Pseudocode

```
----------------------------------------------------------
Review Network Construction and Analysis
-----------------------------------------------------------
1. Initialize spark session, spark context and sql context
2. Read input json files.
3. create a separate list of elite and non elite users.
4. Initialize a new networkx graph.
5. for every row in the reviews data frame do:
    a. create a Node with user_id.
    b. set node type to elite if user is elite user else set user to regular user.
    c. create a node of business type.
    d. create review network by adding edges between user and business node

6. Separate node based on type - regular user, elite user or business.
7. create degree plot for review network with user nodes.
8. create degree plot for review network with business nodes.
9. create degree plot for review network with elite user nodes.
10. Find clustering coefficient for the review network.
11. Find the degree centrality for the review network.
12. Calculate eigen value for regular user network
13. Calculate eigen value for elite user network.
14. Calculate eigen value for all users network
15. Calculate degree centrality value for regular user network
16. Calculate degree centrality for elite user network.
17. Calculate degree centrality for all users network
18. Insert eigen value and degree centrality values in the dataframe
19. Plot graph of eigen value vs rating
```

```
------------------------------------------------------------------------------
Concrete Differences in Ratings of Elite vs Regular users
------------------------------------------------------------------------------
1. Retrieve set of all business ids
2. Create a sub graph of all regular user reviews. Node data is the (weight / degree)
3. Create a sub graph of all elite user reviews. Node data is the (weight / degree)
4. Calculate difference in rating for regular and elite users
5. Calculate difference in rating for all users and elite users.
6. Calculate difference in rating for regular and all users.
7. Plot graph of elite user ratings vs regular user ratings
8. Plot graph of elite user ratings vs all user ratings.
```

```
-------------------------------------------------------
Social Network Construction and Analysis
-------------------------------------------------------
1. Initialize spark session, spark context and sql context
2. Read input json files.
3. create a separate list of elite and non elite users.
4. Get a set of elite user ids
5. Filter out users that do not have any friends.
6. Get a list of friends for each user.
7. Initialize Node class with Data and Type os node.
8. Initialize a new networkx graph.
5. for every row in the user data frame do:
   a. create a Node with user_id.
   b. set node type to elite if user is elite user else set user to regular user.
   c. for every friend in the user's friend list do:
      i. if the friend is in list of user ids
         a. create a node with friends user id and set type of user for the node
         b. add an edge in the graph between the two nodes.
   d. repeat for all users in the list of users with friends.
6. Compute the number of nodes and eadges in the graph
7. create degree plot for social network with all user nodes.
8. create degree plot for social network with regular user nodes only.
9. create degree plot for social network with elite user nodes only.
10. Find the connected component subgraphs for the cluster.
11. Find clustering coefficient for the social network.
12. Find the degree centrality for the social network.
13. Calculate eigen vector centrality for the complete network
14. Calculate eigen value for elite user network only.
15. Calculate eigen value for all users network only.
16. Calculate degree centrality value for regular user network only
17. Calculate degree centrality for elite user network only.
18. Calculate degree centrality for all users network
18. Insert eigen value and degree centrality values in the dataframe
19. Plot graphs of eigen value centralities.
20. Plot graphs of degree centralities.
```

Robustness Analysis

```
1. Initialize graphs of elite users, non elite users, and all users.
2. For every user in the data frame
   a. get friend list and elite status
   b. if friend list is not empty do
      i. for every friend in the list
         a. add an edge from u to v
3. For the graph of elite users
   a. calculate 25% of the number of nodes
   b. compute the connected component subgraphs.
   c. find the max connect component subgraph. append to list of subgraphs.
   d. for the range from 0 to (25% of number of nodes) do:
      i. remove a user form list of users
      ii. remove corresponding node from the graph
4. save the list of subgraphs.
5. Repeat steps 3 to 4 for non elite user only graph.
6. For all users graph do
   a. calculate 25% of the number of elite nodes
   b. compute the connected component subgraphs.
   c. find the max connect component subgraph. append to list of subgraphs.
   d. for the range from 0 to (25% of number of elite nodes) do:
      i. remove elite user from list of users
      ii. remove corresponding node from the graph
7. For all the graphs computed above do:
   a. find the size of the max connected components for each each iteration
   b. store the sizes corresponding to the graph
8. For all the graphs computed above do:
   a. plot a graph of the number of nodes removed at each stage vs the size of teh max connected component.
   c. we see the size of the max connected component decreasing for the graph with all users where the elite users are removed in parts.
```

Score for influence

1. Read the data into dataframes
2.Take the users dataframe create a new column number of friends
3. Create a target column with Boolean values of whether the user is an elite user or not
4. Use RandomForestClassifier to create a classifier
5.Do 10 fold cross validation
6.Print features and their weights

7. Create a new column called score with hold dot product of features and their weights
8. Plot all the users score
9. Plot elite users score
10. Create a new column and take log of the score column values into it
11. Plot all the users score
12. Plot the elite users score
13. Use the trained model to predict probability of non elite user becoming elite users
14. Take only the user whose probability is greater than 80% and not equal to 1
15. Count number of users.