

HW3

Zihan Lin

2024-10-24

```
library(sna)
library(network)
library(networkdata)
library(tidyverse)
library(ergm)
data(coldwar)
data(hansell)
```

Question 1

```
coldwarcc <- coldwar$cc[,1]
coldwarcc[coldwarcc < 0] <- 0
# coldwarcc[is.na(coldwarcc)] <- 0

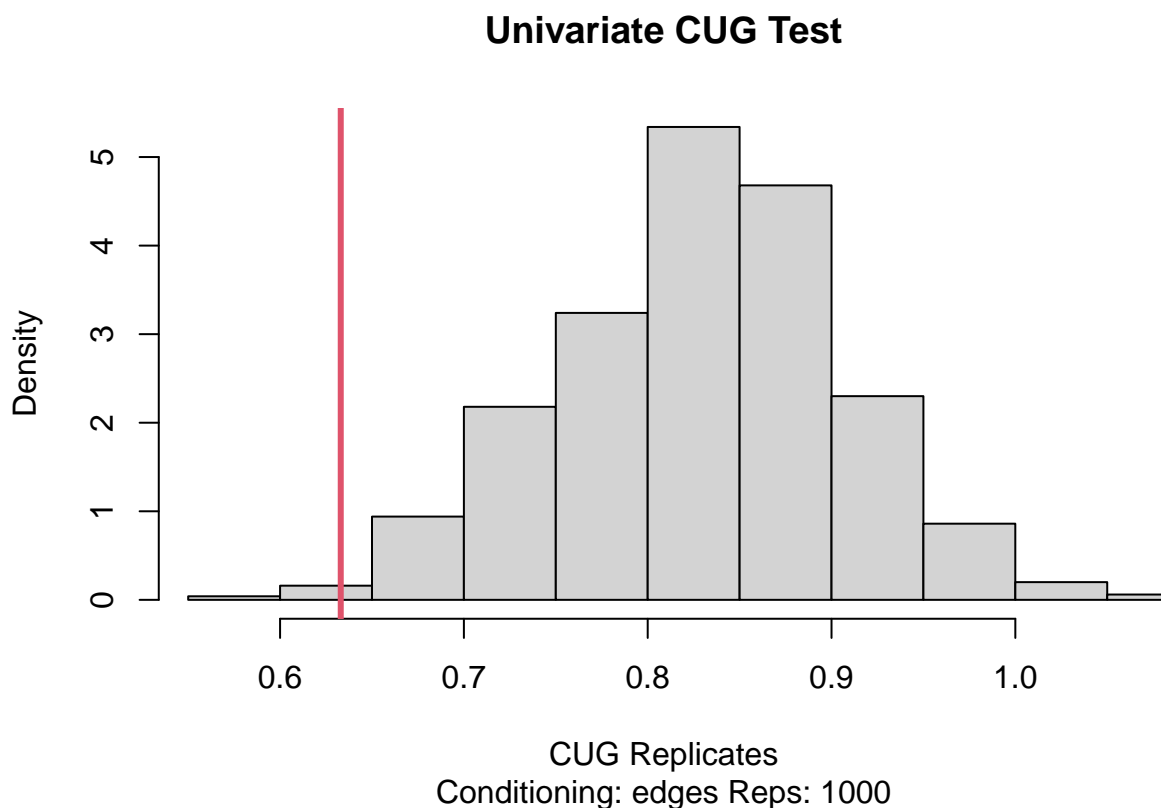
network_cw <- network(coldwarcc, directed = FALSE)
degree_cw <- sna::centralization(dat = network_cw, FUN = sna::degree)
degree_cw
```

```
## [1] 0.09423077
```

since the value of degree centralization is 0.0942308, the null hypothesis cannot be rejected, which means it is a simple random graph model.

```
Cd = function(Y) {
  n = nrow(Y)
  d = apply(Y,1,sum,na.rm=TRUE)
  sum(max(d)-d)/( (n-1)*(n-2) )
}

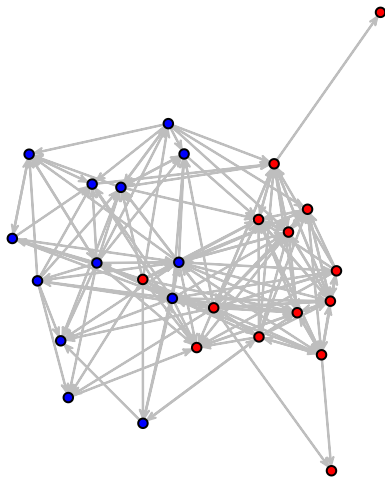
cug_test <- cug.test(dat = network_cw, FUN = Cd, cmode = "edges", reps = 1000)
plot(cug_test)
```



Question 2

a)

```
plot(hansell,  
     vertex.col = ifelse(get.vertex.attribute(hansell, "sex") == "male", "blue", "red"),  
     edge.col = "grey")
```



based on the plot, there is a general preference for same-sex friendship ties, it is easy to see that blue nodes are predominantly connected the other blue nodes, same for red nodes which represents girls.

b)

```
fit.er <- ergm(hansell ~ edges)
```

```
## Starting maximum pseudolikelihood estimation (MPLE):
```

```
## Obtaining the responsible dyads.
```

```
## Evaluating the predictor and response matrix.
```

```
## Maximizing the pseudolikelihood.
```

```
## Finished MPLE.
```

```
## Evaluating log-likelihood at the estimate.
```

```
summary(fit.er)
```

```
## Call:
```

```
## ergm(formula = hansell ~ edges)
```

```
##
```

```
## Maximum Likelihood Results:
```

```
##
```

```
##      Estimate Std. Error MCMC % z value Pr(>|z|)
## edges -1.24454    0.09058      0  -13.74  <1e-04 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
##      Null Deviance: 973.2 on 702 degrees of freedom
```

```
## Residual Deviance: 746.2 on 701 degrees of freedom
```

```
##
```

```
## AIC: 748.2 BIC: 752.8 (Smaller is better. MC Std. Err. = 0)
```

```
log_odds_b <- -1.24454
```

```
cat("the log-odds of a tie between two randomly chosen students is", log_odds_b)
```

```
## the log-odds of a tie between two randomly chosen students is -1.24454
```

```
probability_b <- exp(log_odds_b) / (1 + exp(log_odds_b))
```

```
cat("the MLE of the probability of a tie between two randomly chosen students is",
    probability_b)
```

```
## the MLE of the probability of a tie between two randomly chosen students is 0.2236467
```

```
hansell
```

```
## Network attributes:
##   vertices = 27
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   total edges= 157
##   missing edges= 0
##   non-missing edges= 157
##
## Vertex attribute names:
##   sex
##
## No edge attributes
```

```
vertices <- 27
tot_edges <- 157
density_b <- tot_edges / (vertices * (vertices - 1))
cat("the density of the network is", density_b)
```

```
## the density of the network is 0.2236467
```

```
# H_0: The null hypothesis is that all networks among the students are equally likely
# H_1: The alternative hypothesis is that the Erdos-Renyi model fits the data better

Null_Deviance <- 973.2
Residual_Deviance <- 746.2
p_value_b <- 1 - pchisq(Null_Deviance - Residual_Deviance, df = 1)
p_value_b
```

```
## [1] 0
```

the MLE probability under the Erdos-Renyi model is lower than the actual observed density of ties in the network (MLE = 0.2236467, density = 0.2236467) and the p-value is 0, suggesting that the Erdos-Renyi model might underestimate the number of ties in this specific network.

c)

```
fit.homo <- ergm(hansell~edges + nodematch("sex"))
```

```
## Starting maximum pseudolikelihood estimation (MPLE):
```

```
## Obtaining the responsible dyads.
```

```
## Evaluating the predictor and response matrix.
```

```
## Maximizing the pseudolikelihood.
```

```
## Finished MPLE.
```

```
## Evaluating log-likelihood at the estimate.
```

```
summary(fit.homo)
```

```
## Call:
```

```
## ergm(formula = hansell ~ edges + nodematch("sex"))
```

```
##
```

```
## Maximum Likelihood Results:
```

```
##
```

	Estimate	Std. Error	MCMC %	z value	Pr(> z)	
## edges	-1.9841	0.1608	0	-12.341	<1e-04	***
## nodematch.sex	1.2954	0.1979	0	6.547	<1e-04	***

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Null Deviance: 973.2 on 702 degrees of freedom
```

```
## Residual Deviance: 699.1 on 700 degrees of freedom
```

```
##
```

```
## AIC: 703.1 BIC: 712.2 (Smaller is better. MC Std. Err. = 0)
```

since p-value for nodematch.sex is <1e-04, which is highly significant result that indicates strong evidence of homophily based on sex

```
log_odds_c <- -1.9841 + 1.2954  
cat("the log-odds of a tie between two students of the same sex is", log_odds_c)
```

```
## the log-odds of a tie between two students of the same sex is -0.6887
```

```
probability_c <- exp(log_odds_c) / (1 + exp(log_odds_c))  
cat("the MLE of the probability of a tie between two students of the same sex is",  
    probability_c)
```

```
## the MLE of the probability of a tie between two students of the same sex is 0.3343223
```

```
probability_oppo <- exp(-1.9841) / (1 + exp(-1.9841))  
cat("the MLE of the probability of a tie between two students of the opposite sex is",  
    probability_oppo)
```

```
## the MLE of the probability of a tie between two students of the opposite sex is 0.1208825
```

```
Null_Deviance <- 973.2  
Residual_Deviance <- 699.1  
p_value_c <- 1 - pchisq(Null_Deviance - Residual_Deviance, df = 1)  
p_value_c
```

```
## [1] 0
```

The p-value is extremely small (p-value = 0), indicating that the model with homophily based on sex fits significantly better than the Erdos-Renyi model.

d)

```
fit.diff.homo <- ergm(hansell~edges + nodematch("sex", diff=TRUE))
```

```
## Starting maximum pseudolikelihood estimation (MPLE):
```

```
## Obtaining the responsible dyads.
```

```
## Evaluating the predictor and response matrix.
```

```
## Maximizing the pseudolikelihood.
```

```
## Finished MPLE.
```

```
## Evaluating log-likelihood at the estimate.
```

```
summary(fit.diff.homo)
```

```
## Call:
```

```
## ergm(formula = hansell ~ edges + nodematch("sex", diff = TRUE))
```

```
##
```

```
## Maximum Likelihood Results:
```

```
##
```

	Estimate	Std. Error	MCMC %	z value	Pr(> z)	
## edges	-1.9841	0.1608	0	-12.341	<1e-04	***
## nodematch.sex.female	1.4674	0.2221	0	6.607	<1e-04	***
## nodematch.sex.male	1.0813	0.2389	0	4.526	<1e-04	***

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Null Deviance: 973.2 on 702 degrees of freedom
```

```
## Residual Deviance: 696.4 on 699 degrees of freedom
```

```
##
```

```
## AIC: 702.4 BIC: 716 (Smaller is better. MC Std. Err. = 0)
```

the significant z-tests for both coefficients, combined with the higher estimate for nodematch.sex.female, support the conclusion that homophily is indeed stronger among females than among males in this network.

e)

```
fit.mix <- ergm(hansell~edges+nodemix("sex"))
```

```
## Starting maximum pseudolikelihood estimation (MPLE):
```

```
## Obtaining the responsible dyads.
```

```
## Evaluating the predictor and response matrix.
```

```
## Maximizing the pseudolikelihood.
```

```
## Finished MPLE.
```

```
## Evaluating log-likelihood at the estimate.
```

```
summary(fit.mix)
```

```
## Call:
```

```
## ergm(formula = hansell ~ edges + nodemix("sex"))
```

```
##
```

```
## Maximum Likelihood Results:
```

```
##
```

	Estimate	Std. Error	MCMC %	z value	Pr(> z)	
## edges	-0.5167	0.1532	0	-3.372	0.000746	***
## mix.sex.male.female	-1.3207	0.2643	0	-4.997	< 1e-04	***
## mix.sex.female.male	-1.6326	0.2868	0	-5.693	< 1e-04	***
## mix.sex.male.male	-0.3862	0.2339	0	-1.651	0.098731	.

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
##      Null Deviance: 973.2  on 702  degrees of freedom
```

```
## Residual Deviance: 695.5  on 698  degrees of freedom
```

```
##
```

```
## AIC: 703.5  BIC: 721.7  (Smaller is better. MC Std. Err. = 0)
```

```
log_odds_e <- -1.8374
cat("the log-odds of a tie from a male to a female student is", log_odds_e)
```

```
## the log-odds of a tie from a male to a female student is -1.8374
```

```
probability_e <- exp(-2.1493) / (1 + exp(-2.1493))
cat("the MLE of the probability of a tie from a female to a male student is",
    probability_e)
```

```
## the MLE of the probability of a tie from a female to a male student is 0.1043967
```

```
sex_attr <- get.vertex.attribute(hansell, "sex")
edge_list <- as.matrix.network.edgelist(hansell)

female_to_male_count <- 0
for (i in 1:nrow(edge_list)) {
  from <- edge_list[i, 1]
  to <- edge_list[i, 2]

  if (sex_attr[from] == "female" && sex_attr[to] == "male") {
    female_to_male_count <- female_to_male_count + 1
  }
}

female_to_male_count / (13 * 14)
```

```
## [1] 0.1043956
```

they are almost the same, the difference between them is $-1.0491962 \times 10^{-6}$ (empirical frequency - MLE of the probability of a tie from a female to a male student)

```
Residual_Deviance_bernoulli <- 746.2
Residual_Deviance_mixing <- 695.5
p_value_e <- 1 - pchisq(Residual_Deviance_bernoulli - Residual_Deviance_mixing, df = 2)
p_value_e
```

```
## [1] 9.786616e-12
```

the p-value is very small, indicating that the full mixing model fits significantly better than the Erdos-Renyi model.

```
Residual_Deviance_homophily <- 699.1
Residual_Deviance_mixing <- 695.5
p_value_e <- 1 - pchisq(Residual_Deviance_homophily - Residual_Deviance_mixing, df = 2)
p_value_e
```

```
## [1] 0.1652989
```

the p-value is 0.1652989, indicating that the full mixing model does not fit better than the homophily model.

```
Residual_Deviance_diff <- 696.4
Residual_Deviance_mixing <- 695.5
p_value_e <- 1 - pchisq(Residual_Deviance_diff - Residual_Deviance_mixing, df = 2)
p_value_e
```

```
## [1] 0.6376282
```

the p-value is 0.6376282, indicating that the full mixing model does not fit better than the differential homophily model.

Question 3

a)

```
fit <- ergm(hansell ~ edges + nodematch("sex") + sender + receiver + mutual,
            control=control.ergm(MCMLE.maxit=4))
```

```
## Observed statistic(s) sender10, sender26, sender27, and receiver20 are at their smallest attainable
```

```
## Starting maximum pseudolikelihood estimation (MPLE):
```

```
## Obtaining the responsible dyads.
```

```
## Evaluating the predictor and response matrix.
```

```
## Maximizing the pseudolikelihood.
```

```
## Finished MPLE.
```

```
## Starting Monte Carlo maximum likelihood estimation (MCMLE):
```

```
## Iteration 1 of at most 4:
```

```
## Optimizing with step length 1.0000.
```

```
## The log-likelihood improved by 0.2359.
```

```
## Estimating equations are not within tolerance region.
```

```
## Iteration 2 of at most 4:
```

```
## Optimizing with step length 1.0000.
```

```
## The log-likelihood improved by 0.0667.
```

```
## Convergence test p-value: 0.9977. Not converged with 99% confidence; increasing sample size.
## Iteration 3 of at most 4:
## Optimizing with step length 1.0000.
## The log-likelihood improved by 0.0816.
## Convergence test p-value: 1.0000. Not converged with 99% confidence; increasing sample size.
## Iteration 4 of at most 4:
## Optimizing with step length 1.0000.
## The log-likelihood improved by 0.1306.
## Estimating equations are not within tolerance region.
## Estimating equations did not move closer to tolerance region more than 1 time(s) in 4 steps; increasing sample size.
## MCMLE estimation did not converge after 4 iterations. The estimated coefficients may not be accurate.
## Finished MCMLE.
## Evaluating log-likelihood at the estimate. Fitting the dyad-independent submodel...
## Bridging between the dyad-independent submodel and the full model...
## Setting up bridge sampling...
## Using 16 bridges: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 .
## Bridging finished.
##
## This model was fit using MCMC. To examine model diagnostics and check
## for degeneracy, use the mcmc.diagnostics() function.
```

```
fit0 <- ergm(hansell ~ edges + nodematch("sex") + sender + receiver + mutual,
            estimate="MPLE", control=control.ergm(MCMLE.maxit=4))
```

```
## Observed statistic(s) sender10, sender26, sender27, and receiver20 are at their smallest attainable values.
## Starting maximum pseudolikelihood estimation (MPLE):
## Obtaining the responsible dyads.
## Evaluating the predictor and response matrix.
## Maximizing the pseudolikelihood.
## Finished MPLE.
## Evaluating log-likelihood at the estimate.
```

```
fit$mle.lik <- fit0$mple.lik
summary_fit <- summary(fit)
summary_fit
```

```
## Call:
## ergm(formula = hansell ~ edges + nodematch("sex") + sender +
##       receiver + mutual, control = control.ergm(MCMLE.maxit = 4))
##
## Monte Carlo Maximum Likelihood Results:
##
##           Estimate Std. Error MCMC % z value Pr(>|z|)
## edges          -2.07613    0.77876    0 -2.666 0.00768 **
## nodematch.sex    1.81981    0.28086    0  6.479 < 1e-04 ***
## sender2         -1.48987    0.92745    0 -1.606 0.10818
## sender3           0.28446    0.73920    0  0.385 0.70037
## sender4           1.34116    0.69966    0  1.917 0.05525 .
## sender5           1.45814    0.73329    0  1.988 0.04676 *
## sender6           0.83099    0.70241    0  1.183 0.23679
## sender7          -0.51765    0.82669    0 -0.626 0.53120
## sender8           0.21153    0.73555    0  0.288 0.77367
## sender9          -1.00949    0.84328    0 -1.197 0.23127
```

```

## sender10      -Inf    0.00000    0    -Inf < 1e-04 ***
## sender11     -2.14185  1.25128    0   -1.712 0.08695 .
## sender12     -0.90660  0.86946    0   -1.043 0.29708
## sender13     -1.46866  0.97029    0   -1.514 0.13012
## sender14     -0.08062  0.82059    0   -0.098 0.92174
## sender15     -0.81053  0.85413    0   -0.949 0.34264
## sender16      3.29765  0.79015    0    4.173 < 1e-04 ***
## sender17     -3.00159  1.26849    0   -2.366 0.01797 *
## sender18     -2.13130  0.97831    0   -2.179 0.02936 *
## sender19      1.17541  0.77498    0    1.517 0.12934
## sender20      1.21022  0.77357    0    1.564 0.11771
## sender21     -0.72626  0.85646    0   -0.848 0.39645
## sender22      0.28069  0.79982    0    0.351 0.72564
## sender23      0.01243  0.81456    0    0.015 0.98782
## sender24     -1.59863  0.92665    0   -1.725 0.08450 .
## sender25      0.28514  0.79381    0    0.359 0.71944
## sender26      -Inf    0.00000    0    -Inf < 1e-04 ***
## sender27      -Inf    0.00000    0    -Inf < 1e-04 ***
## receiver2    -0.25083  0.80679    0   -0.311 0.75588
## receiver3    -1.53960  0.95527    0   -1.612 0.10703
## receiver4    -1.49635  0.92365    0   -1.620 0.10523
## receiver5    -0.12432  0.82215    0   -0.151 0.87981
## receiver6    -0.04480  0.82233    0   -0.054 0.95655
## receiver7    -1.49024  0.94925    0   -1.570 0.11644
## receiver8    -2.05554  1.01151    0   -2.032 0.04214 *
## receiver9    -0.24790  0.81593    0   -0.304 0.76126
## receiver10    0.08339  0.76470    0    0.109 0.91316
## receiver11   -0.95131  0.92201    0   -1.032 0.30218
## receiver12   -1.44490  0.88385    0   -1.635 0.10210
## receiver13   -0.29661  0.84500    0   -0.351 0.72558
## receiver14    0.55613  0.80041    0    0.695 0.48718
## receiver15    1.44758  0.82121    0    1.763 0.07794 .
## receiver16   -0.82385  0.87985    0   -0.936 0.34909
## receiver17    1.22834  0.82104    0    1.496 0.13463
## receiver18    0.93215  0.85969    0    1.084 0.27824
## receiver19   -1.61071  0.98879    0   -1.629 0.10332
## receiver20    -Inf    0.00000    0    -Inf < 1e-04 ***
## receiver21    0.90452  0.81202    0    1.114 0.26531
## receiver22   -0.36858  0.85903    0   -0.429 0.66787
## receiver23   -0.75453  0.93941    0   -0.803 0.42186
## receiver24    0.63054  0.85912    0    0.734 0.46299
## receiver25    0.81153  0.80727    0    1.005 0.31476
## receiver26   -3.04521  1.37764    0   -2.210 0.02707 *
## receiver27   -2.07797  1.10025    0   -1.889 0.05894 .
## mutual       0.78116  0.46025    0    1.697 0.08965 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Null Deviance: 973.2 on 702 degrees of freedom
## Residual Deviance: 451.8 on 647 degrees of freedom
##
## AIC: 553.8 BIC: 778.1 (Smaller is better. MC Std. Err. = NA)
##
## Warning: The following terms have infinite coefficient estimates:

```

```
## sender10 sender26 sender27 receiver20
```

the coefficient for `nodematch.sex` is 1.8198132 with a very small p-value ($< 1e-04$), which suggests that there is a statistically significant preference for same-sex friendships in this network.

The coefficient for `mutual` is 0.7811571 with a p-value of 0.07782, which is marginally significant at the 10% level, which suggests a weak, positive preference for mutual friendship ties.

b)

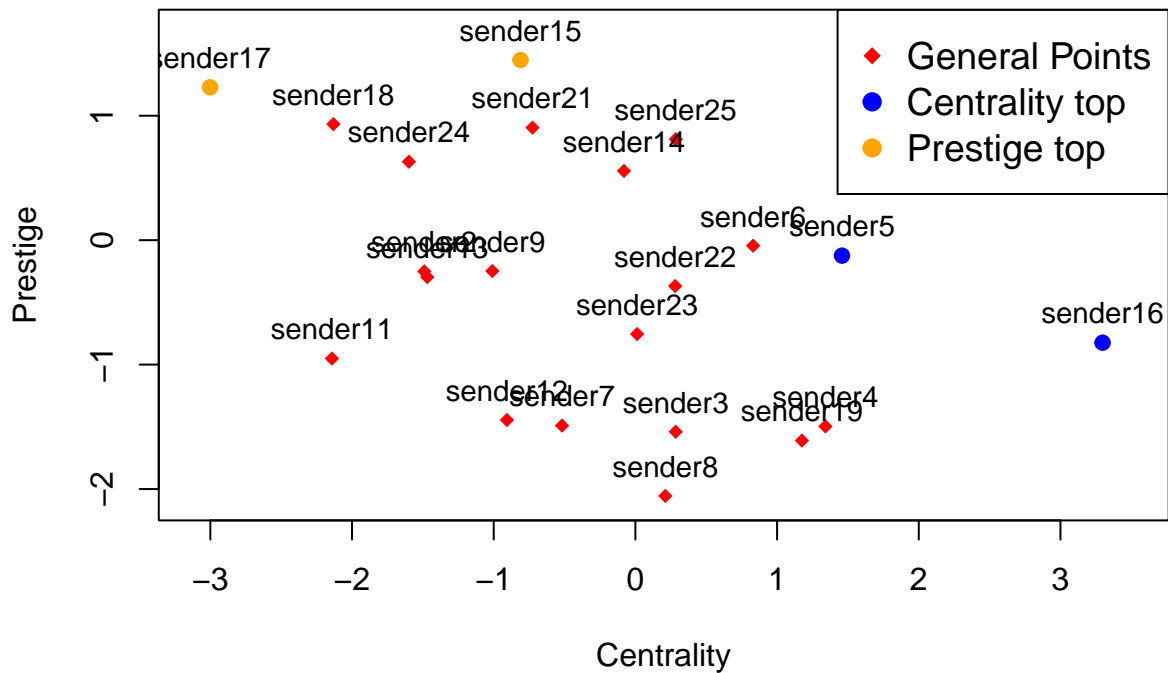
```
centrality_scores <- coef(fit)[grep("sender", names(coef(fit)))]
prestige_scores <- coef(fit)[grep("receiver", names(coef(fit)))]

plot(centrality_scores, prestige_scores,
     xlab = "Centrality", ylab = "Prestige",
     xlim = c(-3.1, 3.5), ylim = c(-2.1, 1.7),
     pch = 18, col = "red")

text(centrality_scores, prestige_scores,
     labels = names(centrality_scores), pos = 3, cex = 0.88)

points(x = centrality_scores[15], y = prestige_scores[15], pch = 19, col = "blue")
points(x = centrality_scores[4], y = prestige_scores[4], pch = 19, col = "blue")
points(x = centrality_scores[14], y = prestige_scores[14], pch = 19, col = "orange")
points(x = centrality_scores[16], y = prestige_scores[16], pch = 19, col = "orange")

legend("topright",
     legend = c("General Points", "Centrality top", "Prestige top"),
     col = c("red", "blue", "orange"),
     pch = c(18, 19, 19),
     cex = 1.2)
```



```
sort(centrality_scores, decreasing = TRUE)[1:2]
```

```
## sender16 sender5  
## 3.297650 1.458141
```

```
sort(prestige_scores, decreasing = TRUE)[1:2]
```

```
## receiver15 receiver17  
## 1.447583 1.228335
```

yes, in the figure,

Sender16 and Sender5 are clearly positioned on the far right, reflecting their high centrality.

Receiver15 and Receiver17 are positioned near the top, indicating their high prestige.

```
in_degree <- degree(hansell, cmode = "indegree")  
out_degree <- degree(hansell, cmode = "outdegree")  
  
node_indices <- as.numeric(gsub("sender|receiver", "", names(centrality_scores)))  
  
calculate_correlation <- function(scores, matching_degree) {  
  filtered_indices <- which(is.finite(scores) & is.finite(matching_degree))  
  scores_filtered <- scores[filtered_indices]  
  matching_degree_filtered <- matching_degree[filtered_indices]  
  cor(scores_filtered, matching_degree_filtered)  
}  
  
cor_centrality <- calculate_correlation(centrality_scores, out_degree[node_indices])  
cor_prestige <- calculate_correlation(prestige_scores, in_degree[node_indices])  
  
cat("Correlation between ERGM centrality and Freeman's out-degree centrality:", cor_centrality, "\n")  
  
## Correlation between ERGM centrality and Freeman's out-degree centrality: 0.972006  
  
cat("Correlation between ERGM prestige and Freeman's in-degree centrality:", cor_prestige, "\n")  
  
## Correlation between ERGM prestige and Freeman's in-degree centrality: 0.9844776
```


c)

```
sender_coefficients <- coef(fit)[grep("sender", names(coef(fit)))]
receiver_coefficients <- coef(fit)[grep("receiver", names(coef(fit)))]

typical_sender_magnitude <- mean(abs(sender_coefficients[is.finite(sender_coefficients)]))
typical_receiver_magnitude <- mean(abs(receiver_coefficients[is.finite(receiver_coefficients)]))

cat("Typical magnitude of sender coefficients:", typical_sender_magnitude, "\n")

## Typical magnitude of sender coefficients: 1.142212

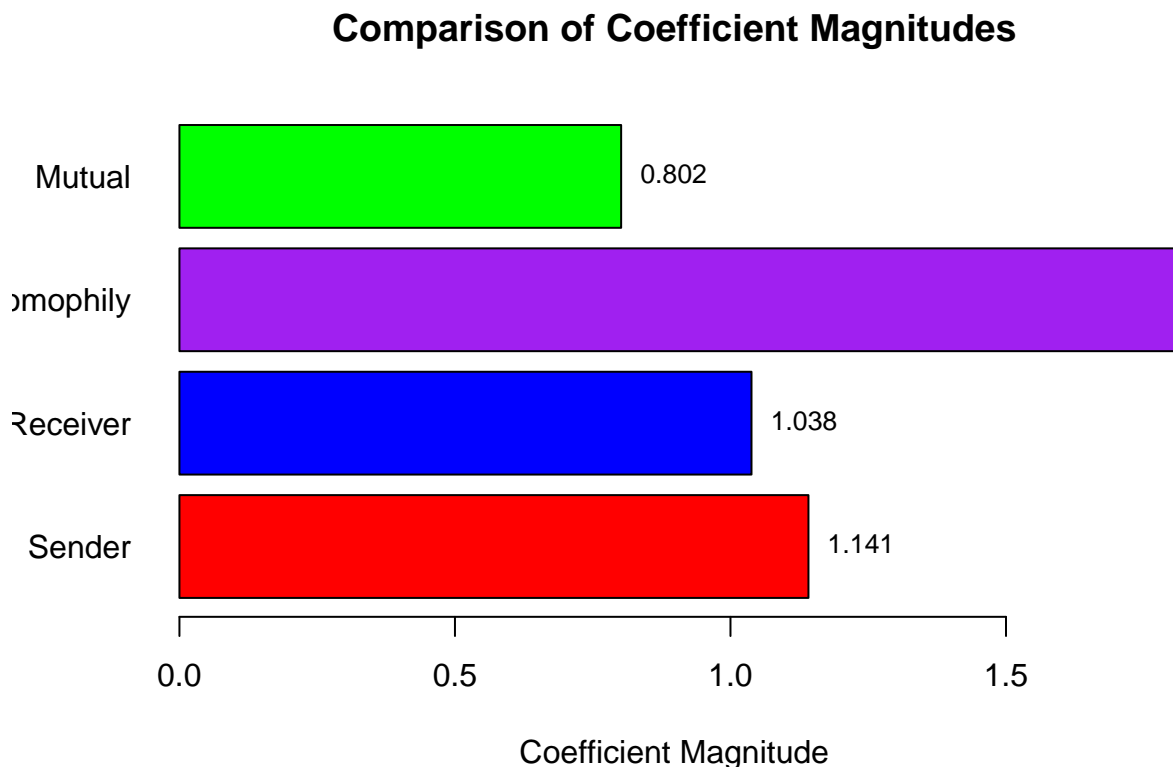
cat("Typical magnitude of receiver coefficients:", typical_receiver_magnitude, "\n")

## Typical magnitude of receiver coefficients: 1.008696

coefficients <- c(Sender = 1.141488, Receiver = 1.038135, Homophily = 1.81462, Mutual = 0.80159)

barplot(coefficients,
        horiz = TRUE,
        col = c("red", "blue", "purple", "green"),
        main = "Comparison of Coefficient Magnitudes",
        xlab = "Coefficient Magnitude",
        las = 1)

text(coefficients, barplot(coefficients, horiz = TRUE, plot = FALSE),
     labels = round(coefficients, 3), pos = 4, cex = 0.8)
```



The typical magnitudes of the sender 1.141488 and receiver 1.038135 coefficients are smaller than both the same-sex preference coefficient 1.81462 and the mutual tie coefficient 0.80159, indicating that homophily has the strongest effect on network ties, followed by mutuality, with sender and receiver effects being comparatively weaker.