# Data Assignment 2

2022-11-15

## Introduction

This project focuses on 3 networks reaching across Europe: cultural similarity, trading partnership, and militaristic alliances. The goal is to examine the prominence of actors across these networks. One interesting finding throughout the project is that some networks favor small countries for prominence, while others favor large countries.

### A Note on Data Collection and Preperation

This project relied heavily on public data sources from the internet. These data were scraped and prepared using Python scripts, which can be found in a folder sumitted wirh this assignment.Discussion of the data sources (except for the Correlates of War source) can be found in Assignment one. A brief discussion of Correlates of War data can be found in the militaristic alliances section of this paper.

## Cultural Similarity

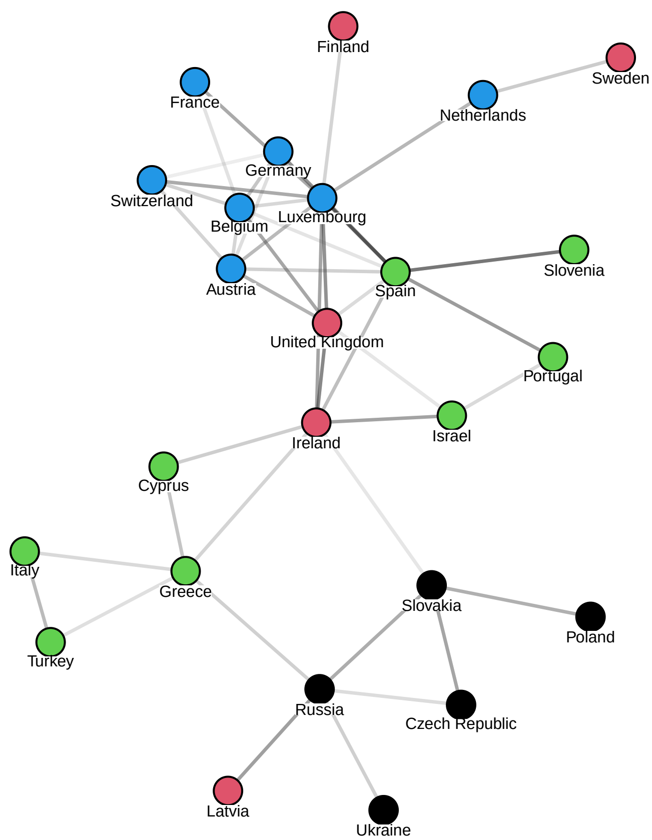
```
library(statnet)
library(intergraph)
library(tidyverse)
```

Import and join cultural similarity network data:

```
cultural_exchange_edgelist <- read.csv("cultural_exchange_edgelist")
cultural_strength <- read.csv("cultural_exchange_strength")
cultural_node_attributes <- read_csv("node_attributes_cultural_exchange")
cultural_exchange_network <- as.network(cultural_exchange_edgelist, directed = FALSE)

set.edge.attribute(cultural_exchange_network, "weight", cultural_strength[['strength']])

set.vertex.attribute(cultural_exchange_network, "democracy_index",
                     cultural_node_attributes$democracy_index)
set.vertex.attribute(cultural_exchange_network, "press_freedoom",
                     cultural_node_attributes$press_freedom)
set.vertex.attribute(cultural_exchange_network, "population",
                     cultural_node_attributes$population)
set.vertex.attribute(cultural_exchange_network, "GDP",
                     cultural_node_attributes$GDP)
set.vertex.attribute(cultural_exchange_network, "region",
                     cultural_node_attributes$region)
```

There is an issue with this network in that it is almost a clique, with differently weighted edges. To examine centrality, we must consider a sub-graph of the network. For now we'll choose edges with a weight greater than .78. We will also set the opacity of the edge based on it's weight to the fourth power. We will display the region of each country by color.

```
cultural_similarity_subgraph <- get.inducedSubgraph(
  cultural_exchange_network, eid=which(cultural_exchange_network %e% "weight" > .78))

cultural_exchange_network.ecol <- gray(1 - (cultural_exchange_network %e% "weight"),
                                       alpha = (cultural_exchange_network %e% "weight")^4)
cultural_exchange_network.vcol <- as.factor(cultural_similarity_subgraph %v% "region")

op <- par(mar=c(0,0,0,0))
gplot(cultural_similarity_subgraph, gmode = "graph", displaylabels = TRUE, label.pos = 1,
      boxed.labels = T, label.border = 0, label.cex = .5, label.pad = 0,
      edge.col = cultural_exchange_network.ecol,
      vertex.col = cultural_exchange_network.vcol )
```

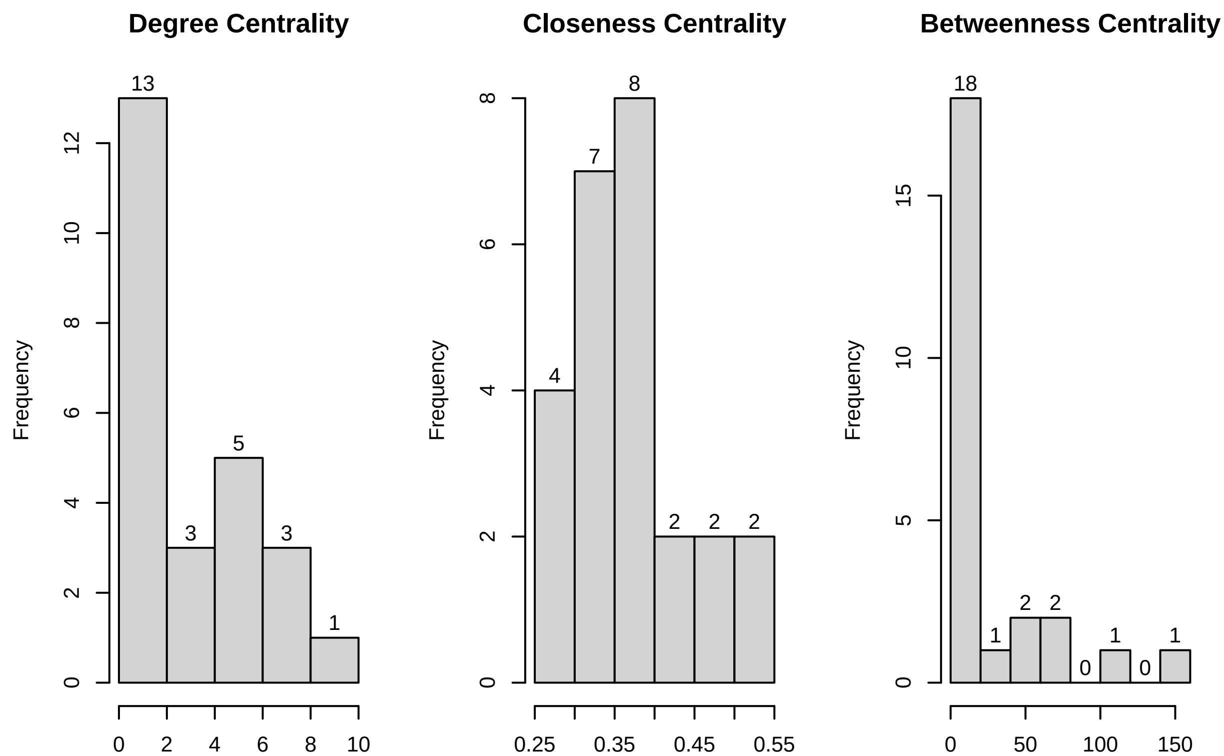



```
par(op)
```

A histogram is a good way to see the distribution of degree, closeness, and betweenness centrality.

```
par(mfrow=c(1, 3))
hist(degree(cultural_similarity_subgraph, gmode="graph"), labels = T, breaks = "Sturges",
     main = "Degree Centrality", xlab = "")
```

```r
hist(closeness(cultural_similarity_subgraph, gmode="graph"), labels = T,
     breaks = "Sturges", main = "Closeness Centrality", xlab = "")
hist(betweenness(cultural_similarity_subgraph, gmode="graph"), labels = T,
     breaks = "Sturges", main = "Betweenness Centrality", xlab = "")
```




Each of these distributions are skewed right, which indicates that the majority of countries are not connected strongly by cultural similarity. The degree centrality chart shows us that one key player seems to hoold the most strong culturak ties,

To examine the centrality by country, we will print out a dataframe with the three types of centrality for each node. We will also include the network level statistics for these measures.

```r
vertex <- cultural_similarity_subgraph %v% "vertex.names"
degree_centrality <- degree(cultural_similarity_subgraph, gmode="graph")
closeness_centrality <- closeness(cultural_similarity_subgraph, gmode="graph")
betweenness_centrality <- betweenness(cultural_similarity_subgraph, gmode="graph")

centrality_df <- data.frame(vertex, degree_centrality, closeness_centrality,
                            betweenness_centrality)
centrality_df[nrow(centrality_df) + 1,] =
  c("Network", centralization(cultural_similarity_subgraph,degree),
               centralization(cultural_similarity_subgraph,closeness),
               centralization(cultural_similarity_subgraph,betweenness))

print(centrality_df)
```

```
##              vertex degree_centrality closeness_centrality betweenness_centrality
```

```
## 1         Austria                 6     0.387096774193548        1.72619047619048
## 2         Belgium                 7      0.39344262295082        5.55952380952381
## 3  Czech Republic                 2     0.324324324324324                       0
## 4         Finland                 1     0.342857142857143                       0
## 5          France                 2     0.347826086956522                       0
## 6         Germany                 5                 0.375                    0.75
## 7          Greece                 5     0.436363636363636                      70
## 8         Ireland                 7     0.545454545454545        149.035714285714
## 9          Israel                 3     0.387096774193548                       6
## 10          Italy                 2     0.311688311688312                       0
## 11         Latvia                 1     0.263736263736264                       0
## 12     Luxembourg                10      0.51063829787234        106.119047619048
## 13    Netherlands                 2     0.352941176470588                      23
## 14         Poland                 1                   0.3                       0
## 15       Portugal                 2     0.338028169014085       0.809523809523809
## 16         Russia                 5     0.352941176470588                    51.5
## 17       Slovakia                 4     0.421052631578947                      61
## 18       Slovenia                 1     0.328767123287671                       0
## 19          Spain                 8                  0.48        51.5119047619048
## 20         Sweden                 1     0.263736263736264                       0
## 21    Switzerland                 4     0.358208955223881                       0
## 22         Turkey                 2     0.311688311688312                       0
## 23 United Kingdom                 6     0.461538461538462        13.9880952380952
## 24        Ukraine                 1     0.263736263736264                       0
## 25         Cyprus                 2      0.39344262295082                       0
## 26        Network 0.289855072463768     0.190310664065831        0.48081111973775
```

Interestingly, Luxembourg emerges as the actor with the highest degree centrality, and high closeness centrality and betweenness centrality. Interestingly, large countries such as Germany and France don't have many strong cultural ties as one might expect. This juxtaposition seems to suggest that small country's cultures may be much more easily influenced by their neighbors, and therefore often have higher amounts of strong cultural ties.

Ireland too plays an important role. It has the highest betweenness score and clearly acts as a cultural bridge between Western, Southern, and Eastern European countries. This is curious considering the geographic location of the island.

The characteristics of grouping are also interesting. It seems that Western and Eastern countries have a tendency to stick together culturally, while Northern countries are more dispersed in their cultural similarity. This likely hints at the rugged individualism of Northern European countries. Southern countries attempt to group, but are culturally bisected by Ireland, giving rise to it's high centrality score.

At the network level low centrality on all acounts. This makes sense, considering the weight threshold for ties of .78 we've chosen. If we chose a higher threshold it would go up. If we had no threshold at all it would likely drop to 0.

**Use Modularity to Detect Community Structure:** To prepare we detach the statnet package, load the igraph and intergraph packages, and change the statnet network object to an igraph object.

```
detach(name = package:statnet)
library(intergraph)
library(igraph)

icultural_similarity_subgraph <- asIgraph(cultural_similarity_subgraph)
```

Now we will use the modularity function of igraph to measure modularity by region. Before doing this we need to convert our region attribute to a numerical vector.

```
V(icultural_similarity_subgraph)$numeric_region <-
  case_when(V(icultural_similarity_subgraph)$region == "N" ~ 1,
            V(icultural_similarity_subgraph)$region == "E" ~ 2,
            V(icultural_similarity_subgraph)$region == "S" ~ 3,
            V(icultural_similarity_subgraph)$region == "W" ~ 4)
modularity(icultural_similarity_subgraph,V(icultural_similarity_subgraph)$numeric_region,
           weights = E(icultural_similarity_subgraph)$weight)
```

```
## [1] 0.2854439
```

The modularity function shows that region is not an incredibly good predictor for clustering. This is due to the disperse Northern countries and bisected Southern countries which were previously discussed.

Now we will attempt to use the walktrap and louvain algorithm to detect community.

```
cw <- cluster_walktrap(icultural_similarity_subgraph,
                       weights = E(icultural_similarity_subgraph)$weight)
modularity(cw)
```

```
## [1] 0.3771873
```

```
cl <- cluster_louvain(icultural_similarity_subgraph,
                      weights = E(icultural_similarity_subgraph)$weight)
modularity(cl)
```

```
## [1] 0.4306485
```

From the modularity of these algorithms, .37 and .43 respectively, we can infer that modularity would likely remain around 0.4 and never rise above 0.5. Thus the extent of subgroup structure within European cultural similarity are not, based on any variable, resounding but is significant.

### Economic trading:

We will start with statnet again and then move to igraph, we we must start by unloading and reloading packages.

```
detach(package:igraph)
library(statnet)
```

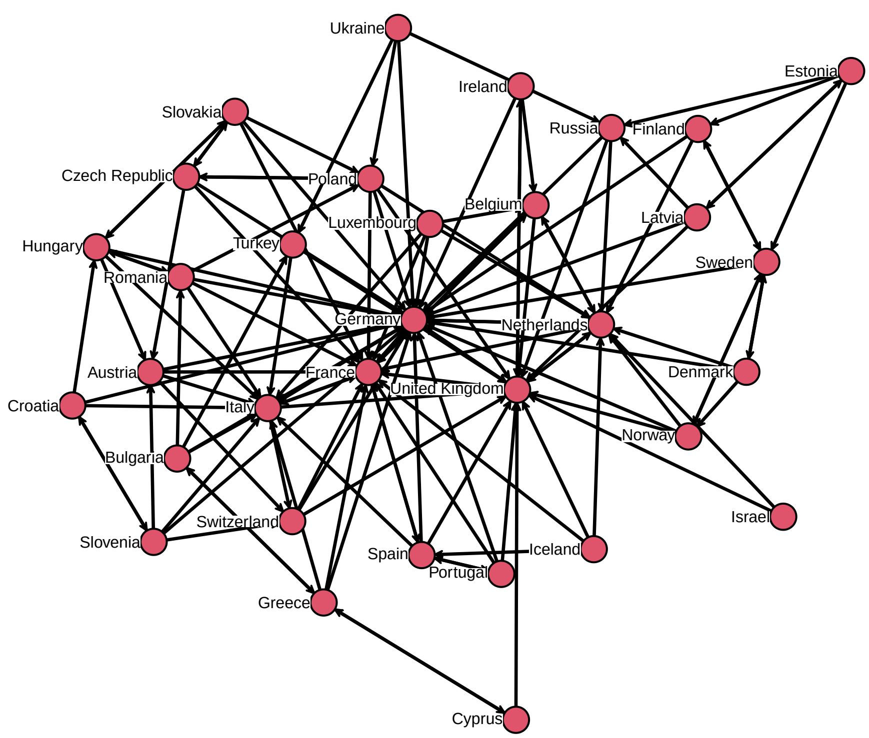
We will join our economic trading network data into a statnet network object

```
economic_exchange_edgelist <- read.csv("trade_edge_list")
economic_exchange_edge_attributes <- read_csv("trade_edge_attributes")
economic_node_attributes <- read_csv("node_attributes_economic_exchange")
economic_exchange_network <- as.network(economic_exchange_edgelist)

set.edge.attribute(economic_exchange_network, "trade_in_USD",
```

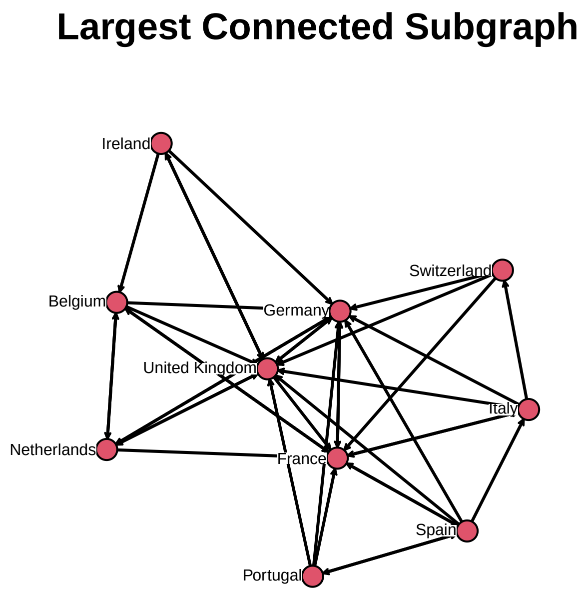
```
                       economic_exchange_edge_attributes$Trade)
set.edge.attribute(economic_exchange_network, "share_percent",
                       economic_exchange_edge_attributes$`Share Percent`)
set.vertex.attribute(economic_exchange_network, "democracy_index",
                       economic_node_attributes$democracy_index)
set.vertex.attribute(economic_exchange_network, "press_freedom",
                       economic_node_attributes$press_freedom)
set.vertex.attribute(economic_exchange_network, "population",
                       economic_node_attributes$population)
set.vertex.attribute(economic_exchange_network, "GDP",
                       economic_node_attributes$GDP)
set.vertex.attribute(economic_exchange_network, "region",
                       economic_node_attributes$region)
```

We will plot the network, to gain some intuition for what the network looks like, and consider the centrality histograms associated with it. Note that for measuring closeness we must only consider the largest component of the graph, so as not to cause errors due to disconnected components.

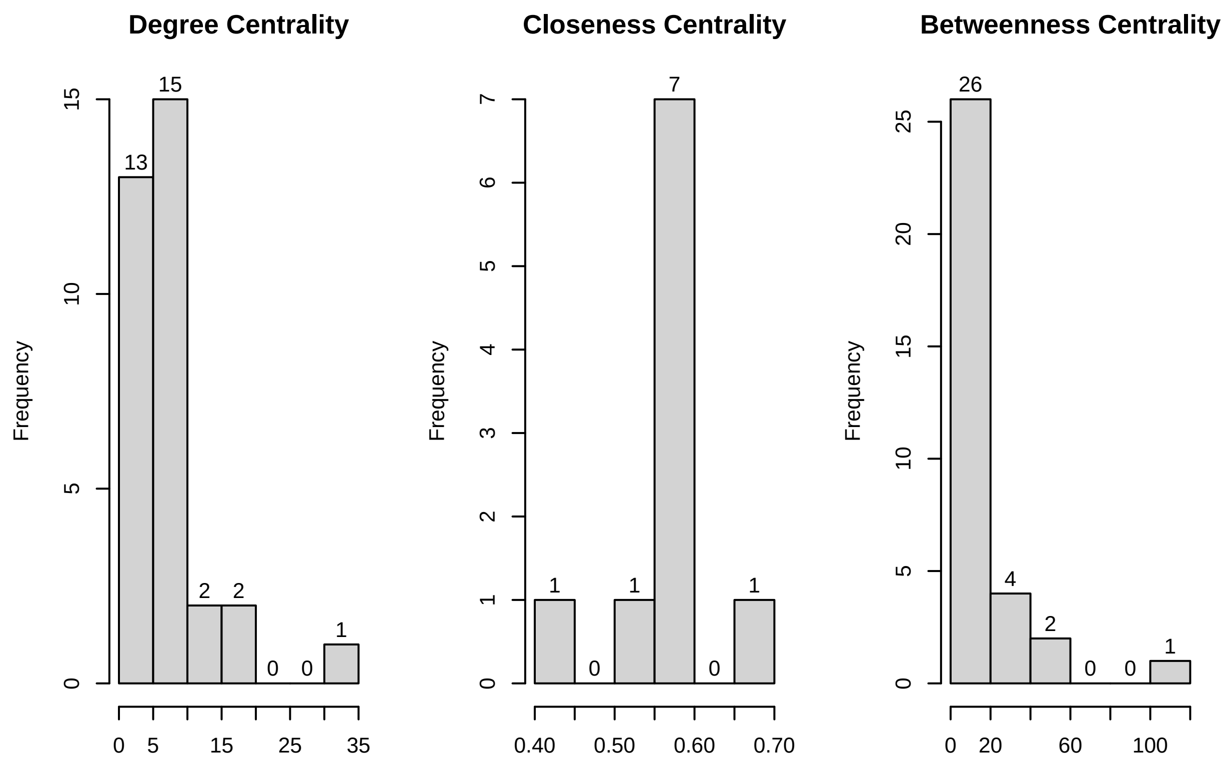
```
op <- par(mar=c(0,1,0,0))
gplot(economic_exchange_network, gmode = "digraph", displaylabels = TRUE, label.pos = 2,
      boxed.labels = T, label.border = 0, label.cex = .5, label.pad = 0,
      arrowhead.cex = .3)
```



```
par(op)


lgc <- component.largest(economic_exchange_network,result="graph")
gplot(lgc, gmode = "digraph", displaylabels = TRUE, label.pos = 2, boxed.labels = T,
      label.border = 0, label.cex = .5, label.pad = 0, arrowhead.cex = .3,
      main = "Largest Connected Subgraph")
```




```
par(op)

par(mfrow=c(1, 3))
hist(degree(economic_exchange_network, gmode="digraph"), labels = T, breaks = "Sturges",
     main = "Degree Centrality", xlab = "")
hist(closeness(lgc, gmode="digraph"), labels = T, breaks = "Sturges",
     main = "Closeness Centrality", xlab = "")
hist(betweenness(economic_exchange_network, gmode="digraph"), labels = T,
     breaks = "Sturges", main = "Betweenness Centrality", xlab = "")
```




Similarly to in our cultural similarity network, betweennness and degree centrality are skewed right. This means that most countries are not in the top 5 trading partners of other countries, and that most countries do not lie in between trading relations. Closeness centrality is roughly normally distributed, which suggests most countries are about equidistant from one another in terms of economic trading partners.

Especially to investigate which countries have high betweenness and high degree centrality, we will look at a dataframe. Closeness centrality is not included in the dataframe because of the difficulty measuring it and comparing it across connected subgraphs.

```
vertex <- economic_exchange_network %v% "vertex.names"
degree_centrality <- degree(economic_exchange_network, gmode="digraph")
betweenness_centrality <- betweenness(economic_exchange_network, gmode="digraph")

centrality_df <- data.frame(vertex, degree_centrality, betweenness_centrality)
centrality_df[nrow(centrality_df) + 1,] =
  c("Network", centralization(economic_exchange_network,degree),
    centralization(economic_exchange_network,betweenness))

print(centrality_df)
```

```
##            vertex degree_centrality betweenness_centrality
## 1         Austria                 7       4.71666666666667
## 2         Belgium                 8                    3.5
## 3        Bulgaria                 6                     14
## 4         Croatia                 5                      5
## 5  Czech Republic                 7                   9.25
## 6          Cyprus                 3      0.666666666666667
```

```
## 7          Denmark                  5                 0.45
## 8          Estonia                  5                    4
## 9          Finland                  5                 2.95
## 10          France                 20     105.866666666667
## 11         Germany                 31     52.6166666666667
## 12          Greece                  7                 14.2
## 13         Hungary                  8                   22
## 14         Iceland                  4                    0
## 15         Ireland                  4    0.916666666666667
## 16          Israel                  2                    0
## 17           Italy                 15                   38
## 18          Latvia                  5                  2.5
## 19      Luxembourg                  5                    0
## 20     Netherlands                 15     26.2166666666667
## 21          Norway                  6     2.11666666666667
## 22          Poland                  9                 15.9
## 23        Portugal                  5                    0
## 24         Romania                  7     18.2166666666667
## 25          Russia                  6     5.65833333333333
## 26        Slovakia                  7     11.2666666666667
## 27        Slovenia                  6                    1
## 28           Spain                  8                33.75
## 29          Sweden                  8                9.875
## 30     Switzerland                  6     2.28333333333333
## 31          Turkey                  5     3.16666666666667
## 32  United Kingdom                 20     51.9166666666667
## 33         Ukraine                  4                    0
## 34         Network  0.382560483870968   0.0955015120967742
```

To no one's surprise we see that Germany has the highest degree centrality within the network at 31 ties. This is consistent with Germany as Europe's largest national economy. Germany is followed by France and the UK, both at 20 ties, and Italy and the Netherlands, both at 15 ties. Other than the Netherlands, this follows the pattern of European economy by size exactly.

There is little overlap between the prominent countries of our cultural similarity network and the prominent countries in our economic trading partner network. This seems to have to do with a country's size. Small countries seem culturally similar to many other countries but don't have a lot of trading partners, while large countries have have a lot of trading partners but a lot of cultural individuality.

At the network level, trading partnership seems to be reasonably centralized. This makes sense considering the prominence of Germany, France, and the UK.

```r
detach(name = package:statnet)
library(igraph)

ieconomic_exchange_network <- asIgraph(economic_exchange_network)
```

**Use Modularity to Detect Community Structure** We will look at the modularity of the network based on GDP, and region. We will split GDP and population into 5 group categories based on their value.

```r
V(ieconomic_exchange_network)$numeric_region <-
case_when(V(ieconomic_exchange_network)$region == "N" ~ 1,
          V(ieconomic_exchange_network)$region == "E" ~ 2,
          V(ieconomic_exchange_network)$region == "S" ~ 3,
          V(ieconomic_exchange_network)$region == "W" ~ 4)

V(ieconomic_exchange_network)$GDP_level <-
case_when(V(ieconomic_exchange_network)$GDP >= 4223116  ~ 5,
          V(ieconomic_exchange_network)$GDP >= 3173702  ~ 4,
          V(ieconomic_exchange_network)$GDP >= 2124288  ~ 3,
          V(ieconomic_exchange_network)$GDP >= 1074874  ~ 2,
          V(ieconomic_exchange_network)$GDP >= 25459  ~ 1)

modularity(ieconomic_exchange_network,V(ieconomic_exchange_network)$numeric_region)
modularity(ieconomic_exchange_network,V(ieconomic_exchange_network)$GDP_level)
```

```
## [1] 0.2125803
## [1] 0.00327135
```

The issue with calculating modularity by GDP and population split into five even groups is that you end up with most countries being in the first group while only a small fraction of countries are in the other groups. This is due to some countries massive GDP and population compared to most countries.[1] This is why our modularity measure for GDP is so low. Interestingly modularity for economic ties considering region is quite similar to that of cultural ties considering region.

We will use the walktrap cluster edge betweenness algorithm to detect community.

```r
cw <- cluster_walktrap(ieconomic_exchange_network)
modularity(cw)
```

```
## [1] 0.288424
```

```r
ceb <- cluster_edge_betweenness(ieconomic_exchange_network)
modularity(ceb)
```

```
## [1] 0.05274334
```

From these inconsistent scores it appears that our community detection algorithms are rather unreliable. This means it is quite difficult to discern distinct subgroups within our network of trading partners.

## Miliataristic Alliances:

Alliance data was prepared in python. The dyadic ties and the attributes of these dyadic ties were extracted from a larger data set into two separate CSVs. One big disadvantage to the data found is that it is missing data from Austria, Ireland, Israel, Slovenia, Sweden, Switzerland, Cyprus.

```r
detach(package:igraph)
library(statnet)
```

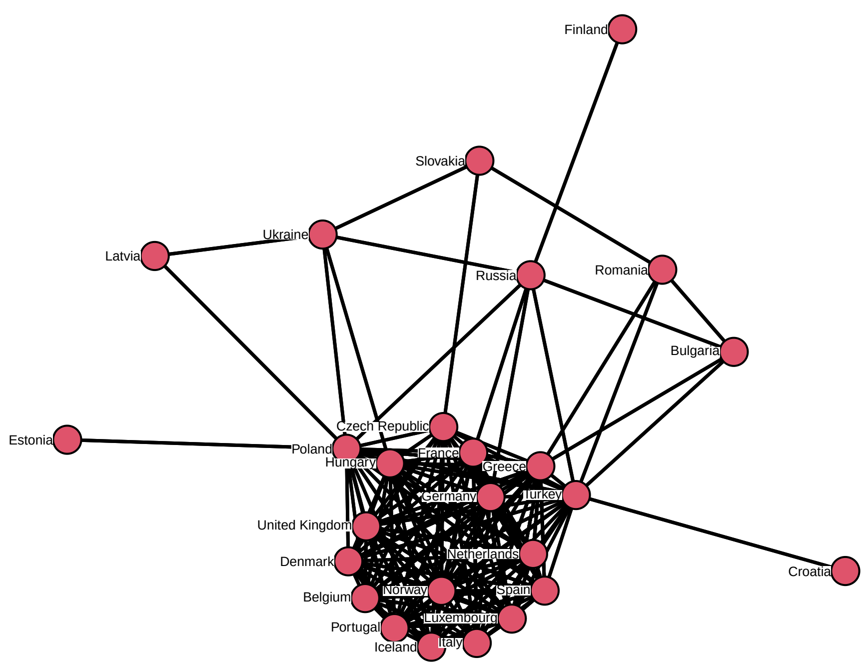
[1] I tried to run the same code but using the 4th route of GDP, however R always crashed when I did this.

We will now examine our last network, that of alliances.

```
alliance_edgelist <- read.csv("alliance_edge_list")
alliance_edge_attributes <- read.csv("alliance_edge_attributes")
alliance_node_attributes <- read.csv("node_attributes_alliance_network")

alliance_network <- as.network(alliance_edgelist, multiple = T, directed = F)

set.edge.attribute(alliance_network, "defense", alliance_edge_attributes$defense)
set.edge.attribute(alliance_network, "neutrality", alliance_edge_attributes$neutrality)
set.edge.attribute(alliance_network, "nonaggression",
                   alliance_edge_attributes$nonaggression)
set.edge.attribute(alliance_network, "entente", alliance_edge_attributes$entente)
set.edge.attribute(alliance_network, "asymmetric", alliance_edge_attributes$asymmetric)

set.vertex.attribute(alliance_network, "region", alliance_node_attributes$region)
```

Note that our multiple = TRUE in our network object. This must be true because dyads in the data set represent the general establishment of alliances between two countries, which have not been dissolved. Many alliances have been re-established over the years with slightly different attributes, while not dissolving previous alliances. Thus the centrality of nodes are biased by the amount of times they have reestablished relationships.

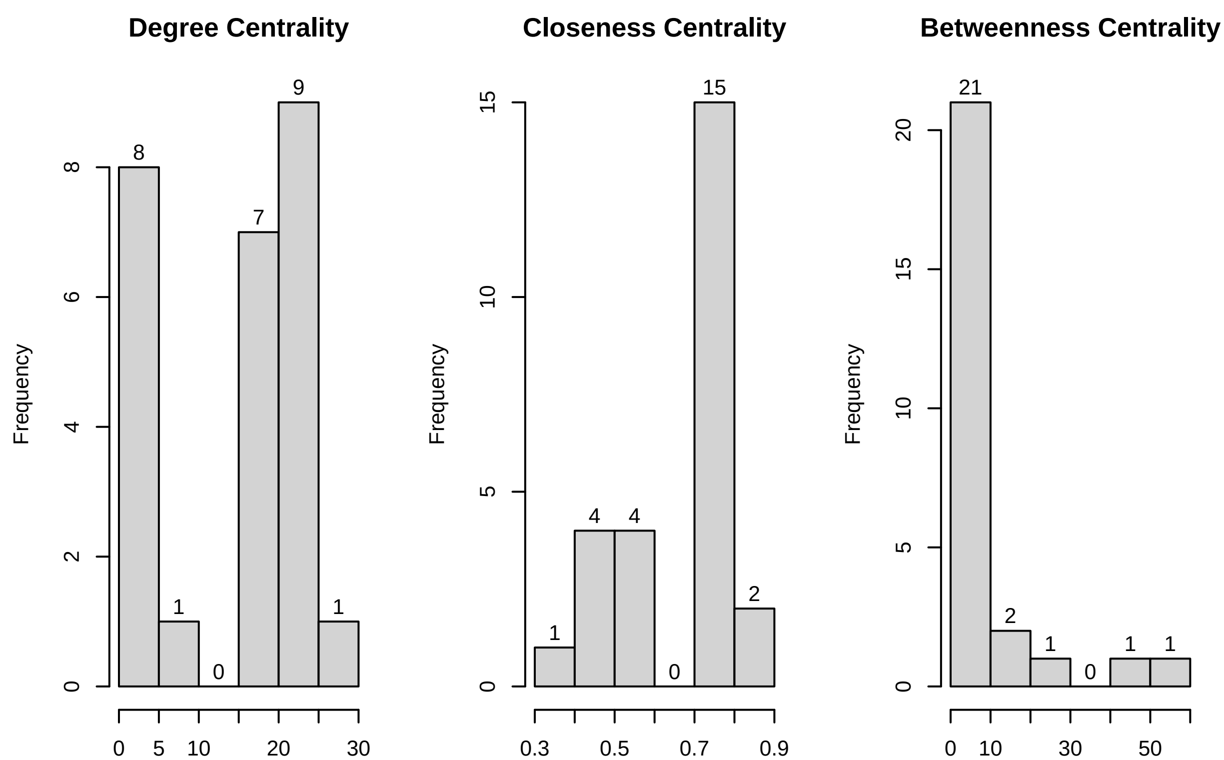
```
op <- par(mar=c(0,0,0,0))
gplot(alliance_network, gmode = "graph", displaylabels = TRUE, label.pos = 2,
      boxed.labels = T, label.border = 0, label.cex = .4, label.pad = 0)
```



```
par(op)
```

Before examining sub-graphs by tie type, we will explore the prominence of actors within the entire alliance network. We display the distribution of degree centrality, closeness centrality, and between centrality.

```
par(mfrow=c(1, 3))
hist(degree(alliance_network, gmode="graph"), labels = T,
     breaks = "Sturges", main = "Degree Centrality", xlab = "")
hist(closeness(alliance_network, gmode="graph"), labels = T,
     breaks = "Sturges", main = "Closeness Centrality", xlab = "")
hist(betweenness(alliance_network, gmode="graph"), labels = T,
     breaks = "Sturges", main = "Betweenness Centrality", xlab = "")
```




Unlike in the previous networks, there are many nodes which have a high degree centrality. This is due to the densely connected cluster of nodes which obviously represents the Nato alliance. This alliance also skews closeness centrality to the left, as many nodes in the network are very close together, and skews betweeness to rhe right, as most nodes do not lie between onther nodes but rather in the dense Nato cluster.

We examine centrality statistics at the network level and by specific country in a dataframe:

```
vertex <- alliance_network %v% "vertex.names"
degree_centrality <- degree(alliance_network, gmode="graph")
closeness_centrality <- closeness(alliance_network, gmode="graph")
betweenness_centrality <- betweenness(alliance_network, gmode="graph")

centrality_df <-
```

```
  data.frame(vertex, degree_centrality, closeness_centrality, betweenness_centrality)
centrality_df[nrow(centrality_df) + 1,] =
  c("Network", centralization(alliance_network,degree), centralization(
    alliance_network,closeness), centralization(alliance_network,betweenness))
print(centrality_df)
```

```
##            vertex degree_centrality closeness_centrality betweenness_centrality
## 1  United Kingdom                25    0.714285714285714                      0
## 2     Netherlands                24    0.714285714285714                      0
## 3         Belgium                24    0.714285714285714                      0
## 4      Luxembourg                24    0.714285714285714                      0
## 5          France                25    0.757575757575758       8.57142857142857
## 6           Spain                24    0.714285714285714                      0
## 7        Portugal                25    0.714285714285714                      0
## 8           Italy                24    0.714285714285714                      0
## 9         Germany                17    0.757575757575758       5.48571428571429
## 10         Poland                21    0.833333333333333       56.1547619047619
## 11        Hungary                17    0.735294117647059                      7
## 12 Czech Republic                17    0.735294117647059                     15
## 13         Greece                27    0.757575757575758                   19.4
## 14         Norway                16    0.714285714285714                      0
## 15        Denmark                17    0.714285714285714                      0
## 16        Iceland                17    0.714285714285714                      0
## 17        Romania                 4    0.510204081632653                    2.5
## 18         Russia                 7    0.581395348837209       27.8166666666667
## 19       Bulgaria                 4    0.520833333333333       1.28571428571429
## 20       Slovakia                 3                  0.5       1.91666666666667
## 21        Croatia                 1    0.462962962962963                      0
## 22         Latvia                 2    0.480769230769231                      0
## 23         Turkey                20    0.833333333333333       43.0190476190476
## 24        Ukraine                 5    0.543478260869565                   6.85
## 25        Finland                 1    0.373134328358209                      0
## 26        Estonia                 1    0.462962962962963                      0
## 27        Network 0.516666666666667    0.194608412518846      0.168669841269841
```

A lot of the actors with high degree centrality in this network are major Nato leaders: France, the UK, etc. And many of the countries within Nato have a similar number of connections, from 16 to 25. The counties within Nato also share a similar closeness centrality statistic, around 0.75. Germany has a rather low degree for country within Nato, which suggests that not all big countries have a lot of military alliances.

Greece is the most prominent actor in terms of degree, at 27 ties. Maybe this can be understood though Greece's relativity high betweenness score in cultural similarity. Maybe it's the case that due to being culturally similar many other countries, Greece is also in a good position to make many militaristic ties with other nations. This hypothesis is supported by Luxembourg's high military alliance degree centrality, at 24 ties. Luxembourg too was a central player in the cultural similarity network.

At the network level, we see that degree centrality is quite high, meaning the network is quite dense. Many types of militaristic ties are included in this network, so it may be interesting to examine some sub-graphs.

```
detach(name = package:statnet)
library(igraph)
```

```
ialliance_network <- asIgraph(alliance_network)
```

**Use Modularity to Detect Community Structure** We will examine modularity once again by our faithful categorical variable: region.

```
V(ialliance_network)$numeric_region <-
case_when(V(ialliance_network)$region == "N" ~ 1,
          V(ialliance_network)$region == "E" ~ 2,
          V(ialliance_network)$region == "S" ~ 3,
          V(ialliance_network)$region == "W" ~ 4)

modularity(ialliance_network,V(ialliance_network)$numeric_region)
```

```
## [1] 0.00524521
```

We can tell from this modularity score that geographic region has almost nothing to do with the creation of alliance ties. This reflects a modern globalized world.

We will try walktrap and louvain algorithm to detect community.

```
cw <- cluster_walktrap(ialliance_network)
modularity(cw)
```

```
## [1] 0.0578535
```

```
cl <- cluster_louvain(icultural_similarity_subgraph)
modularity(cl)
```

```
## [1] 0.4306485
```

The inconsistency in these modularity scores is curious. The louvain algorithm's finding of a .42 modularity seems significant, although when compared to the cluster walktrap algortihm it seems dubious. More investigation into these commuity detection algorithms, or analysis of subgraphs might put this into perspective.

**Alliance Sub-Graphs** In the network object, several edge attributes exist to represent different types of alliances. These alliance types include defense, neutrality, nonaggression, entente, and asymmetric. The definitions for each alliance type are as follows: 1. defense: one or both states in the dyad established a defense contract with the other 2. neutrality: one or both states in the dyad established a neutrality pact with the other 3. nonaggression: one or both states in the dyad had a non-aggression pact with the other 4. entente: if one or both states in the dyad had an understanding that consultations with the other state would take place if a crisis occurred 5. asymmetric: if any of the terms applied to only one state of the dyadic pair Full analysis of these graphs will not be included, as such has already been done for the three previous networks.

```
detach(package:igraph)
library(statnet)
```

```
##
## 'statnet' 2019.6 (2019-06-13), part of the Statnet Project
## * 'news(package="statnet")' for changes since last version
## * 'citation("statnet")' for citation information
## * 'https://statnet.org' for help, support, and other information
```

```
## unable to reach CRAN
```

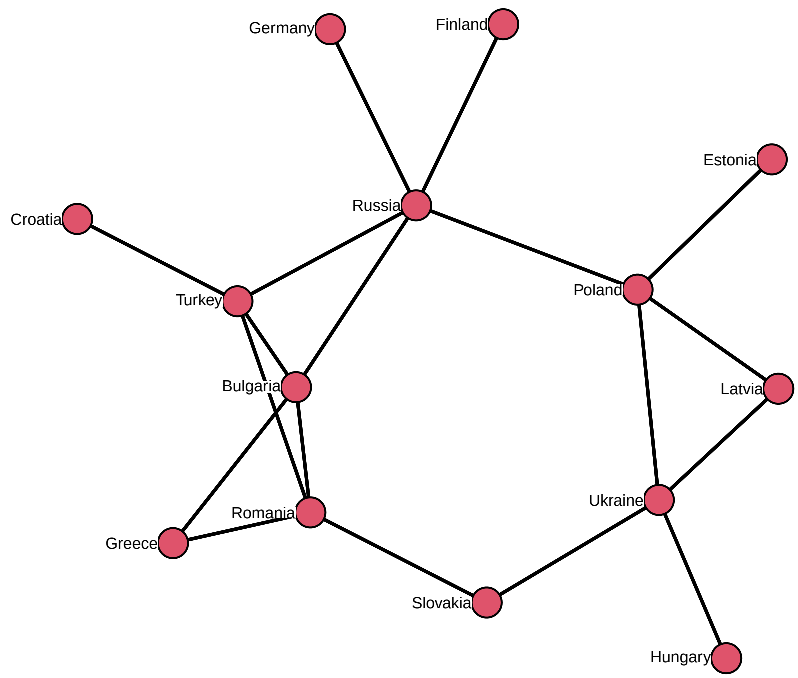
We will create a function to display based the parameter edge_type, as well as to display centrality distributions via histogram for each subgraph.

```
display_alliance_subgraph <- function(edge_type, displaylabels = T, name = "") {
  subgraph <- get.inducedSubgraph(alliance_network, eid=which(
    alliance_network%e% edge_type==1))
  gplot(subgraph, gmode = "graph", displaylabels = displaylabels, label.pos = 2,
        boxed.labels = T, label.border = 0, label.cex = .5, label.pad = 0, main = name)
}

display_graph_centrality <- function(graph, name = "" ,show_degree_centrality = T,
                   show_closeness_centrality = T, show_betweenness_centrality = T) {
  degree_centrality <- degree(graph, gmode="graph")
  closeness_centrality <- closeness(graph, gmode="graph")
  betweenness_centrality <- betweenness(graph, gmode="graph")

  hist(degree_centrality, labels = F, breaks = "Sturges",
       main = paste(name, "Degree"), xlab = "")
  hist(closeness_centrality, labels = F, breaks = "Sturges",
       main = paste(name, "Closeness"), xlab = "")
  hist(betweenness_centrality, labels = F, breaks = "Sturges",
       main = paste(name, "Betweenness"), xlab = "")
}
```

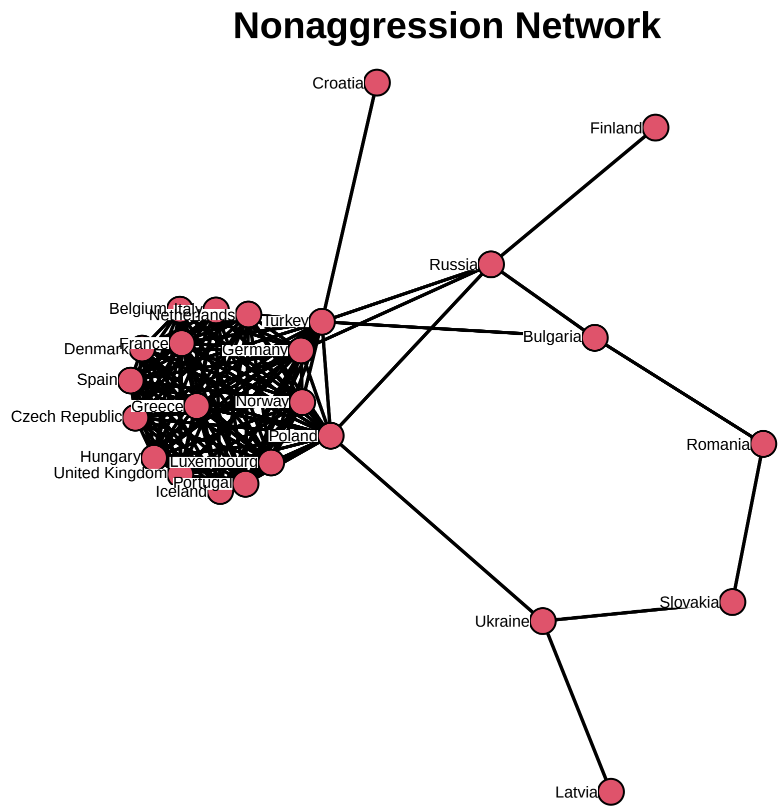
We will call this function to view some of the more interesting sub-graphs of this network.

```
op <- par(mar=c(1,1,1,1))
display_alliance_subgraph("neutrality", displaylabels = T, name = "Neutrality Network")
```

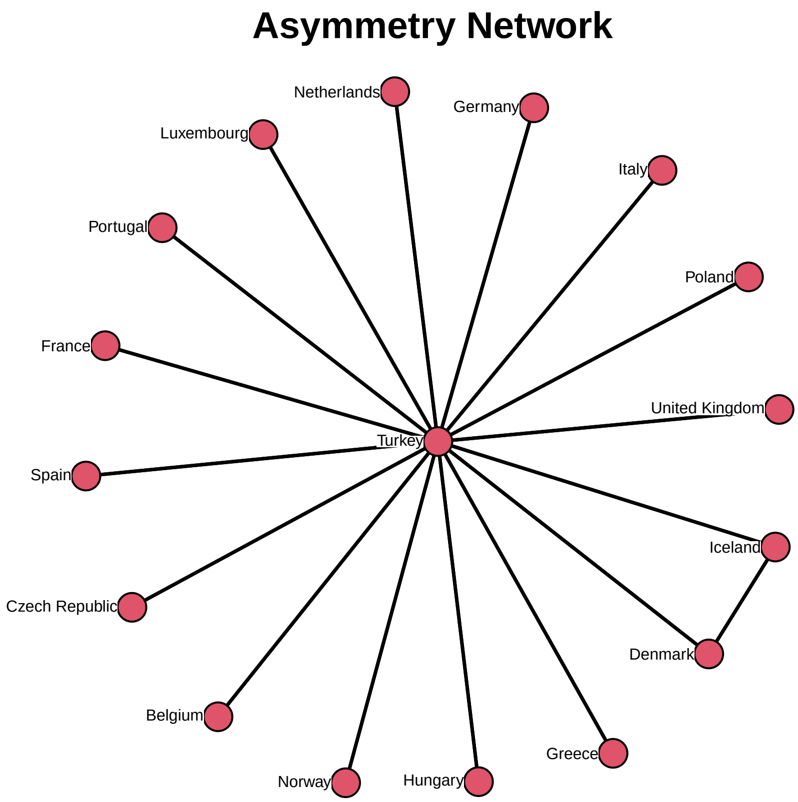
## Neutrality Network




```
display_alliance_subgraph("nonaggression", displaylabels = T, name = "Nonaggression Network")
```



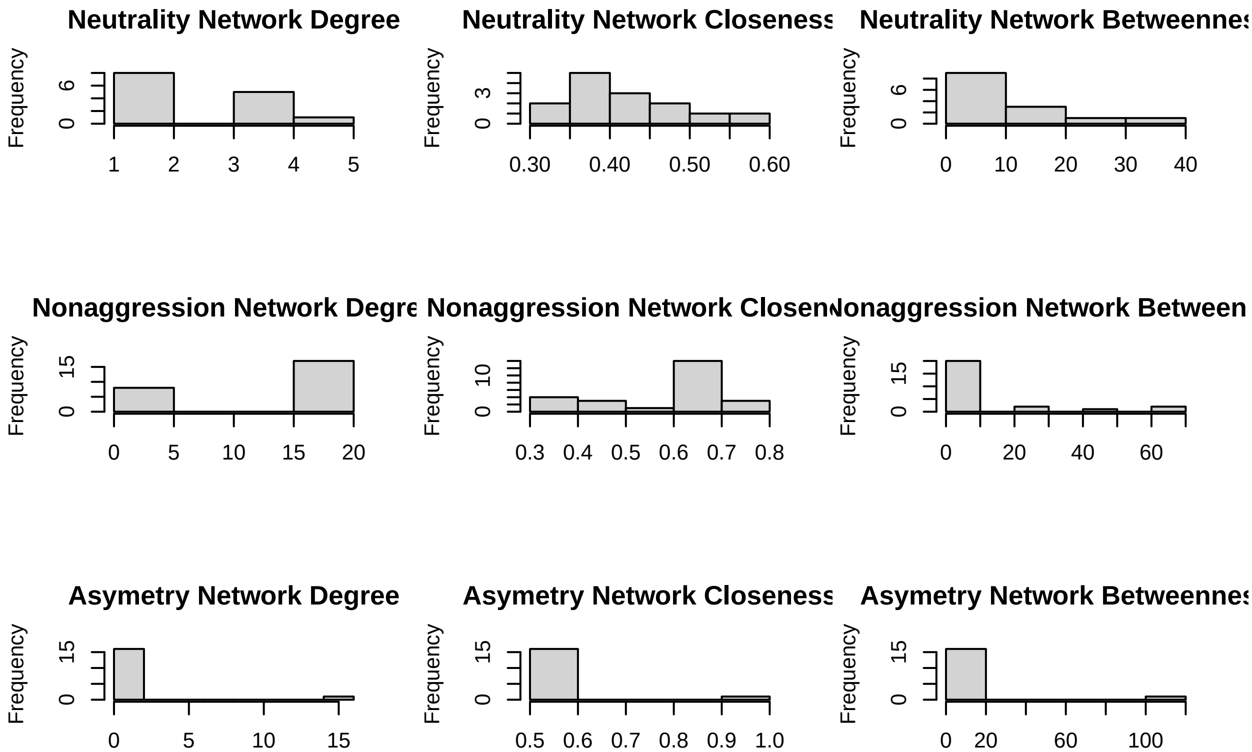

```
display_alliance_subgraph("asymmetric", displaylabels = T, name = "Asymmetry Network")
```




```
par(op)
```

In the defense network and the non-aggression network

```
par(mfrow=c(3,3))
display_graph_centrality(get.inducedSubgraph(
  alliance_network, eid=which(alliance_network%e%"neutrality"==1)),
  name = "Neutrality Network")
display_graph_centrality(get.inducedSubgraph(
  alliance_network, eid=which(alliance_network%e%"nonaggression"==1)),
  name = "Nonaggression Network")
display_graph_centrality(get.inducedSubgraph(
  alliance_network, eid=which(alliance_network%e%"asymmetric"==1)),
  name = "Asymetry Network")
```




### Data Sources

https://en.wikipedia.org/wiki/United_Nations_geoscheme_for_Europe

https://correlatesofwar.org/data-sets/formal-alliances/

http://userpage.fu-berlin.de/~jroose/index_en/main_indexvaluesaz.htm

https://wits.worldbank.org/countrysnapshot/en/RUS

https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)