1. First, consider the social and task ties as separate networks.

(A) Use igraph to generate indegree, outdegree, closeness, betweenness, and PageRank centrality statistics for each individual the social and task networks.

Social:

```
> indegree_social
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
 3  1  0  1  3  3  1  1  1  2  2  3  0  0  1  5  4  7  5  3  5  6

> outdegree_social
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
 3  1  0  1  3  3  1  1  1  2  1  3  0  0  1  5  4  6  6  3  4  8

> betweenness_social
          1           2           3           4           5           6           7           8           9          10          11 
 24.0000000   0.0000000   0.0000000   0.0000000  24.0000000  28.0000000   0.0000000   0.0000000   0.0000000  28.0000000  15.0000000 
         12          13          14          15          16          17          18          19          20          21          22 
 52.0000000   0.0000000   0.0000000   0.0000000  45.8333333   0.8333333  33.0000000  14.7500000   0.2500000  13.5000000 126.8333333 

> closeness_social
          1           2           3           4           5           6           7           8           9          10          11 
0.006756757 0.006535948 0.002164502 0.002267574 0.006756757 0.006250000 0.005617978 0.002267574 0.005714286 0.006134969 0.002267574 
         12          13          14          15          16          17          18          19          20          21          22 
0.006666667 0.002164502 0.002164502 0.002267574 0.007042254 0.006578947 0.006756757 0.006993007 0.006535948 0.006896552 0.007246377 

> pageRank_social
$`vector`
          1           2           3           4           5           6           7           8           9          10          11 
0.041770956 0.015776938 0.007712082 0.051413882 0.041770956 0.049972638 0.026310414 0.051413882 0.021870996 0.043760780 0.080476425 
         12          13          14          15          16          17          18          19          20          21          22 
0.048299457 0.007712082 0.007712082 0.076117044 0.060409878 0.050543283 0.082724924 0.059917284 0.040273604 0.058135886 0.075904525 
```

Task:

```
> indegree_task
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 
 1  1  0  1  1  1  1  1  1  1  1  0  2  1  1  2  3  4  3  2  3 17 
> outdegree_task
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 
 1  1  0  1  1  1  1  1  1  1  1  0  2  1  1  2  3  4  3  2  3 17 
> betweenness_task
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22 
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   1   0   0 257 
> closeness_task
          1           2           3           4           5           6           7           8           9          10          11 
0.008264463 0.008264463 0.002164502 0.002267574 0.008264463 0.008264463 0.008264463 0.002267574 0.008264463 0.008264463 0.008264463 
         12          13          14          15          16          17          18          19          20          21          22 
0.002164502 0.008333333 0.008264463 0.008264463 0.008333333 0.008403361 0.008474576 0.008403361 0.008333333 0.008403361 0.009523810 
> pageRank_task
$`vector`
          1           2           3           4           5           6           7           8           9          10          11 
0.022987269 0.022987269 0.007389163 0.049261084 0.022987269 0.022987269 0.022987269 0.049261084 0.022987269 0.022987269 0.022987269 
         12          13          14          15          16          17          18          19          20          21          22 
0.007389163 0.037580433 0.022987269 0.022987269 0.038858829 0.052437813 0.068673713 0.056017273 0.038858829 0.052437813 0.311962119 
```

(B) Compute the correlations of the five centrality measures you generate for the social network with the five measures generated for the task network. Which measures in the task network are most closely related to those in the socializing network? Name at least one insight can you draw from the relationships between these five measures across the two networks.

Betweenness has highest correlation. Page Rank has lowest corelation

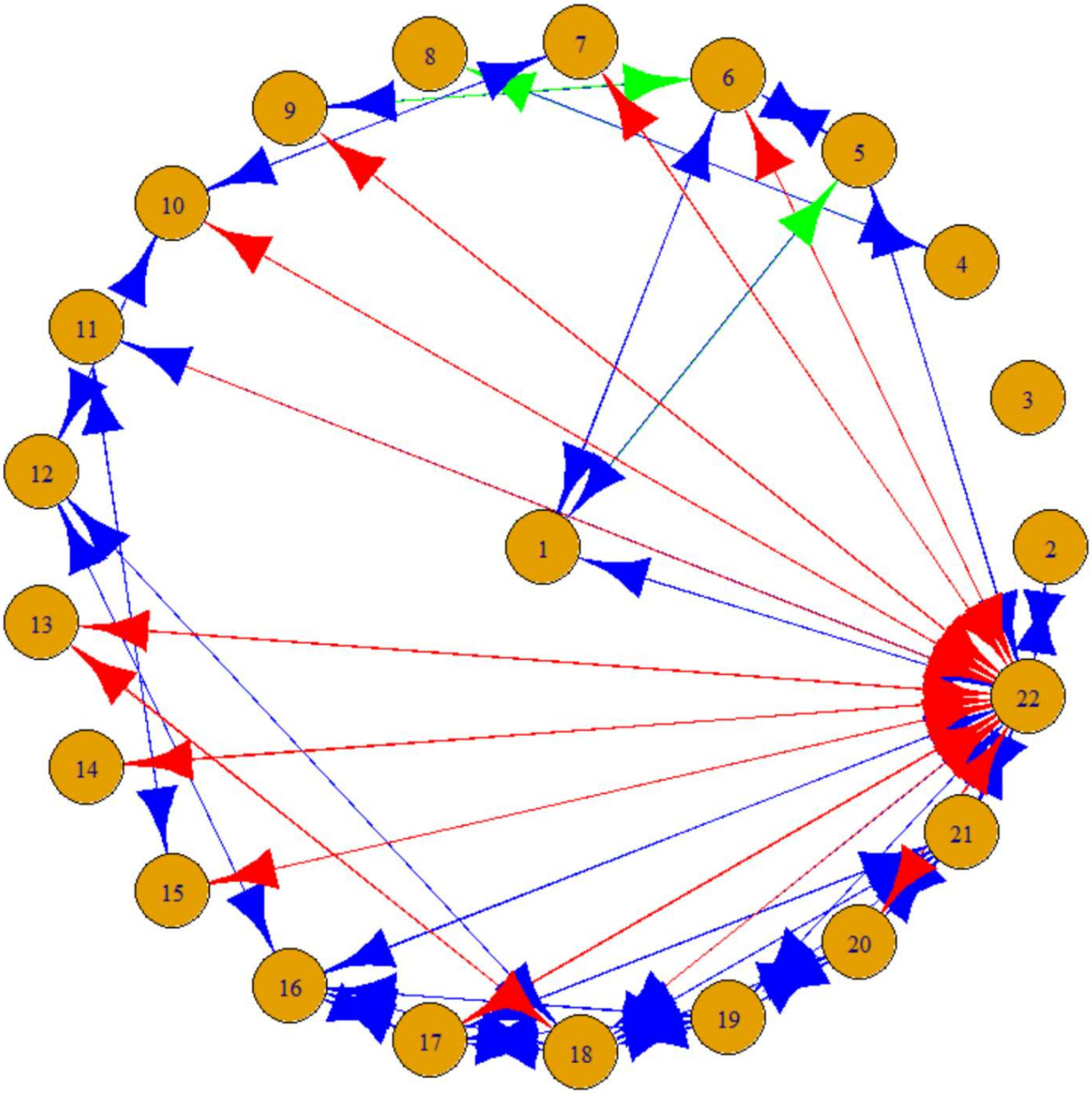
	measures	Corelations
4	Betweenness	0.8318598
2	Outdegree	0.6996636
1	Indgree	0.5578869
3	Closeness	0.4434346
5	Page Rank	0.3975701

After combing two graphs in one graph the final graph we get is as below.

Blue = Social Tie

Red = Task Tie

Green = Both social and task tie



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

I have considered **undirected graphs for answers below**. Logic of creating edge weights while combining the edges is as below. For both social and task tie, Maximum value from the in or out is considered as the weight of the edge.

```
ego <- dataset[row, "ego"]
alter  <- dataset[row, "alter"]
social_value <- dataset[row, "social_tie"]
task_value <- dataset[row, "task_tie"]

if(social_value != 0 || task_value != 0){

  rev_relation <- dataset[which((dataset$ego == alter) & (dataset$alter == ego)), ]
  rev_social_value <- rev_relation[,"social_tie"]
  rev_task_value <- rev_relation[,"task_tie"]

  debug <- FALSE
  if(ego == 21 & alter == 19){
    print('found')
    debug = TRUE
    print(rev_relation)
  }

  if(rev_social_value > social_value){
    social_value <- rev_social_value
  }
  if(rev_task_value > task_value){
    task_value <- rev_task_value
  }

  is_reverse_present <- FALSE
  for(value in which(egos==alter)){
    if(alters[value] == ego){
      print("Reverse")
      print(ego)
      print(alter)
      is_reverse_present <- TRUE
      reverse_present[value] <- "YES"
    }
  }
  if(!is_reverse_present)
  {
    egos <- c(egos, ego)
    alters <- c(alters, alter)
    social_values <- c(social_values, social_value)
    task_values <- c(task_values, task_value)

    reverse_present <- c(reverse_present, "NO")
  }
}
```

2. Next, consider the social and task ties together, as two distinct types of ties comprising one network.

For the sake of simplicity, I have cleaned the dataframe to have values of tie maximum if reverse relation exists. Below is the cleaned data frame.

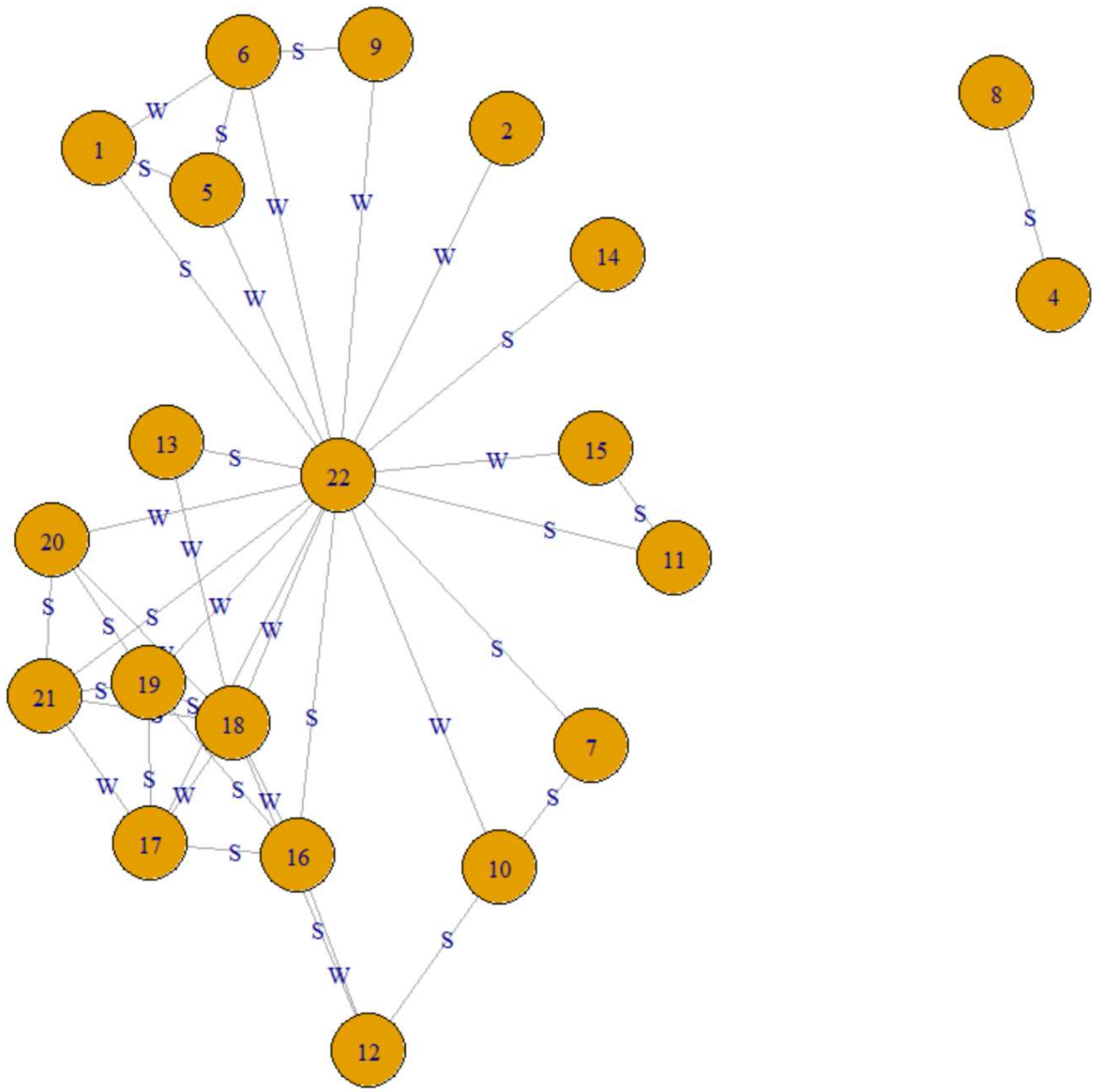
(A) Suppose that a tie is strong if it is above the mean strength for that type, conditional on the tie existing—do not include weights of 0 in the calculation of the mean. Under this definition, does the network satisfy Strong Triadic Closure? Come up with a solution that illustrates this (1) visually, in a plot, as well as (2) programmatically, by giving the number or proportion of ties that are violation of Strong Triadic Closure.

(B) Now suppose that a tie is strong if it is above the median strength for that type, conditional on the tie existing. Under this definition, does the network satisfy Strong Triadic Closure? What insights does this illustrate about these interactions within the network?

```
> (mean_social_tie <- mean(clean_undirected_graph_df[which(clean_undirected_graph_df$social_values !=0), 'social_values']))
[1] 1.825

> (mean_task_tie <- mean(clean_undirected_graph_df[which(clean_undirected_graph_df$task_values !=0), 'task_values']))
[1] 3.453125

> (median_social_tie <- median(clean_undirected_graph_df[which(clean_undirected_graph_df$social_values !=0), 'social_values']))
[1] 1.125

> (median_task_tie <- median(clean_undirected_graph_df[which(clean_undirected_graph_df$task_values !=0), 'task_values']))
[1] 1.875
```

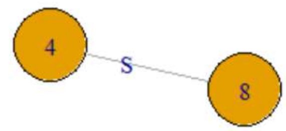
	egos	alters	social_values	task_values	ties_strength_mean	ties_strength_median	reverse_present
1	1	5	5.625	0.000	S	S	YES
2	1	6	1.500	0.000	W	S	YES
3	1	22	1.875	11.250	S	S	YES
4	2	22	0.375	2.250	W	W	YES
5	4	8	1.875	0.750	S	S	YES
6	5	6	1.500	0.000	S	S	YES
7	5	22	0.750	7.125	W	S	YES
8	6	9	0.375	0.000	S	S	YES
9	6	22	0.000	5.250	W	W	YES
10	7	10	2.250	0.000	S	S	YES
11	7	22	0.000	1.500	S	S	YES
12	9	22	0.000	2.250	W	S	YES
13	10	12	0.750	0.000	S	S	YES
14	10	22	0.000	1.125	W	W	YES
15	11	15	3.000	0.000	S	S	YES
16	11	22	0.375	2.625	S	S	YES
17	12	16	0.375	0.000	W	S	YES
18	12	18	0.375	0.000	S	S	YES
19	13	18	0.000	0.750	W	W	YES
20	13	22	0.000	1.500	S	S	YES
21	14	22	0.000	0.750	S	S	YES
22	15	22	0.000	2.250	W	S	YES
23	16	17	0.750	0.000	S	S	YES
24	16	18	0.375	0.000	W	W	YES
25	16	19	5.250	1.125	S	S	YES
26	16	22	0.750	10.500	S	S	YES
27	17	18	1.125	0.375	W	S	YES
28	17	19	1.125	0.000	S	S	YES
29	17	21	0.375	0.375	W	W	YES
30	17	22	0.000	1.125	S	S	YES
31	18	19	2.250	0.000	S	S	YES
32	18	20	1.125	0.000	W	S	YES
33	18	21	14.625	1.125	S	S	YES
34	18	22	0.375	3.375	W	W	YES
35	19	20	1.875	0.375	S	S	YES
36	19	21	0.375	0.000	S	S	NO
37	19	22	1.125	10.125	W	S	YES
38	20	21	0.750	0.000	S	S	YES
39	20	22	0.000	3.375	W	W	YES
40	21	22	1.500	11.625	S	S	YES

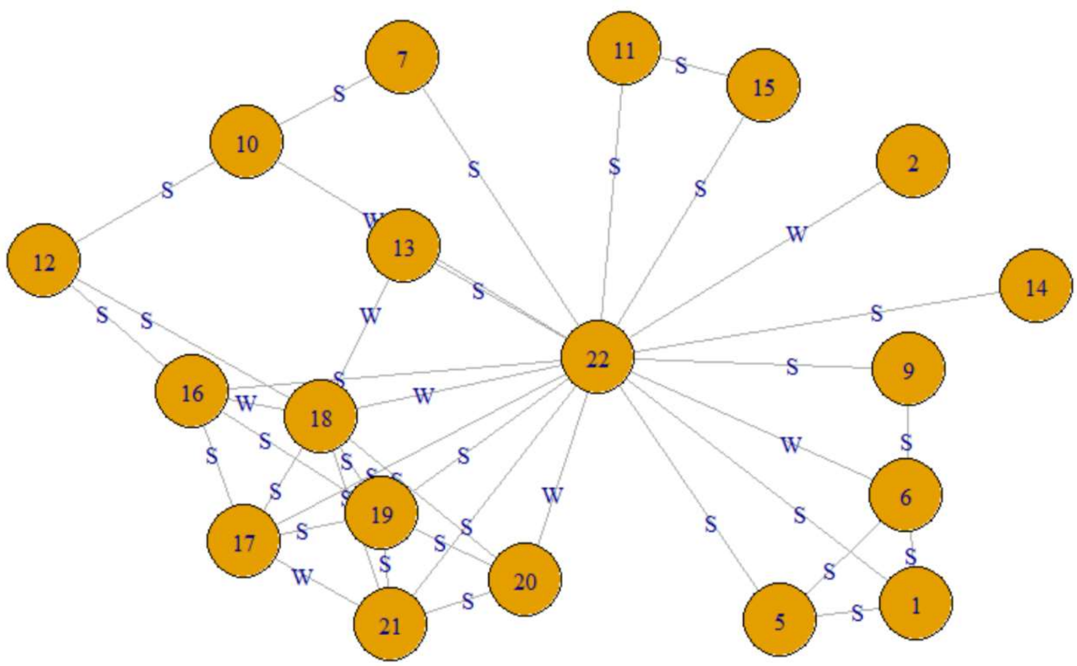
USING MEAN AS CUTOFF



Some of the violations of strong triads are triangles = 22-7-14, 22 – 16 - 1

USING MEDIAN AS CUTOFF





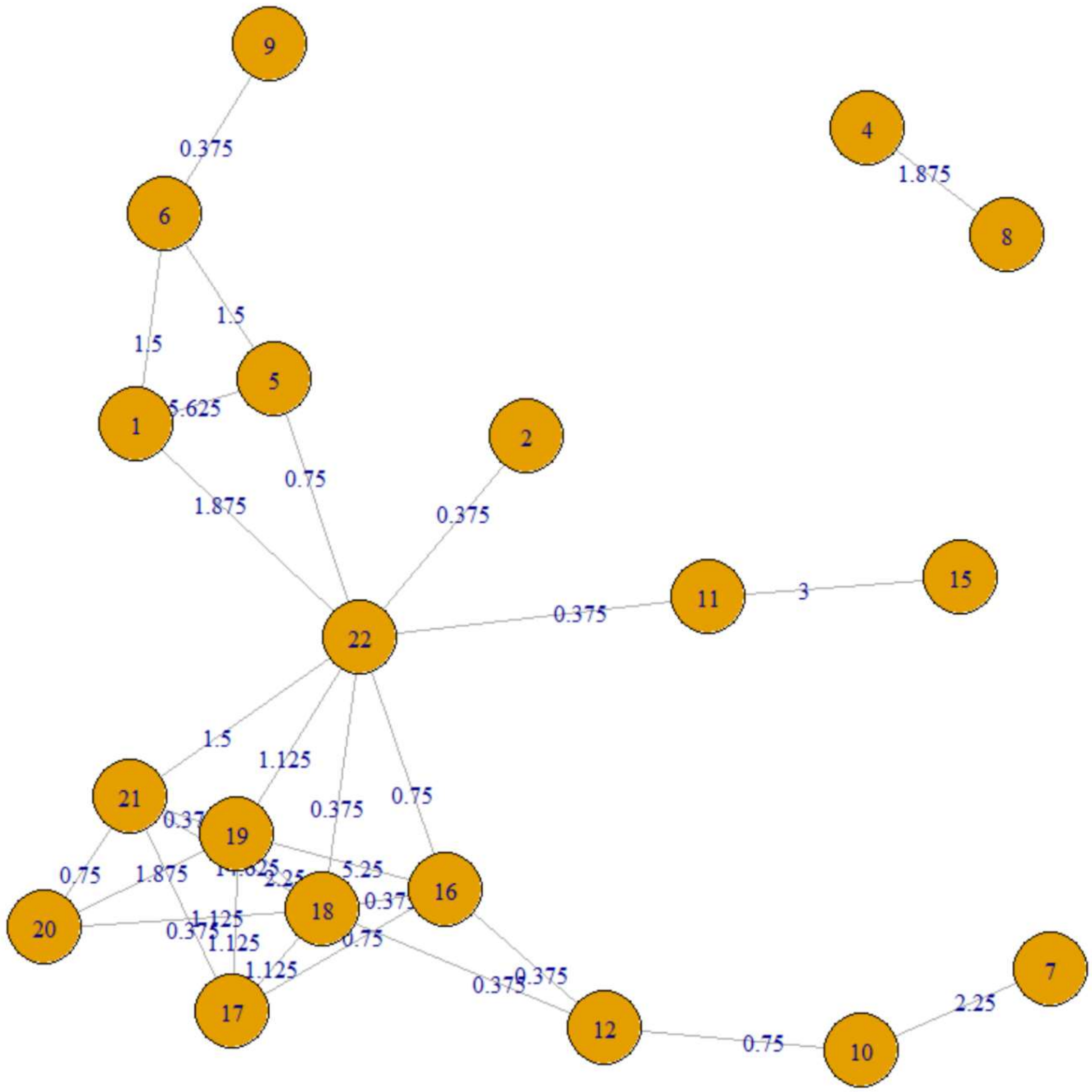
Some of the violations of strong triads are triangles = 22-11-5, 22 – 7 - 11

3. Continue to treat the social and task ties as two distinct ties comprising one network.

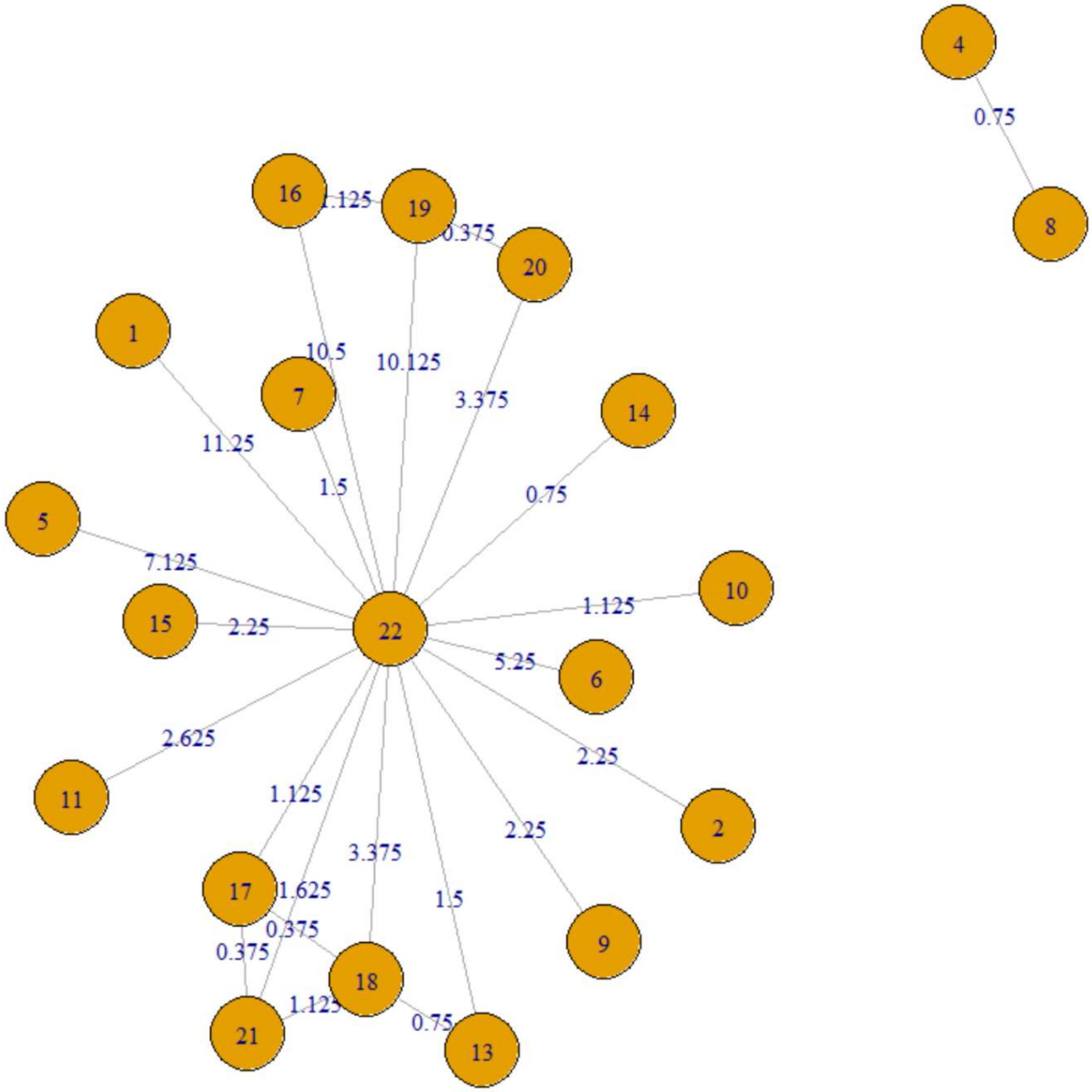
(A) It is also possible to compute betweenness on the edges in a network, as well as the vertices. This is a good measure of the flow of information and resources through a network. Calculate the edge-level betweenness for both of the types of tie.

Yes it is possible to calculate edge level and vertex level betweenness.

SOCIAL UNDIRECTED GRAPH



TASK UNDIRECTED GRAPH



(B) Does it seem like edges with high betweenness tend to be strong or weak ties, according to our two definitions above? Does this result make sense?

SOCIAL TIES GRAPH EDGEWISE BETWEENNESS

egos	alters	social_values	task_values	ties_strength_mean	ties_strength_median	reverse_present	social_betweenness
10	12	0.750	0.000	S	S	YES	30.000000
11	22	0.375	2.625	S	S	YES	30.000000
1	22	1.875	11.250	S	S	YES	26.000000
5	22	0.750	7.125	W	S	YES	26.000000
18	22	0.375	3.375	W	W	YES	24.666667
12	18	0.375	0.000	S	S	YES	24.000000
16	22	0.750	10.500	S	S	YES	22.250000
12	16	0.375	0.000	W	S	YES	18.000000
2	22	0.375	2.250	W	W	YES	16.000000
6	9	0.375	0.000	S	S	YES	16.000000
7	10	2.250	0.000	S	S	YES	16.000000
11	15	3.000	0.000	S	S	YES	16.000000
1	6	1.500	0.000	W	S	YES	15.000000
5	6	1.500	0.000	S	S	YES	15.000000
21	22	1.500	11.625	S	S	YES	12.916667
19	22	1.125	10.125	W	S	YES	12.666667
18	20	1.125	0.000	W	S	YES	7.500000
17	18	1.125	0.375	W	S	YES	4.833333
16	17	0.750	0.000	S	S	YES	4.750000
19	20	1.875	0.375	S	S	YES	4.500000
18	21	14.625	1.125	S	S	YES	4.250000
20	21	0.750	0.000	S	S	YES	4.000000
17	21	0.375	0.375	W	W	YES	3.583333
17	19	1.125	0.000	S	S	YES	3.333333
16	19	5.250	1.125	S	S	YES	3.250000
18	19	2.250	0.000	S	S	YES	2.500000
16	18	0.375	0.000	W	W	YES	1.750000
19	21	0.375	0.000	S	S	NO	1.250000
1	5	5.625	0.000	S	S	YES	1.000000
4	8	1.875	0.750	S	S	YES	1.000000

TASK TIES GRAPH EDGEWISE BETWEENNESS

egos	alters	social_values	task_values	ties_strength_mean	ties_strength_median	reverse_present	task_betweenness
1	22	1.875	11.250	S	S	YES	17.0
2	22	0.375	2.250	W	W	YES	17.0
5	22	0.750	7.125	W	S	YES	17.0
6	22	0.000	5.250	W	W	YES	17.0
7	22	0.000	1.500	S	S	YES	17.0
9	22	0.000	2.250	W	S	YES	17.0
10	22	0.000	1.125	W	W	YES	17.0
11	22	0.375	2.625	S	S	YES	17.0
14	22	0.000	0.750	S	S	YES	17.0
15	22	0.000	2.250	W	S	YES	17.0
16	22	0.750	10.500	S	S	YES	15.5
20	22	0.000	3.375	W	W	YES	15.5
13	22	0.000	1.500	S	S	YES	15.0
19	22	1.125	10.125	W	S	YES	15.0
17	22	0.000	1.125	S	S	YES	14.5
21	22	1.500	11.625	S	S	YES	14.5
18	22	0.375	3.375	W	W	YES	14.0
13	18	0.000	0.750	W	W	YES	2.0
16	19	5.250	1.125	S	S	YES	1.5
17	18	1.125	0.375	W	S	YES	1.5
18	21	14.625	1.125	S	S	YES	1.5
19	20	1.875	0.375	S	S	YES	1.5
4	8	1.875	0.750	S	S	YES	1.0
17	21	0.375	0.375	W	W	YES	1.0

The strength depends on the cutoff value set. Looking at above two tables we can not say if high betweenness yields to strong or weak ties.

4. Continue to treat the social and task ties as two distinct ties comprising one network. How many pairs of nodes do not have walks between one another? Find a solution that performs this calculation directly on the matrix—it is possible to verify this solution via igraph afterward.

Since we have consider the ties as two distinct but in one network, that means if social tie is present one way but task tie is present other way still we say that the tie is both ways.

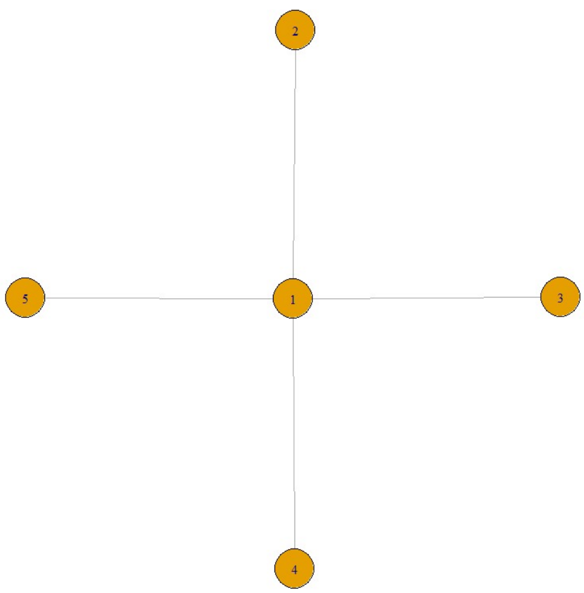
I have already coded the logic to find edges that do not have reverse relations also present. The edge marked as red is the one that does not have reverse relation all others have a reverse relation present.

	egos	alters	social_values	task_values	ties_strength_mean	ties_strength_median	reverse_present
1	1	5	5.625	0.000	S	S	YES
2	1	6	1.500	0.000	W	S	YES
3	1	22	1.875	11.250	S	S	YES
4	2	22	0.375	2.250	W	W	YES
5	4	8	1.875	0.750	S	S	YES
6	5	6	1.500	0.000	S	S	YES
7	5	22	0.750	7.125	W	S	YES
8	6	9	0.375	0.000	S	S	YES
9	6	22	0.000	5.250	W	W	YES
10	7	10	2.250	0.000	S	S	YES
11	7	22	0.000	1.500	S	S	YES
12	9	22	0.000	2.250	W	S	YES
13	10	12	0.750	0.000	S	S	YES
14	10	22	0.000	1.125	W	W	YES
15	11	15	3.000	0.000	S	S	YES
16	11	22	0.375	2.625	S	S	YES
17	12	16	0.375	0.000	W	S	YES
18	12	18	0.375	0.000	S	S	YES
19	13	18	0.000	0.750	W	W	YES
20	13	22	0.000	1.500	S	S	YES
21	14	22	0.000	0.750	S	S	YES
22	15	22	0.000	2.250	W	S	YES
23	16	17	0.750	0.000	S	S	YES
24	16	18	0.375	0.000	W	W	YES
25	16	19	5.250	1.125	S	S	YES
26	16	22	0.750	10.500	S	S	YES
27	17	18	1.125	0.375	W	S	YES
28	17	19	1.125	0.000	S	S	YES
29	17	21	0.375	0.375	W	W	YES
30	17	22	0.000	1.125	S	S	YES
31	18	19	2.250	0.000	S	S	YES
32	18	20	1.125	0.000	W	S	YES
33	18	21	14.625	1.125	S	S	YES
34	18	22	0.375	3.375	W	W	YES
35	19	20	1.875	0.375	S	S	YES
36	19	21	0.375	0.000	S	S	NO
37	19	22	1.125	10.125	W	S	YES
38	20	21	0.750	0.000	S	S	YES
39	20	22	0.000	3.375	W	W	YES
40	21	22	1.500	11.625	S	S	YES

5. The network-level measure of degree centrality is a good indicator of the dispersion of the degreedistributioninanetwork. GenerateandplotanetworkinRinwhichthenetwork-level measure of degree centrality, is equal to 1, and another where it is equal to 0. Would this relationship hold true for other measures of centrality, such as closeness or betweenness?

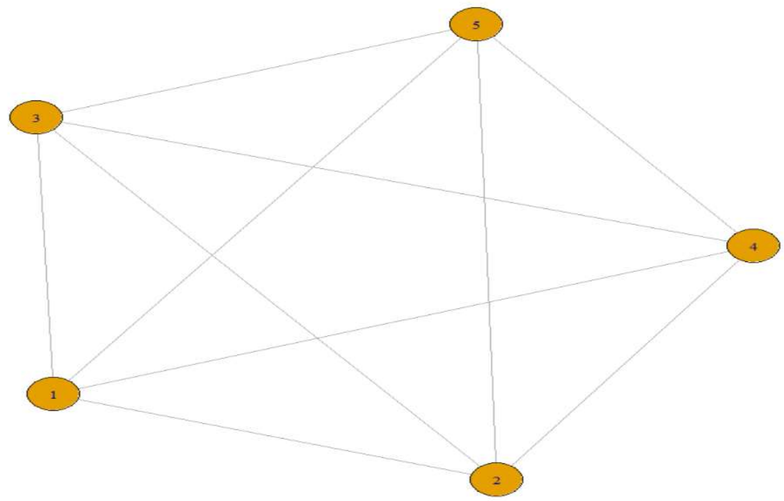
I have considered undirected graph for this example.

For network level degree centrality to be **1:** All the vertices should be connected to central node and no other node



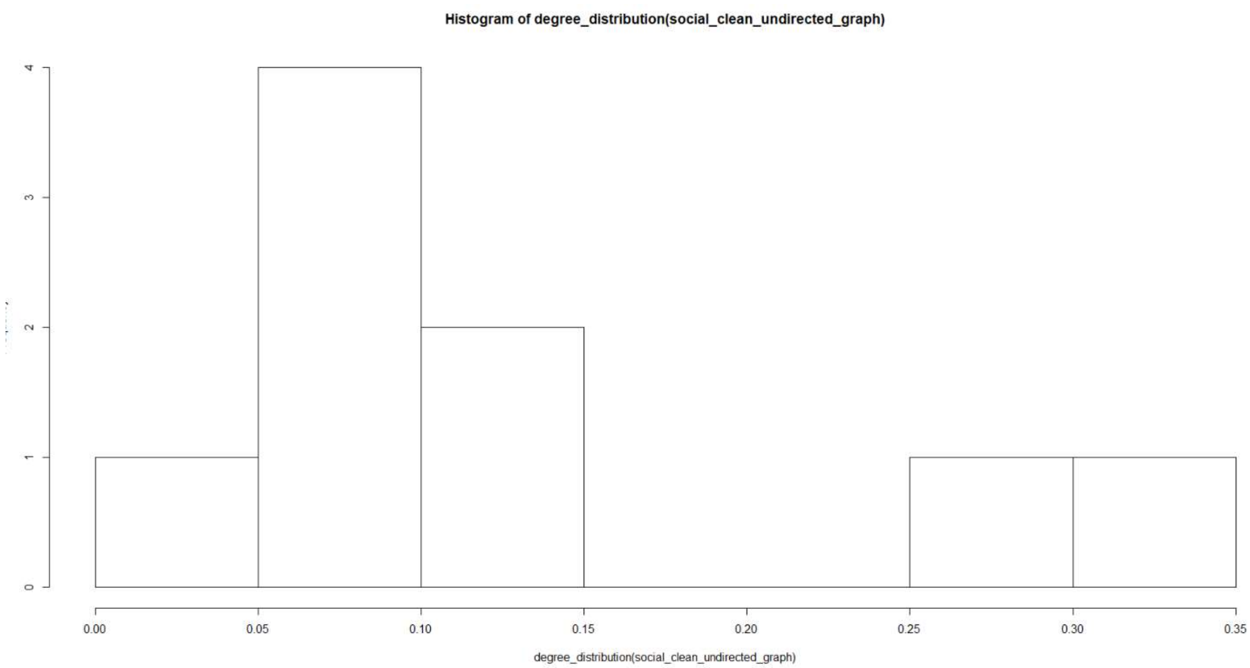
All the vertices have degree as 1, except for central with degree as 3. Hence from the formula given in question the Network level degree centrality = **1**

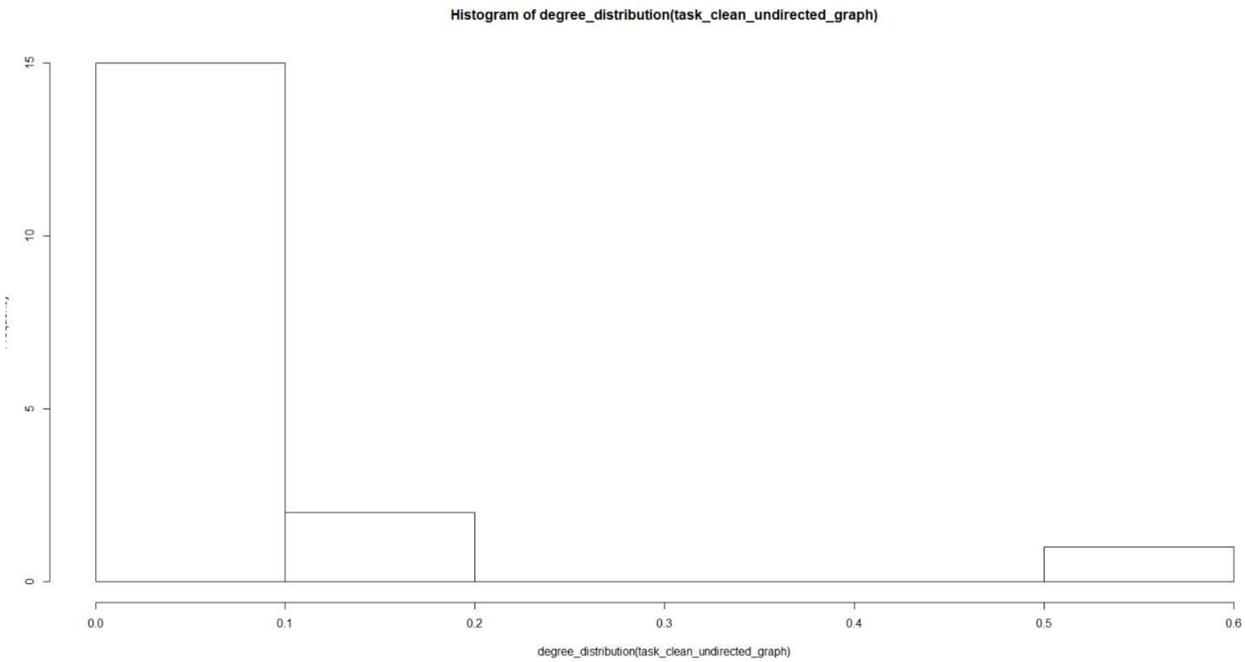
For network level degree centrality to be **0:** All the vertices should be connected to each other.

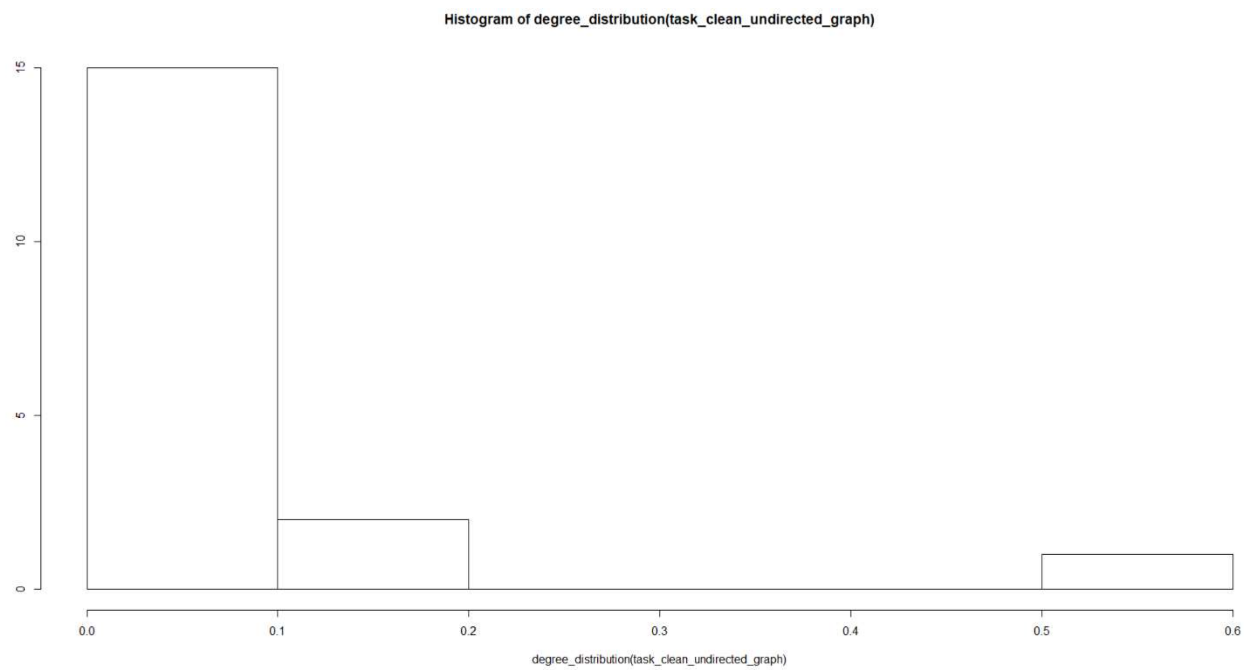


All the vertices have degree as 4. Hence from the formula given in question the Network level degree centrality = **0**

Histogram shows degree distribution after scaling the values(degrees) to between 0 to 1 is




Histogram of degree_distribution(task_clean_undirected_graph)
15
10
5
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
degree_distribution(task_clean_undirected_graph)


Histogram of degree_distribution(task_clean_undirected_graph)
15
10
5
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
degree_distribution(task_clean_undirected_graph)