Final_AY

2024-06

Analysis

if you would like to read the analysis report scroll down to discussion section

```
nxs_edge_list <- read.csv("NexusEdgeList.csv")

#oko_edge_list <- read.csv("Oko2MdEdgeList.csv")

NexusOko_nodeLst <- read.csv("NexusOko_attr_final.csv")

#just nexus

graphNexus <- graph_from_data_frame(nxs_edge_list, directed = TRUE, vertices = NexusOko_nodeLst)

NexusAdjMxt <- as_adjacency_matrix(graphNexus, type = "both", attr = "weight", sparse = FALSE)

#write.csv(as.data.frame(NexusAdjMxt), "NexusAdjMxt.csv", row.names = FALSE)
```

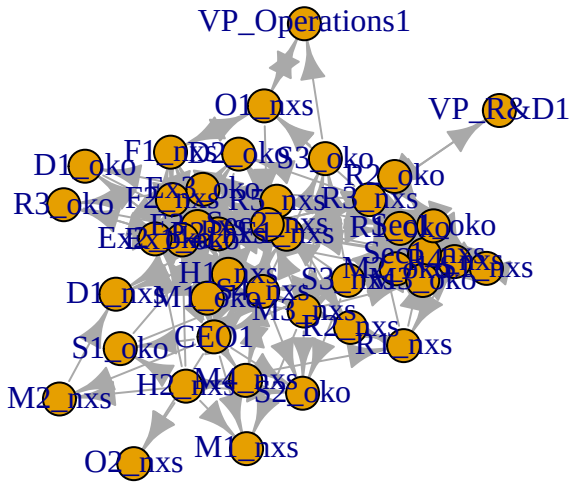
```
oko_edge_list_dir <- read.csv("OkoEdgeList.csv")

graphOko_dir <- graph_from_data_frame(oko_edge_list_dir, directed = TRUE, vertices = NexusOko_nodeLst)

OkoAdjMxt_dir <- as_adjacency_matrix(graphOko_dir, type = "both", attr = "weight", sparse = FALSE)
#graphOko_dir
#OkoAdjMxt_dir
```

```
final_edge_list <- read.csv("final_NexusOko_EdgeLst.csv")

graphFinal <- graph_from_data_frame(final_edge_list, directed = TRUE, vertices = NexusOko_nodeLst)

FinalAdjMxt <- as_adjacency_matrix(graphFinal, type = "both", attr = "Weight", sparse = FALSE)
#graphOko_dir
#OkoAdjMxt_dir
```

```
# all_nodes <- union(rownames(NexusAdjMxt), rownames(OkoAdjMxt_dir))
#
# combined_NexusAdjMxt <- matrix(0, nrow = length(all_nodes), ncol = length(all_nodes), dimnames = list
# combined_OkoAdjMxt_dir <- matrix(0, nrow = length(all_nodes), ncol = length(all_nodes), dimnames = li
#
# combined_NexusAdjMxt[rownames(NexusAdjMxt), colnames(NexusAdjMxt)] <- NexusAdjMxt
# combined_OkoAdjMxt_dir[rownames(OkoAdjMxt_dir), colnames(OkoAdjMxt_dir)] <- OkoAdjMxt_dir
#
# combinedAdjMxt_dir <- combined_NexusAdjMxt + combined_OkoAdjMxt_dir
#
```

```
# write.csv(as.data.frame(combinedAdjMxt_dir), "NexusOkoAdjMtx_dir.csv", row.names = FALSE)
#
# combined_graph_dir <- graph_from_adjacency_matrix(combinedAdjMxt_dir, mode = "directed", weighted = F
#
# #print(combinedAdjMxt_dir)
#
# plot(combined_graph_dir, vertex.label = V(combined_graph_dir)$name, main = "Combined Nexus and Oko Di



all_nodes <- union(union(rownames(NexusAdjMxt), rownames(OkoAdjMxt_dir)), rownames(FinalAdjMxt))

combined_FinalAdjMxt <- matrix(0, nrow = length(all_nodes), ncol = length(all_nodes), dimnames = list(al

combined_FinalAdjMxt[rownames(NexusAdjMxt), colnames(NexusAdjMxt)] <- NexusAdjMxt
combined_FinalAdjMxt[rownames(OkoAdjMxt_dir), colnames(OkoAdjMxt_dir)] <- combined_FinalAdjMxt[rownames
combined_FinalAdjMxt[rownames(FinalAdjMxt), colnames(FinalAdjMxt)] <- combined_FinalAdjMxt[rownames(Fina

# Create final combined graph
combined_graph_dir <- graph_from_adjacency_matrix(combined_FinalAdjMxt, mode = "directed", weighted = TR


plot(combined_graph_dir, vertex.label = V(combined_graph_dir)$name, main = "Combined NexusOko CEO chang
```

Combined NexusOko CEO change Directed Network



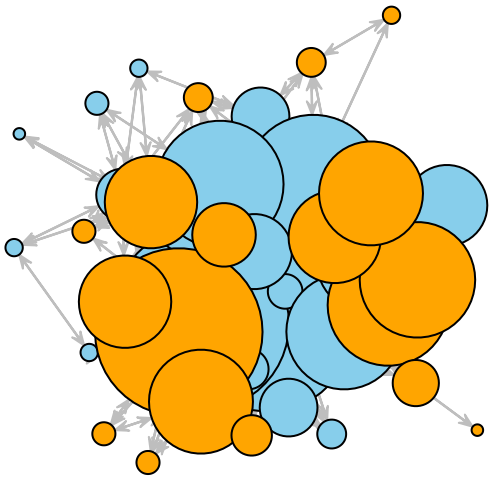

```
#write.csv(as.data.frame(combined_FinalAdjMxt), "FinalCombinedAdjMtx.csv", row.names = FALSE)
```

```
# Convert the combined adjacency matrix to a network object for sna
network <- network::as.network(combined_FinalAdjMxt, directed = FALSE)

# Set colors based on the group (Nexus or Oko)
vertex_colors <- ifelse(grepl("_nxs$", V(combined_graph_dir)$name), "skyblue", "orange")

# Set vertex sizes based on degree
vertex_sizes <- igraph::degree(combined_graph_dir) * 2
```

```
# Define a color palette for the edges
edge_colors <- rep("gray", ecount(combined_graph_dir))
```

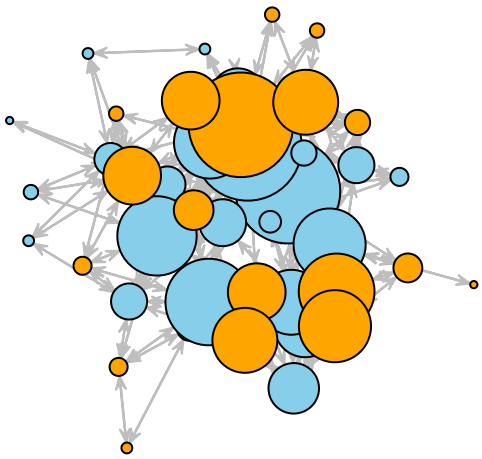
```
# Kamada-Kawai Layout
gplot(network, displaylabels = FALSE, label.cex = 0.7,
      vertex.col = vertex_colors, vertex.cex = vertex_sizes / 5,
      edge.col = edge_colors, edge.lwd = 0.5,
      main = "Kamada-Kawai Layout")
```

Kamada−Kawai Layout



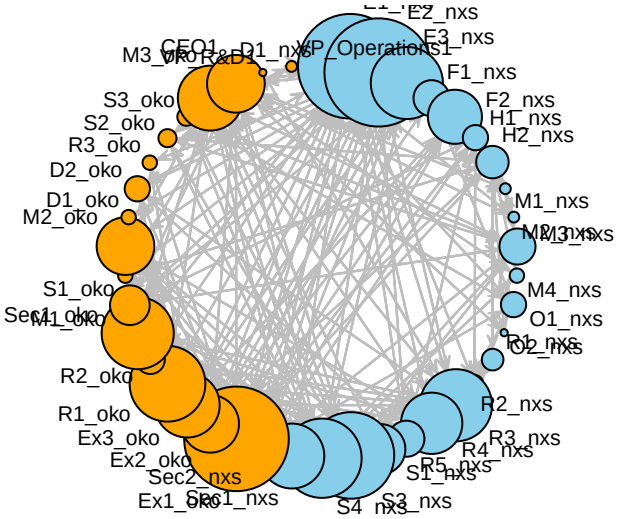
```
# Fruchterman-Reingold Layout
gplot(network, displaylabels = FALSE, label.cex = 0.7,
      vertex.col = vertex_colors, vertex.cex = vertex_sizes / 8,
      edge.col = edge_colors, edge.lwd = 0.5,
      mode = "fruchtermanreingold",
      main = "Fruchterman-Reingold Layout")
```

Fruchterman−Reingold Layout



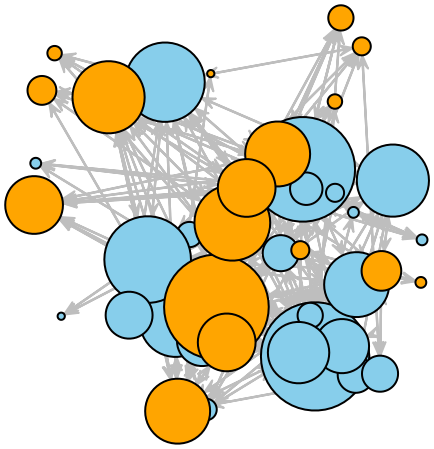
```
# Circle Layout
gplot(network, displaylabels = TRUE, label.cex = 0.7,
      vertex.col = vertex_colors, vertex.cex = vertex_sizes / 8,
      edge.col = edge_colors, edge.lwd = 0.5,
      mode = "circle",
      main = "Circle Layout")
```

Circle Layout




```
# Random Layout
gplot(network, displaylabels = FALSE, label.cex = 0.7,
      vertex.col = vertex_colors, vertex.cex = vertex_sizes / 8,
      edge.col = edge_colors, edge.lwd = 0.5,
      mode = "random",
      main = "Random Layout")
```

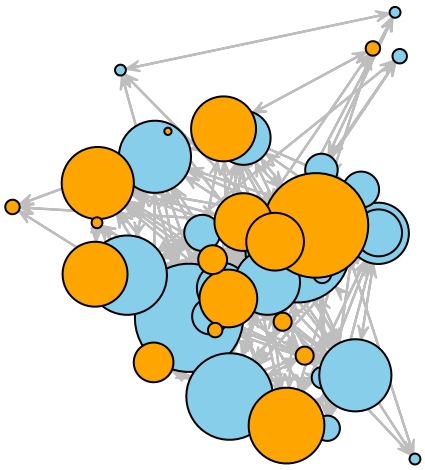
Random Layout



```
# Spring Layout
gplot(network, displaylabels = FALSE, label.cex = 0.7,
```

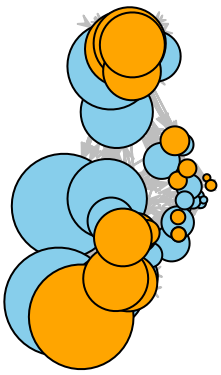
```
      vertex.col = vertex_colors, vertex.cex = vertex_sizes / 8,
      edge.col = edge_colors, edge.lwd = 0.5,
      mode = "spring",
      main = "Spring Layout")
```

Spring Layout



```
# Eigen Layout
gplot(network, displaylabels = FALSE, label.cex = 0.7,
      vertex.col = vertex_colors, vertex.cex = vertex_sizes / 8,
      edge.col = edge_colors, edge.lwd = 0.5,
      mode = "eigen",
      main = "Eigen Layout")
```

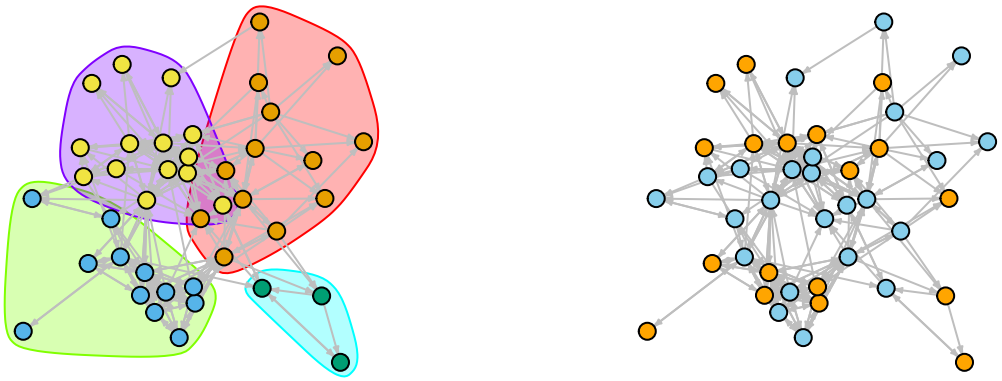
Eigen Layout



#HW5 additions (MOST OF THEM REMOVED FOR VISUAL SIMPLICITY OF THE REPORT)

```
#x <- asIgraph(network)

cd <- cluster_walktrap(combined_graph_dir)
#membership(cd)
```

```r
#modularity(cd)
#plot(cd,combined_graph_dir)

vertex_colors <- ifelse(grepl("_nxs$", V(combined_graph_dir)$name), "skyblue", "orange")

layout <- layout_with_fr(combined_graph_dir)

par(mfrow=c(1, 2))

# Plotting with adjustments
plot(cd, combined_graph_dir, layout=layout, vertex.size=10, edge.arrow.size=.2,
     vertex.label=NA, # Hide vertex labels to reduce clutter
     vertex.color=vertex_colors,  #i was hoping this would work but no :(
     edge.color="gray")

plot(combined_graph_dir, layout=layout, vertex.size=10, edge.arrow.size=.2,
     vertex.label=NA, # Hide vertex labels to reduce clutter
     vertex.color=vertex_colors,  #i was hoping this would work but no :(
     edge.color="gray")
```



```r
coreness<- coreness(combined_graph_dir)
coreness
```

```
##         D1_nxs         E1_nxs         E2_nxs         E3_nxs         F1_nxs
##              3             12             12             12              8
##         F2_nxs         H1_nxs         H2_nxs         M1_nxs         M2_nxs
##             12              6              4              3              3
##         M3_nxs         M4_nxs         O1_nxs         O2_nxs         R1_nxs
##              7              4              4              2              5
##         R2_nxs         R3_nxs         R4_nxs         R5_nxs         S1_nxs
##              5             12             14              7             14
##         S3_nxs         S4_nxs       Sec1_nxs       Sec2_nxs        Ex1_oko
##             14             10             14             10             12
##        Ex2_oko        Ex3_oko         R1_oko         R2_oko       Sec1_oko
##             12             12             14              8             14
##         M1_oko         S1_oko         M2_oko         D1_oko         D2_oko
##              9              4             14              4              7
##         R3_oko         S2_oko         S3_oko         M3_oko           CEO1
##              4              4              4             14              9
##        VP_R&D1 VP_Operations1
##              2              3
```

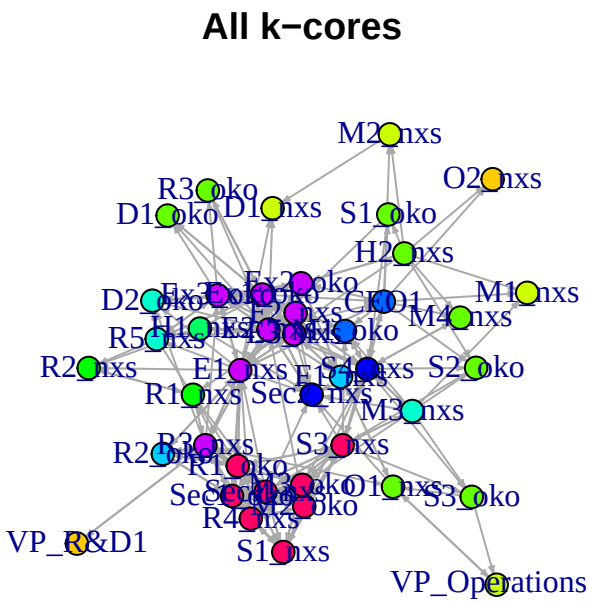
```
table(coreness)
```

```
## coreness
##  2  3  4  5  6  7  8  9 10 12 14
##  2  4  8  2  1  3  2  2  2  8  8
```

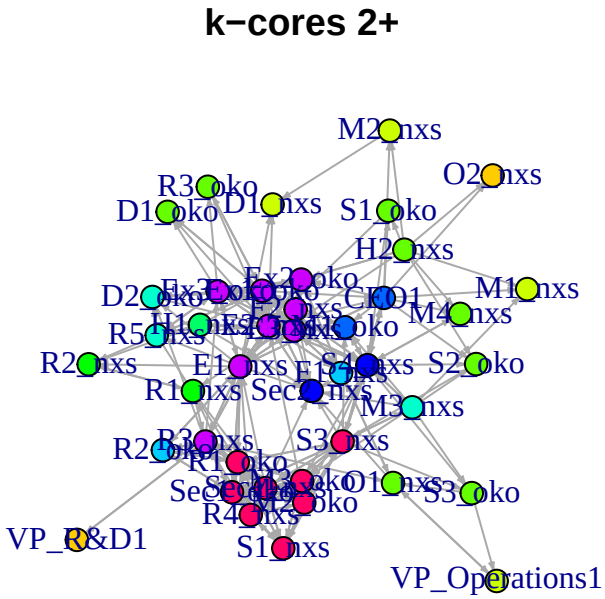
```
max(coreness)
```

```
## [1] 14
```

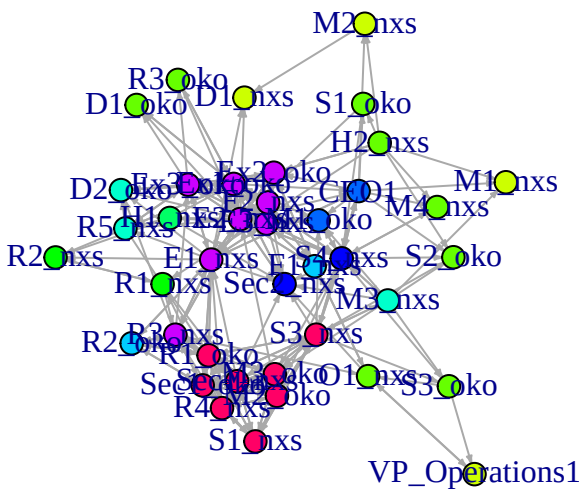
```
coreness <- coreness(combined_graph_dir)

colors <- rainbow(max(coreness) + 1)

V(combined_graph_dir)$color <- colors[coreness + 1]  # +1 because R indexing starts at 1

combined_graph_dir2 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 1))
combined_graph_dir3 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 2))
combined_graph_dir4 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 3))
combined_graph_dir5 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 4))
combined_graph_dir6 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 5))
combined_graph_dir7 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 6))
combined_graph_dir8 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 7))
combined_graph_dir9 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 8))
combined_graph_dir10 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 9))
combined_graph_dir11 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 10))
combined_graph_dir12 <- induced_subgraph(combined_graph_dir, vids=which(coreness > 11))


plot(combined_graph_dir, layout=layout, main="All k-cores", vertex.size=10, edge.arrow.size=.2)
```



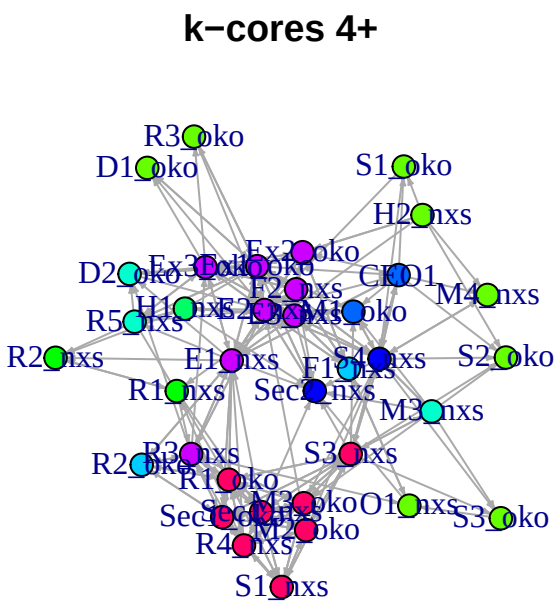

```
plot(combined_graph_dir2, layout=layout[which(coreness > 1), ],vertex.size=10, edge.arrow.size=.2, main=
```



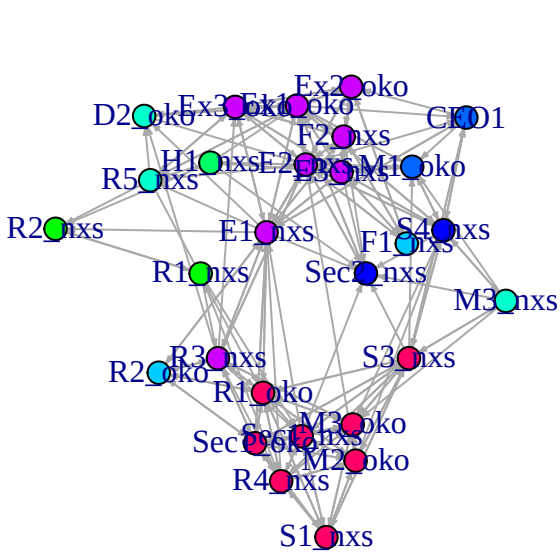

```
plot(combined_graph_dir3, layout=layout[which(coreness > 2), ],vertex.size=10, edge.arrow.size=.2, main=
```



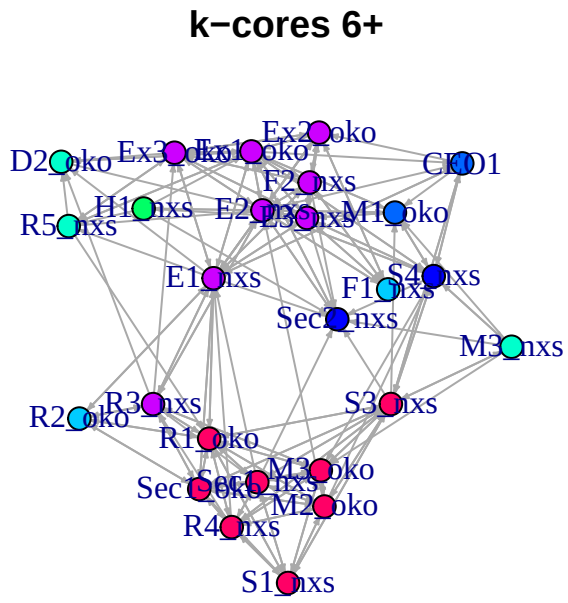

```
plot(combined_graph_dir4, layout=layout[which(coreness > 3), ],vertex.size=10, edge.arrow.size=.2, main=
```



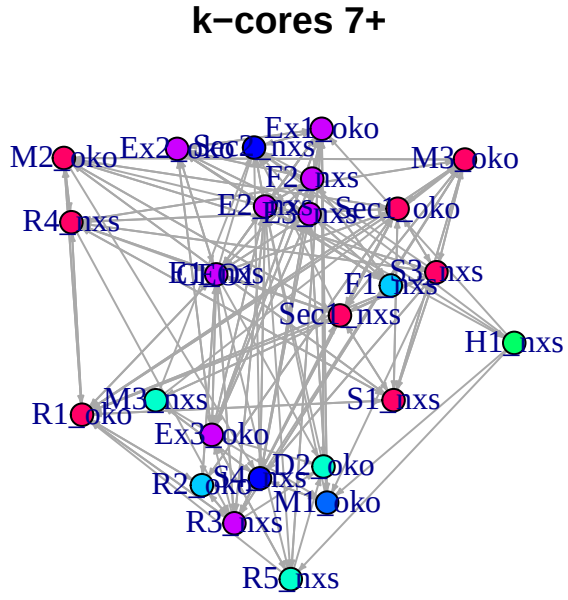

```
plot(combined_graph_dir5, layout=layout[which(coreness > 4), ],vertex.size=10, edge.arrow.size=.2, main=
```




```
plot(combined_graph_dir6, layout=layout[which(coreness > 5), ],vertex.size=10, edge.arrow.size=.2, main=
```

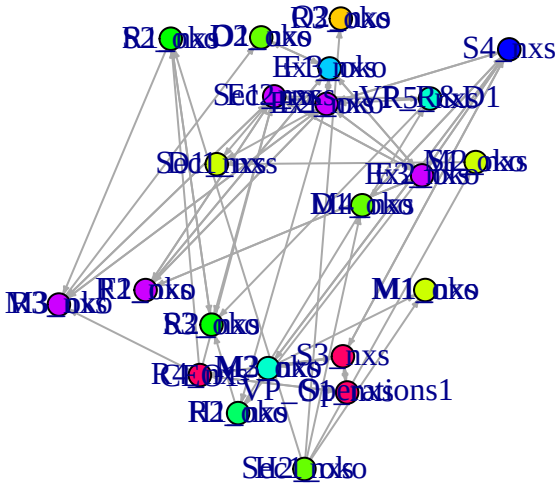



```
plot(combined_graph_dir6, layout=layout[which(coreness > 6), ],vertex.size=10, edge.arrow.size=.2, main=
```



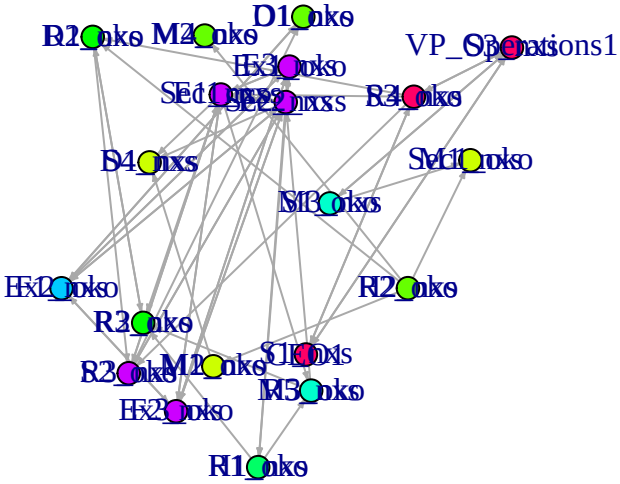

```
plot(combined_graph_dir2, layout=layout[which(coreness > 7), ],vertex.size=10, edge.arrow.size=.2, main=
```

k−cores 8+

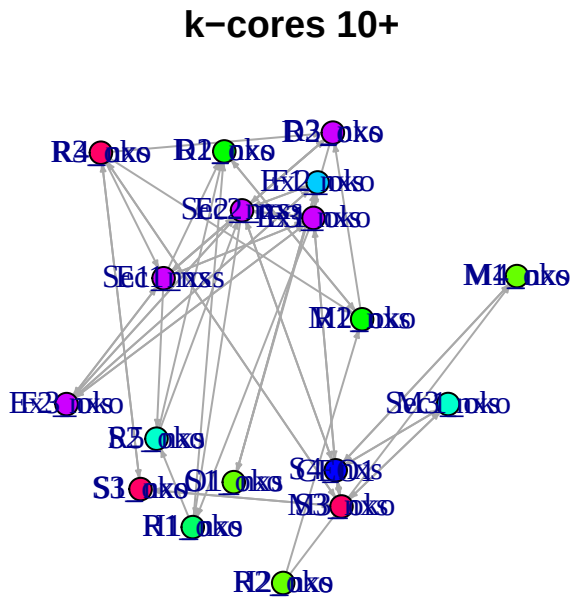


```
plot(combined_graph_dir3, layout=layout[which(coreness > 8), ],vertex.size=10, edge.arrow.size=.2, main=
```

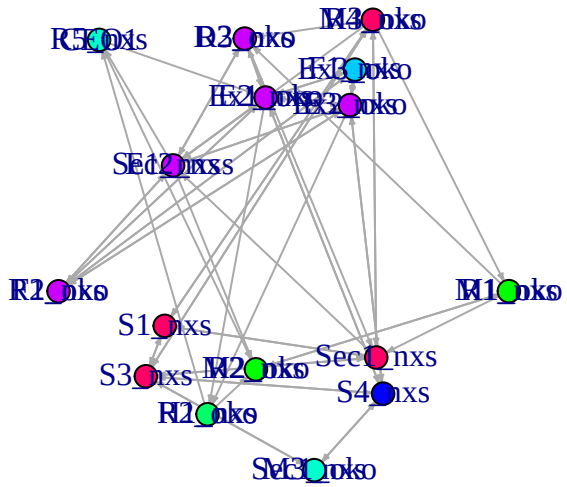
k−cores 9+



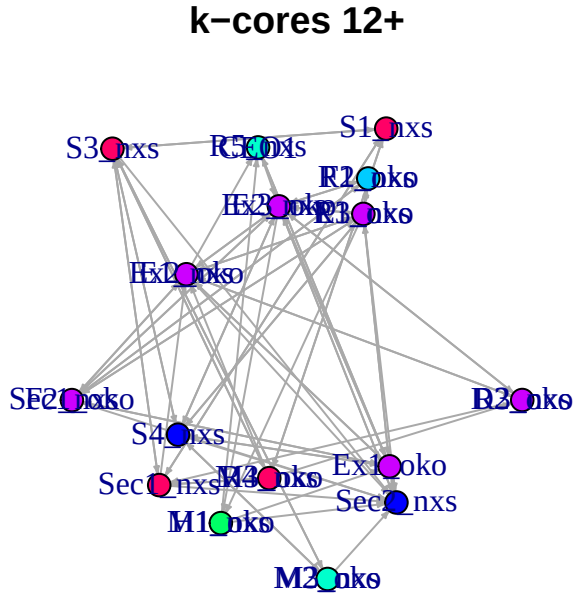
```
plot(combined_graph_dir4, layout=layout[which(coreness > 9), ],vertex.size=10, edge.arrow.size=.2, main=
```




```
plot(combined_graph_dir5, layout=layout[which(coreness > 10), ],vertex.size=10, edge.arrow.size=.2, mai
```




```
plot(combined_graph_dir6, layout=layout[which(coreness > 11), ],vertex.size=10, edge.arrow.size=.2, mai
```




```
NOc <- fastgreedy.community(as.undirected(combined_graph_dir))
```

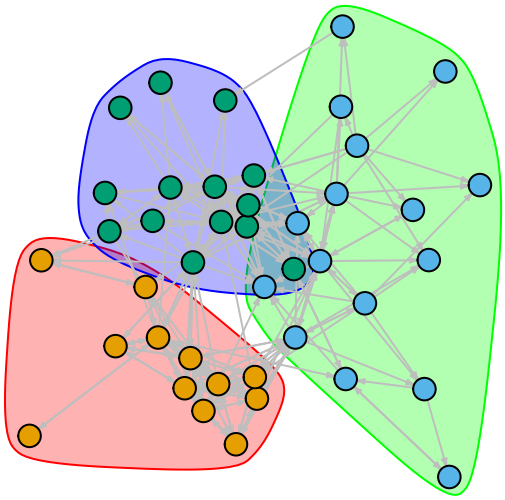
```
## Warning: `fastgreedy.community()` was deprecated in igraph 2.0.0.
## i Please use `cluster_fast_greedy()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
NOc_bw <- cluster_edge_betweenness(combined_graph_dir)
```

```
## Warning in cluster_edge_betweenness(combined_graph_dir): At
## vendor/cigraph/src/community/edge_betweenness.c:497 : Membership vector will be
## selected based on the highest modularity score.
```

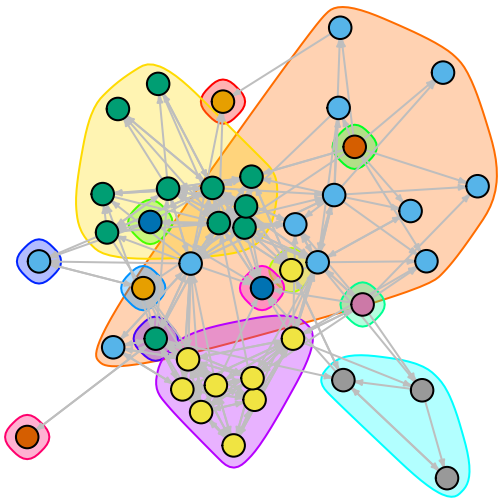
```
NOc_eigen<- cluster_leading_eigen(as.undirected(combined_graph_dir))

# membership(NOc)
# membership(NOc_bw)
# membership(NOc_eigen)


plot(NOc, combined_graph_dir, layout=layout, vertex.size=10, edge.arrow.size=.2,
     vertex.label=NA, # Hide vertex labels to reduce clutter
     vertex.color=vertex_colors,  #i was hoping this would work but no :(
     edge.color="gray", main = "Fast Greedy Clustering")
```

Fast Greedy Clustering



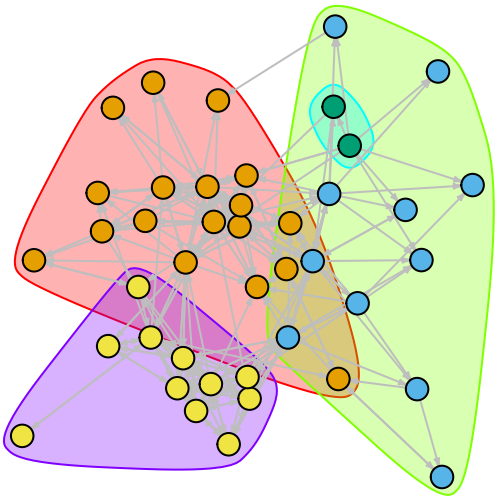
```
plot(NOc_bw, combined_graph_dir, layout=layout, vertex.size=10, edge.arrow.size=.2,
     vertex.label=NA, # Hide vertex labels to reduce clutter
     vertex.color=vertex_colors,  #i was hoping this would work but no :(
     edge.color="gray", main = "Edge Betweeness Clustering")
```

Edge Betweeness Clustering



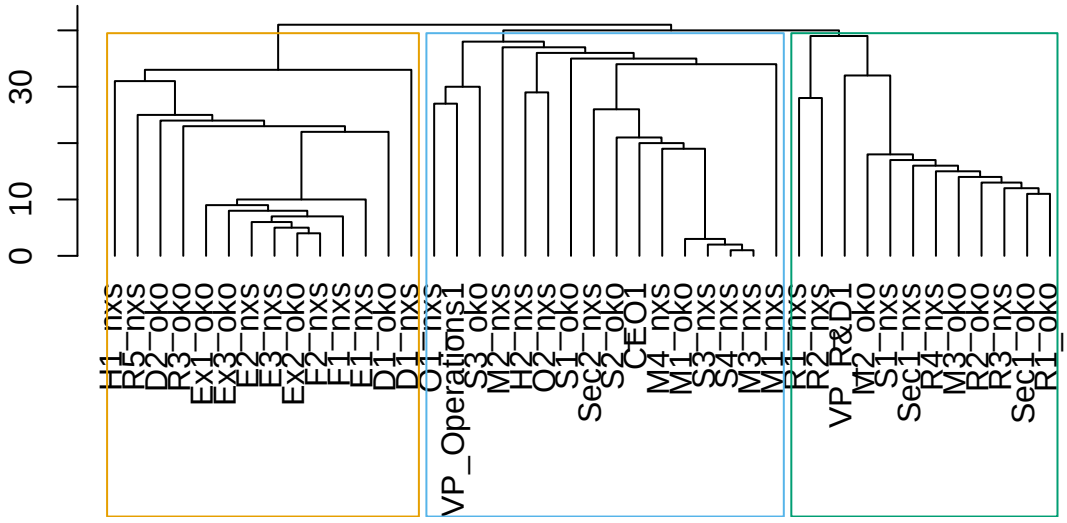
```
plot(NOc_eigen, combined_graph_dir, layout=layout, vertex.size=10, edge.arrow.size=.2,
     vertex.label=NA, # Hide vertex labels to reduce clutter
     vertex.color=vertex_colors,  #i was hoping this would work but no :(
     edge.color="gray", main = "Leading Eigenvector Clustering")
```

Leading Eigenvector Clustering

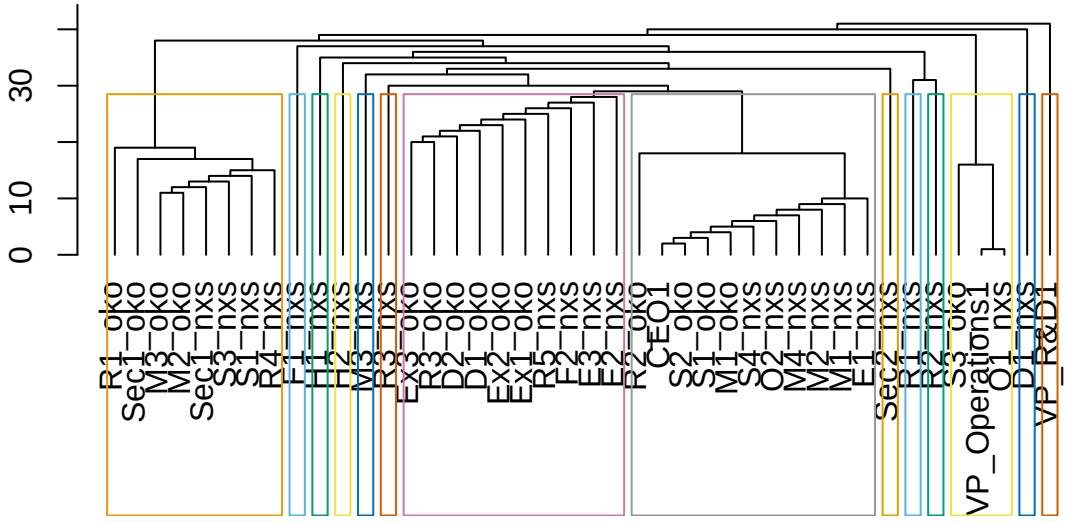


```
dendPlot(NOc)
```

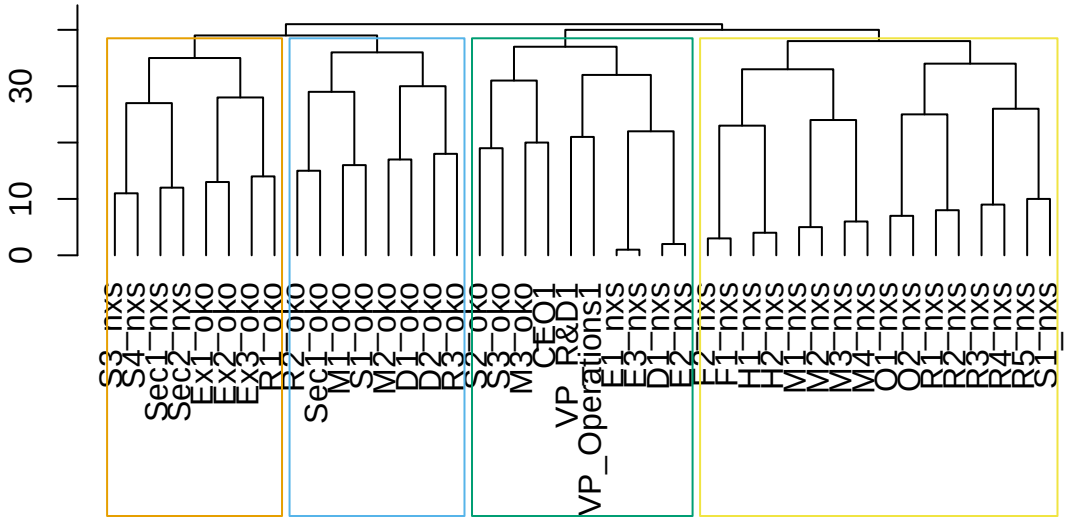
```
## Warning: `dendPlot()` was deprecated in igraph 2.0.0.
## i Please use `plot_dendrogram()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
dendPlot(NOc_bw)
```




```
dendPlot(NOc_eigen)
```




```
NxsOkoTbl <- data.frame(deg=igraph::degree(combined_graph_dir), btw = igraph::betweenness(combined_graph
```

```
## Warning: `evcent()` was deprecated in igraph 2.0.0.
## i Please use `eigen_centrality()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
NxsOkoTbl
```

```
##                deg        btw         evc
## D1_nxs           3   0.000000 0.097283626
## E1_nxs          29 201.870884 0.718105242
## E2_nxs          30 231.772615 1.000000000
## E3_nxs          20  56.882540 0.874542906
## F1_nxs          10  68.815148 0.325318180
## F2_nxs          15  10.642857 0.733197473
## H1_nxs           7  10.309325 0.180489569
## H2_nxs           9   0.000000 0.067036574
## M1_nxs           3   0.000000 0.024164876
```

```
## M2_nxs           3   3.333333 0.016823275
## M3_nxs          10   0.000000 0.282404154
## M4_nxs           4   0.000000 0.119217111
## O1_nxs           7  94.700000 0.084932720
## O2_nxs           2   0.000000 0.015623722
## R1_nxs           6  52.712698 0.031515523
## R2_nxs           5  19.369048 0.045641793
## R3_nxs          20 237.867014 0.265940005
## R4_nxs          17  25.261447 0.177901002
## R5_nxs          10  41.251774 0.189244679
## S1_nxs          14   0.622558 0.133511514
## S3_nxs          24 124.570574 0.396590116
## S4_nxs          22 110.994726 0.669793189
## Sec1_nxs        18  64.736844 0.192511524
## Sec2_nxs        13  72.837799 0.412239940
## Ex1_oko         29 248.115865 0.688577619
## Ex2_oko         16  22.505700 0.633157383
## Ex3_oko         18  59.032907 0.505079469
## R1_oko          21  48.700370 0.279470572
## R2_oko           8   1.068590 0.176157599
## Sec1_oko        20  58.327171 0.239310424
## M1_oko          11  31.110815 0.340847875
## S1_oko           4   2.876263 0.079186637
## M2_oko          16  21.549939 0.194866505
## D1_oko           4   0.000000 0.114305090
## D2_oko           7   1.966667 0.160836002
## R3_oko           4   0.000000 0.114305090
## S2_oko           5   0.000000 0.066456033
## S3_oko           5  22.021429 0.057216337
## M3_oko          18  61.166699 0.202614883
## CEO1            16 245.733000 0.259883370
## VP_R&D1          2   0.000000 0.021093561
## VP_Operations1   3   0.000000 0.009005721
```

```
cor(NxsOkoTbl)
```

```
##           deg       btw       evc
## deg 1.0000000 0.7487872 0.7991832
## btw 0.7487872 1.0000000 0.5705138
## evc 0.7991832 0.5705138 1.0000000
```

Discussion from HW6 on model (before the new CEO came into picture) FOR CONTEXT:

In model 2 I focused just on position level ERGM. We see significant baseline propensity for forming ties according to the edges estimate and strong positive coefficient value to form mutual ties. This aligns with model1 in that there are lots reciprocated relationships. looking and the nodefactor coefficients we can see than coefficients are highest in levels 1 and 2 and go lower in 3 and 4 (even though lower levels are insignificant). this tells us that individuals closer to the exec levels are forming ties more often reaffirming the finding from model1.

Looking at the graphs as well we can see than the simulated models didn't look much like the real one, but model 2 (to me personally) seemed closest to. Admittedly, my network is likely too entangled making it

difficult to locate more interesting metrics such as popularity for example due no non-convergence.

```
network <- asNetwork(combined_graph_dir)

# Add attributes to the network
network %v% "Age" <- NexusOko_nodeLst$AgeLvl
network %v% "PositionLvl" <- NexusOko_nodeLst$PositionLvl
```

```
set.seed(12345)


model1 <- ergm(network ~ edges +
               mutual +
               nodeicov("Age") +
               nodeocov("Age") +
               absdiff("Age") +
               nodeicov("PositionLvl") +
               nodeocov("PositionLvl") +
               absdiff("PositionLvl"),
               control = control.ergm(MCMC.samplesize = 10000, MCMC.burnin = 2000, MCMLE.maxit = 20),
               verbose = FALSE)
```

```
## 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 20:

## Warning: 'glpk' selected as the solver, but package 'Rglpk' is not available;
## falling back to 'lpSolveAPI'. This should be fine unless the sample size and/or
## the number of parameters is very big.

## Optimizing with step length 1.0000.

## The log-likelihood improved by 0.1205.

## Estimating equations are not within tolerance region.

## Iteration 2 of at most 20:

## Optimizing with step length 1.0000.

## The log-likelihood improved by 0.0010.

## Convergence test p-value: < 0.0001. Converged with 99% confidence.
## 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.
```

```
model2 <- ergm(network ~ edges + mutual + nodefactor("PositionLvl"),
               control = control.ergm(MCMC.samplesize = 10000, MCMC.burnin = 2000, MCMLE.maxit = 20),
               verbose = FALSE)
```

```
## 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 20:
## Optimizing with step length 1.0000.
## The log-likelihood improved by 0.2103.
## Estimating equations are not within tolerance region.
## Iteration 2 of at most 20:
## Optimizing with step length 1.0000.
## The log-likelihood improved by 0.0045.
## Convergence test p-value: < 0.0001. Converged with 99% confidence.
## 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.
```

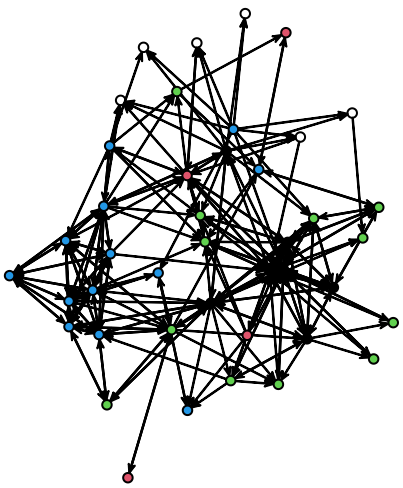
```
model3 <- ergm(network ~ edges + mutual + nodefactor("Age"),
               control = control.ergm(MCMC.samplesize = 10000, MCMC.burnin = 2000, MCMLE.maxit = 20),
               verbose = FALSE)
```

```
## 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 20:
## Optimizing with step length 1.0000.
## The log-likelihood improved by 0.0004.
## Convergence test p-value: < 0.0001. Converged with 99% confidence.
## 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.
```

```
# Summarize the model results
summary(model2)
```

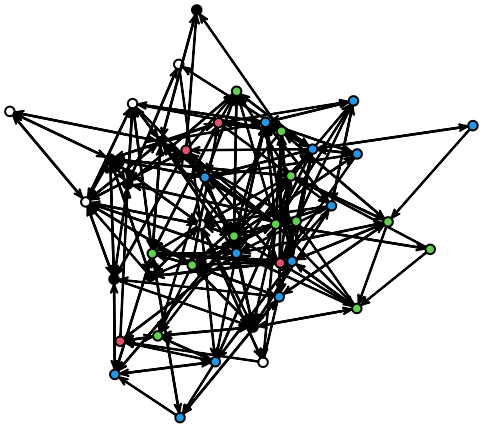
```
## Call:
## ergm(formula = network ~ edges + mutual + nodefactor("PositionLvl"),
##     control = control.ergm(MCMC.samplesize = 10000, MCMC.burnin = 2000,
##         MCMLE.maxit = 20), verbose = FALSE)
##
## Monte Carlo Maximum Likelihood Results:
##
##                           Estimate Std. Error MCMC % z value Pr(>|z|)
## edges                      -4.6389     0.3800      0 -12.207  < 1e-04 ***
## mutual                      3.4231     0.2579      0  13.274  < 1e-04 ***
## nodefactor.PositionLvl.1    1.4540     0.2146      0   6.777  < 1e-04 ***
## nodefactor.PositionLvl.2    0.6574     0.2386      0   2.755 0.005877 **
## nodefactor.PositionLvl.3    0.7254     0.2095      0   3.463 0.000534 ***
## nodefactor.PositionLvl.4    1.0590     0.2028      0   5.223  < 1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 2387  on 1722  degrees of freedom
##  Residual Deviance: 1091  on 1716  degrees of freedom
##
## AIC: 1103  BIC: 1136  (Smaller is better. MC Std. Err. = 0.5212)
```

```
plot(network, vertex.col="PositionLvl", main="real")
```



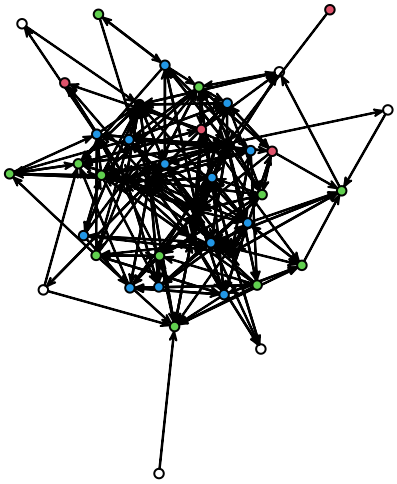

```
plot(simulate(model1), vertex.col="PositionLvl", main="simulated on 2vars")
```

simulated on 2vars



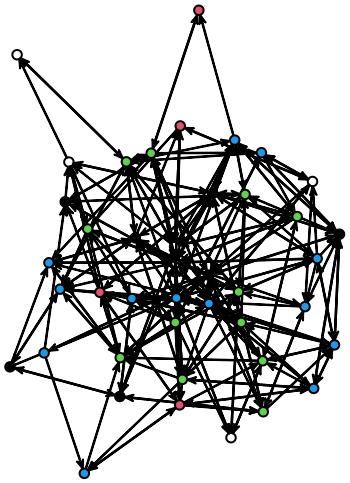
```
plot(simulate(model2), vertex.col="PositionLvl", main="simulated on position")
```

simulated on position



```
plot(simulate(model3), vertex.col="PositionLvl", main="simulated on Age")
```

simulated on Age



```
#gof_model1 <- gof(model1)
gof_model2 <- gof(model2)
#gof_model3 <- gof(model3)

#print(gof_model1)
print(gof_model2)
```

```
##
## Goodness-of-fit for in-degree
##
##           obs min mean max MC p-value
## idegree0    1   0 0.95   3       1.00
## idegree1    1   0 2.26   8       0.56
## idegree2    8   0 3.59   8       0.06
## idegree3    5   1 4.19  10       0.78
## idegree4    4   0 4.82  10       0.82
## idegree5    4   0 4.51   9       1.00
## idegree6    0   1 4.59   9       0.00
## idegree7    5   0 3.82   8       0.74
## idegree8    1   0 3.27   8       0.22
## idegree9    4   0 2.47   5       0.56
## idegree10   4   0 2.20   4       0.30
## idegree11   2   0 1.70   5       1.00
## idegree12   0   0 1.24   5       0.60
## idegree13   1   0 0.87   5       1.00
## idegree14   0   0 0.73   4       0.98
## idegree15   1   0 0.36   2       0.66
## idegree16   0   0 0.21   2       1.00
## idegree17   1   0 0.15   2       0.28
## idegree18   0   0 0.04   1       1.00
## idegree19   0   0 0.02   1       1.00
## idegree23   0   0 0.01   1       1.00
##
## Goodness-of-fit for out-degree
```

```
##
##           obs min mean max MC p-value
## odegree0    4   0 0.91   5       0.02
## odegree1    5   0 2.61   6       0.22
## odegree2    6   0 3.44   7       0.16
## odegree3    2   0 4.20  10       0.48
## odegree4    2   0 4.65  11       0.26
## odegree5    2   0 4.72  10       0.26
## odegree6    1   0 4.32  10       0.18
## odegree7    2   0 3.80   8       0.48
## odegree8    3   0 3.10   7       1.00
## odegree9    3   0 2.83   8       1.00
## odegree10   4   0 1.99   6       0.30
## odegree11   3   0 1.79   5       0.50
## odegree12   1   0 1.12   4       1.00
## odegree13   1   0 0.91   4       1.00
## odegree14   2   0 0.69   3       0.24
## odegree15   0   0 0.45   3       1.00
## odegree16   1   0 0.24   2       0.40
## odegree17   0   0 0.10   1       1.00
## odegree18   0   0 0.10   1       1.00
## odegree19   0   0 0.03   1       1.00
##
## Goodness-of-fit for edgewise shared partner
##
##           obs min  mean max MC p-value
## esp.OTP0   28  41 67.24  98       0.00
## esp.OTP1   38  57 75.49 103       0.00
## esp.OTP2   22  36 54.56  77       0.00
## esp.OTP3   44   9 32.62  48       0.16
## esp.OTP4   11   2 15.55  35       0.56
## esp.OTP5   24   0  6.80  19       0.00
## esp.OTP6   55   0  2.57  13       0.00
## esp.OTP7   18   0  0.99   8       0.00
## esp.OTP8    6   0  0.29   5       0.00
## esp.OTP9    7   0  0.11   2       0.00
## esp.OTP10   0   0  0.04   1       1.00
## esp.OTP12   1   0  0.00   0       0.00
##
## Goodness-of-fit for minimum geodesic distance
##
##     obs min   mean max MC p-value
## 1   254 215 256.26 309       0.92
## 2   576 661 833.11 994       0.00
## 3   493 388 489.49 598       0.98
## 4   111  12  62.35 123       0.08
## 5    40   0   3.24  28       0.00
## 6     8   0   0.13   6       0.00
## Inf 240   0  77.42 348       0.04
##
## Goodness-of-fit for model statistics
##
##                          obs min   mean max MC p-value
## edges                    254 215 256.26 309       0.92
```

```
## mutual                     88  69  88.70 108       1.00
## nodefactor.PositionLvl.1 157 124 156.62 185       0.96
## nodefactor.PositionLvl.2  34  21  33.80  52       0.94
## nodefactor.PositionLvl.3 111  78 112.07 152       0.96
## nodefactor.PositionLvl.4 185 146 188.13 234       0.82
```

```
#print(gof_model3)
```

Final Project Discussion on model focusing on position levels:

Analysis of ERGM model1 which includes age and position level variables after the new CEO team came in:

The new network (post CEO change) MLE results in the edges indicate that formation of ties in the network is a lot less likely to happen by pure chance so this is technically a sparce network. Mutual is very positive and do theres still a strong reciprocity tendency just like in the other network if a node is connected to another one. We still see that exec positions (level 1) are most likely to send of receive ties compared to lowere levels. level 4 now has a much higher chance in that regard than before, more likely then even position level 2 and 3. this could be explained by the internal tool initiative, but also by the fact that exec were talking to lowest levels much more often than in other edge lists.

The Goodness-of-Fit (GOF) results suggest several places where the model may be lagging in accurately representing the network's underlying structure. There are noticeable differences between the simulated and observed values of the GOF statistics for in-degree, out-degree, ESP, and geodesic distance. This is indicated by examples of odegree1 and odegree11 where pvalue is poor and moderate respectively. But also more clearly by p-values of edgewise shared partners in esp.OTP0,1,2,6,7,8,9,10, and 13; this particularly tells us that the model doesn't capture clustering and transitivity well. The fit for geodesic distances of 2 and 3 are poor also. But over all the model generally has a good fit especially for edges, mutual ties, and node factors for position levels, but higher order structures seem to be more difficult.

Lets run gwesp model

```
model2_gwesp <- ergm(network ~ edges + mutual + nodefactor("PositionLvl") +
                       gwesp(0.5, fixed = TRUE),
                     control = control.ergm(MCMC.samplesize = 5000, MCMC.burnin = 1000, MCMLE.maxit = 10)
                     verbose = FALSE)
```

```
## 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 10:

## Optimizing with step length 0.2979.

## The log-likelihood improved by 4.9891.

## Estimating equations are not within tolerance region.

## Iteration 2 of at most 10:
```

```
## Optimizing with step length 0.5811.
## The log-likelihood improved by 4.7456.
## Estimating equations are not within tolerance region.
## Iteration 3 of at most 10:
## Optimizing with step length 0.2582.
## The log-likelihood improved by 14.8960.
## Estimating equations are not within tolerance region.
## Iteration 4 of at most 10:
## Optimizing with step length 0.1010.
## The log-likelihood improved by 4.9753.
## Estimating equations are not within tolerance region.
## Iteration 5 of at most 10:
## Optimizing with step length 0.1252.
## The log-likelihood improved by 4.9083.
## Estimating equations are not within tolerance region.
## Iteration 6 of at most 10:
## Optimizing with step length 0.1106.
## The log-likelihood improved by 3.4271.
## Estimating equations are not within tolerance region.
## Iteration 7 of at most 10:
## Optimizing with step length 0.1977.
## The log-likelihood improved by 5.1051.
## Estimating equations are not within tolerance region.
## Iteration 8 of at most 10:
## Optimizing with step length 0.4091.
## The log-likelihood improved by 6.7200.
## Estimating equations are not within tolerance region.
## Iteration 9 of at most 10:
## Optimizing with step length 0.6466.
## The log-likelihood improved by 2.7133.
## Estimating equations are not within tolerance region.
## Iteration 10 of at most 10:
## Optimizing with step length 0.0396.
## The log-likelihood improved by 10.8627.
## Estimating equations are not within tolerance region.
## MCMLE estimation did not converge after 10 iterations. The estimated coefficients may not be accurat
```

```
## 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.
```

```
# Assess the Goodness-of-Fit for the model with GWESP
gof_model2_gwesp <- gof(model2_gwesp)
print(gof_model2_gwesp)
```

```
##
## Goodness-of-fit for in-degree
##
##           obs min mean max MC p-value
## idegree0    1   0 0.00   0       0.00
## idegree1    1   0 0.00   0       0.00
## idegree2    8   0 0.00   0       0.00
## idegree3    5   0 0.00   0       0.00
## idegree4    4   0 0.00   0       0.00
## idegree5    4   0 0.00   0       0.00
## idegree7    5   0 0.00   0       0.00
## idegree8    1   0 0.00   0       0.00
## idegree9    4   0 0.00   0       0.00
## idegree10   4   0 0.00   0       0.00
## idegree11   2   0 0.00   0       0.00
## idegree13   1   0 0.00   0       0.00
## idegree15   1   0 0.00   0       0.00
## idegree17   1   0 0.00   0       0.00
## idegree19   0   0 0.01   1       1.00
## idegree21   0   0 0.03   1       1.00
## idegree22   0   0 0.10   1       1.00
## idegree23   0   0 0.07   2       1.00
## idegree24   0   0 0.20   2       1.00
## idegree25   0   0 0.33   2       1.00
## idegree26   0   0 0.52   3       1.00
## idegree27   0   0 0.62   3       1.00
## idegree28   0   0 0.98   6       0.72
## idegree29   0   0 1.07   4       0.66
## idegree30   0   0 1.71   5       0.28
## idegree31   0   0 2.33   7       0.18
## idegree32   0   0 3.30   8       0.04
## idegree33   0   0 4.51   9       0.02
## idegree34   0   1 5.56  12       0.00
## idegree35   0   2 6.22  14       0.00
## idegree36   0   1 5.09  10       0.00
## idegree37   0   1 4.51  11       0.00
## idegree38   0   0 2.89   7       0.10
## idegree39   0   0 1.34   4       0.42
## idegree40   0   0 0.57   3       1.00
## idegree41   0   0 0.04   1       1.00
```

```
##
## Goodness-of-fit for out-degree
##
##           obs min mean max MC p-value
## odegree0    4   0 0.00   0       0.00
## odegree1    5   0 0.00   0       0.00
## odegree2    6   0 0.00   0       0.00
## odegree3    2   0 0.00   0       0.00
## odegree4    2   0 0.00   0       0.00
## odegree5    2   0 0.00   0       0.00
## odegree6    1   0 0.00   0       0.00
## odegree7    2   0 0.00   0       0.00
## odegree8    3   0 0.00   0       0.00
## odegree9    3   0 0.00   0       0.00
## odegree10   4   0 0.00   0       0.00
## odegree11   3   0 0.00   0       0.00
## odegree12   1   0 0.00   0       0.00
## odegree13   1   0 0.00   0       0.00
## odegree14   2   0 0.00   0       0.00
## odegree16   1   0 0.00   0       0.00
## odegree19   0   0 0.01   1       1.00
## odegree20   0   0 0.01   1       1.00
## odegree21   0   0 0.02   1       1.00
## odegree22   0   0 0.10   2       1.00
## odegree23   0   0 0.10   1       1.00
## odegree24   0   0 0.29   2       1.00
## odegree25   0   0 0.32   2       1.00
## odegree26   0   0 0.53   3       1.00
## odegree27   0   0 0.58   4       1.00
## odegree28   0   0 0.86   4       0.78
## odegree29   0   0 1.14   4       0.62
## odegree30   0   0 1.64   5       0.38
## odegree31   0   0 2.42   7       0.12
## odegree32   0   0 3.53   9       0.06
## odegree33   0   0 4.59  10       0.02
## odegree34   0   1 5.14  12       0.00
## odegree35   0   1 5.88  14       0.00
## odegree36   0   1 5.29  12       0.00
## odegree37   0   0 4.59   8       0.02
## odegree38   0   0 2.84   6       0.10
## odegree39   0   0 1.43   5       0.44
## odegree40   0   0 0.57   3       1.00
## odegree41   0   0 0.12   2       1.00
##
## Goodness-of-fit for edgewise shared partner
##
##           obs min   mean max MC p-value
## esp.OTP0   28   0   0.00   0       0.00
## esp.OTP1   38   0   0.00   0       0.00
## esp.OTP2   22   0   0.00   0       0.00
## esp.OTP3   44   0   0.00   0       0.00
## esp.OTP4   11   0   0.00   0       0.00
## esp.OTP5   24   0   0.00   0       0.00
## esp.OTP6   55   0   0.00   0       0.00
```

```
## esp.OTP7   18   0   0.00   0       0.00
## esp.OTP8    6   0   0.00   0       0.00
## esp.OTP9    7   0   0.00   0       0.00
## esp.OTP10   0   0   0.01   1       1.00
## esp.OTP11   0   0   0.03   1       1.00
## esp.OTP12   1   0   0.06   2       0.10
## esp.OTP13   0   0   0.20   4       1.00
## esp.OTP14   0   0   0.47   5       1.00
## esp.OTP15   0   0   1.21   7       1.00
## esp.OTP16   0   0   2.53  13       0.54
## esp.OTP17   0   0   4.14  18       0.30
## esp.OTP18   0   0   7.76  22       0.08
## esp.OTP19   0   0  13.00  34       0.02
## esp.OTP20   0   1  19.88  39       0.00
## esp.OTP21   0   6  28.80  57       0.00
## esp.OTP22   0  12  41.19  74       0.00
## esp.OTP23   0  18  55.93  91       0.00
## esp.OTP24   0  34  75.89 123       0.00
## esp.OTP25   0  47  98.71 155       0.00
## esp.OTP26   0  72 121.12 166       0.00
## esp.OTP27   0  95 143.24 194       0.00
## esp.OTP28   0 116 156.85 200       0.00
## esp.OTP29   0 110 162.16 222       0.00
## esp.OTP30   0  92 149.34 210       0.00
## esp.OTP31   0  72 124.27 203       0.00
## esp.OTP32   0  36  93.99 159       0.00
## esp.OTP33   0  18  61.66 130       0.00
## esp.OTP34   0   6  36.83  84       0.00
## esp.OTP35   0   0  17.53  50       0.02
## esp.OTP36   0   0   7.51  24       0.12
## esp.OTP37   0   0   2.64  18       0.62
## esp.OTP38   0   0   0.69   7       1.00
## esp.OTP39   0   0   0.08   2       1.00
## esp.OTP40   0   0   0.02   2       1.00
##
## Goodness-of-fit for minimum geodesic distance
##
##     obs  min    mean  max MC p-value
## 1   254 1388 1427.74 1469          0
## 2   576  253  294.26  334          0
## 3   493    0    0.00    0          0
## 4   111    0    0.00    0          0
## 5    40    0    0.00    0          0
## 6     8    0    0.00    0          0
## Inf 240    0    0.00    0          0
##
## Goodness-of-fit for model statistics
##
##                                obs      min     mean      max MC p-value
## edges                      254.000 1388.000 1427.740 1469.000          0
## mutual                      88.000  596.000  631.950  666.000          0
## nodefactor.PositionLvl.1 157.000  481.000  503.980  534.000          0
## nodefactor.PositionLvl.2  34.000  199.000  225.750  252.000          0
## nodefactor.PositionLvl.3 111.000  783.000  817.020  849.000          0
```

```
## nodefactor.PositionLvl.4 185.000  889.000  922.040  956.000          0
## gwesp.OTP.fixed.0.5      336.729 2288.425 2353.945 2421.972          0
```

```
# Summarize the model with GWESP
summary(model2_gwesp)
```

```
## Call:
## ergm(formula = network ~ edges + mutual + nodefactor("PositionLvl") +
##     gwesp(0.5, fixed = TRUE), control = control.ergm(MCMC.samplesize = 5000,
##     MCMC.burnin = 1000, MCMLE.maxit = 10), verbose = FALSE)
##
## Monte Carlo Maximum Likelihood Results:
##
##                          Estimate Std. Error MCMC % z value Pr(>|z|)
## edges                     -5.2024    17.9575      0  -0.290    0.772
## mutual                     1.6419    19.6443      0   0.084    0.933
## nodefactor.PositionLvl.1   0.5488    10.2279      0   0.054    0.957
## nodefactor.PositionLvl.2  -0.4031     7.5671      0  -0.053    0.958
## nodefactor.PositionLvl.3   0.2468    12.2541      0   0.020    0.984
## nodefactor.PositionLvl.4   0.4592    10.2723      0   0.045    0.964
## gwesp.OTP.fixed.0.5        3.0840     6.5480      0   0.471    0.638
##
##      Null Deviance: 2387  on 1722  degrees of freedom
##  Residual Deviance: 4654  on 1715  degrees of freedom
##
## AIC: 4668  BIC: 4707  (Smaller is better. MC Std. Err. = 79.42)
```

Analysis of GWESP model stats:

the gwesp term accounts for the clustering tendency which is something our baseline model steruggled with. From our MLE param estimates we can see that edges still has a negative tendency, mutual has positive tendency hence the indication of a sparce and highly transitive network. position levels 1 and 4 respectively still hold the strongest positive estimates, but their estimate values decreased. Interestingly level 2 is now insignificant. Now the GWESP factor coefficient shows strong clustering ability which tells us that nodes that share neighbors are probably going to form ties thus pushing for increase in transitivity and tradic closures. from the GOF values though we can clearly see that gwesp term has a really positive impact on the model compared to the previous one. ESP GOF in particular now has only Opt 6 and 7 with 0 p-value, as well as 9 and 2 being moderate. the indegree and out degree both have no poor node representations

```
# Define the model with GWDSP
# model1_gwdsp <- ergm(network ~ edges + mutual +
#                        nodeicov("Age") + nodeocov("Age") + absdiff("Age") +
#                        nodeicov("PositionLvl") + nodeocov("PositionLvl") + absdiff("PositionLvl") +
#                        gwdsp(0.5, fixed = TRUE),
#                      control = control.ergm(MCMC.samplesize = 10000, MCMC.burnin = 2000, MCMLE.maxit =
#                      verbose = FALSE)

model2_gwdsp <- ergm(network ~ edges + mutual + nodefactor("PositionLvl") + gwdsp(1, fixed = FALSE, cut
                control = control.ergm(MCMC.samplesize = 5000, MCMC.burnin = 1000, MCMLE.maxit = 10),
                verbose = FALSE)
```

```
## Warning: In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter
## 'decay' has no effect. To specify an initial value for 'decay', use the
## 'control.ergm()' parameter 'init='.
```

```
## 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 10:
## Optimizing with step length 0.0845.
## The log-likelihood improved by 5.6374.
## Estimating equations are not within tolerance region.
## Iteration 2 of at most 10:
## Optimizing with step length 0.4532.
## The log-likelihood improved by 2.9618.
## Estimating equations are not within tolerance region.
## Iteration 3 of at most 10:
## Optimizing with step length 0.0670.
## The log-likelihood improved by 3.0958.
## Estimating equations are not within tolerance region.
## Iteration 4 of at most 10:
## Optimizing with step length 0.3372.
## The log-likelihood improved by 5.1197.
## Estimating equations are not within tolerance region.
## Iteration 5 of at most 10:
## Optimizing with step length 0.0021.
## The log-likelihood improved by 0.5569.
## Estimating equations are not within tolerance region.
## Estimating equations did not move closer to tolerance region more than 1 time(s) in 4 steps; increas
## Iteration 6 of at most 10:
## Optimizing with step length 0.0590.
## The log-likelihood improved by 2.9446.
## Estimating equations are not within tolerance region.
## Iteration 7 of at most 10:
## Optimizing with step length 0.0012.
## The log-likelihood improved by 0.1799.
## Estimating equations are not within tolerance region.
## Iteration 8 of at most 10:
## Optimizing with step length 0.0059.
```

```
## The log-likelihood improved by 0.9670.

## Estimating equations are not within tolerance region.

## Estimating equations did not move closer to tolerance region more than 1 time(s) in 4 steps; increas

## Iteration 9 of at most 10:

## Optimizing with step length 0.0265.

## The log-likelihood improved by 2.6599.

## Estimating equations are not within tolerance region.

## Iteration 10 of at most 10:

## Optimizing with step length 0.1395.

## The log-likelihood improved by 3.1577.

## Estimating equations are not within tolerance region.

## MCMLE estimation did not converge after 10 iterations. The estimated coefficients may not be accurat

## Finished MCMLE.

## Evaluating log-likelihood at the estimate.

## Warning: In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter
## 'decay' has no effect. To specify an initial value for 'decay', use the
## 'control.ergm()' parameter 'init='.

## Fitting the dyad-independent submodel...
## Bridging between the dyad-independent submodel and the full model...
## Setting up bridge sampling...

## Warning: In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter
## 'decay' has no effect. To specify an initial value for 'decay', use the
## 'control.ergm()' parameter 'init='.

## 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.
```

```
# Assess the Goodness-of-Fit for the model with GWDSP
gof_model2_gwdsp <- gof(model2_gwdsp)
```

```
## Warning: In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To s
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
```

```
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
```

```
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
## In term 'gwdsp' in package 'ergm': When 'fixed=FALSE' parameter 'decay' has no effect. To specify an
```

```
print(gof_model2_gwdsp)
```

```
##
## Goodness-of-fit for in-degree
##
##           obs min mean max MC p-value
## idegree0    1   0 3.76   7       0.12
## idegree1    1   0 3.14   9       0.32
## idegree2    8   0 1.07   6       0.00
## idegree3    5   0 0.27   3       0.00
## idegree4    4   0 0.03   1       0.00
## idegree5    4   0 0.01   1       0.00
## idegree7    5   0 0.00   0       0.00
## idegree8    1   0 0.00   0       0.00
## idegree9    4   0 0.00   0       0.00
## idegree10   4   0 0.00   0       0.00
## idegree11   2   0 0.00   0       0.00
## idegree12   0   0 0.07   1       1.00
## idegree13   1   0 0.15   4       0.24
## idegree14   0   0 0.20   2       1.00
```

```
## idegree15   1   0 0.47   4       0.64
## idegree16   0   0 0.85   7       0.84
## idegree17   1   0 1.34   4       1.00
## idegree18   0   0 2.21   5       0.20
## idegree19   0   0 2.74   8       0.06
## idegree20   0   0 3.18   9       0.14
## idegree21   0   0 3.53   9       0.08
## idegree22   0   0 3.50   8       0.04
## idegree23   0   0 2.98  12       0.16
## idegree24   0   0 2.75   9       0.22
## idegree25   0   0 2.62   7       0.12
## idegree26   0   0 2.09   5       0.22
## idegree27   0   0 1.76   5       0.34
## idegree28   0   0 1.29   5       0.68
## idegree29   0   0 0.89   3       0.82
## idegree30   0   0 0.54   3       1.00
## idegree31   0   0 0.34   3       1.00
## idegree32   0   0 0.16   2       1.00
## idegree33   0   0 0.05   2       1.00
## idegree34   0   0 0.01   1       1.00
##
## Goodness-of-fit for out-degree
##
##           obs min mean max MC p-value
## odegree0    4   0 3.74   7       1.00
## odegree1    5   0 3.08   8       0.42
## odegree2    6   0 1.22   5       0.00
## odegree3    2   0 0.19   2       0.02
## odegree4    2   0 0.04   1       0.00
## odegree5    2   0 0.00   0       0.00
## odegree6    1   0 0.01   1       0.02
## odegree7    2   0 0.01   1       0.00
## odegree8    3   0 0.00   0       0.00
## odegree9    3   0 0.00   0       0.00
## odegree10   4   0 0.00   0       0.00
## odegree11   3   0 0.04   2       0.00
## odegree12   1   0 0.02   1       0.04
## odegree13   1   0 0.13   3       0.22
## odegree14   2   0 0.21   2       0.02
## odegree15   0   0 0.52   3       1.00
## odegree16   1   0 0.80   5       0.98
## odegree17   0   0 1.32   6       0.54
## odegree18   0   0 2.13   6       0.38
## odegree19   0   0 2.72   8       0.16
## odegree20   0   0 3.13   8       0.04
## odegree21   0   0 3.48  10       0.02
## odegree22   0   1 3.52   8       0.00
## odegree23   0   0 3.36   9       0.06
## odegree24   0   0 2.87   9       0.16
## odegree25   0   0 2.57   7       0.26
## odegree26   0   0 2.08   7       0.26
## odegree27   0   0 1.48   5       0.44
## odegree28   0   0 1.16   4       0.52
## odegree29   0   0 1.09   5       0.86
```

```
## odegree30   0   0 0.56   4       1.00
## odegree31   0   0 0.35   3       1.00
## odegree32   0   0 0.14   1       1.00
## odegree33   0   0 0.03   1       1.00
##
## Goodness-of-fit for edgewise shared partner
##
##           obs min  mean max MC p-value
## esp.OTP0   28   0  6.06  18       0.00
## esp.OTP1   38   0  0.26   3       0.00
## esp.OTP2   22   0  0.02   2       0.00
## esp.OTP3   44   0  0.08   4       0.00
## esp.OTP4   11   0  0.13   5       0.00
## esp.OTP5   24   0  0.52   5       0.00
## esp.OTP6   55   0  1.57  12       0.00
## esp.OTP7   18   0  4.26  19       0.02
## esp.OTP8    6   0  8.95  43       0.90
## esp.OTP9    7   1 17.21  52       0.28
## esp.OTP10   0   7 29.12  59       0.00
## esp.OTP11   0  17 45.98  78       0.00
## esp.OTP12   1  31 59.61  86       0.00
## esp.OTP13   0  35 73.47 109       0.00
## esp.OTP14   0  38 82.52 108       0.00
## esp.OTP15   0  33 83.23 108       0.00
## esp.OTP16   0  21 77.83 110       0.00
## esp.OTP17   0  20 70.26 112       0.00
## esp.OTP18   0  13 56.66  94       0.00
## esp.OTP19   0   6 44.87 101       0.00
## esp.OTP20   0   5 34.76  78       0.00
## esp.OTP21   0   2 24.20  65       0.00
## esp.OTP22   0   0 15.76  46       0.04
## esp.OTP23   0   0  9.84  28       0.12
## esp.OTP24   0   0  6.19  23       0.26
## esp.OTP25   0   0  3.31  16       0.58
## esp.OTP26   0   0  1.45  12       0.90
## esp.OTP27   0   0  0.91   8       1.00
## esp.OTP28   0   0  0.40   5       1.00
## esp.OTP29   0   0  0.07   2       1.00
## esp.OTP30   0   0  0.03   1       1.00
##
## Goodness-of-fit for minimum geodesic distance
##
##     obs min   mean max MC p-value
## 1   254 528 759.53 858       0.00
## 2   576 243 358.01 420       0.00
## 3   493   0   2.92  82       0.00
## 4   111   0   1.92  98       0.00
## 5    40   0   1.35  63       0.04
## 6     8   0   0.97  61       0.04
## 7     0   0   0.75  73       1.00
## 8     0   0   0.43  42       1.00
## Inf 240 493 596.12 879       0.00
##
## Goodness-of-fit for model statistics
```

```
##
##                           obs min   mean max MC p-value
## edges                     254 528 759.53 858       0.00
## mutual                     88 231 340.75 394       0.00
## nodefactor.PositionLvl.1 157 319 376.75 409       0.00
## nodefactor.PositionLvl.2  34  58 112.19 144       0.00
## nodefactor.PositionLvl.3 111 278 472.88 557       0.00
## nodefactor.PositionLvl.4 185 394 550.55 637       0.00
## dsp.OTP#1                352   0   4.61  33       0.00
## dsp.OTP#2                198   0   0.55  16       0.00
## dsp.OTP#3                 90   0   0.32  16       0.00
## dsp.OTP#4                 42   0   0.37   9       0.00
## dsp.OTP#5                 33   0   0.86   9       0.00
## dsp.OTP#6                 55   0   2.80  29       0.00
## dsp.OTP#7                 18   0   7.22  40       0.14
## dsp.OTP#8                  6   0  15.22  81       0.58
## dsp.OTP#9                  7   1  29.59  93       0.08
## dsp.OTP#10                 0   9  48.71 116       0.00
## dsp.OTP#11                 0  31  75.62 118       0.00
## dsp.OTP#12                 1  48  96.64 135       0.00
## dsp.OTP#13                 0  67 116.43 174       0.00
## dsp.OTP#14                 0  46 125.01 165       0.00
## dsp.OTP#15                 0  41 123.72 172       0.00
## dsp.OTP#16                 0  28 112.34 162       0.00
## dsp.OTP#17                 0  21  97.94 148       0.00
## dsp.OTP#18                 0  16  76.29 134       0.00
## dsp.OTP#19                 0  10  58.72 126       0.00
## dsp.OTP#20                 0   5  44.19 103       0.00
## dsp.OTP#21                 0   3  29.29  78       0.00
## dsp.OTP#22                 0   0  19.32  49       0.02
## dsp.OTP#23                 0   0  11.34  34       0.10
## dsp.OTP#24                 0   0   7.32  33       0.22
## dsp.OTP#25                 0   0   3.87  18       0.48
## dsp.OTP#26                 0   0   1.65  12       0.80
## dsp.OTP#27                 0   0   1.03  10       1.00
## dsp.OTP#28                 0   0   0.41   5       1.00
## dsp.OTP#29                 0   0   0.07   2       1.00
## dsp.OTP#30                 0   0   0.03   1       1.00
```

```
# Summarize the model with GWDSP
summary(model2_gwdsp)
```

```
## Call:
## ergm(formula = network ~ edges + mutual + nodefactor("PositionLvl") +
##     gwdsp(1, fixed = FALSE, cutoff = 50), control = control.ergm(MCMC.samplesize = 5000,
##     MCMC.burnin = 1000, MCMLE.maxit = 10), verbose = FALSE)
##
## Monte Carlo Maximum Likelihood Results:
##
##                            Estimate Std. Error MCMC % z value Pr(>|z|)
## edges                    -3.195e+00  4.877e-01      2  -6.552  < 1e-04 ***
## mutual                    3.387e+00  2.920e-01      4  11.600  < 1e-04 ***
## nodefactor.PositionLvl.1  1.472e+00  3.635e-01      2   4.050  < 1e-04 ***
## nodefactor.PositionLvl.2  7.481e-01  3.802e-01      3   1.968 0.049089 *
## nodefactor.PositionLvl.3  8.166e-01  2.826e-01      3   2.890 0.003856 **
```

```
## nodefactor.PositionLvl.4  9.654e-01  2.708e-01      2   3.565 0.000364 ***
## gwdsp.OTP                -3.601e-01  6.173e-02      2  -5.834  < 1e-04 ***
## gwdsp.OTP.decay           1.303e-10  3.500e-01      2   0.000 1.000000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 2387  on 1722  degrees of freedom
##  Residual Deviance: 1296  on 1714  degrees of freedom
##
## AIC: 1312  BIC: 1356  (Smaller is better. MC Std. Err. = 13.57)
```

Analysis of GWDSP model stats:

In the MLE parameters we see general similarity in significant coefficients across all modes. The negative coeff for gwdsp (OTP) tells us there is a tendency to avoid dyadic shared parnerships (which i'm slightly skeptical about in all honesty, but it could be again a by product of the non execs lacking interactions). The decay term is insignificant. The GOF for in degree and out degree shows reasobly good ability to capture the distributions, its also capturing geodecis distance distribution pretty well, but there are still some deviations; this model is better than Baseline, but not as good as GWESP model. Also it stuggles with ESP especialy with higher values.