**Defining sets**

We define a set of developers, D; a set of reviews R; a set of approvals A; and a set of comments C. The members of the set are calculated as given below:

D = 678 unique developers who own at least one review that is in the MERGED status.

R = 3,038 reviews either in MERGED or ABANDONED status which are owned by members of D and which have been commented by its owner plus two other developers in D.

A = All the approvals on members of R, which are approved by members of D: how many?

C = All the comments on R, made by members of D: how many?

3.Generate DRCN for members of D, RSN for members of R, and DRON for members of D, as specified below.

DRCN

----

Generate a developer review communication network (DRCN) in Pajek

format whose vertices are developers (D)(each developer is uniquely

identified by a PersonID). Two developers are connected by an edge if

both have commented on at least one review. The weight of the edge is

the number of reviews both have commented on.

RSN

---

Generate Review Similarity Network(RSN) based on LDA method in pajek format

whose vertices are Reviews (R)(each Review is uniqely identified by a ReviewId)

.Two Review will be connected if the similarity value is higher or equal to the

similarity threshold 't' .Similarity threshold is measured using 95th percentile

of the KL value distribution.The wight will be KL value.

DRON

----

Generate Developer Review Ownership Network(DRON) based on LDA method in pajek format whose vertices are Developers (D)(each Developer is uniqely identified by a ReviewId).Two Developer will be connected if Review which they own is connected in RSN.The wight of the edge is the number of review both own and they have connection in RSN.

***All the following parameters should be calculated based on the members of sets D, R, A, C only.***

**Developer master file**

1. Person id (person.id)
2. Name (name)
3. Email (email)
4. IsBotAccount (is.bot)
5. Number of reviews owned by this person (noof.rev.owned)
6. Number of reviews owned by this person that are in status = merged (noof.rev.merged)
7. Number of reviews merged divided by number of reviews owned (merged.by.owned)
8. Number of reviews owned by this person that are in status = abandoned (noof.rev.abandoned)
9. Average elapsed time of all reviews owned by the person(average.elapsed.time)
10. Median elapsed time of all reviews owned by the person(median.elapsed.time)
11. Average elapsed time of all reviews owned by the person which are in merged status (average.elapsed.time.merged)
12. Median elapsed time of all reviews owned by the person which are in merged status (median.elapsed.time.merged)
13. Number of reviews commented upon by this person (noof.rev.commented)
14. Number of comments by this person across all reviews (noof.comments.by.dev)
15. Average number of comments per review by this person (avg.noof.comments.by.dev)
16. Number of comments by this person on reviews that she owns (noof.comments.owned)
17. Number of comments by this person on reviews that she does not own (noof.comments.nonowned)
18. Number of approvals by this person (noof.approvals)
19. Average approval score by this person (avg.approval.score.by.dev)
20. Median approval score by this person (median.approval.score.by.dev)
21. Time stamp of first comment (dev.comment.time.1)
22. Time stamp of last comment (dev.comment.time.2)
23. Comment span (comment.span)
24. Time stamp of first approval (approval.time.1)
25. Time stamp of last approval (approval.time.2)
26. Approval span (approval.span)
27. Congruence (congruence)
28. DRCN degree (drcn.degree)
29. DRCN Betweenness (drcn.betweenness)
30. DRCN clustering coefficient (drcn.clustcoeff)
31. DRCN closeness centrality (drcn.closenesscentrality)
32. DRCN eigenvector centrality (drcn.eigencentrality)
33. DRCN pagerank (drcn.pagerank)
34. DRON degree (dron.degree)
35. DRON Betweenness (dron.betweenness)
36. DRON clustering coefficient (dron.clustcoeff)
37. DRON closeness centrality (dron.closenesscentrality)
38. DRON eigenvector centrality (dron.eigencentrality)
39. DRON pagerank (dron.pagerank)

**Review master file**

1. ReviewId (review.id)
2. CreatedOn (created.on)
3. LastUpdatedOn (last.updated.on)
4. ElapsedTime (elapsed.time)
5. NumberOfPatches (noof.patches)
6. Project (project)
7. Branch (branch)
8. SubmitType (submit.type)
9. Status (status)
10. Number of approvals (noof.approvals)
11. Net approval score (Sum of Approval.ReviewedStatus field)
12. Average approval score (average.approval.score)
13. Median approval score (median.approval.score)
14. Number of comments (noof.comments.on.this.review)
15. Time stamp of first comment (comment.time.1)
16. Time stamp of last comment (comment.time.2)
17. Comment span (comment.span.on.this.review)
18. OwnerId (owner.id)
19. Number of total reviews owned by owner (reviews.owned.by.owner)
20. Number of total reviews owned by owner in merged status (reviews.owned.by.owner.merged)
21. Number of total approvals given by owner (approvals.by.owner)
22. Number of comments by owner on this review (noof.comments.by.owner)
23. Number of total comments by owner across all reviews (total.comments.owner)
24. Number of comments by non-owners on this review (noof.comments.by.nonowners)
25. Number of reviews of owner which have merged status (noof.reviews.merged.owner)
26. Average approval score of owner (avg.aprpoval.score.owner)
27. Median approval score of owner (median.approval.score.owner)
28. RSN degree (rsn.degree)
29. RSN Betweenness (rsn.betweenness)
30. RSN clustering coefficient (rsn.clustcoeff)
31. RSN closeness centrality (rsn.closenesscentrality)
32. RSN eigenvector centrality (rsn.eigencentrality)
33. RSN pagerank (rsn.pagerank)