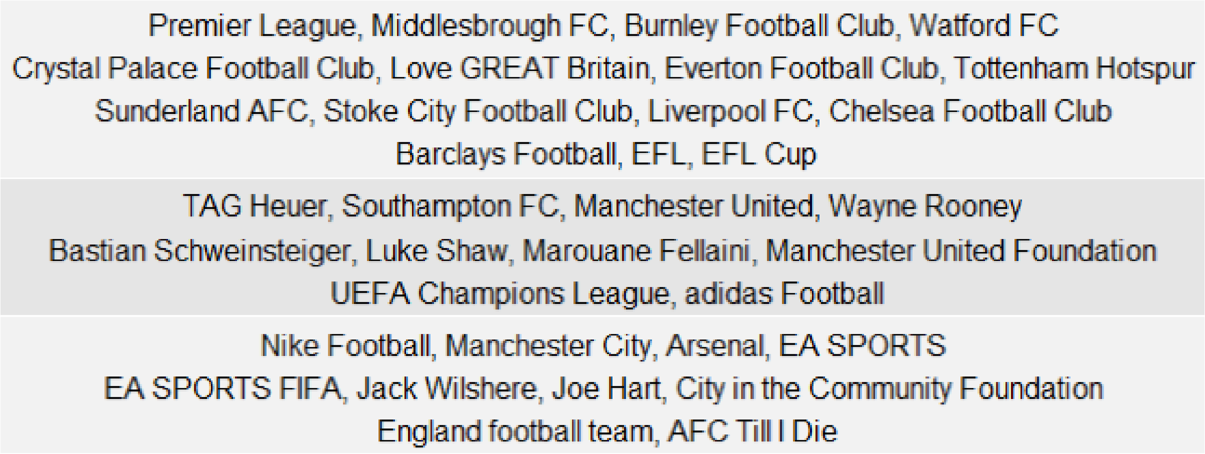
XXXI) Fast greedy clustering → total pages in each cluster



```
table(communities$cluster)
```

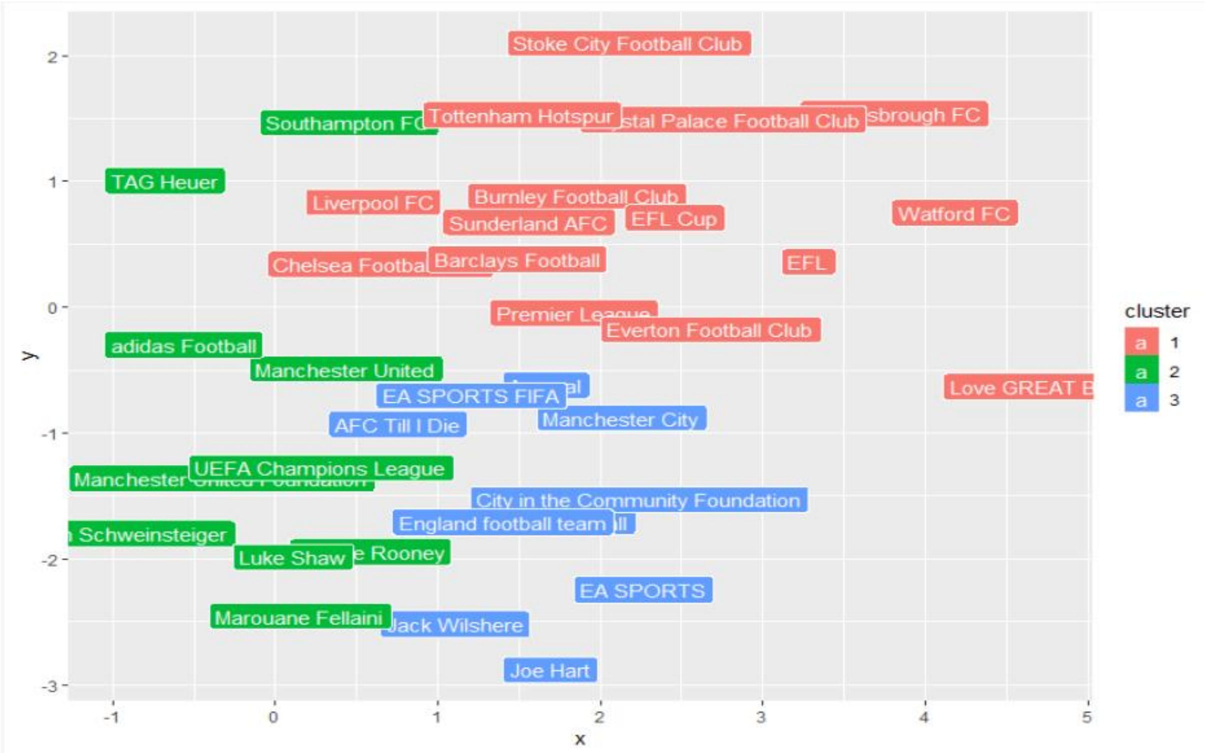
```
 1  2  3
15 10 10
```

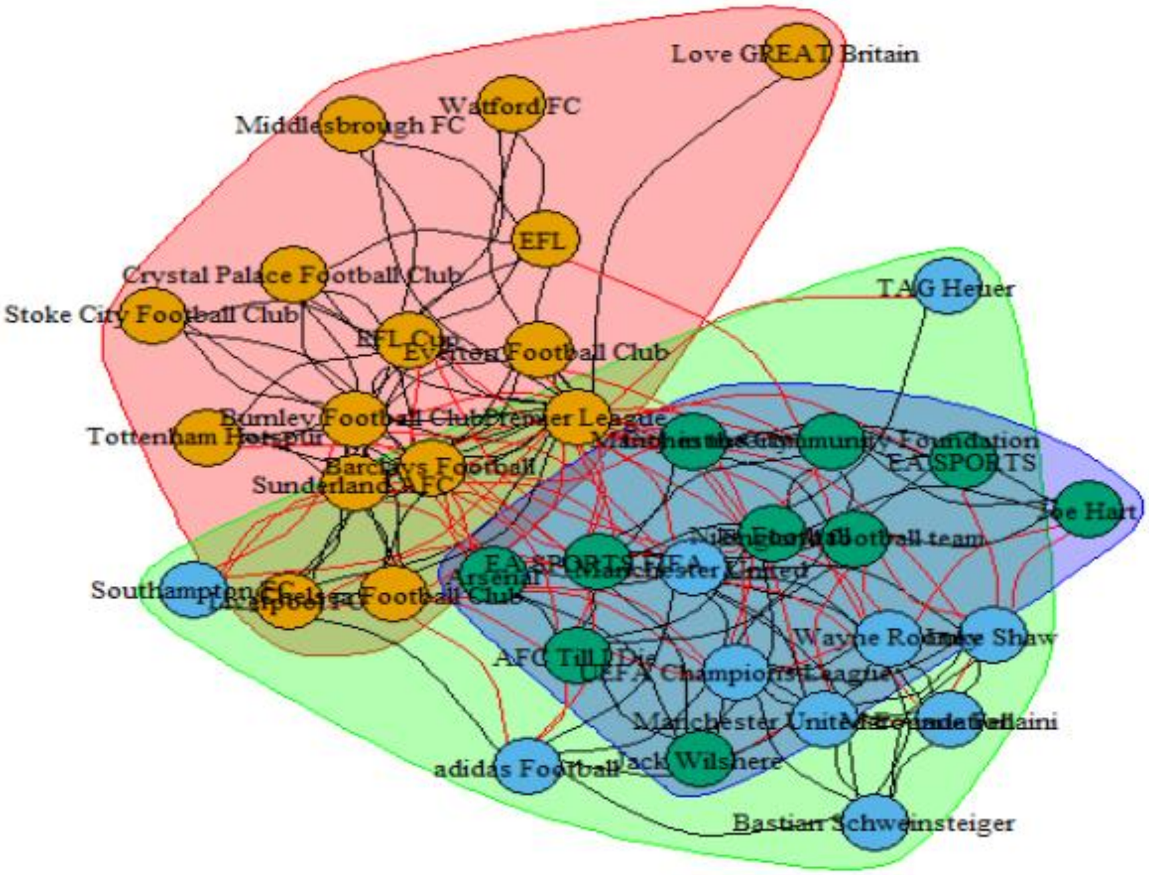
XXXII) Get page names in each cluster



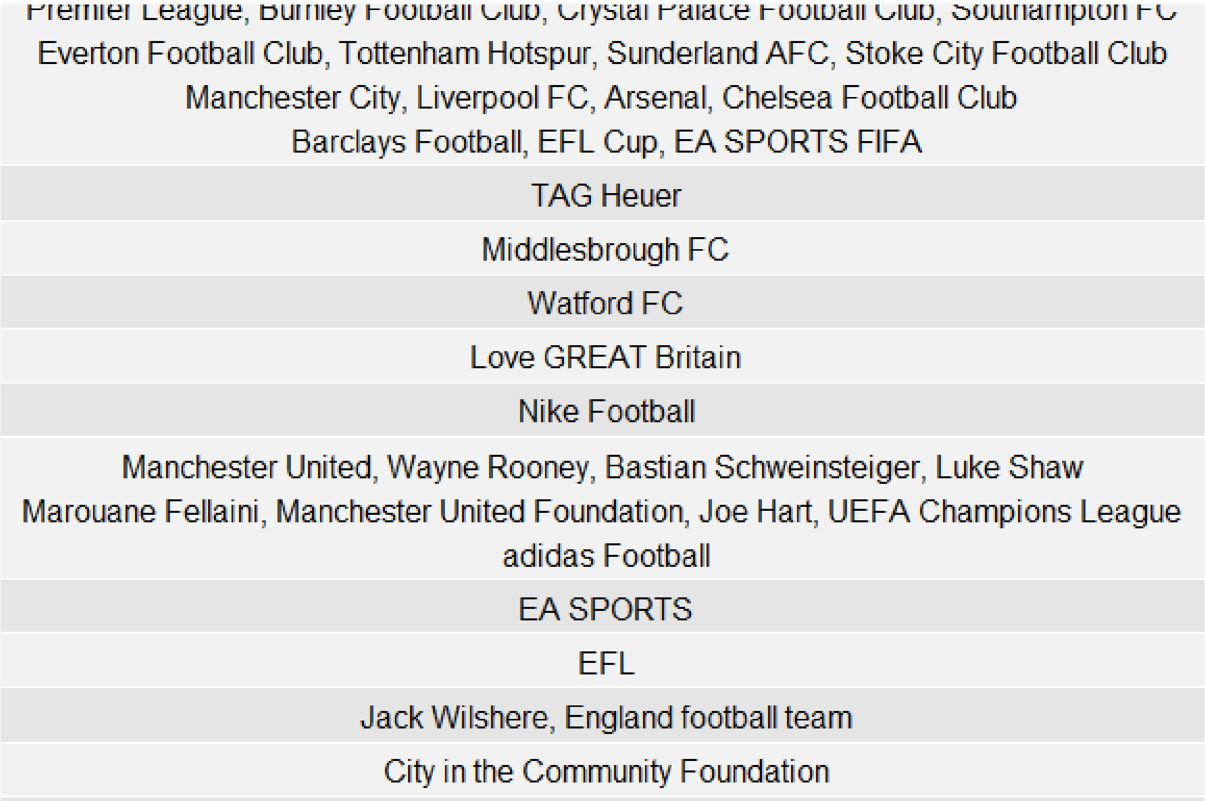
Premier League, Middlesbrough FC, Burnley Football Club, Watford FC Crystal Palace Football Club, Love GREAT Britain, Everton Football Club, Tottenham Hotspur Sunderland AFC, Stoke City Football Club, Liverpool FC, Chelsea Football Club Barclays Football, EFL, EFL Cup
TAG Heuer, Southampton FC, Manchester United, Wayne Rooney Bastian Schweinsteiger, Luke Shaw, Marouane Fellaini, Manchester United Foundation UEFA Champions League, adidas Football
Nike Football, Manchester City, Arsenal, EA SPORTS EA SPORTS FIFA, Jack Wilshere, Joe Hart, City in the Community Foundation England football team, AFC Till I Die

XXXIII) Modularity score

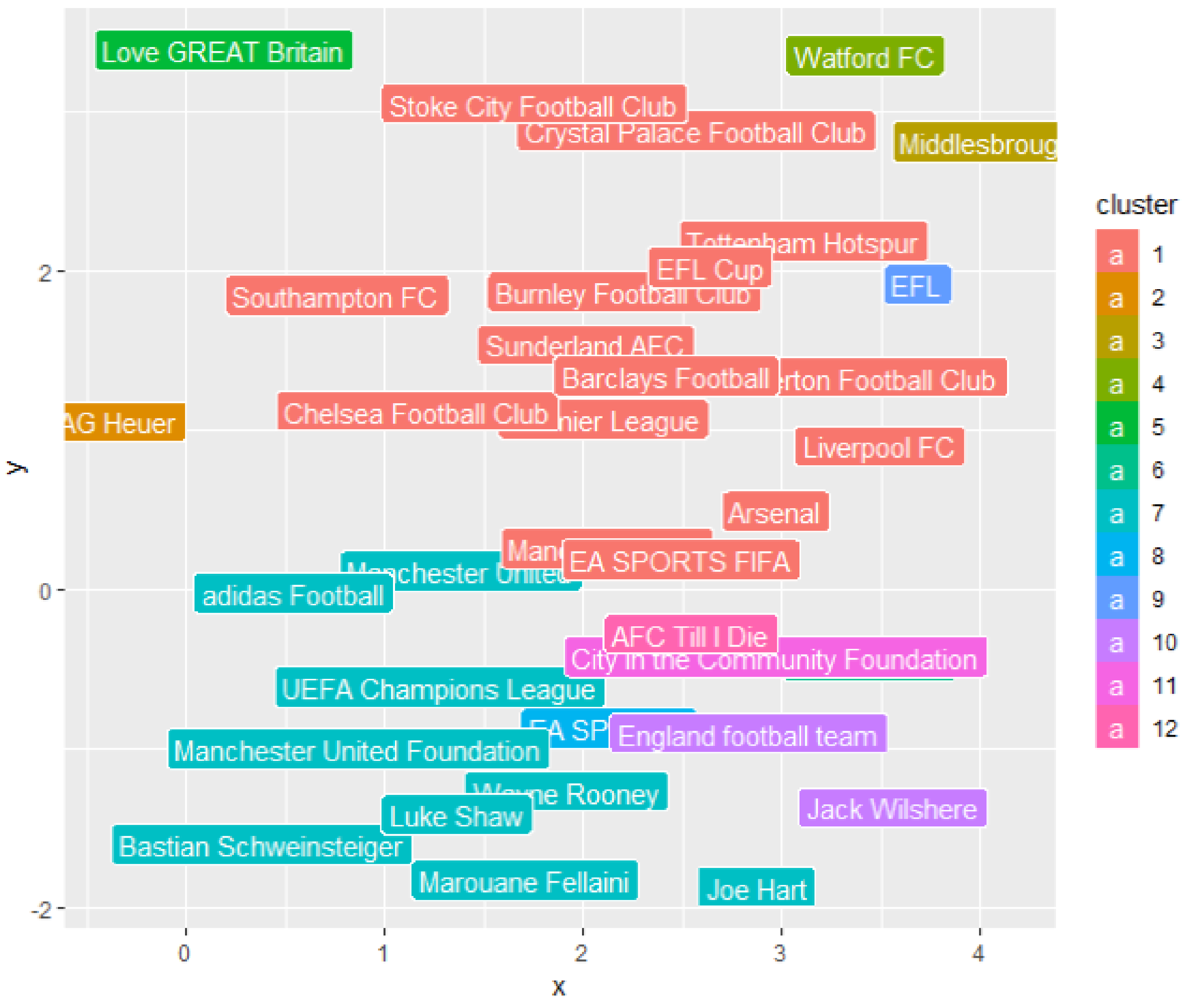

Stoke City Football Club
Southampton FC
Tottenham Hotspur
stal Palace Football Club
sbrough FC
TAG Heuer
Liverpool FC
Burnley Football Club
Sunderland AFC
EFL Cup
Watford FC
Chelsea Footbal
Barclays Football
EFL
Premier League
Everton Football Club
adidas Football
Manchester United
EA SPORTS FIFA
Love GREAT B
AFC Till I Die
Manchester City
Manchester
UEFA Champions League
City in the Community Foundation
England football team
Schweinsteiger
Luke Shaw
e Rooney
EA SPORTS
Marouane Fellaini
Jack Wilshere
Joe Hart
x
y
2
1
0
-1
-2
-3
-1
0
1
2
3
4
5
cluster
a 1
a 2
a 3

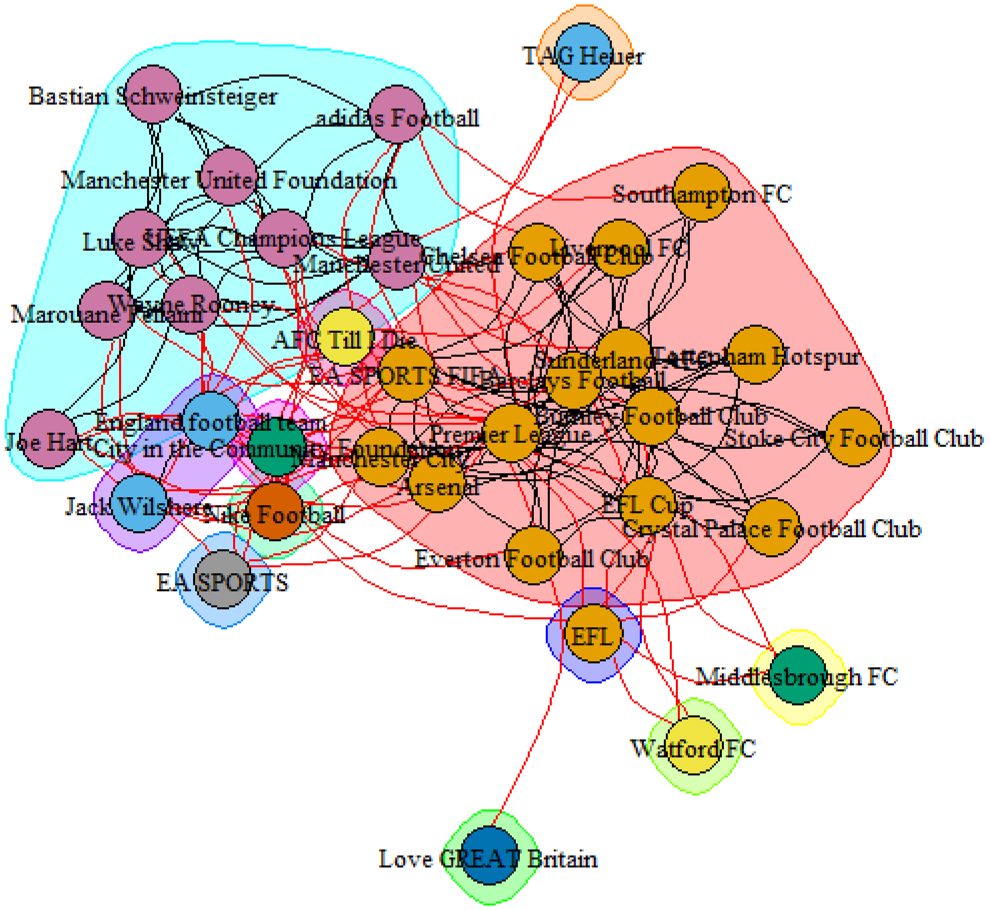

Love GREAT Britain
Watford FC
Middlesbrough FC
EFL
Crystal Palace Football Club
Stoke City Football Club
TAG Heuer
EFL Cup
Everton Football Club
Burnley Football Club
Premier League
Tottenham Hotspur
City in the Community Foundation
EA SPORTS
Barclays Football
Sunderland AFC
Joe Hart
England football team
EA SPORTS FIFA
Manchester United
Arsenal
Southampton FC
Chelsea Football Club
Liverpool FC
Wayne Rooney
Luke Shaw
AFC Till I Die
UEFA Champions League
Manchester United Foundation
Marouane Fellaini
adidas Football
Jack Wilshere
Bastian Schweinsteiger

XXXIV) Edge betweenness clustering



Premier League, Burnley Football Club, Crystal Palace Football Club, Southampton FC Everton Football Club, Tottenham Hotspur, Sunderland AFC, Stoke City Football Club Manchester City, Liverpool FC, Arsenal, Chelsea Football Club Barclays Football, EFL Cup, EA SPORTS FIFA
TAG Heuer
Middlesbrough FC
Watford FC
Love GREAT Britain
Nike Football
Manchester United, Wayne Rooney, Bastian Schweinsteiger, Luke Shaw Marouane Fellaini, Manchester United Foundation, Joe Hart, UEFA Champions League adidas Football
EA SPORTS
EFL
Jack Wilshere, England football team
City in the Community Foundation


Love GREAT Britain
Watford FC
Stoke City Football Club
Crystal Palace Football Club
Middlesboroug
Tottenham Hotspur
EFL Cup
EFL
Southampton FC
Burnley Football Club
Sunderland AFC
Barclays Football
rton Football Club
AG Heuer
Chelsea Football Club
nier League
Liverpool FC
Arsenal
Manc
EA SPORTS FIFA
Manchester United
adidas Football
AFC Till I Die
City in the Community Foundation
UEFA Champions League
EA SP
England football team
Manchester United Foundation
Wayne Rooney
Luke Shaw
Jack Wilshere
Bastian Schweinsteiger
Marouane Fellaini
Joe Hart
y
2
0
-2
x
0
1
2
3
4
cluster
a 1
a 2
a 3
a 4
a 5
a 6
a 7
a 8
a 9
a 10
a 11
a 12


TAG Heuer
Bastian Schweinsteiger
adidas Football
Manchester United Foundation
Southampton FC
Luke Shaw
UEFA Champions League
Manchester United
Chelsea Football Club
Liverpool FC
Marouane Fellaini
Wayne Rooney
AFC Till I Die
EA SPORTS FIFA
Sunderland AFC
Tottenham Hotspur
Barclays Football
Burnley Football Club
England football team
Joe Hart
City in the Community Foundation
Manchester City
Premier League
Stoke City Football Club
Arsenal
Jack Wilshere
Nike Football
EFL Cup
Crystal Palace Football Club
Everton Football Club
EA SPORTS
EFL
Middlesbrough FC
Watford FC
Love GREAT Britain

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 (Google)

4) www.github.com (Github)

5) 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)
```
~~~