Introduction:
GitHub is one of the most popular open source project hosting service, it also introduced the social functions to help the programmers to develop projects together.

Data Description:
In this example, I choose to study the social network of a group of top developers on a specific programming language. 22 individuals who specialise in programming using Objective-C are identified. Since they may work on same projects, we consider there exist some social links between them, thus we put them together as a group.

Note: Github also provides “following” function, however, I think it is not a good idea to analyse the links generated by following/followers since the correlation is to low since the nature of the social network of Github is “project based”.

Relational Measure Definition:
We then define 2 kinds of measures for the group of people.

1. **Starring others’ project**: if A stars a project that B has contributions to, there will be a *directed* edge from A to B.

2. **Co-working on a project**: if A and B both have contribution to a given project, there will be an *undirected* edge between A and B

Example matrix for starring links:

	steipete	*0xced*	*rs*
steipete	-	36	3
0xced	26	-	5
rs	4	4	-

Example matrix for co-working links:

	steipete	*0xced*	*rs*
steipete	-	8	3
0xced	8	-	3
rs	3	3	-

The number represents the weight of the edge, which is calculated by the #links from A to B.

Data Collection:
In this case, top developers who specialise in Objective-C are identified by parsing the list of trending developers ranked by Github daily, weekly, or monthly (probably measured by the popularity of their projects) and union the results and filter out non-human users (such as Facebook, Google). We then have a set of users as our nodes.

With the Github API, we can easily access to the list of projects a user starred. We can build up the links easily since the data is literally public accessible.
However, GitHub API does not provide something like “a list of projects a user has contribution to”, so we turn to build a set of projects that contains all projects of each users, all projects they have ever starred, and all projects of the organisations they work in.

We then collected:

- 23 users
- 342 starring links
- 103 co-working links

for trending users of Objective-C on 2014/11/08.

Visualisation and Analysis:

Tool Selection
The tool we choose to visualise the network data is *d3.js (data-driven documents)*. The reason for why we choose this tool is that it provides a rich set of templates and it is flexible since we could control the output by tuning the parameters in the javascript. Also, it renders the graph elegantly, and provides the functionality of interactive animations which is useful if we want to manipulate the data and visualise it real-time.

Data Visualisation

Fig.1 shows the visualisation of the network using the first measure “starring links”. The node size represents degree of the node, and the stroke width of the edges represent the link strength between two nodes.
We found that the user: *0xced* and *steipete* received a lot of stars on the project they worked on. We then check their Github profiles and observe the projects they worked on, and found some differences between these two users. For 0xced, he tends to enjoy contributing to others’ projects thus he received a lot of star on the project he has contribution to but not owned by him. On the other hand, steipete has a lot of highly starred projects started by him.

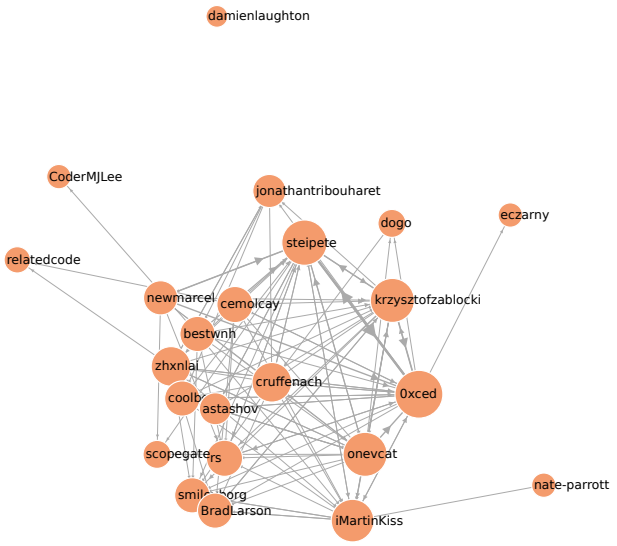

Fig. 1 - Starring Links

Fig. 2 shows the visualisation of the network using the second measure “co-working links”. The node size and the edge stroke width represent exactly the same as the previous graph.

We found that some of the users never collaborate with others.
Take *CoderMJLee* and *nate-parrot* as examples, we found that they are both new stars in this area (might be classified as daily trending developers), so they might not have the experiences working with others.

For other developers who are the “gurus” in this field, they are tied together strongly since they would like to contribute to others’ projects. Which shows the social coding scheme provided by GitHub works really great for experienced users. Another interesting point is that there’re no significant size differences between users. Suggesting the developers would rather make contributions than just “watch” it.

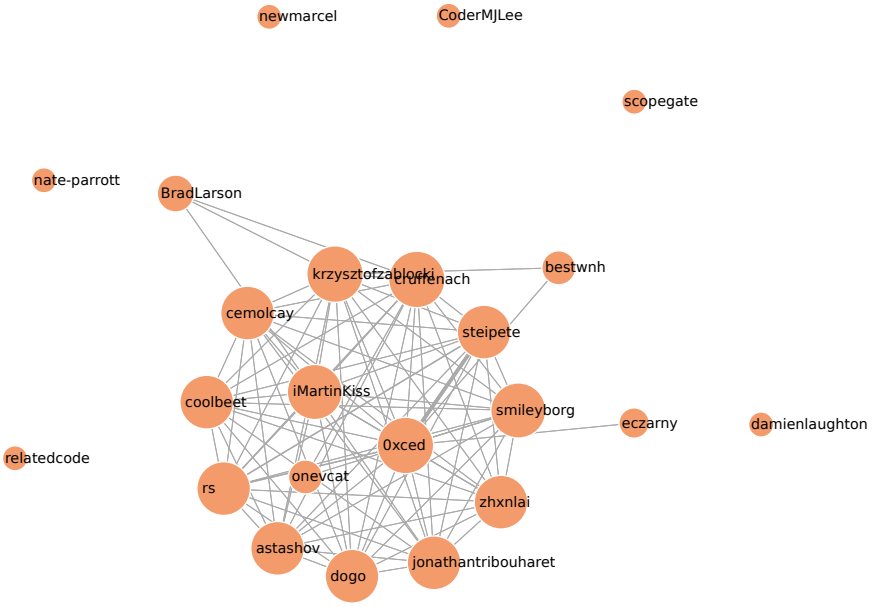

Fig. 2 - Co-working Links

Conclusions:

In summary, we choose GitHub as the community as our target of analysis and found some interesting facts that may not be seen easily. However, we still doubt the correctness of the relations and our findings. By collecting the data from other domain (language) and scale up the sampling size may be more beneficial for validating our findings.

Another future work is that we can find the interactions of the users from two different domains, such as a group of users who specialise in Ruby and another group of users who specialise in Python and find the relationships and interactions between them. This might be helpful to identify the prosperity and violence of a language.

References:
GitHub: https://github.com/
d3.js: http://d3js.org/

Appendix:
All of the source code and data (stored in a sqlite database) for this assignment can be found in this repository: https://github.com/ChengYuHsu/sc-assignment-2