INFX 576: Problem Set 1 - Network Data and Node-Level Indices*

Jay Chauhan

Due: Thursday, January 19, 2017

Collaborators: Avanti Chande, Gosuddin Siddiqi

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the `problemset1.Rmd` file from Canvas. Open `problemset1.Rmd` in RStudio and supply your solutions to the assignment by editing `problemset1.Rmd`.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps1.Rmd`, knit a PDF and submit the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries
library(statnet)

# Load data
load("problemset1_data.Rdata")
ls() # Print objects in workspace to see what is available
```

```
## [1] "silsys.ad.ilas" "silsys.fr.ilas" "sw.incidence"
```

Problem 1: Two-Mode Network Data

After loading the data for this problem set, you can use the `ls()` command to reveal the object `sw.incidence` This is the incidence matrix for the famous "Southern Women" dataset from Davis, Gardner, and Gardner's 1941 study of class and social interaction in the Deep South[1]. The matrix shows the attendance of 18 women at 14 informal social events during a nine-month observation period, based on various data sources such as interviews, guest lists, and participant observation. This is clearly two-mode data, with individuals as the "row vertices" and events as the "column vertices".

*Problems originally written by C.T. Butts (2009)

[1] Davis, Gardner, and Gardner. (1941) *Deep South*. Chicago: The University of Chicago Press.

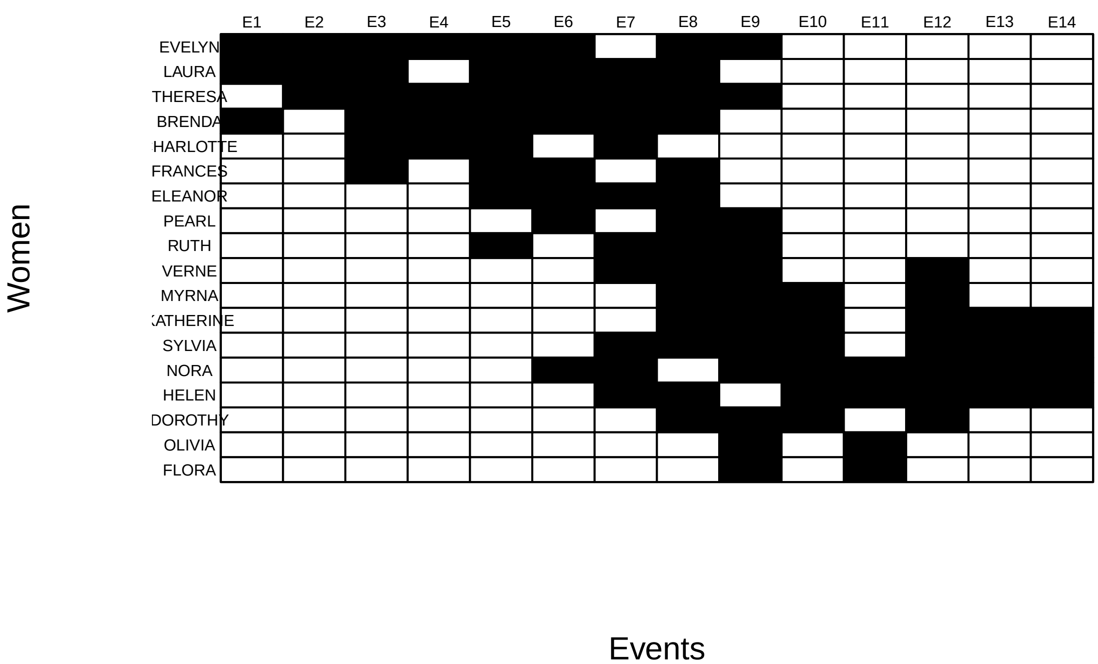
(a) Exploring Network Data

Begin by printing the matrix, and plotting it using `plot.sociomatrix`. Who seems to be the most active? Are all the women active in the same events? Describe what you observe.

```r
# Printing the incidence matrix
sw.incidence
```

```
##           E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14
## EVELYN     1  1  1  1  1  1  0  1  1   0   0   0   0   0
## LAURA      1  1  1  0  1  1  1  1  0   0   0   0   0   0
## THERESA    0  1  1  1  1  1  1  1  1   0   0   0   0   0
## BRENDA     1  0  1  1  1  1  1  1  0   0   0   0   0   0
## CHARLOTTE  0  0  1  1  1  0  1  0  0   0   0   0   0   0
## FRANCES    0  0  1  0  1  1  0  1  0   0   0   0   0   0
## ELEANOR    0  0  0  0  1  1  1  1  0   0   0   0   0   0
## PEARL      0  0  0  0  0  1  0  1  1   0   0   0   0   0
## RUTH       0  0  0  0  1  0  1  1  1   0   0   0   0   0
## VERNE      0  0  0  0  0  0  1  1  1   0   0   1   0   0
## MYRNA      0  0  0  0  0  0  0  1  1   1   0   1   0   0
## KATHERINE  0  0  0  0  0  0  0  1  1   1   0   1   1   1
## SYLVIA     0  0  0  0  0  0  1  1  1   1   0   1   1   1
## NORA       0  0  0  0  0  1  1  0  1   1   1   1   1   1
## HELEN      0  0  0  0  0  0  1  1  0   1   1   1   1   1
## DOROTHY    0  0  0  0  0  0  0  1  1   1   0   1   0   0
## OLIVIA     0  0  0  0  0  0  0  0  1   0   1   0   0   0
## FLORA      0  0  0  0  0  0  0  0  1   0   1   0   0   0
```

```r
# Plotting the sociomatrix
plot.sociomatrix(sw.incidence, cex.lab = 0.5, xlab = "Events", ylab = "Women")
```



- Evelyn, Theresa and Nora seems to be amongst the women who were most active in the social event during the nine month period, although their activities seems to be concentrated. Evelyn and Theresa were pretty active in the initial months, while on the other hand Nora wasn't so active at the start, but became a regular as since the early mid till the end of the nine month research period • Another intereseting observation that can be made is that the women which were studied during the research were divided into 2 groups, where one group was active in the social events during the start to mid of the reseach, while the other group of women became active during the mid to end of the research. • There are just 3 events in the middle of the research, events 7, 8 and 9, in which most of the women from both the groups attended together. In this event 8 was the most attended event, whereas the events 7 and 9 saw less number of women even though these were attended by women from both the groups

(b) One-Mode Projections

Consider how these women are connected through events. To do this, form the (valued) row projection of `sw.incidence` and say it as `sw.p2p`. You might find it helpfull to know that `%*%` is R's inner product operator, and `t()` is a function to transpose a matrix. `sw.p2p[i,j]` should now be the number of events that i and j have in common. Plot this matrix as in part (a) and answer the following:

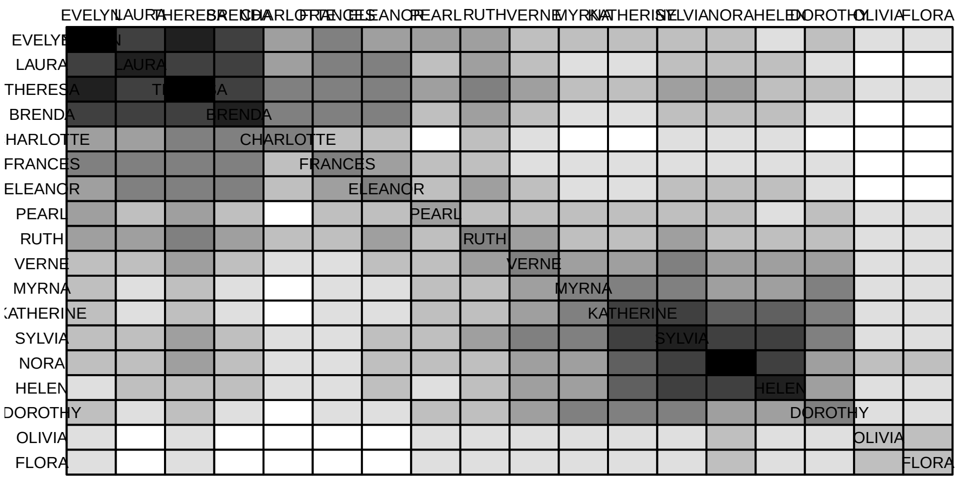
- What does the row projection tell us about how people are connected in this social group?
- Does the group seem to have subdivision?
- Do some members seem more "central" than others? If so, who?

```r
# caluclate row projections by matrix multiplication of the incidence matrix with its transpose
sw.p2p<-sw.incidence%*%t(sw.incidence)
sw.p2p
```

```
##           EVELYN LAURA THERESA BRENDA CHARLOTTE FRANCES ELEANOR PEARL RUTH
## EVELYN         8     6       7      6         3       4       3     3    3
## LAURA          6     7       6      6         3       4       4     2    3
## THERESA        7     6       8      6         4       4       4     3    4
## BRENDA         6     6       6      7         4       4       4     2    3
## CHARLOTTE      3     3       4      4         4       2       2     0    2
## FRANCES        4     4       4      4         2       4       3     2    2
## ELEANOR        3     4       4      4         2       3       4     2    3
## PEARL          3     2       3      2         0       2       2     3    2
## RUTH           3     3       4      3         2       2       3     2    4
## VERNE          2     2       3      2         1       1       2     2    3
## MYRNA          2     1       2      1         0       1       1     2    2
## KATHERINE      2     1       2      1         0       1       1     2    2
## SYLVIA         2     2       3      2         1       1       2     2    3
## NORA           2     2       3      2         1       1       2     2    2
## HELEN          1     2       2      2         1       1       2     1    2
## DOROTHY        2     1       2      1         0       1       1     2    2
## OLIVIA         1     0       1      0         0       0       0     1    1
## FLORA          1     0       1      0         0       0       0     1    1
##           VERNE MYRNA KATHERINE SYLVIA NORA HELEN DOROTHY OLIVIA FLORA
## EVELYN        2     2         2      2    2     1       2      1     1
## LAURA         2     1         1      2    2     2       1      0     0
## THERESA       3     2         2      3    3     2       2      1     1
## BRENDA        2     1         1      2    2     2       1      0     0
## CHARLOTTE     1     0         0      1    1     1       0      0     0
## FRANCES       1     1         1      1    1     1       1      0     0
## ELEANOR       2     1         1      2    2     2       1      0     0
## PEARL         2     2         2      2    2     1       2      1     1
## RUTH          3     2         2      3    2     2       2      1     1
## VERNE         4     3         3      4    3     3       3      1     1
## MYRNA         3     4         4      4    3     3       4      1     1
## KATHERINE     3     4         6      6    5     5       4      1     1
```

```
## SYLVIA       4    4       6     7    6     6      4     1    1
## NORA         3    3       5     6    8     6      3     2    2
## HELEN        3    3       5     6    6     7      3     1    1
## DOROTHY      3    4       4     4    3     3      4     1    1
## OLIVIA       1    1       1     1    2     1      1     2    2
## FLORA        1    1       1     1    2     1      1     2    2
```

```
plot.sociomatrix(sw.p2p,cex.lab = 0.5)
```



- The row projection tells us that not all the people participating in the research show a strong relation with each other. There seems to be a concentration of a connection among people at two seperate points where most of the women can be categorized. While some women seems to be connected with all the other women to some extent • As explained above, the row projection tells us that people attending these social events were mainly categorized into two groups. The sociomatrix plot of the one-mode matrix shows that Evelyn, Laura, Theresa and Brenda, Harlotte and Francis were the women who could be categorized into the first group as they had many event common between them. On the other hand, Myrna, Catherie, Sylvia, Nora, Helen and Dorothy may be considered as another group having many common events between them.

- Some members do seem more central than others in terms of betweenness where some women like Helen, Nora, Evelyn seem to have a high betweenness as they attend social events which are attended by women from both the groups, as well as events which are attended by women of their own group, thus serving as a gateway between those 2 groups.

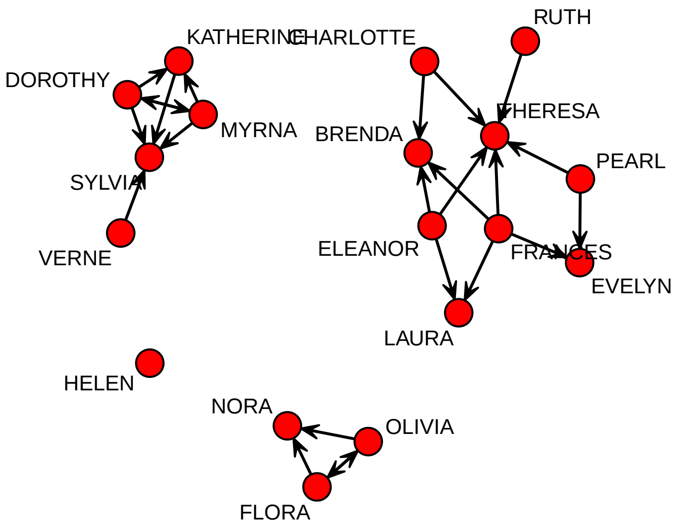
(c) Entailment Structures

Now, we are going to explore the *entailment structures* of women and events. We can construct a row-wise entailment matrix using the following code. The new matrix will be a person by person matrix such that `sw.r.entail[i,j]==1` if person j attends all of person i's events.

Use this function to create the entailment matrices (row-wise and column-wise) and produce a visualization of the entailment network for each case.

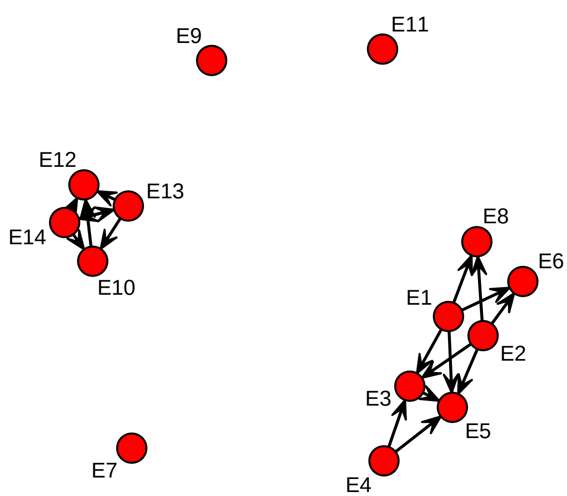
```
# Code to determine the row-wise entailment structure
# Create a new empty matrix
sw.r.entail <- matrix(0, nc=nrow(sw.incidence), nr=nrow(sw.incidence))
# Populate the matrix using a nested 'for' loop
for (i in 1:nrow(sw.incidence)){ # Pick an women i
  for (j in 1:nrow(sw.incidence)){ # And and women j
    sw.r.entail[i,j] <- all(sw.incidence[j,] >= sw.incidence[i,]) # Compare them
  }
}
```

```r
rownames(sw.r.entail) <- rownames(sw.incidence) # Renames the nodes
colnames(sw.r.entail) <- rownames(sw.incidence)
# Plot the row-wise entailment structure
gplot(sw.r.entail, label=rownames(sw.r.entail), label.cex=.7,
      boxed.labels=FALSE, vertex.cex=1.5)
```




```r
# Code to determine the column-wise entailment structure
# Create a new empty matrix
sw.c.entail <- matrix(0, nc=ncol(sw.incidence), nr=ncol(sw.incidence))
# Populate the matrix using a nested 'for' loop
for (i in 1:ncol(sw.incidence)){ # Pick an women i
  for (j in 1:ncol(sw.incidence)){ # And and women j
    sw.c.entail[i,j] <- all(sw.incidence[,j] >= sw.incidence[,i]) # Compare them
  }
}
rownames(sw.c.entail) <- colnames(sw.incidence) # Renames the nodes
colnames(sw.c.entail) <- colnames(sw.incidence)

# Plot the columne-wise entailment structure
gplot(sw.c.entail, label=rownames(sw.c.entail), label.cex=.7,
      boxed.labels=FALSE, vertex.cex=1.5)
```



Use the matrices and visualizations to answer:

- What does a path tell us?
- What do mutual (i.e. bidirectional) dyads mean?
- What is special about isolates?

• A path in the entailment network, first we take the row wise entailment network, tells us when one woman attends all of another women's events but the reverse may not be true. Explaining it with an example in the row entailment network, if we take an example of Myrna, Catherine and Sylvia, the path is directed from Myrna to Catherine and from Catherine to Sylvia indicates that the events attended by Olivia are a subset of all the events attended by Nora. Taking the case of column wise entailment network, a path indicates the association that all the women in that event is a subset of all the women attending another event but the reverse may not be true. Taking an example in this case, all the women attending the event E4 are included in the the total number of women attending event E3 and all the women attending the event E3 are included in the the total number of women attending event E5

• A bidirectional dyad in terms of entailment, specifically row wise entailment, mean that the events attended by a woman are included in all of the events attended by another woman, and the reverse is also true, which in short implies that both the women have attended the same set of event. Taking the example of Olivia an FLora, all the events attended by Olivia and Flora are the same. Similarly in case of column wise entailment, a birectional dyad imply that all the women who attended one event is inlcuded in all the women who attended the second event, that is both the events are attended by the same women.

• A Spciality about isolates in terms of women isolates is that these women have attended events that were not all commom to any one particular women. Thus they might have a high degree of relation to a diverse group of women. In terms of events, the isolates signify that the women who attended these events were not common to other events, that is a diverse group of women have attended those events.

Problem 2: Node-Level Indices and Hypothesis Tests

In the data for this assignment, you will find the following network objects: `silsys.ad.ilas` and `silsys.fr.ilas`. These are network objects containing data from David Krackhardt's famous Silican Valley Systems study.[2] The two networks consist of advice-seeking ties and friendship ties (respectively). In addition each network contains several other attributes.

(a) Computing Node-Level Indices

Compute indegree, outdegree, betweenness and eigenvector centrality scores for all individuals in each of the two networks. A useful trick to combine vectors or matrices `a`, `b`, and `c` into a single matrix using the `cbind` command as follows: `cbind(a,b,c)`. Print the centrality scores.

- Who are some of the most central individuals in the advice-seeking network? In the friendship network?

```
#calculate indegree, outdegree, betweenness and eigenvector centrality scores in advice-seeking network
ideg<-degree(silsys.ad.ilas, cmode="indegree")
odeg<-degree(silsys.ad.ilas, cmode="outdegree")
bet<-betweenness(silsys.ad.ilas,gmode="graph")
evc<-evcent(silsys.ad.ilas,gmode = "digraph")

#combine all centrality scores into a matrix form
ad_cent<-cbind(Indegree=ideg,Outdegree=odeg,Betweenness=bet,EigenvectorCentrality=evc)
ad_cent
```

[2] Krackhardt, David. (1990) "Assessing the Political Landscape: Structure, Cognition, and Power in Organizations." *ASQ*, 35(2): 342-369.

```
##       Indegree Outdegree  Betweenness EigenvectorCentrality
##  [1,]        0         4   0.00000000           0.085137912
##  [2,]        1         2   5.00000000           0.046021177
##  [3,]        1         1   0.00000000           0.010723431
##  [4,]        0         0   0.00000000           0.000000000
##  [5,]       19         1  24.56388889           0.035112155
##  [6,]        9         3  16.50000000           0.052946912
##  [7,]        0         2   0.00000000           0.017211866
##  [8,]        2         3   0.29166667           0.028558189
##  [9,]        0         1   0.00000000           0.009826219
## [10,]        5         2   1.13333333           0.013404775
## [11,]        0         0   0.00000000           0.000000000
## [12,]        3         7  40.14166667           0.296327696
## [13,]        5        15  62.31388889           0.464123784
## [14,]        0         1   0.00000000           0.007111326
## [15,]        2         4   3.23055556           0.092011133
## [16,]        7         9  71.74722222           0.337127823
## [17,]       10         3  24.08333333           0.046259283
## [18,]        2         3  12.75000000           0.049871387
## [19,]       19         7 185.46388889           0.173366170
## [20,]        4         3   2.60555556           0.081133332
## [21,]        7         3   8.29166667           0.052946912
## [22,]        1         6   4.16666667           0.074929595
## [23,]        0         4   0.00000000           0.116399857
## [24,]        6        11  95.35555556           0.174128372
## [25,]        5        15  28.66388889           0.472714197
## [26,]        1         2   0.08333333           0.045835586
## [27,]       11         2  18.30555556           0.019926759
## [28,]        2         4   0.00000000           0.293130687
## [29,]        7         6  36.83333333           0.192117282
## [30,]        7         3  19.25000000           0.048516930
## [31,]        0         2   0.00000000           0.103391245
## [32,]        0         2   0.00000000           0.046021177
## [33,]        0         5   0.00000000           0.130231731
## [34,]        0         2   0.00000000           0.012083866
## [35,]        6         5   9.78333333           0.242644917
## [36,]        5         4   4.44166667           0.161903520
```

Advice-seeking network: • For indegree centrality, person 5 and 19 are most central as they have a high number of ties coming in, in this case 19 each • For outdegree centrality, person 13 and 25 are the most central, as they have a high number of ties going out to other people, in this case 15 each. • Betweenness is the highest in person 19 with a score or 185.46 while it is also high for person 24 with a value 95.35, which indicates that they serve in the position of a gatekeeper. • Eigenvector centrality is high for person 25 and 13

```r
#calculate indegree, outdegree, betweenness and eigenvector centrality scores in friendship network
ideg<-degree(silsys.fr.ilas, cmode="indegree")
odeg<-degree(silsys.fr.ilas, cmode="outdegree")
bet<-betweenness(silsys.fr.ilas,gmode="graph")
evc<-evcent(silsys.fr.ilas,gmode = "digraph")

#combine all centrality scores into a matrix form
fr_cent<-cbind(Indegree=ideg,Outdegree=odeg,Betweenness=bet,EigenvectorCentrality=evc)
fr_cent
```

```
##       Indegree Outdegree Betweenness EigenvectorCentrality
```

```
## [1,]       0       1   0.0000000          0.00521893
## [2,]       6       4  45.4000000          0.12763229
## [3,]       2       4  23.0142857          0.04106213
## [4,]       8       7  50.8100108          0.24710489
## [5,]       2       1   0.0000000          0.02122349
## [6,]       4       2  23.3145022          0.03520181
## [7,]       3       2   0.4880952          0.07582730
## [8,]       1       2   0.0000000          0.01130669
## [9,]       3       2   3.4247294          0.11750703
## [10,]      1       1   0.0000000          0.01892243
## [11,]      9       7  15.1009380          0.25394931
## [12,]      2       4  33.7774351          0.07752845
## [13,]      6       7  37.5816739          0.23063646
## [14,]      5       7   9.8318723          0.21676149
## [15,]      4       3   1.6250000          0.14235488
## [16,]      5       4  37.9661797          0.07984479
## [17,]      0       0   0.0000000          0.00000000
## [18,]      4       4   3.7466270          0.15354700
## [19,]      8       6  62.3512266          0.14315298
## [20,]     10      10  52.3765512          0.34008707
## [21,]      5       4  30.8626082          0.09428414
## [22,]      1       3   0.0000000          0.13120207
## [23,]      1       2   0.0000000          0.10627877
## [24,]     11       9  54.3242424          0.26435214
## [25,]      0       0   0.0000000          0.00000000
## [26,]      3       2   3.3916667          0.03665105
## [27,]      3       3   1.2660714          0.09859178
## [28,]      0       1   0.0000000          0.01892243
## [29,]     12      16 101.9499639          0.45250083
## [30,]      5       5  10.6783550          0.15493182
## [31,]      0       2   0.0000000          0.03306107
## [32,]      0       0   0.0000000          0.00000000
## [33,]      8       7  44.5743867          0.29229007
## [34,]      6       7   8.0500000          0.29092490
## [35,]      8       7  84.5935786          0.15292812
## [36,]      1       1   0.0000000          0.01183758
```

Friendship network: • For indegree centrality, person 29 is most central as it has a high number of ties coming in. • For outdegree centrality, person 20 and 29 are the most central, as they have a high number of ties going out to other people. • Betweenness is the highest in person 29 with a score or 101.94 while it is also high for person 35 with a value 84.59, which indicates that they serve in the position of a gatekeeper. • Eigenvector centrality is high for person 29 and 20

(b) Comparing Node-Level Indices

The `cor` command calculates correlations. You can apply this function to a matrix to compute the correlation matrix - correlations for all pairs of columns. Compute the within and between network correlation matrices for the centrality scores you computed in part (a). Print this table and answer the following:

- Does centrality in the advice-seeking network correspond (or not) to centrality in the friendship network?
- What centrality measures are most strong correlated? Least strongly correlated?

```
#calculate correlation within advice-seeking network correlation matrix
cat("correlation within advice-seeking network correlation matrix\n")
```

```
## correlation within advice-seeking network correlation matrix
```

```r
cor(ad_cent)
```

```
##                        Indegree Outdegree Betweenness
## Indegree              1.0000000 0.2047587   0.6534849
## Outdegree             0.2047587 1.0000000   0.5548932
## Betweenness           0.6534849 0.5548932   1.0000000
## EigenvectorCentrality 0.1608585 0.8927812   0.4282096
##                       EigenvectorCentrality
## Indegree                          0.1608585
## Outdegree                         0.8927812
## Betweenness                       0.4282096
## EigenvectorCentrality             1.0000000
```

```r
#calculate correlation within friendship network correlation matrix
cat("\ncorrelation within friendship network correlation matrix\n")
```

```
##
## correlation within friendship network correlation matrix
```

```r
cor(fr_cent)
```

```
##                        Indegree Outdegree Betweenness
## Indegree              1.0000000 0.9109728   0.8230471
## Outdegree             0.9109728 1.0000000   0.8146123
## Betweenness           0.8230471 0.8146123   1.0000000
## EigenvectorCentrality 0.8834378 0.9414224   0.6628399
##                       EigenvectorCentrality
## Indegree                          0.8834378
## Outdegree                         0.9414224
## Betweenness                       0.6628399
## EigenvectorCentrality             1.0000000
```

```r
#calculate correlation between advice-seeking and friendship network correlation matrices
cat("\ncorrelation between advice-seeking and friendship network correlation matrices\n")
```

```
##
## correlation between advice-seeking and friendship network correlation matrices
```

```r
cor(ad_cent,fr_cent)
```

```
##                        Indegree  Outdegree Betweenness
## Indegree              0.1326214 0.02212860   0.2334395
## Outdegree             0.1179450 0.14838411   0.2772911
## Betweenness           0.3551649 0.26241725   0.4448274
## EigenvectorCentrality 0.0288127 0.09014183   0.2913855
##                       EigenvectorCentrality
## Indegree                       -0.098459851
## Outdegree                       0.094943928
## Betweenness                     0.142671419
## EigenvectorCentrality           0.006682372
```

• The result of the correlation suggests that the centrality measure in the advice seeking network does not correspond to the centrality measure in the friendship network. This is due to the low correlation results. This means that the people who are more central in the advice-seeking network might not be equally central in the friendship network and vice-versa. • Here we can see that in genral all the centrality measures of the friendship network are strongly correlated to each other. If we take some of the most strongly correlated centrality measures, they are the correlation between Enigenvector centrality and Outdegree(0.94) in the advice-seeking network, Enigenvector centrality and Outdegree(0.91) which are among the friendship network, Betweenness and Betweenness in the correaltion between both networks. Also, the least correlated centrality measures are indegree and eigenvector centrality(0.16) in the advice-seeking network, betweenness and eigenvector centrality(0.66) in the friendship network, eigenvector centrality and eigenvector centrality(0.006) in the correlation between advice-seeking and friendship network.

(c) Relating Node-Level Indices to Covariates

In the in-class demo you were given a function for testing the correlation between vectors using a permutation test. Using this function, assess the relationship between the "Charisma" (charisma, as rated by fellow employees) and "Potency" (ability to overcome opposition in order to achieve goals, as rated by fellow employees) vertex attributes and the centrality scores you computed in part (a).

Remember you can extract vertex attributes from network objects with the `%v%` operator or the `get.vertex.attribute` function. Report the results of these tests as a table showing the observed correlation of each attribute with each centrality measure, along with the two-sided p-value for the appropriate test in each case.

```r
#retrience potency and charisma values from advice-seeking network
advice_network<-silsys.ad.ilas
advice_potency<-get.vertex.attribute(x = advice_network,attrname = "Potency")
advice_Charisma<-get.vertex.attribute(x = advice_network,attrname = "Charisma")
cat("Charisma in advice seeking netwrok:\n",advice_Charisma)
```

```
## Charisma in advice seeking netwrok:
##  3 5.281 3.625 4.838 3.843 4.875 4.468 2.258 4.906 4.656 3.986 2.218 0 3.516 3.566 5.29 4.906 4.468 4
```

```r
cat("\nPotency in advice seeking netwrok:\n",advice_potency)
```

```
## 
## Potency in advice seeking netwrok:
##  2.193 2.843 2.75 3.612 6.187 5.625 3.531 2.354 2.871 3.437 2.531 1.718 0 2.58 2.766 6.225 5.322 4.48
```

```r
#retrience potency and charisma values from friends network
fr_network<-silsys.fr.ilas
fr_potency<-get.vertex.attribute(x = fr_network,attrname = "Potency")
fr_charisma<-get.vertex.attribute(x = fr_network,attrname = "Charisma")
cat("\nCharisma in friendship netwrok:\n",fr_charisma)
```

```
## 
## Charisma in friendship netwrok:
##  3 5.281 3.625 4.838 3.843 4.875 4.468 2.258 4.906 4.656 3.986 2.218 0 3.516 3.566 5.29 4.906 4.468 4
```

```r
cat("\nPotency in friendship netwrok:\n",fr_potency)
```

```
## 
## Potency in friendship netwrok:
##  2.193 2.843 2.75 3.612 6.187 5.625 3.531 2.354 2.871 3.437 2.531 1.718 0 2.58 2.766 6.225 5.322 4.48
```

```r
perm.cor.test<-function(x,y,niter=5000){  #Define a simple test function
  c.obs<-cor(x,y,use="complete.obs")
  c.rep<-vector()
  c.two_sided<-vector()
  for(i in 1:niter)
    c.rep[i]<-cor(x,sample(y),use="complete.obs")
  cat("Vector Permutation Test:\n\tObserved correlation: ",
      c.obs,"\tReplicate quantiles (niter=",niter,")\n",sep="")
```

```r
  cat("\t\tPr(rho>=obs):",mean(c.rep>=c.obs),"\n")
  cat("\t\tPr(rho<=obs):",mean(c.rep<=c.obs),"\n")
  cat("\t\tPr(|rho|>=|obs|):",mean(abs(c.rep)>=abs(c.obs)),"\n")
  invisible(list(obs=c.obs, rep=c.rep, two_sided_prob=mean(abs(c.rep)>=abs(c.obs))))
}
```

```r
#correlation between friendship network and Charisma vectors using permutation text
fr_c_in<-perm.cor.test(fr_charisma,fr_cent[,"Indegree"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.02825184   Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.5738
##      Pr(rho<=obs): 0.4262
##      Pr(|rho|>=|obs|): 0.8632
```

```r
fr_c_out<-perm.cor.test(fr_charisma,fr_cent[,"Outdegree"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.06023858   Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.6588
##      Pr(rho<=obs): 0.3412
##      Pr(|rho|>=|obs|): 0.7214
```

```r
fr_c_bet<-perm.cor.test(fr_charisma,fr_cent[,"Betweenness"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.1143221    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.7572
##      Pr(rho<=obs): 0.2428
##      Pr(|rho|>=|obs|): 0.5076
```

```r
fr_c_eig<-perm.cor.test(fr_charisma,fr_cent[,"EigenvectorCentrality"])
```

```
## Vector Permutation Test:
##  Observed correlation: 0.0005403347  Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.5208
##      Pr(rho<=obs): 0.4792
##      Pr(|rho|>=|obs|): 0.9972
```

```r
#correlation between friendship network and Potency vectors using permutation text
fr_pot_in<-perm.cor.test(fr_potency,fr_cent[,"Indegree"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.05101107   Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.6134
##      Pr(rho<=obs): 0.3866
##      Pr(|rho|>=|obs|): 0.7784
```

```r
fr_pot_out<-perm.cor.test(fr_potency,fr_cent[,"Outdegree"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.1096314    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.7316
##      Pr(rho<=obs): 0.2684
##      Pr(|rho|>=|obs|): 0.5278
```

```r
fr_pot_bet<-perm.cor.test(fr_potency,fr_cent[,"Betweenness"])
```

```
## Vector Permutation Test:
```

```
##  Observed correlation: -0.06166778   Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.6422
##      Pr(rho<=obs): 0.3578
##      Pr(|rho|>=|obs|): 0.7172
```

```r
fr_pot_eig<-perm.cor.test(fr_potency,fr_cent[,"EigenvectorCentrality"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.1364308    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.7828
##      Pr(rho<=obs): 0.2172
##      Pr(|rho|>=|obs|): 0.4264
```

```r
#correlation between advice-seeking network and potency vectors using permutation text
ad_pot_in<-perm.cor.test(advice_potency,ad_cent[,"Indegree"])
```

```
## Vector Permutation Test:
##  Observed correlation: 0.5834776 Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0
##      Pr(rho<=obs): 1
##      Pr(|rho|>=|obs|): 6e-04
```

```r
ad_pot_out<-perm.cor.test(advice_potency,ad_cent[,"Outdegree"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.2492979    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.9342
##      Pr(rho<=obs): 0.0658
##      Pr(|rho|>=|obs|): 0.1404
```

```r
ad_pot_bet<-perm.cor.test(advice_potency,ad_cent[,"Betweenness"])
```

```
## Vector Permutation Test:
##  Observed correlation: 0.2411634 Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.0866
##      Pr(rho<=obs): 0.9134
##      Pr(|rho|>=|obs|): 0.159
```

```r
ad_pot_eig<-perm.cor.test(advice_potency,ad_cent[,"EigenvectorCentrality"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.2113803    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.894
##      Pr(rho<=obs): 0.106
##      Pr(|rho|>=|obs|): 0.2178
```

```r
#correlation between advice-seeking network and charisma vectors using permutation text
ad_c_in<-perm.cor.test(advice_Charisma,ad_cent[,"Indegree"])
```

```
## Vector Permutation Test:
##  Observed correlation: 0.08059178    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.3468
##      Pr(rho<=obs): 0.6532
##      Pr(|rho|>=|obs|): 0.6374
```

```r
ad_c_out<-perm.cor.test(advice_Charisma,ad_cent[,"Outdegree"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.4556335    Replicate quantiles (niter=5000)
```

```
##      Pr(rho>=obs): 0.9898
##      Pr(rho<=obs): 0.0102
##      Pr(|rho|>=|obs|): 0.0104
```

```r
ad_c_bet<-perm.cor.test(advice_Charisma,ad_cent[,"Betweenness"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.1343587    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.808
##      Pr(rho<=obs): 0.192
##      Pr(|rho|>=|obs|): 0.4216
```

```r
ad_c_eig<-perm.cor.test(advice_Charisma,ad_cent[,"EigenvectorCentrality"])
```

```
## Vector Permutation Test:
##  Observed correlation: -0.3875669    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.9842
##      Pr(rho<=obs): 0.0158
##      Pr(|rho|>=|obs|): 0.0166
```

```r
#form a table of all the the observed correlation of each attribute with each centrality measure, along
rbind(Indegree_Potency_Friends=fr_pot_in,Outdegree_Potency_Friends=fr_pot_out,
      Betweenness_Potency_friends=fr_pot_bet,Eigenvector_Potency_Freinds=fr_pot_eig,
      Indegree_Charisma_Friends=fr_c_in,Outdegree_Charisma_Friends=fr_c_out,
      Betweenness_Charsima_Friends=fr_c_bet,Eigenvector_Charisma_Friends=fr_c_eig,
      Indegree_Potency_Advisor=ad_pot_in,Outdegree_Potency_Advisor=ad_pot_out,
      Betweenness_Potency_Advisor=ad_pot_bet,Eigenvector_Potency_Advisor=ad_pot_eig,
      Indegree_Charisma_Advisor=ad_c_in,Outdegree_Charisma_Advisor=ad_c_out,
      Betweenness_Charisma_Advisor=ad_c_bet, Eigenvector_Charisma_Advisor=ad_c_eig)
```

```
##                              obs          rep          two_sided_prob
## Indegree_Potency_Friends     -0.05101107  Numeric,5000 0.7784
## Outdegree_Potency_Friends    -0.1096314   Numeric,5000 0.5278
## Betweenness_Potency_friends  -0.06166778  Numeric,5000 0.7172
## Eigenvector_Potency_Freinds  -0.1364308   Numeric,5000 0.4264
## Indegree_Charisma_Friends    -0.02825184  Numeric,5000 0.8632
## Outdegree_Charisma_Friends   -0.06023858  Numeric,5000 0.7214
## Betweenness_Charsima_Friends -0.1143221   Numeric,5000 0.5076
## Eigenvector_Charisma_Friends 0.0005403347 Numeric,5000 0.9972
## Indegree_Potency_Advisor     0.5834776    Numeric,5000 6e-04
## Outdegree_Potency_Advisor    -0.2492979   Numeric,5000 0.1404
## Betweenness_Potency_Advisor  0.2411634    Numeric,5000 0.159
## Eigenvector_Potency_Advisor  -0.2113803   Numeric,5000 0.2178
## Indegree_Charisma_Advisor    0.08059178   Numeric,5000 0.6374
## Outdegree_Charisma_Advisor   -0.4556335   Numeric,5000 0.0104
## Betweenness_Charisma_Advisor -0.1343587   Numeric,5000 0.4216
## Eigenvector_Charisma_Advisor -0.3875669   Numeric,5000 0.0166
```

- How to charisma and potency appear to relate to positional structure at Silicon Valley Systems?

- The positional structure at Silicon Valley Systems relate to the charisma and potency such that the in the advice seeking ties, based on the probability and correlation values, the person rated highly potent by its fellow employees has a high indegree values