

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

*Avanti Chande*

*Due: Thursday, January 19, 2017*

**Collaborators: Jay Chauhan, Gosuddin Siddiqi, Suchitra Sundararaman**

## **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 chunks, 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)
```

```
## Warning: package 'statnet' was built under R version 3.2.5
```

```
## Warning: package 'tergm' was built under R version 3.2.5
```

```
## Warning: package 'statnet.common' was built under R version 3.2.5
```

```
## Warning: package 'ergm' was built under R version 3.2.5
```

```
## Warning: package 'network' was built under R version 3.2.5
```

```
## Warning: package 'networkDynamic' was built under R version 3.2.5
```

```
## Warning: package 'ergm.count' was built under R version 3.2.5
```

```
## Warning: package 'sna' was built under R version 3.2.5
```

```
# Load data
```

```
load("problemset1_data.Rdata")
```

```
ls() # Print objects in workspace to see what is available
```

---

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

```
## [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<sup>1</sup>. 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”.

### (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
```

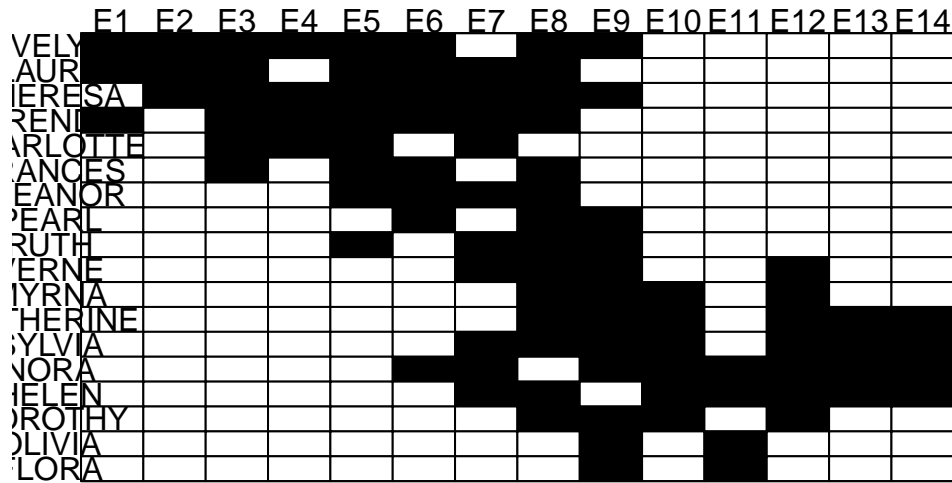
```
## starting httpd help server ...
```

```
## done
```

---

<sup>1</sup>Davis, Gardner, and Gardner. (1941) *Deep South*. Chicago: The University of Chicago Press.

```
plot.sociomatrix(sw.incidence)
```



Evelyn, Theresa and Nora seem particularly active. There is a division of the events these women are interested in. The upper half section of the women are clearly interested in events E1 to E9. Lower half of the women are interested in events E7 to E14. That said, there is also an overlap in the middle where women are interested in events E5 through E9.

## (b) One-Mode Projections

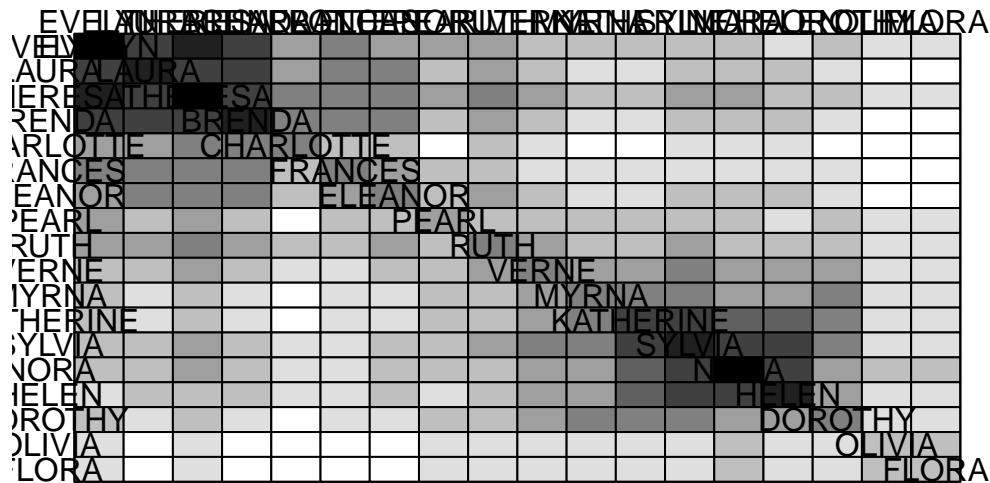
Consider how these women are connected through events. To do this, form the (valued) row projection of `sw.incidence` and say it as `sw.p2p`. You might find it helpful to know that `%*` is R's inner product operator, and `t()` is a function to transpose a matrix. `sw.p2p[i,j]` should now be the number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



```
?plot.sociomatrix
```

- What does the row projection tell us about how people are connected in this social group? The projection tells us that there are groups of women who attend the same events. And there are groups of women who don't attend the same events. There is a specially noticable difference between the upper and lower half sections of the women. Women in the center form an important part of the connection from the 'vitality' point of view. In addition, it tells us about the 'reach' of certain women in terms of similar interests they share which makes them attend the same events. (Hypothesis).
- Does the group seem to have subdivision? Yes. The group seems to have a clear subdivision. Evelyn, laura, theresa and Brenda make the most significant subdivision followed by Katherine, Sylvia, Nora and Helen.
- Do some members seem more "central" than others? If so, who? In terms of degree centrality, Evelyn and Theresa seem to be the most central. In terms of betweenness, Ruth seems to be central as she is the bridge between the two groups. Also, Helen and Nora have high betweenness. In terms of closeness, again, Evelyn, Theresa, Nora, Helen, Ruth, Verne seem to be more central than the others.

```
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

```

```
degree(sw.p2p, cmode="indegree")
```

```
## [1] 50 45 57 46 24 32 36 31 40 39 35 41 50 47 42 35 14 14
```

```
degree(sw.p2p, cmode="outdegree")
```

```
## [1] 50 45 57 46 24 32 36 31 40 39 35 41 50 47 42 35 14 14
```

```
betweenness(sw.p2p)
```

```

## [1] 2.7532468 0.7272727 2.7532468 0.7272727 0.0000000 0.7272727 0.7272727
## [8] 1.4545455 2.7532468 2.7532468 1.4545455 1.4545455 2.7532468 2.7532468
## [15] 2.7532468 1.4545455 0.0000000 0.0000000

```

```
closeness(sw.p2p)
```

```

## [1] 1.0000000 0.8947368 1.0000000 0.8947368 0.7391304 0.8947368 0.8947368
## [8] 0.9444444 1.0000000 1.0000000 0.9444444 0.9444444 1.0000000 1.0000000
## [15] 1.0000000 0.9444444 0.7727273 0.7727273

```

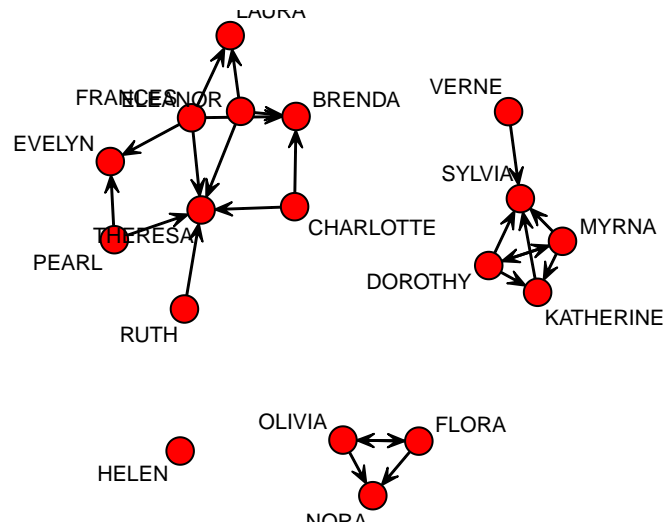
### (c) Entailment Structures

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
  }
}
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)
```



```
sw.r.entail
```

##	EVELYN	LAURA	THERESA	BREND	CHARLOTTE	FRANCES	ELEANOR	PEARL	RUTH
## EVELYN	1	0	0	0	0	0	0	0	0
## LAURA	0	1	0	0	0	0	0	0	0
## THERESA	0	0	1	0	0	0	0	0	0
## BREND	0	0	0	1	0	0	0	0	0
## CHARLOTTE	0	0	1	1	1	0	0	0	0
## FRANCES	1	1	1	1	0	1	0	0	0

## ELEANOR	0	1	1	1	0	0	1	0	0
## PEARL	1	0	1	0	0	0	0	1	0
## RUTH	0	0	1	0	0	0	0	0	1
## VERNE	0	0	0	0	0	0	0	0	0
## MYRNA	0	0	0	0	0	0	0	0	0
## KATHERINE	0	0	0	0	0	0	0	0	0
## SYLVIA	0	0	0	0	0	0	0	0	0
## NORA	0	0	0	0	0	0	0	0	0
## HELEN	0	0	0	0	0	0	0	0	0
## DOROTHY	0	0	0	0	0	0	0	0	0
## OLIVIA	0	0	0	0	0	0	0	0	0
## FLORA	0	0	0	0	0	0	0	0	0
##	VERNE	MYRNA	KATHERINE	SYLVIA	NORA	HELEN	DOROTHY	OLIVIA	FLORA
## EVELYN	0	0	0	0	0	0	0	0	0
## LAURA	0	0	0	0	0	0	0	0	0
## THERESA	0	0	0	0	0	0	0	0	0
## BRENDA	0	0	0	0	0	0	0	0	0
## CHARLOTTE	0	0	0	0	0	0	0	0	0
## FRANCES	0	0	0	0	0	0	0	0	0
## ELEANOR	0	0	0	0	0	0	0	0	0
## PEARL	0	0	0	0	0	0	0	0	0
## RUTH	0	0	0	0	0	0	0	0	0
## VERNE	1	0	0	1	0	0	0	0	0
## MYRNA	0	1	1	1	0	0	1	0	0
## KATHERINE	0	0	1	1	0	0	0	0	0
## SYLVIA	0	0	0	1	0	0	0	0	0
## NORA	0	0	0	0	1	0	0	0	0
## HELEN	0	0	0	0	0	1	0	0	0
## DOROTHY	0	1	1	1	0	0	1	0	0
## OLIVIA	0	0	0	0	1	0	0	1	1
## FLORA	0	0	0	0	1	0	0	1	1

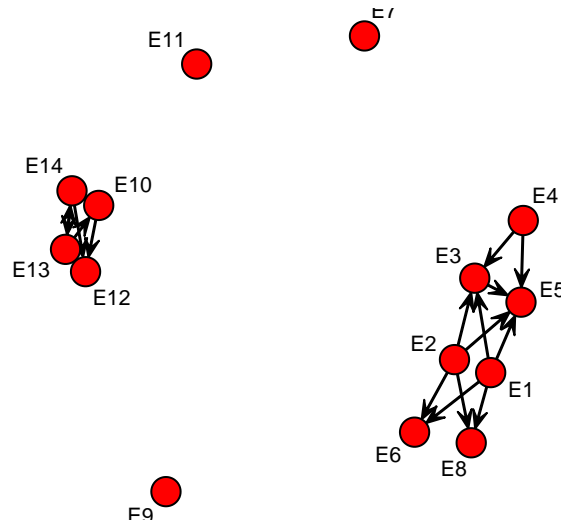
```

# 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 column-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? Row-wise Entailment: The path tells us the direction in which a particular woman has attended another woman's events. For example, the path between Myrna, Katherine and Sylvia suggest that Myrna has attended all of the events that Katherine has attended and Evelyn has attended all the events that Katherine has attended, but the opposite is not true.

Column-wise Entailment: The path in the column-wise entailment tells us that all who women who attend a particular event also attend another event. For example, all women who attend event 4 also attend event 3 and all women who attended event 3 also attended event 5. but the opposite is not true.

- What do mutual (i.e. bidirectional) dyads mean? For row wise entailment, mutual dyads mean that both women have attended the other's events. In other words, they attended the same events. For columnwise entailment, it means that the events were attended by the same women.
- What is special about isolates? Special thing about Isolates is that they are connected with both of the subdivisions of this social group. For row-wise entailment, Helen is an isolate. It means that she attends events that are attended by both subdivisions. For example, Evelyn attends some of Helen's events and Nora also attends some of Helen's events. This establishes a connection for Helen from a vitality and reach point of view.

For coulumn wise entailment, Events E7, E9 and E11 are isolates. It means that these events are attended by diverse women. We can say that these events are meeting points for the women from the upper-right and lower-left subdivisions of this social group.

## 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.<sup>2</sup> 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.

```
#Network 1 - Advise Seeking network - silsys.ad.ilas
#Indegree
indeg_silsys_ad_ilas = degree(silsys.ad.ilas, cmode="indegree")

#OutDegree
outdeg_silsys_ad_ilas = degree(silsys.ad.ilas, cmode="outdegree")
all(degree(silsys.ad.ilas) == indeg_silsys_ad_ilas+outdeg_silsys_ad_ilas)

## [1] TRUE

#Betweenness
bet_silsys_ad_ilas = betweenness(silsys.ad.ilas)

#Eigen Vector Centrality
evcent_silsys_ad_ilas = evcent(silsys.ad.ilas)

#Combining the vectors
centrality_scores_Ad = cbind(indeg_silsys_ad_ilas,outdeg_silsys_ad_ilas, bet_silsys_ad_ilas, evcent_sil.

colnames(centrality_scores_Ad) = c('Indegree', 'OutDegree', 'Betweenness', 'Eigen-Vector-Centrality')
centrality_scores_Ad
```

##	Indegree	OutDegree	Betweenness	Eigen-Vector-Centrality
## [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

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

```
## [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
```

```
#Network 2 - Friendship Ties network - silsys.fr.ilas
```

```
#Indegree
```

```
indeg_silsys_fr_ilas = degree(silsys.fr.ilas, cmode="indegree")
```

```
#OutDegree
```

```
outdeg_silsys_fr_ilas = degree(silsys.fr.ilas, cmode="outdegree")
```

```
all(degree(silsys.fr.ilas) == indeg_silsys_fr_ilas+outdeg_silsys_fr_ilas)
```

```
## [1] TRUE
```

```
#Betweenness
```

```
bet_silsys_fr_ilas = betweenness(silsys.fr.ilas)
```

```
#Eigen Vector Centrality
```

```
evcent_silsys_fr_ilas = evcent(silsys.fr.ilas)
```

```
#Combining the vectors
```

```
centrality_scores_Fr = cbind(indeg_silsys_fr_ilas,outdeg_silsys_fr_ilas, bet_silsys_fr_ilas, evcent_sil
```

```
colnames(centrality_scores_Fr) = c('Indegree', 'OutDegree', 'Betweenness', 'Eigen-Vector-Centrality')
```

```
centrality_scores_Fr
```

```
##      Indegree OutDegree Betweenness Eigen-Vector-Centrality
## [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?

The most central individual in the advice-seeking network are the 19th and 5th individuals according to indegree. The 19th individual also has the highest betweenness(370) and also a good eigen vector centrality measure. According to outdegree, 25th and 13th individuals are the most central. 25th individual also has a high eigen vector centrality.

The most central individual in the friendship network is the 29th individual. She has the highest indegree(12) and the highest outdegree(16). She also has the highest betweenness(203) and a high eigen vector centrality.

## (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(centrality_scores_Ad)
```

```
##               Indegree OutDegree Betweenness
## Indegree      1.0000000 0.2047587  0.6534849
## OutDegree      0.2047587 1.0000000  0.5548932
## Betweenness    0.6534849 0.5548932  1.0000000
## Eigen-Vector-Centrality 0.1608585 0.8927812  0.4282096
##               Eigen-Vector-Centrality
## Indegree              0.1608585
## OutDegree              0.8927812
## Betweenness            0.4282096
## Eigen-Vector-Centrality 1.0000000
```

```
cor(centrality_scores_Fr)
```

```
##               Indegree OutDegree Betweenness
## Indegree      1.0000000 0.9109728  0.8230471
## OutDegree      0.9109728 1.0000000  0.8146123
## Betweenness    0.8230471 0.8146123  1.0000000
```

```
## Eigen-Vector-Centrality 0.8834378 0.9414224 0.6628399
##                               Eigen-Vector-Centrality
## Indegree                    0.8834378
## OutDegree                    0.9414224
## Betweenness                  0.6628399
## Eigen-Vector-Centrality      1.0000000
```

```
cor(centrality_scores_Fr, centrality_scores_Ad)
```

```
##                               Indegree OutDegree Betweenness
## Indegree                    0.13262144 0.11794496 0.3551649
## OutDegree                    0.02212860 0.14838411 0.2624173
## Betweenness                  0.23343953 0.27729110 0.4448274
## Eigen-Vector-Centrality     -0.09845985 0.09494393 0.1426714
##                               Eigen-Vector-Centrality
## Indegree                    0.028812701
## OutDegree                    0.090141827
## Betweenness                  0.291385454
## Eigen-Vector-Centrality      0.006682372
```

- Does centrality in the advice-seeking network correspond (or not) to centrality in the friendship network?
  1. The centrality measures in the advises-seeking network correspond to the same in the friendship network in some cases and doesnot correspond in some other cases. For example, indegree and betweenness in the advise-seeking network are correlated, the same are strongly correlated in the friendship network.
  2. However, eigen-vector centrality and indegree are not correlated in the advise-seeking network, but highly correlated in the friendship network.
  3. Similarly, outdegree and indegree are strongly correlated in the friendship network, but not correlated in the advise-seeking network. This means that the those who advise, do not go out seeking advise. But those who're friends with many people, also seek out new friendship ties.
- What centrality measures are most strong correlated? Least strongly correlated? When considering within the friendship network centralities, eigen vector centrality and outdegree are the most strongly correlated. (0.94) and Eigen vector centrality and betweenness are the least strongly correlated.

When considering within the advising network centralities, Eigenvector centrality and outdegree are the most strongly correlated, and Indegree and Eigen vector centrality are the least strongly correlated.

When considering between networks centralities, both their betweenness are the most strongly correlated; whereas both their eigen vector centralities are the 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.

```
charisma = get.vertex.attribute(silsys.fr.ilas, 'Charisma')
potency = get.vertex.attribute(silsys.fr.ilas, 'Potency')
charisma
```

```
## [1] 3.000 5.281 3.625 4.838 3.843 4.875 4.468 2.258 4.906 4.656 3.986
```

```
## [12] 2.218 0.000 3.516 3.566 5.290 4.906 4.468 4.875 4.593 4.482 4.218
## [23] 2.500 0.000 3.354 5.000 4.187 4.281 5.625 4.281 3.892 2.225 4.625
## [34] 4.187 0.000 4.800
```

potency

```
## [1] 2.193 2.843 2.750 3.612 6.187 5.625 3.531 2.354 2.871 3.437 2.531
## [12] 1.718 0.000 2.580 2.766 6.225 5.322 4.483 6.875 3.687 3.586 3.093
## [23] 1.687 0.000 2.645 3.031 5.031 2.812 5.062 4.125 3.200 1.741 2.906
## [34] 2.937 0.000 6.069
```

```
perm.cor.test<-function(x,y,alternative ="two-sided", 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")

  pvalue <- mean(abs(c.rep)>=abs(c.obs))

  result <- c(c.obs, pvalue)
  return(result)
}
```

*#Charisma and Centrality in Friendship Network*

```
p1 = perm.cor.test(charisma, indeg_silsys_fr_ilas)
p2 = perm.cor.test(charisma, outdeg_silsys_fr_ilas)
p3 = perm.cor.test(charisma, bet_silsys_fr_ilas)
p4 = perm.cor.test(charisma, evcent_silsys_fr_ilas)
```

*#Potency and Centrality in Friendship Network*

```
p5 = perm.cor.test(potency, indeg_silsys_fr_ilas)
p6 = perm.cor.test(potency, indeg_silsys_fr_ilas)
p7 = perm.cor.test(potency, bet_silsys_fr_ilas)
p8 = perm.cor.test(potency, evcent_silsys_fr_ilas)
```

*#Charisma and Centrality in Advising Network*

```
p13 = perm.cor.test(charisma, indeg_silsys_ad_ilas)
p14 = perm.cor.test(charisma, outdeg_silsys_ad_ilas)
p15 = perm.cor.test(charisma, bet_silsys_ad_ilas)
p16 = perm.cor.test(charisma, evcent_silsys_ad_ilas)
```

*#Potency and Centrality in Advising Network*

```
p9 = perm.cor.test(potency, indeg_silsys_ad_ilas)
p10 = perm.cor.test(potency, outdeg_silsys_ad_ilas)
p11 = perm.cor.test(potency, bet_silsys_ad_ilas)
p12 = perm.cor.test(potency, evcent_silsys_ad_ilas)
```

*#Combining them in a Table*

```
table = data.frame(rbind(p1,p2,p3,p4,p5,p6,p7,p8,p13,p14,p15,p16,p9,p10,p11,p12))
table
```

```
##          X1      X2
```

```
## p1 -0.0282518358 0.8636
## p2 -0.0602385845 0.7166
## p3 -0.1143221278 0.5194
## p4 0.0005403347 0.9980
## p5 -0.0510110711 0.7590
## p6 -0.0510110711 0.7702
## p7 -0.0616677795 0.7202
## p8 -0.1364308262 0.4252
## p13 0.0805917794 0.6364
## p14 -0.4556335297 0.0094
## p15 -0.1343586873 0.4068
## p16 -0.3875668713 0.0200
## p9 0.5834775737 0.0002
## p10 -0.2492979455 0.1424
## p11 0.2411633798 0.1588
## p12 -0.2113802600 0.2190
```

```
colnames(table) = c("Observed Correlation", "Two-sided Pvalue")
row.names(table) = c("charisma_fr_indegree", "charisma_fr_outdegree", "charisma_fr_betweenness", "charisma_fr_eigenvectorcen", "potency_fr_indegree", "potency_fr_outdegree", "potency_fr_betweenness", "potency_fr_eigenvectorcen", "charisma_ad_indegree", "charisma_ad_outdegree", "charisma_ad_betweenness", "charisma_ad_eigenvectorcen", "potency_ad_indegree", "potency_ad_outdegree", "potency_ad_betweenness", "potency_ad_eigenvectorcen")
```

```
table
```

##	Observed Correlation	Two-sided Pvalue
## charisma_fr_indegree	-0.0282518358	0.8636
## charisma_fr_outdegree	-0.0602385845	0.7166
## charisma_fr_betweenness	-0.1143221278	0.5194
## charisma_fr_eigenvectorcen	0.0005403347	0.9980
## potency_fr_indegree	-0.0510110711	0.7590
## potency_fr_outdegree	-0.0510110711	0.7702
## potency_fr_betweenness	-0.0616677795	0.7202
## potency_fr_eigenvectorcen	-0.1364308262	0.4252
## charisma_ad_indegree	0.0805917794	0.6364
## charisma_ad_outdegree	-0.4556335297	0.0094
## charisma_ad_betweenness	-0.1343586873	0.4068
## charisma_ad_eigenvectorcen	-0.3875668713	0.0200
## potency_ad_indegree	0.5834775737	0.0002
## potency_ad_outdegree	-0.2492979455	0.1424
## potency_ad_betweenness	0.2411633798	0.1588
## potency_ad_eigenvectorcen	-0.2113802600	0.2190

- How to charisma and potency appear to relate to positional structure at Silicon Valley Systems? It's interesting to see that the potency is correlated with the advertising structure's indegree centralities (0.58) at silicon valley systems. This means that potency matters in the advise seeking network. There is also a slight negative correlation between charisma and outdegree of the advise seeking network, but we can't reject our null hypothesis that there is no association between charisma and outdegree since the two-sided p-value is greater than 0.05.(0.09)