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

Suchitra

Due: Thursday, January 19, 2017

Collaborators: Suchitra,Avanti, Gos, Jay.

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)

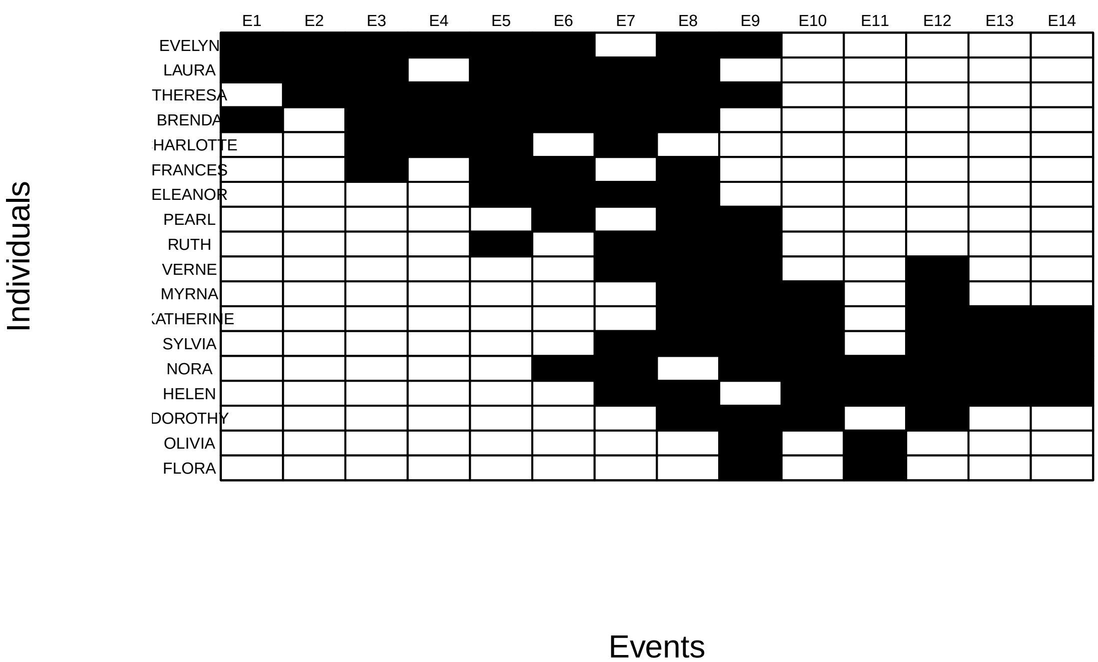
[1] Davis, Gardner, and Gardner. (1941) *Deep South*. Chicago: The University of Chicago Press.

(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.

```
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
```

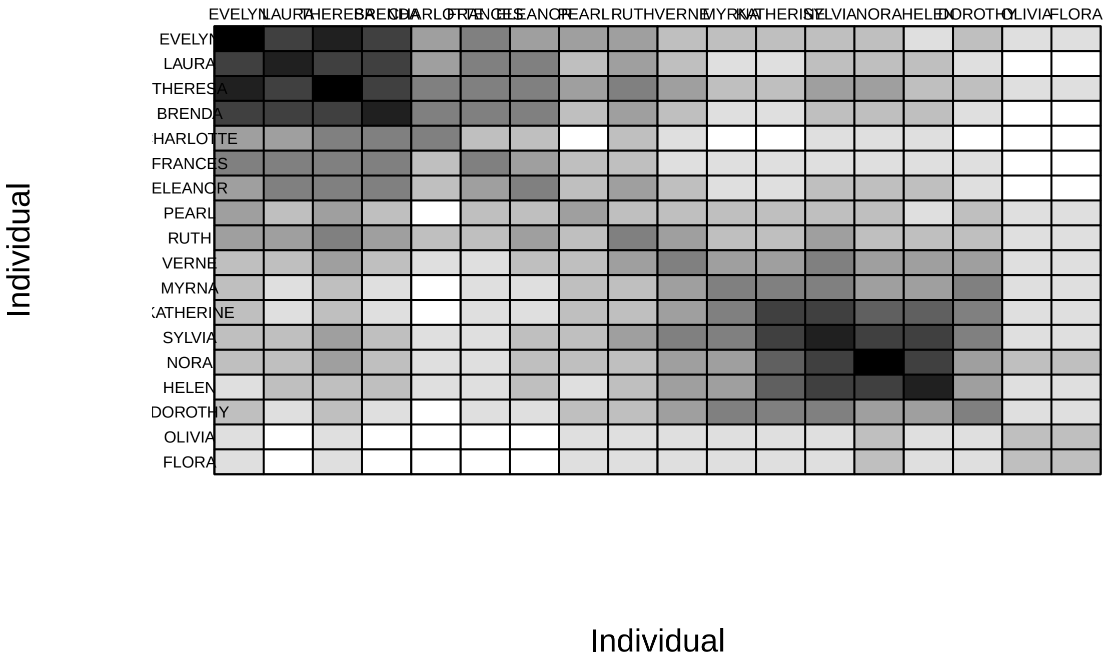
```
plot.sociomatrix(sw.incidence,xlab = "Events", ylab= "Individuals", cex.lab = 0.5)
```



Answer:
From the plot, it can observed that Evelyn, Theresa and Nora are the most active among the given set of people. Though, not all the women are active in the same events, higher participation can be observed in few events as compared to others. Events like E8 and E9 have higher participation while events like E1,E2,E13 and E14 have fewer particpation. The graph also shows concentration at the upper left and lower right regions, which may indicate some kind of groups within the social structure.

(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:

```
sw.p2p <- sw.incidence %*% t(sw.incidence)
plot.sociomatrix(sw.p2p, xlab= "Individual", ylab= "Individual", cex.lab = 0.5, diaglab = FALSE)
```



- What does the row projection tell us about how people are connected in this social group? The sociomatrix function helps in visualizing the matrix. Here, the cells within the matrix are shaded according to their value in the dataset. This explains the reason for fewer cells having darker shades and the others having lighter ones. From the plot it can be seen that there are two groups of people who attended the same events. Evelyn, Laura, Theresand and Brenda form the first group, and Katherine, Sylvia, Nora and Helen form the second group. These women may have closer ties as they have participated in mostly the same events. But barring these few, there doesn't seem to be strong social ties within the network.
- Does the group seem to have subdivision? The group does seem to have a sub-division. There are two distinctly dense region indicating higher participation by the same set of women in the same events.
- Do some members seem more "central" than others? If so, who? From the plot, it can be observed that Evelyn, Nora and Theresa have a higher degree centrality. This is because they have higher number of events common with other women, and this may indicate more number of ties. Ruth and Verne seem to bridge the two active groups and this may indicate a high betweenness centrality measure.

(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.

```r
# 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
  }
}
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? Entailment structure for Women: A path from W1 to W2 to W3 tells that, W3 participates in all the events that W2 does, and W2 participates in all the event that W1 does. Considerng the example of Sylvia, Katherine and Myrna, events attended by Myrna is a subset of those attended by Katherine, and events attended by Katherine is a subset of those attended by Sylvia.

Entailment structure for Events: A path from E13 to E10 to E12 tells that, E12 has all the women participants as E10 does, and E10 has all the women participants as E13 does.

- What do mutual (i.e. bidirectional) dyads mean? Dyad in general means the relationship between two nodes. Considering the case of Dorothy and Myrna, they both attended the same events. Considering E14 and E13, they have the same women paticipants.
- What is special about isolates? Isolate in women(Helen) indicate that she is connected to a diverse group of people. This is because the set of events she attends and that other attend is quite different. This does not mean that she attends less number of events, but simply that she doesn't meet the same people quite often. Isolates in events(E7, E9, E11) indicate that diverse set of women attend these events. These would be the people who wouldn't have that many common 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.

```
#Advice-seeking ties
summary(silsys.ad.ilas)
```

```
## Network attributes:
##   vertices = 36
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##  total edges = 147
##    missing edges = 0
##    non-missing edges = 147
##  density = 0.1166667
##
## Vertex attributes:
##
##  Bargaining.Unit:
##    logical valued attribute
##    attribute summary:
##    Mode   FALSE    TRUE    NA's
## logical      21      15       0
##
##  Charisma:
##    numeric valued attribute
##    attribute summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.476   4.249   3.801   4.810   5.625
##
##  ID:
##    integer valued attribute
##    36 values
##
##  Potency:
##    numeric valued attribute
##    attribute summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.568   2.984   3.320   4.214   6.875
##
##  Rank:
##    numeric valued attribute
```

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

```
##    attribute summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   1.000   1.222   1.000   3.000
##
##  Responded:
##    logical valued attribute
##    attribute summary:
##    Mode   FALSE    TRUE    NA's
## logical       3      33       0
##
##  Role:
##    numeric valued attribute
##    attribute summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.194   3.000   3.000
##   vertex.names:
##    character valued attribute
##    36 valid vertex names
##
## No edge attributes
##
## Network edgelist matrix:
##        [,1] [,2]
##   [1,]   24    2
##   [2,]   12    3
##   [3,]    2    5
##   [4,]    6    5
##   [5,]    7    5
##   [6,]    8    5
##   [7,]   13    5
##   [8,]   14    5
##   [9,]   15    5
##  [10,]   16    5
##  [11,]   17    5
##  [12,]   19    5
##  [13,]   20    5
##  [14,]   21    5
##  [15,]   22    5
##  [16,]   23    5
##  [17,]   24    5
##  [18,]   25    5
##  [19,]   29    5
##  [20,]   32    5
##  [21,]   33    5
##  [22,]    1    6
##  [23,]    3    6
##  [24,]    8    6
##  [25,]   12    6
##  [26,]   21    6
##  [27,]   24    6
##  [28,]   25    6
##  [29,]   26    6
##  [30,]   35    6
##  [31,]   13    8
```

```
## [32,]   25    8
## [33,]   13   10
## [34,]   22   10
## [35,]   24   10
## [36,]   29   10
## [37,]   34   10
## [38,]   13   12
## [39,]   24   12
## [40,]   25   12
## [41,]   12   13
## [42,]   16   13
## [43,]   25   13
## [44,]   28   13
## [45,]   29   13
## [46,]   13   15
## [47,]   33   15
## [48,]   13   16
## [49,]   19   16
## [50,]   25   16
## [51,]   28   16
## [52,]   31   16
## [53,]   35   16
## [54,]   36   16
## [55,]   10   17
## [56,]   13   17
## [57,]   16   17
## [58,]   19   17
## [59,]   22   17
## [60,]   24   17
## [61,]   25   17
## [62,]   30   17
## [63,]   34   17
## [64,]   36   17
## [65,]    7   18
## [66,]   27   18
## [67,]    1   19
## [68,]    5   19
## [69,]    6   19
## [70,]   13   19
## [71,]   16   19
## [72,]   17   19
## [73,]   18   19
## [74,]   20   19
## [75,]   21   19
## [76,]   23   19
## [77,]   24   19
## [78,]   25   19
## [79,]   26   19
## [80,]   28   19
## [81,]   29   19
## [82,]   30   19
## [83,]   31   19
## [84,]   35   19
## [85,]   36   19
```

```
##  [86,]   13   20
##  [87,]   22   20
##  [88,]   25   20
##  [89,]   33   20
##  [90,]    6   21
##  [91,]    8   21
##  [92,]   15   21
##  [93,]   18   21
##  [94,]   19   21
##  [95,]   24   21
##  [96,]   25   21
##  [97,]   24   22
##  [98,]    1   24
##  [99,]   12   24
## [100,]   15   24
## [101,]   19   24
## [102,]   22   24
## [103,]   23   24
## [104,]   12   25
## [105,]   13   25
## [106,]   16   25
## [107,]   28   25
## [108,]   35   25
## [109,]   12   26
## [110,]    1   27
## [111,]   10   27
## [112,]   13   27
## [113,]   16   27
## [114,]   17   27
## [115,]   18   27
## [116,]   22   27
## [117,]   24   27
## [118,]   25   27
## [119,]   29   27
## [120,]   30   27
## [121,]   13   28
## [122,]   25   28
## [123,]    2   29
## [124,]   13   29
## [125,]   15   29
## [126,]   20   29
## [127,]   23   29
## [128,]   32   29
## [129,]   33   29
## [130,]    9   30
## [131,]   13   30
## [132,]   16   30
## [133,]   19   30
## [134,]   24   30
## [135,]   25   30
## [136,]   27   30
## [137,]   12   35
## [138,]   16   35
## [139,]   25   35
```

```
## [140,]   29   35
## [141,]   33   35
## [142,]   36   35
## [143,]   13   36
## [144,]   16   36
## [145,]   19   36
## [146,]   25   36
## [147,]   35   36
```

```r
# Degree
indeg_ad <- degree(silsys.ad.ilas, cmode = "indegree")
outdeg_ad <- degree(silsys.ad.ilas, cmode = "outdegree")
#Betweenness
?betweenness
bet_ad <- betweenness(silsys.ad.ilas, gmode = "digraph")
#Eigen Vector
?evcent
eig_ad <- evcent(silsys.ad.ilas)

advice_matrix <- cbind(indeg_ad,outdeg_ad,bet_ad,eig_ad)
advice_matrix
```

```
##       indeg_ad outdeg_ad      bet_ad      eig_ad
##  [1,]        0         4   0.0000000 0.085137912
##  [2,]        1         2  10.0000000 0.046021177
##  [3,]        1         1   0.0000000 0.010723431
##  [4,]        0         0   0.0000000 0.000000000
##  [5,]       19         1  49.1277778 0.035112155
##  [6,]        9         3  33.0000000 0.052946912
##  [7,]        0         2   0.0000000 0.017211866
##  [8,]        2         3   0.5833333 0.028558189
##  [9,]        0         1   0.0000000 0.009826219
## [10,]        5         2   2.2666667 0.013404775
## [11,]        0         0   0.0000000 0.000000000
## [12,]        3         7  80.2833333 0.296327696
## [13,]        5        15 124.6277778 0.464123784
## [14,]        0         1   0.0000000 0.007111326
## [15,]        2         4   6.4611111 0.092011133
## [16,]        7         9 143.4944444 0.337127823
## [17,]       10         3  48.1666667 0.046259283
## [18,]        2         3  25.5000000 0.049871387
## [19,]       19         7 370.9277778 0.173366170
## [20,]        4         3   5.2111111 0.081133332
## [21,]        7         3  16.5833333 0.052946912
## [22,]        1         6   8.3333333 0.074929595
## [23,]        0         4   0.0000000 0.116399857
## [24,]        6        11 190.7111111 0.174128372
## [25,]        5        15  57.3277778 0.472714197
## [26,]        1         2   0.1666667 0.045835586
## [27,]       11         2  36.6111111 0.019926759
## [28,]        2         4   0.0000000 0.293130687
## [29,]        7         6  73.6666667 0.192117282
## [30,]        7         3  38.5000000 0.048516930
## [31,]        0         2   0.0000000 0.103391245
```

```
## [32,]        0         2   0.0000000 0.046021177
## [33,]        0         5   0.0000000 0.130231731
## [34,]        0         2   0.0000000 0.012083866
## [35,]        6         5  19.5666667 0.242644917
## [36,]        5         4   8.8833333 0.161903520
```

```
#Friendship ties
summary(silsys.fr.ilas)
```

```
## Network attributes:
##   vertices = 36
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##  total edges = 147
##    missing edges = 0
##    non-missing edges = 147
##  density = 0.1166667
##
## Vertex attributes:
##
##  Bargaining.Unit:
##    logical valued attribute
##    attribute summary:
##    Mode   FALSE    TRUE    NA's
## logical      21      15       0
##
##  Charisma:
##    numeric valued attribute
##    attribute summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.476   4.249   3.801   4.810   5.625
##
##  ID:
##    integer valued attribute
##    36 values
##
##  Potency:
##    numeric valued attribute
##    attribute summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.568   2.984   3.320   4.214   6.875
##
##  Rank:
##    numeric valued attribute
##    attribute summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   1.000   1.222   1.000   3.000
##
##  Responded:
##    logical valued attribute
##    attribute summary:
```

```
##    Mode   FALSE    TRUE    NA's
## logical       3      33       0
##
##  Role:
##    numeric valued attribute
##    attribute summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.194   3.000   3.000
##   vertex.names:
##    character valued attribute
##    36 valid vertex names
##
## No edge attributes
##
## Network edgelist matrix:
##        [,1] [,2]
##   [1,]    4    2
##   [2,]   10    2
##   [3,]   13    2
##   [4,]   14    2
##   [5,]   28    2
##   [6,]   29    2
##   [7,]    8    3
##   [8,]   12    3
##   [9,]    2    4
##  [10,]    7    4
##  [11,]   13    4
##  [12,]   20    4
##  [13,]   24    4
##  [14,]   27    4
##  [15,]   29    4
##  [16,]   35    4
##  [17,]   19    5
##  [18,]   24    5
##  [19,]    1    6
##  [20,]    3    6
##  [21,]    8    6
##  [22,]   21    6
##  [23,]   14    7
##  [24,]   24    7
##  [25,]   29    7
##  [26,]    3    8
##  [27,]   14    9
##  [28,]   21    9
##  [29,]   29    9
##  [30,]    2   10
##  [31,]   14   11
##  [32,]   18   11
##  [33,]   20   11
##  [34,]   22   11
##  [35,]   24   11
##  [36,]   29   11
##  [37,]   30   11
##  [38,]   33   11
```

```
## [39,]   34   11
## [40,]    3   12
## [41,]   35   12
## [42,]    4   13
## [43,]   16   13
## [44,]   18   13
## [45,]   20   13
## [46,]   29   13
## [47,]   30   13
## [48,]   11   14
## [49,]   15   14
## [50,]   29   14
## [51,]   33   14
## [52,]   34   14
## [53,]    2   15
## [54,]   14   15
## [55,]   29   15
## [56,]   34   15
## [57,]   13   16
## [58,]   19   16
## [59,]   31   16
## [60,]   35   16
## [61,]   36   16
## [62,]   11   18
## [63,]   13   18
## [64,]   27   18
## [65,]   29   18
## [66,]    5   19
## [67,]    6   19
## [68,]   16   19
## [69,]   21   19
## [70,]   24   19
## [71,]   30   19
## [72,]   31   19
## [73,]   35   19
## [74,]    4   20
## [75,]    9   20
## [76,]   11   20
## [77,]   13   20
## [78,]   21   20
## [79,]   22   20
## [80,]   24   20
## [81,]   29   20
## [82,]   33   20
## [83,]   34   20
## [84,]    6   21
## [85,]   20   21
## [86,]   24   21
## [87,]   26   21
## [88,]   29   21
## [89,]   20   22
## [90,]   29   23
## [91,]    4   24
## [92,]    7   24
```

```
##  [93,]   11   24
##  [94,]   19   24
##  [95,]   20   24
##  [96,]   23   24
##  [97,]   27   24
##  [98,]   29   24
##  [99,]   30   24
## [100,]   33   24
## [101,]   34   24
## [102,]   12   26
## [103,]   20   26
## [104,]   35   26
## [105,]    4   27
## [106,]   18   27
## [107,]   29   27
## [108,]    2   29
## [109,]    4   29
## [110,]    9   29
## [111,]   13   29
## [112,]   14   29
## [113,]   15   29
## [114,]   18   29
## [115,]   20   29
## [116,]   23   29
## [117,]   24   29
## [118,]   33   29
## [119,]   34   29
## [120,]   11   30
## [121,]   13   30
## [122,]   19   30
## [123,]   24   30
## [124,]   35   30
## [125,]   11   33
## [126,]   12   33
## [127,]   14   33
## [128,]   19   33
## [129,]   20   33
## [130,]   29   33
## [131,]   34   33
## [132,]   35   33
## [133,]   11   34
## [134,]   15   34
## [135,]   20   34
## [136,]   22   34
## [137,]   29   34
## [138,]   33   34
## [139,]    3   35
## [140,]    4   35
## [141,]   12   35
## [142,]   16   35
## [143,]   19   35
## [144,]   26   35
## [145,]   30   35
## [146,]   33   35
```

```
## [147,]   16   36
```

```r
# Degree
indeg_fr <- degree(silsys.fr.ilas, cmode = "indegree")
outdeg_fr <- degree(silsys.fr.ilas, cmode = "outdegree")
#Betweenness

bet_fr <- betweenness(silsys.fr.ilas, gmode = "digraph")

#Eigen Vector

eig_fr <- evcent(silsys.fr.ilas)

friendship_matrix <- cbind(indeg_fr,outdeg_fr,bet_fr,eig_fr)
friendship_matrix
```

```
##       indeg_fr outdeg_fr      bet_fr     eig_fr
##  [1,]        0         1   0.0000000 0.00521893
##  [2,]        6         4  90.8000000 0.12763229
##  [3,]        2         4  46.0285714 0.04106213
##  [4,]        8         7 101.6200216 0.24710489
##  [5,]        2         1   0.0000000 0.02122349
##  [6,]        4         2  46.6290043 0.03520181
##  [7,]        3         2   0.9761905 0.07582730
##  [8,]        1         2   0.0000000 0.01130669
##  [9,]        3         2   6.8494589 0.11750703
## [10,]        1         1   0.0000000 0.01892243
## [11,]        9         7  30.2018759 0.25394931
## [12,]        2         4  67.5548701 0.07752845
## [13,]        6         7  75.1633478 0.23063646
## [14,]        5         7  19.6637446 0.21676149
## [15,]        4         3   3.2500000 0.14235488
## [16,]        5         4  75.9323593 0.07984479
## [17,]        0         0   0.0000000 0.00000000
## [18,]        4         4   7.4932540 0.15354700
## [19,]        8         6 124.7024531 0.14315298
## [20,]       10        10 104.7531025 0.34008707
## [21,]        5         4  61.7252165 0.09428414
## [22,]        1         3   0.0000000 0.13120207
## [23,]        1         2   0.0000000 0.10627877
## [24,]       11         9 108.6484848 0.26435214
## [25,]        0         0   0.0000000 0.00000000
## [26,]        3         2   6.7833333 0.03665105
## [27,]        3         3   2.5321429 0.09859178
## [28,]        0         1   0.0000000 0.01892243
## [29,]       12        16 203.8999278 0.45250083
## [30,]        5         5  21.3567100 0.15493182
## [31,]        0         2   0.0000000 0.03306107
## [32,]        0         0   0.0000000 0.00000000
## [33,]        8         7  89.1487734 0.29229007
## [34,]        6         7  16.1000000 0.29092490
## [35,]        8         7 169.1871573 0.15292812
## [36,]        1         1   0.0000000 0.01183758
```

- Who are some of the most central individuals in the advice-seeking network? In the friendship network? Advice-seeking Network: Indegree Centrality: 5th and 19th nodes seem to be central according to the indegree centrality measure. Outdegree Centrality: 13th and 25th nodes seem to be central according to the outdegree centrality measure. Betweenness Centrality: 19th node seems to be central according to the betweenness centrality measure. Eigen vector Centrality: 25th node seems to be central according to the eigen vector centrality measure.

Friendship Network: Indegree Centrality: 29th node seems to be central according to the indegree centrality measure. Outdegree Centrality: 29th node seems to be central according to the outdegree centrality measure. Betweenness Centrality: 29th node seems to be central according to the betweenness centrality measure. Eigen vector Centrality: 29th node seems to be central according to the eigen vector centrality measure. In all, all the centrality measure for the 29th node indicate that it has a central position in the network.

(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:

```
cor(advice_matrix)
```

```
##            indeg_ad outdeg_ad    bet_ad    eig_ad
## indeg_ad  1.0000000 0.2047587 0.6534849 0.1608585
## outdeg_ad 0.2047587 1.0000000 0.5548932 0.8927812
## bet_ad    0.6534849 0.5548932 1.0000000 0.4282096
## eig_ad    0.1608585 0.8927812 0.4282096 1.0000000
```

```
cor(friendship_matrix)
```

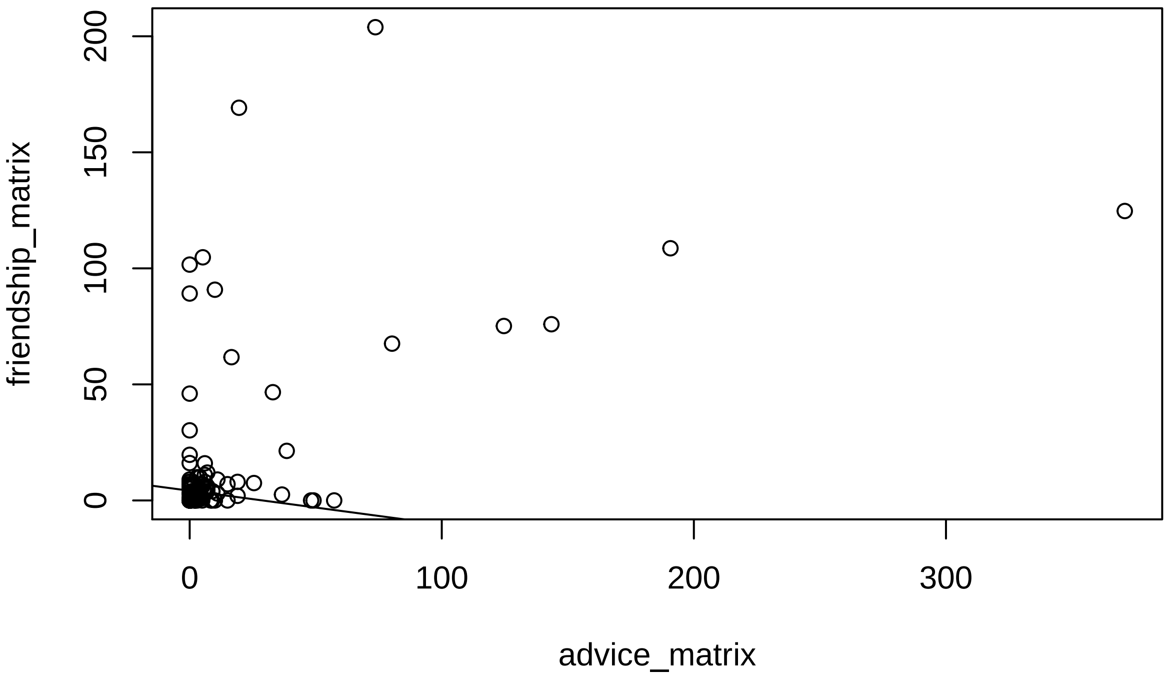
```
##            indeg_fr outdeg_fr    bet_fr    eig_fr
## indeg_fr  1.0000000 0.9109728 0.8230471 0.8834378
## outdeg_fr 0.9109728 1.0000000 0.8146123 0.9414224
## bet_fr    0.8230471 0.8146123 1.0000000 0.6628399
## eig_fr    0.8834378 0.9414224 0.6628399 1.0000000
```

```
cor(advice_matrix, friendship_matrix)
```

```
##            indeg_fr  outdeg_fr    bet_fr       eig_fr
## indeg_ad  0.1326214 0.02212860 0.2334395 -0.098459851
## outdeg_ad 0.1179450 0.14838411 0.2772911  0.094943928
## bet_ad    0.3551649 0.26241725 0.4448274  0.142671419
## eig_ad    0.0288127 0.09014183 0.2913855  0.006682372
```

```
plot(advice_matrix, friendship_matrix)
library(sna)
abline(lm(friendship_matrix ~ advice_matrix))
```

```
## Warning in abline(lm(friendship_matrix ~ advice_matrix)): only using the
## first two of 20 regression coefficients
```



* Does centrality in the advice-seeking network correspond (or not) to centrality in the friendship network? The centrality measures in the advice-seeking network does not seem to correlate much with the centrality measures in the friendship network. The highest correlation can be seen between betweenness of advice-seeking network and betweenness of friendhip network at 0.44. The second highest correlation is between the betweenness of advice-seeking network and indegree of friendship network.

- What centrality measures are most strong correlated? Least strongly correlated? For the advice-seeking network: Outdegree and eigen vector are strongly correlated and eigen vector and indegree are least strongly correlated.

For the Friendship network: Outdegree and eigen vector are strongly correlated and eigen vector and betweenness are least strongly correlated.

For the advice-seeking and friendship network: Betweennes of advice-seeking network and betweenness of friendship network are strongly correlated and eigen vector of friendship and eigen vector of advice-seeking are least strongly correlated.

(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.

```
perm.cor.test<-function(x,y,niter=5000){  #Define a simple test function
  c.obs<-cor(x,y,use="complete.obs")
  c.rep<-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="")
  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_p_value= mean(abs(c.rep)>=abs(c.obs))))
}

?get.vertex.attribute
Charisma_ad <- get.vertex.attribute(silsys.ad.ilas,"Charisma")
Potency_ad <- get.vertex.attribute(silsys.ad.ilas,"Potency")

Charisma_fr <- get.vertex.attribute(silsys.fr.ilas,"Charisma")
Potency_fr <- get.vertex.attribute(silsys.fr.ilas,"Potency")

Char_advice1 <- perm.cor.test(indeg_ad, Charisma_ad)
```

```
## Vector Permutation Test:
##  Observed correlation: 0.08059178    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.3466
##      Pr(rho<=obs): 0.6534
##      Pr(|rho|>=|obs|): 0.6558
```

```
Char_advice2 <- perm.cor.test(outdeg_ad, Charisma_ad)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.4556335    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.991
##      Pr(rho<=obs): 0.009
##      Pr(|rho|>=|obs|): 0.009
```

```
Char_advice3 <- perm.cor.test(bet_ad, Charisma_ad)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.1343587    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.811
##      Pr(rho<=obs): 0.189
##      Pr(|rho|>=|obs|): 0.422
```

```
Char_advice4 <- perm.cor.test(eig_ad, Charisma_ad)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.3875669    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.9816
##      Pr(rho<=obs): 0.0184
##      Pr(|rho|>=|obs|): 0.0192
```

```
Pot_advice1 <- perm.cor.test(indeg_ad, Potency_ad)
```

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

```r
Pot_advice2 <- perm.cor.test(outdeg_ad, Potency_ad)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.2492979    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.9292
##      Pr(rho<=obs): 0.0708
##      Pr(|rho|>=|obs|): 0.1412
```

```r
Pot_advice3 <- perm.cor.test(bet_ad, Potency_ad)
```

```
## Vector Permutation Test:
##  Observed correlation: 0.2411634 Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.0848
##      Pr(rho<=obs): 0.9152
##      Pr(|rho|>=|obs|): 0.1592
```

```r
Pot_advice4 <-perm.cor.test(eig_ad, Potency_ad)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.2113803    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.8938
##      Pr(rho<=obs): 0.1062
##      Pr(|rho|>=|obs|): 0.2108
```

```r
Char_friend1 <- perm.cor.test(indeg_fr, Charisma_fr)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.02825184   Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.5802
##      Pr(rho<=obs): 0.4198
##      Pr(|rho|>=|obs|): 0.8722
```

```r
Char_friend2 <- perm.cor.test(outdeg_fr, Charisma_fr)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.06023858   Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.652
##      Pr(rho<=obs): 0.348
##      Pr(|rho|>=|obs|): 0.7238
```

```r
Char_friend3 <- perm.cor.test(bet_fr, Charisma_fr)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.1143221    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.7614
##      Pr(rho<=obs): 0.2386
##      Pr(|rho|>=|obs|): 0.5004
```

```
Char_friend4 <- perm.cor.test(eig_fr, Charisma_fr)
```

```
## Vector Permutation Test:
##  Observed correlation: 0.0005403347  Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.5264
##      Pr(rho<=obs): 0.4736
##      Pr(|rho|>=|obs|): 0.9978
```

```
Pot_friend1 <- perm.cor.test(indeg_fr, Potency_fr)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.05101107   Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.6222
##      Pr(rho<=obs): 0.3778
##      Pr(|rho|>=|obs|): 0.7694
```

```
Pot_friend2 <- perm.cor.test(outdeg_fr, Potency_fr)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.1096314    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.758
##      Pr(rho<=obs): 0.242
##      Pr(|rho|>=|obs|): 0.51
```

```
Pot_friend3 <- perm.cor.test(bet_fr, Potency_fr)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.06166778   Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.6488
##      Pr(rho<=obs): 0.3512
##      Pr(|rho|>=|obs|): 0.7154
```

```
Pot_friend4 <- perm.cor.test(eig_fr, Potency_fr)
```

```
## Vector Permutation Test:
##  Observed correlation: -0.1364308    Replicate quantiles (niter=5000)
##      Pr(rho>=obs): 0.7884
##      Pr(rho<=obs): 0.2116
##      Pr(|rho|>=|obs|): 0.4216
```

```
rbind(Char_advice1,Char_advice2,Char_advice3,Char_advice4,Pot_advice1,Pot_advice2,Pot_advice3,Pot_advic
```

```
##              obs          rep          two_sided_p_value
## Char_advice1 0.08059178   Numeric,5000 0.6558
## Char_advice2 -0.4556335   Numeric,5000 0.009
## Char_advice3 -0.1343587   Numeric,5000 0.422
## Char_advice4 -0.3875669   Numeric,5000 0.0192
## Pot_advice1  0.5834776    Numeric,5000 0
## Pot_advice2  -0.2492979   Numeric,5000 0.1412
## Pot_advice3  0.2411634    Numeric,5000 0.1592
## Pot_advice4  -0.2113803   Numeric,5000 0.2108
## Char_friend1 -0.02825184  Numeric,5000 0.8722
## Char_friend2 -0.06023858  Numeric,5000 0.7238
## Char_friend3 -0.1143221   Numeric,5000 0.5004
## Char_friend4 0.0005403347 Numeric,5000 0.9978
## Pot_friend1  -0.05101107  Numeric,5000 0.7694
## Pot_friend2  -0.1096314   Numeric,5000 0.51
## Pot_friend3  -0.06166778  Numeric,5000 0.7154
## Pot_friend4  -0.1364308   Numeric,5000 0.4216
```

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

The relation of Charisma and Potency to the positional structure at Silicon Valley Systems can be found out using the p-value test. The null hypothesis considered here is that, there is no relationship between these attributes and the centrality measures. Firstly I considered those observation where a strong correlation exists.Considering the potency attribute and the indegree centrality measure of the advice-seeking network, the probability here is less than 0.05. Thus we can reject the null hypothesis. Thus it can be concluded that Potency attribute and the indegree cenrality measure of the advice-seeking network are strongly correlated.