Social Network Analysis

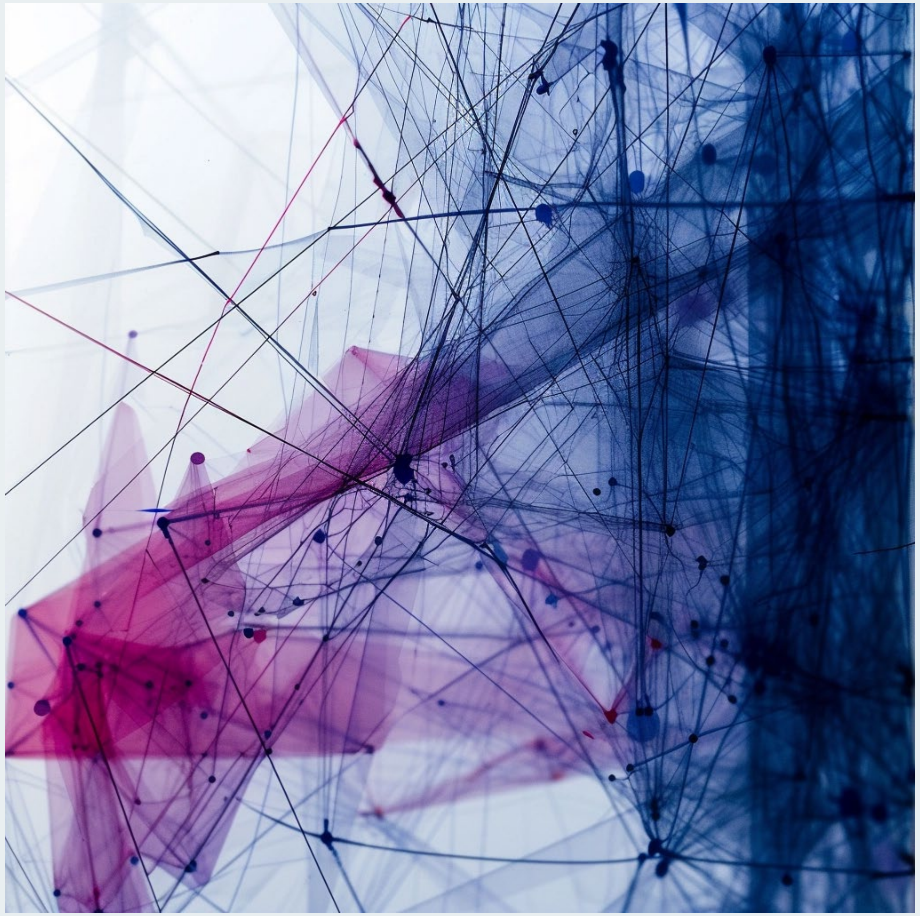
Quadratic Assignment Procedure

Dr. Chun-Hsiang Chan

Department of Geography,
National Taiwan Normal University

Photo credit: midjourney

Outline



Quadratic Assignment Procedure (QAP) Regression

Comparing multiple networks: QAP regression

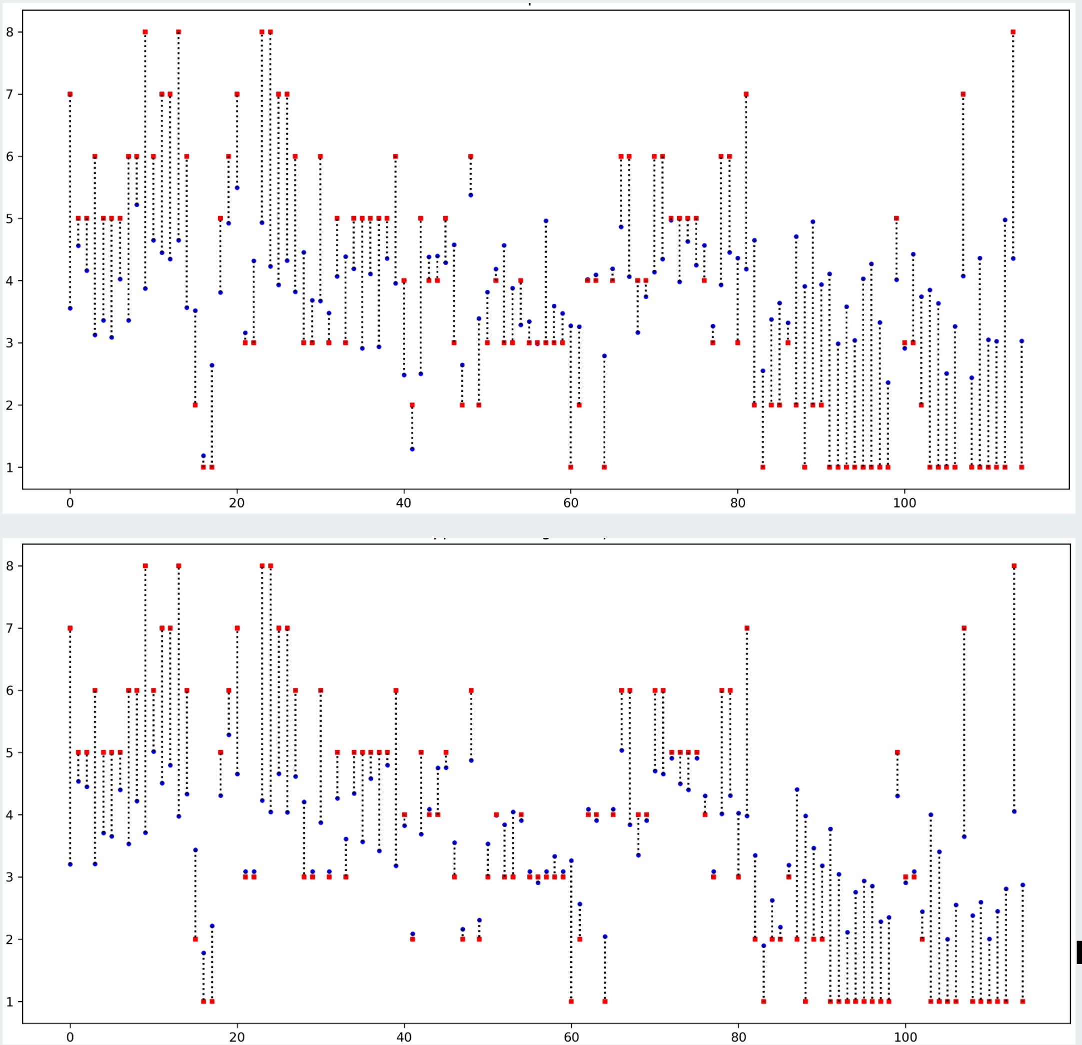
- The substantive question is how one set of relations (or dyadic attributes) relates to another.
- Are country relations correlated with trade networks?
- Do in-person contacts associated with friendship relations from social media?

Issue: QAP as a Graph Regression

- We attempt to understand the potential factors (independent variables) that formulate the target dyadic relations (dependent variables).
- As a regression problem, why do not we use ordinary least squared (OLS) to model this question? If it is a binary problem, shall we use logistic regression to solve this question?
- Why not?

Residuals of OLS

– **Zero mean:**
  $E(u) = 0$
– **Homoskedasticity:**
  $var(u) = \sigma^2, where\ \sigma \in constant$
– **Non-autocorrelation:**
  $cov(ut, ut - s) = 0, \forall s \neq 0$
– **Orthogonality:**
  $cov(x, u) = 0, for\ any\ I$
– **Normality**



Regression in a Graph

– The basic idea is that you create a dataset consisting of "dyad" or "pair" as an unit of analysis.

– The independent variables comprise attributes of one or both members within the pairs, as well as similarities and/or matches between the pairs.

Example of Friendship Ratings

(from William Simpson)

Person	A	B	C	D	E
A	.	0	2	3	1
B	4	.	8	10	6
C	5	5	.	5	5
D	2	8	7	.	3
E	2	4	3	5	.

Example of Friendship Ratings
(from William Simpson)

Pair	Row Number	Column Number	Absolute value of age difference	Friendship Rating
AA	1	1	.	.
AB	1	2	5	0
AC	1	3	25	2
AD	1	4	35	3
AE	1	5	15	1
BA	2	1	5	4

Statistical Problems

– All these factors are likely to affect the probability of co-sponsorship. Politicians, having had more time to cultivate relationships, may engage more actively in co-sponsorship. Additionally, politicians sharing similar ideologies and belonging to the same political party are more inclined to cosponsor bills together.

Statistical Problems

- The challenge lies in the interdependence of observations. For instance, if A cosponsors with B, and B cosponsors with C, it's probable that A may also cosponsor with C.
- Furthermore, the presence of repeated observations introduces correlation among errors. **Observations within rows or columns tend to be closely linked**, thereby either inflating or deflating standard errors.

Statistical Problems

– We might opt for a random effects model, necessitating the modeling and estimation of the covariance matrix.

– However, its **validity** hinges on the accuracy of our model estimation — a formidable challenge to overcome.

Handling Non-independent Observations
(from William Simpson)

- **Fixed effects**
  - Requires dummy for each row and column
  - Maybe have inefficient or parameters may not be estimable
- **Random effects (Generalized least squares)**
  - Requires modeling and estimating covariance matrix
  - If model is wrong, estimates may be inefficent and standard errors may be incorrect

Handling Non-independent Observations
(from William Simpson)

- **Empirical Standard Errors**
  - Use estimation procedure based on **independence** (e.g., OLS), but adjust standard errors
  - In QAP, **standard errors** are estimated by using **permutations of the dataset**

Sample Matrix Permuation

– Values **sharing a row/ column** in the **original** data will **share a row/ Column** in the **permuted** data

– **Diagonal elements** will be moved but still be on the **diagonal locations**

– **Dependent variable** values have been **separated** from the **corresponding independent variables**

Sample Matrix Permuation

Original Matrix

Row/ Column	1	2	3	4
1	$Y_{1,1}$	$Y_{1,2}$	$Y_{1,3}$	$Y_{1,4}$
2	$Y_{2,1}$	$Y_{2,2}$	$Y_{2,3}$	$Y_{2,4}$
3	$Y_{3,1}$	$Y_{3,2}$	$Y_{3,3}$	$Y_{3,4}$
4	$Y_{4,1}$	$Y_{4,2}$	$Y_{4,3}$	$Y_{4,4}$

Sample Matrix Permuation

Original Matrix → (1 ← 3)

Row/ Column	1	2	3	4
1	$\mathbf{Y_{1,3}}$	$\mathbf{Y_{1,2}}$	$\mathbf{Y_{1,1}}$	$\mathbf{Y_{1,4}}$
2	$\mathbf{Y_{2,3}}$	$Y_{2,2}$	$\mathbf{Y_{2,1}}$	$Y_{2,4}$
3	$\mathbf{Y_{3,3}}$	$\mathbf{Y_{3,2}}$	$\mathbf{Y_{3,1}}$	$\mathbf{Y_{3,4}}$
4	$\mathbf{Y_{4,3}}$	$Y_{4,2}$	$\mathbf{Y_{4,1}}$	$Y_{4,4}$

Sample Matrix Permuation

Original Matrix → (1 ←3) Column Exchange

Row/ Column	1	2	3	4
1	**$Y_{1,1}$**	$Y_{1,2}$	**$Y_{1,3}$**	$Y_{1,4}$
2	**$Y_{2,1}$**	$Y_{2,2}$	**$Y_{2,3}$**	$Y_{2,4}$
3	**$Y_{3,1}$**	$Y_{3,2}$	**$Y_{3,3}$**	$Y_{3,4}$
4	**$Y_{4,1}$**	$Y_{4,2}$	**$Y_{4,3}$**	$Y_{4,4}$

Sample Matrix Permuation

Original Matrix ➔ (1 ←3) Row Exchange

Row/ Column	3	2	1	4
3	$Y_{3,3}$	$Y_{3,2}$	$Y_{3,1}$	$Y_{3,4}$
2	$Y_{2,3}$	$Y_{2,2}$	$Y_{2,1}$	$Y_{2,4}$
1	$Y_{1,3}$	$Y_{1,2}$	$Y_{1,1}$	$Y_{1,4}$
4	$Y_{4,3}$	$Y_{4,2}$	$Y_{4,1}$	$Y_{4,4}$

Sample Matrix Permuation

Original Matrix ➔ (1←3) ➔ (2 ←2)

Row/ Column	3	2	1	4
3	$Y_{3,3}$	$Y_{3,2}$	$Y_{3,1}$	$Y_{3,4}$
2	$Y_{2,3}$	$Y_{2,2}$	$Y_{2,1}$	$Y_{2,4}$
1	$Y_{1,3}$	$Y_{1,2}$	$Y_{1,1}$	$Y_{1,4}$
4	$Y_{4,3}$	$Y_{4,2}$	$Y_{4,1}$	$Y_{4,4}$

Sample Matrix Permuation

Original Matrix ➔ (1🡐3) ➔ (2 🡐2) ➔ (3🡐4)

Row/ Column	3	2	1	4
3	$Y_{3,3}$	$Y_{3,2}$	$Y_{3,1}$	$Y_{3,4}$
2	$Y_{2,3}$	$Y_{2,2}$	$Y_{2,1}$	$Y_{2,4}$
1	$Y_{1,3}$	$Y_{1,2}$	$Y_{1,1}$	$Y_{1,4}$
4	$Y_{4,3}$	$Y_{4,2}$	$Y_{4,1}$	$Y_{4,4}$

Sample Matrix Permuation

Original Matrix → (1←3) → (2 ←2) → (3←4)

Row/ Column	3	2	**4**	**1**
3	$Y_{3,3}$	$Y_{3,2}$	$Y_{3,4}$	$Y_{3,1}$
2	$Y_{2,3}$	$Y_{2,2}$	$Y_{2,4}$	$Y_{2,1}$
4	$Y_{4,3}$	$Y_{4,2}$	$Y_{4,4}$	$Y_{4,1}$
1	$Y_{1,3}$	$Y_{1,2}$	$Y_{1,4}$	$Y_{1,1}$

Sample Matrix Permuation

Original Matrix ➔ (1🡐3) ➔ (2 🡐2) ➔ (3🡐4) ➔ (4🡐1)???

Row/ Column	3	2	4	1
3	$Y_{3,3}$	$Y_{3,2}$	$Y_{3,4}$	$Y_{3,1}$
2	$Y_{2,3}$	$Y_{2,2}$	$Y_{2,4}$	$Y_{2,1}$
4	$Y_{4,3}$	$Y_{4,2}$	$Y_{4,4}$	$Y_{4,1}$
1	$Y_{1,3}$	$Y_{1,2}$	$Y_{1,4}$	$Y_{1,1}$

Do it by yourself!

Sample Matrix Permuation

– It have preserved any dependence among elements of the same row and column, but have eliminated any relationship between the dependent variable and the independent variable.

Sample Matrix Permuation

– Permute the dependent variable and merge back with independent variables

– Run the estimation with the new merged dataste, and save the results

– Repeat the permutation and estimation to generate an empirical sampling distribution

Procedure of QAP

	A	B	C
A	-	2	3
B	9	-	5
C	1	5	-

Friendship network

~

	A	B	C
A	-	2	3
B	9	-	5
C	1	5	-

Instagram network

+

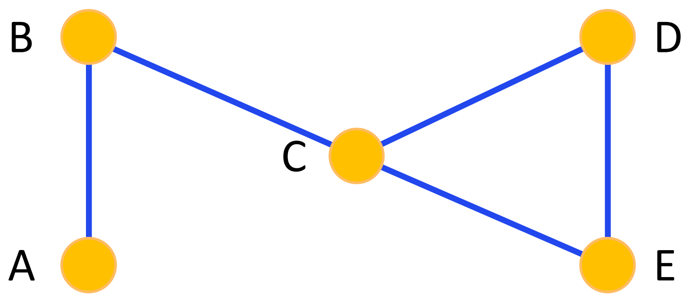
	A	B	C
A	-	2	3
B	9	-	5
C	1	5	-

Threads network

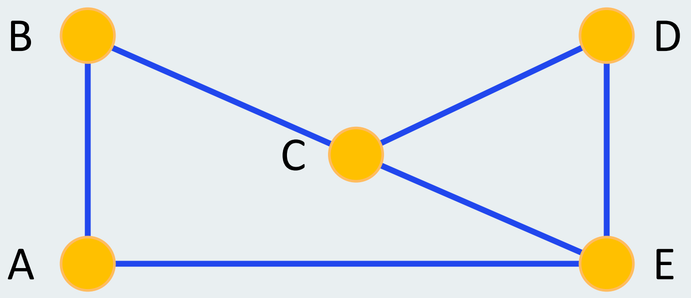
- Regression on response and predictors
- Permute response variables several times to crearte random datasets
  - These give sampling distribution of null hypothesis
  - Preserves dependence between dyads (person A's values stay together during permutation)
  - But removes the relationship between response/ predictors

Procedure of QAP

– Demo of significant test

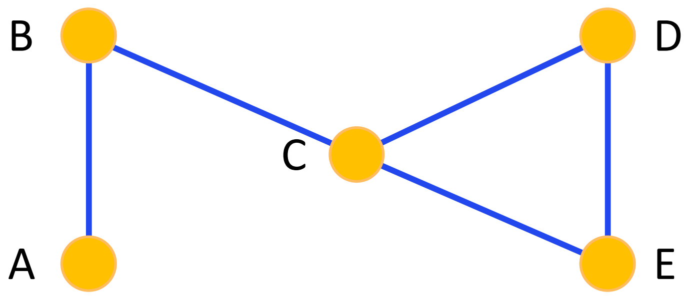


Friendship network

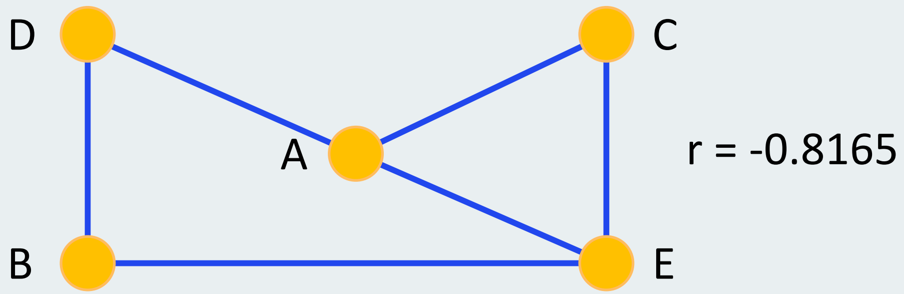


Instagram network

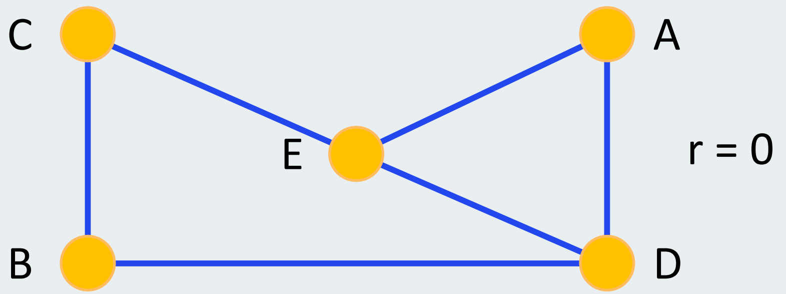
Procedure of QAP



Friendship network



Instagram network

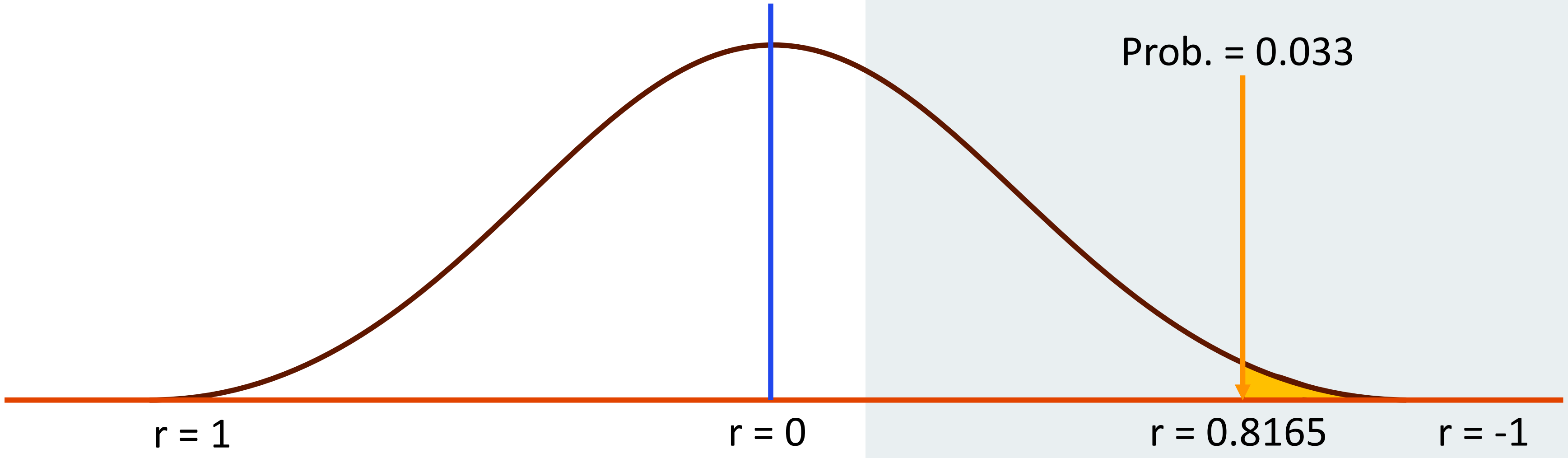


– There are 120 (5!) possible mappings of the five actors to the five actors on the graph.

Procedure of QAP

Measure of similarity (r)	Number of permutaions	Percent
0.8165	4	3.3
0.4082	32	26.7
0.0000	48	40.0
-0.4082	32	26.7
-0.8165	4	3.3
Weighted Average: 0.0	Total: 120	100

Procedure of QAP



– We observed that only 3.3% of occurrence based on random assignment of actors to nodes. The conclusion could be summaried that the results are significant at the 0.033 level against the null hypothesis of random assignment.

QAP Regression

- In information retrieval (IR), it's typical to mitigate some of these challenges by clustering based on a single variable. However, in our case, we have two variables—rows and columns—that require clustering.
- In the Quadratic Assignment Procedure (QAP) for network analysis, standard errors are estimated through permutations of the dataset.

QAP Regression

- Essentially, the QAP "scrambles" the dependent variable data by conducting numerous permutations. Through this process of repeated scrambling, multiple random datasets with the dependent variable are generated, enabling multiple analyses.
- These datasets and analyses collectively constitute an empirical sampling distribution. We can then compare our coefficient with this sampling distribution of coefficients derived from all the permuted datasets.

QAP Regression

– (Note that in the QAP, both rows and columns are permuted. However, for a single node, the row and column remain consistent and are permuted synchronously, ensuring that the relationship between rows and columns for a single node is maintained.)

– Essentially, this process preserves the dependencies within rows/columns while eliminating the association between the dependent and independent variables.

R Code :: QAP Regression

```r
# import package
library(sna)

# load data
load('~/Documents/python/social_network_analysis/studentnets.mrqap173.rda')
# print all data names
ls()
```

– Import “sna” library

– Load data from your local path

– Print all names

```
> ls()
[1] "m173_sem1_FRN"   "m173_sem1_GND"   "m173_sem1_RCE"   "m173_sem1_SEAT" "m173_sem1_SSL"   "m173_sem1_TSL"
[7] "m173_sem2_SSL"   "m173_sem2_TSL"
```

R Code :: QAP Regression

```r
# transform into an matrix
m173_sem1_SSL <- as.matrix(m173_sem1_SSL)
m173_sem1_TSL <- as.matrix(m173_sem1_TSL)
m173_sem1_FRN <- as.matrix(m173_sem1_FRN)
m173_sem1_SEAT <- as.matrix(m173_sem1_SEAT)
m173_sem1_RCE <- as.matrix(m173_sem1_RCE)
m173_sem1_GND <- as.matrix(m173_sem1_GND)

m173_sem2_SSL <- as.matrix(m173_sem2_SSL)
m173_sem2_TSL <- as.matrix(m173_sem2_TSL)
```

– SSL: social interactions per hour

– TSL: task interactions per hour

– FRN: friendship (2=bestfriend; 1=friend; 0=not friend)

– SEAT: who sits next to whom (2=faces; 1=behind; 0=not adjacent)

– RCE: race homophily

– GND: gender homophily

R Code :: QAP Regression

```r
# create a predictor matrix
predictor_matrices <- array(NA, c(6, length(m173_sem1_SSL[1,]),length(m173_sem1_SSL[1,])))
# OR: predictor_matrices <- array(NA, c(6, 26,26))

predictor_matrices[1,,] <- m173_sem1_SSL
predictor_matrices[2,,] <- m173_sem1_TSL
predictor_matrices[3,,] <- m173_sem1_FRN
predictor_matrices[4,,] <- m173_sem1_SEAT
predictor_matrices[5,,] <- m173_sem1_RCE
predictor_matrices[6,,] <- m173_sem1_GND
```

– Create an empty matrix and store all predictors

R Code :: QAP Regression

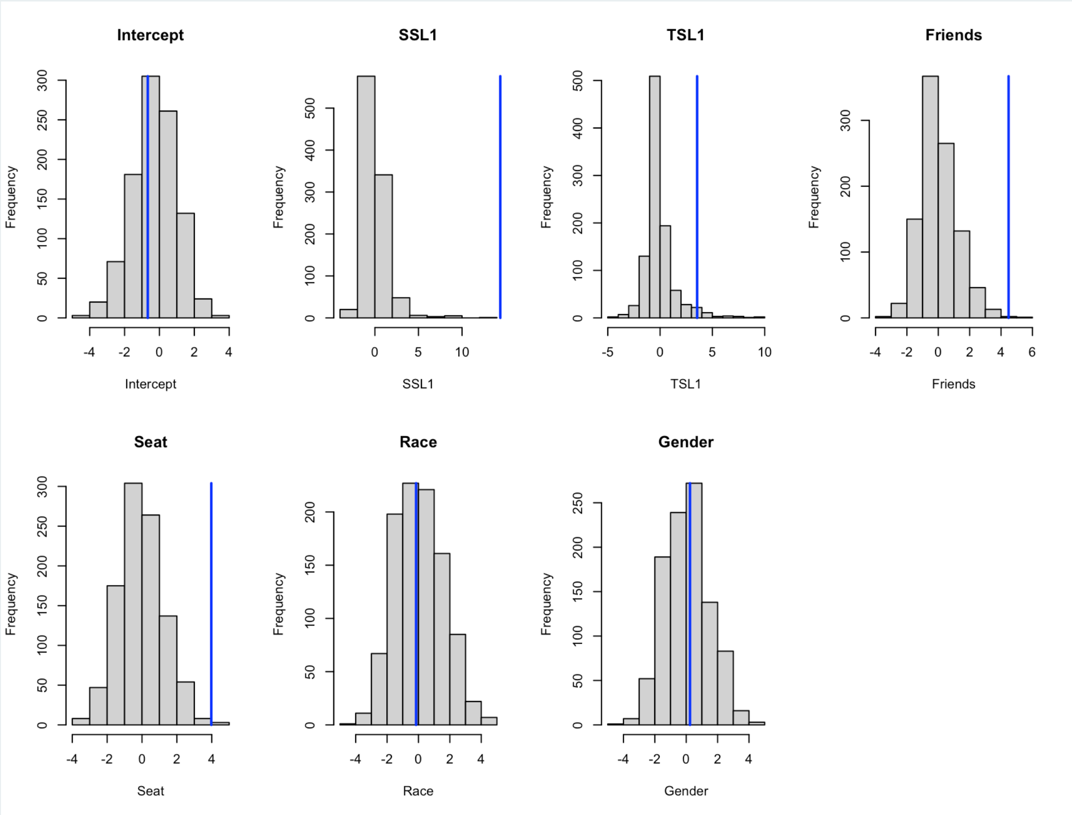
```r
# Fit a netlm model: the response matrix and the array of predictor matrices
nl<-netlm(m173_sem2_SSL, predictor_matrices)

# Make the model easier to read
nlLabeled <- list()
nlLabeled <- summary(nl)

# adding labels
nlLabeled$names <- c("Intercept", "SSL1", "TSL1", "Friends", "Seat","Race","Gender")

# Round the coefficients to two decimals
nlLabeled$coefficients = round(nlLabeled$coefficients, 2)
nlLabeled
```

– Use “netlm” to fit the model

R Code :: QAP Regression

```
OLS Network Model

Residuals:
          0%          25%          50%          75%         100% 
-1.652583881 -0.067206384  0.008678721  0.015216870  2.924942741 

Coefficients:
          Estimate Pr(<=b) Pr(>=b) Pr(>=|b|)
Intercept -0.02    0.383   0.617   0.623    
SSL1       0.45    1.000   0.000   0.000    
TSL1       0.03    0.960   0.040   0.043    
Friends    0.16    0.998   0.002   0.002    
Seat       0.08    0.999   0.001   0.001    
Race       0.00    0.518   0.482   0.932    
Gender     0.01    0.572   0.428   0.860    

Residual standard error: 0.3437 on 643 degrees of freedom
Multiple R-squared: 0.3817      Adjusted R-squared: 0.3759 
F-statistic: 66.16 on 6 and 643 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qap 
        Replications: 1000 
        Coefficient Distribution Summary:

       Intercept       SSL1       TSL1    Friends       Seat       Race     Gender
Min    -4.832186  -2.982899  -5.801952  -3.572400  -3.743904  -3.941144  -3.612336
1stQ   -1.199016  -0.858886  -0.817853  -0.813903  -0.949776  -1.271173  -0.996107
Median -0.231007  -0.183510  -0.376050  -0.150728  -0.157940  -0.237455  -0.007734
Mean   -0.320014  -0.039619  -0.070400  -0.037192  -0.041890  -0.107703   0.015904
3rdQ    0.640560   0.417493   0.126825   0.710464   0.814286   0.998863   0.964724
Max     3.818295  10.274157  10.821725   5.685191   4.964186   5.636270   4.103439
```

```r
# plot distribution
par(mfrow = c(2, 4))
for(i in 1:7){
  hist_obj<-hist(nlLabeled$dist[,i], main = nlLabeled$names[i],
                 xlab = nlLabeled$names[i])
  lines(cbind(nlLabeled$tstat[i], nlLabeled$tstat[i]),
        cbind(0, max(hist_obj$counts)), col='blue', lwd=2)
}
```



R Code :: QAP Regression

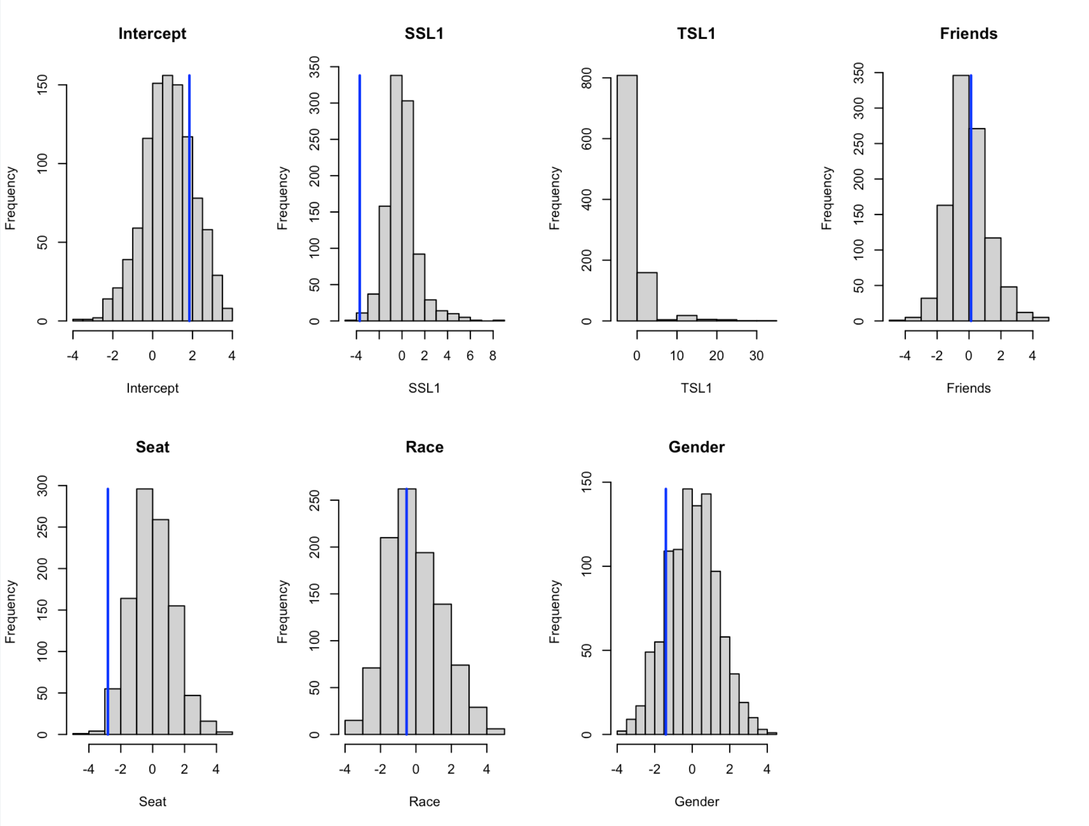
```r
# Fit a netlm model: the response matrix and the array of predictor matrices
n2<-netlm(m173_sem2_TSL, predictor_matrices)

# Make the model easier to read
n2Labeled <- list()
n2Labeled <- summary(n2)

# adding labels
n2Labeled$names <- c("Intercept", "SSL1", "TSL1", "Friends", "Seat","Race","Gender")

# Round the coefficients to two decimals
n2Labeled$coefficients = round(n2Labeled$coefficients, 2)
n2Labeled
```

– Fit another model (the 2nd semester)

R Code :: QAP Regression

```
OLS Network Model

Residuals:
         0%         25%         50%         75%        100% 
-6.79570345 -0.10044585 -0.00923077  0.02628499  7.90740702 

Coefficients:
          Estimate Pr(<=b) Pr(>=b) Pr(>=|b|)
Intercept  0.10    0.828   0.172   0.201    
SSL1      -0.28    0.004   0.996   0.023    
TSL1       1.01    1.000   0.000   0.000    
Friends    0.01    0.570   0.430   0.897    
Seat      -0.14    0.010   0.990   0.026    
Race      -0.04    0.391   0.609   0.741    
Gender    -0.09    0.157   0.843   0.294    

Residual standard error: 0.8241 on 643 degrees of freedom
Multiple R-squared: 0.756      Adjusted R-squared: 0.7538 
F-statistic: 332.1 on 6 and 643 degrees of freedom, p-value:     0 


Test Diagnostics:

        Null Hypothesis: qap 
        Replications: 1000 
        Coefficient Distribution Summary:

       Intercept      SSL1      TSL1   Friends      Seat      Race    Gender
Min    -4.632260 -4.644953 -1.945282 -3.758587 -3.962186 -3.992176 -3.763148
1stQ   -0.215220 -0.763662 -1.030956 -0.799513 -0.821721 -1.203967 -0.976938
Median  0.630094 -0.095071 -0.653983 -0.047572 -0.085257 -0.078445 -0.031490
Mean    0.633431 -0.004339  0.110588  0.061317  0.004628 -0.006635 -0.038267
3rdQ    1.524721  0.582863 -0.108548  0.764359  0.807309  1.113105  0.851510
Max     4.981508  8.608883 28.588922  5.534586  4.023486  4.267597  4.128071
```

```r
# plot distribution
par(mfrow = c(2, 4))
for(i in 1:7){
  hist_obj<-hist(n2Labeled$dist[,i], main = n2Labeled$names[i],
                 xlab = n2Labeled$names[i])
  lines(cbind(n2Labeled$tstat[i], n2Labeled$tstat[i]),
       cbind(0, max(hist_obj$counts)), col='blue', lwd=2)
}
```



References

– "QAP – the Quadratic Assignment Procedure" (William Simpson, Harvard Business School, 2001).
– Decker, Krackhardt, Snijders, "Sensitivity of MRQAP Tests to Collinearity and Autocorrelation Conditions"

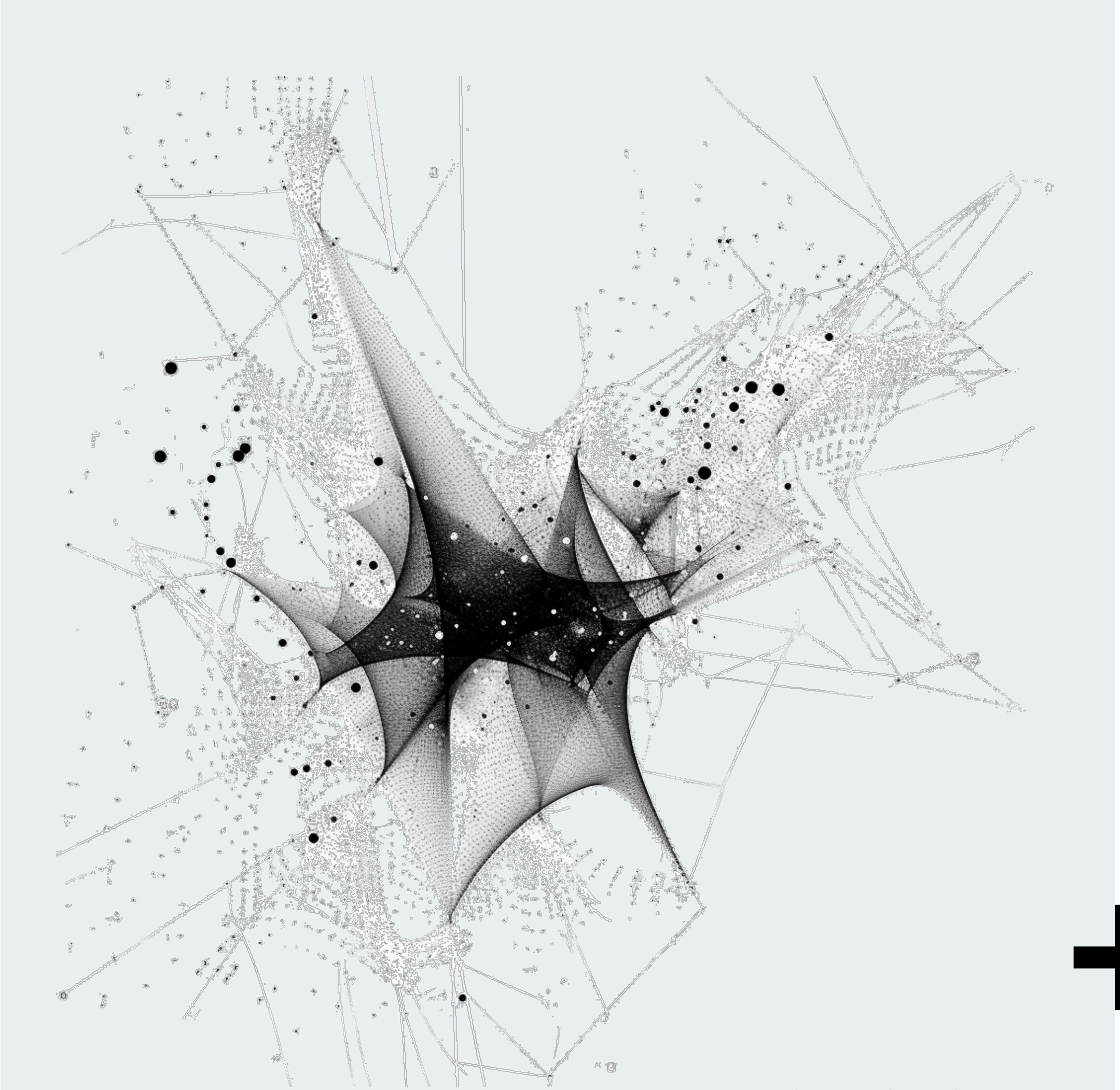


Photo credit: midjourney

The End

Thank you for your attention!

Email: chchan@ntnu.edu.tw
Website: toodou.github.io

Photo credit: midjourney