# Lab 3 (QMSS5062 Social Network Analysis)

**Submitted by**: Gideon Tay
**My UNI**: gt2528

**Overview**: In this lab, I apply community detection algorithms to explore networks of countries (nodes) connected by preferential trade agreements (undirected, unvalued ties). This helps us identify trade blocs of countries that are well-connected with PTAs. We consider PTAs in force between countries as of 2023.

# Import libraries

Let's import the libraries needed for this exercise first.

```
# Load necessary packages
library(readxl)  # to read in excel
library(dplyr)
library(tidyr)
library(igraph)  # for community detection algorithms
library(ggplot2) # for visualization
library(reshape2)
library(mclust)  # for Adjusted Rand Index
library(aricode) # for Normalized Mutual Information
library(clue)  # for comparison of clusterings
library(countrycode) # convert country codes to names
```

# 1- Describe the social network(s) to me, in terms of how it was collected, what it represents and so forth. Also give me basic topography of the network: the nature of the ties; direction of ties; overall density; and if attributes are with the network, the distribution of the categories and variables of those attributes.

## 1a- Describe the social network(s) to me, in terms of how it was collected, what it represents and so forth.

**Preparing the network**

I am interested in how countries are connected by Preferential Trade Agreements (PTAs). PTAs are treaties signed between two or more countries. They often include terms such as preferential (lower) tariffs for goods, lowered export restrictions and barriers to trade, and arrangements that facilitate the movement of capital and people.

This includes free trade agreements (FTA), customs unions (CU), economic integration agreements (EIA), and partial scope agreements (PSA) between countries or groups of countries. For example, the Comprehensive and Progressive Agreement for Trans-Pacific Partnership (CPTPP) is an FTA and the European Economic Area (EEA) is an EIA.

I will thus build a network data set with countries as nodes and PTAs as undirected, unvalued ties. Download the Deep Trade Agreements database 1.0 (horizontal depth) v2.0 released on 2024 from the World Bank website (https://datatopics.worldbank.org/dta/table.html). Save the 'Bilateral Information' tab of the excel spreadsheet as a separate file.

Let's load it in:

```
# Read the Excel file
df <- read_excel("dta_bilateral_data.xlsx")

# Display the first few rows
head(df, 3)
```

```
## # A tibble: 3 × 9
##   iso1  iso2  rta_name                  WBID  year Economy1 Economy2 type  entry
##   <chr> <chr> <chr>                    <dbl> <dbl> <chr>    <chr>    <chr> <dbl>
## 1 ABW   AUT   EU – Overseas Countries…     4  1995 Aruba    Austria  FTA    1995
## 2 ABW   AUT   EU – Overseas Countries…     4  1996 Aruba    Austria  FTA    1996
## 3 ABW   AUT   EU – Overseas Countries…     4  1997 Aruba    Austria  FTA    1997
```

Each row represents a trade agreement between Economy1 and Economy2. The name of the agreement is in the `rta_name` column, the ID number of the agreement is in `WBID`, and the type of agreement is under `type`. Years in which these agreements are in force (active), are indicated under `year`. `iso1` and `iso2` are the country's ISO codes.

The current data file is not formatted in a manner fit for analysis. We need to reformat the data to create a matrix of countries, with cells taking values based on the number of PTAs between each country pair.

The latest year for which data is available in this data set is 2023. We will thus focus on the network (PTAs in force between countries) as of 2023.

```
# Filter for 2023
df_2023 <- df %>% filter(year == 2023)

# Create an undirected version of iso pairs: alphabetically sorted
df_undirected <- df_2023 %>%
  mutate(
    iso_min = pmin(iso1, iso2),
    iso_max = pmax(iso1, iso2)
  ) %>%
  # Counts how many times each pair (order does not matter) appears
  count(iso_min, iso_max, name = "agreement_count")

# Get unique ISO codes from both iso1 and iso2
iso_all <- union(df_2023$iso1, df_2023$iso2)

# Create a square matrix initialized with 0
trade_matrix <- matrix(0, nrow = length(iso_all),
                       ncol = length(iso_all),
                       dimnames = list(iso_all, iso_all))

# Fill in symmetric values for each pair
for (i in seq_len(nrow(df_undirected))) {
  iso_a <- df_undirected$iso_min[i]
  iso_b <- df_undirected$iso_max[i]
  count <- df_undirected$agreement_count[i]
  trade_matrix[iso_a, iso_b] <- count
  trade_matrix[iso_b, iso_a] <- count
}
```

We simplify this matrix by ignoring the value of the ties, as captured by the number of PTAs. As long as at least one PTA is active, we assign the cell for the country pair a value of 1. Otherwise, it has a value of 0.

```
# Convert all non-zero values to 1 (indicating an agreement exists)
trade_matrix_unvalued <- trade_matrix
trade_matrix_unvalued[trade_matrix_unvalued > 0] <- 1

# Show the first few rows of the new matrix
head(trade_matrix_unvalued, 3)
```

```
##     ABW AFG AGO AIA ALB AND ARE ARG ARM ATG AUS AUT AZE BDI BEL BEN BFA BGD BGR
## ABW   0   0   0   0   0   0   0   0   0   0   0   1   0   0   1   0   0   0   1
## AFG   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
## AGO   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   1   0   0
##     BHR BHS BIH BLR BLZ BMU BOL BRA BRB BRN BTN BWA CAF CAN CHE CHL CHN CIV CMR
## ABW   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## AFG   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
## AGO   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0   0   1   1
##     COD COG COK COL COM CPV CRI CUB CYM CYP CZE DEU DJI DMA DNK DOM DZA ECU EGY
## ABW   0   0   0   0   0   0   0   0   0   1   1   1   0   0   1   0   0   0   0
## AFG   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## AGO   1   1   0   0   1   1   0   0   0   0   0   0   1   0   0   0   1   0   1
##     ERI ESP EST ETH FIN FJI FRA FRO FSM GAB GBR GEO GHA GIN GMB GNB GNQ GRC GRD
## ABW   0   1   1   0   1   0   1   0   0   0   0   0   0   0   0   0   0   1   0
## AFG   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## AGO   0   0   0   1   0   0   0   0   0   1   0   0   1   1   0   1   1   0   0
##     GRL GTM GUY HKG HND HRV HTI HUN IDN IND IRL IRN IRQ ISL ISR ITA JAM JOR JPN
## ABW   0   0   0   0   0   1   0   1   0   0   1   0   0   0   0   1   0   0   0
## AFG   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
## AGO   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##     KAZ KEN KGZ KHM KIR KNA KOR KWT LAO LBN LBR LBY LCA LIE LKA LSO LTU LUX LVA
## ABW   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   1
## AFG   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
## AGO   0   1   0   0   0   0   0   0   0   0   1   1   0   0   0   1   0   0   0
##     MAC MAR MDA MDG MDV MEX MHL MKD MLI MLT MMR MNE MNG MOZ MRT MSR MUS MWI MYS
## ABW   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
## AFG   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## AGO   0   1   0   1   0   0   0   0   1   0   0   0   0   1   1   0   1   1   0
##     NAM NCL NER NGA NIC NIU NLD NOR NPL NRU NZL OMN PAK PAN PCN PER PHL PNG POL
## ABW   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   1
## AFG   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0
## AGO   1   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##     PRK PRT PRY PSE PYF QAT ROU RUS RWA SAU SDN SEN SGP SLB SLE SLV SMR SOM SPM
## ABW   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
## AFG   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## AGO   0   0   0   0   0   0   0   0   1   0   1   1   0   0   1   0   0   1   0
##     SRB SSD STP SUR SVK SVN SWE SWZ SYC SYR TCA TCD TGO THA TJK TKM TON TTO TUN
## ABW   0   0   0   0   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0
## AFG   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## AGO   0   1   1   0   0   0   0   1   1   0   0   1   1   0   0   0   0   0   1
##     TUR TUV TWN TZA UGA UKR URY USA UZB VCT VEN VNM VUT WLF WSM XKX YEM YUG ZAF
## ABW   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## AFG   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## AGO   0   0   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   1
##     ZMB ZWE
## ABW   0   0
## AFG   0   0
## AGO   1   1
```

As shown below, this matrix has 211 countries and territories, and is a 211 x 211 square matrix:

```
dim(trade_matrix_unvalued)
```

```
## [1] 211 211
```

# 1b- Also give me basic topography of the network: the nature of the ties; direction of ties; overall density; and if attributes are with the network, the

distribution of the categories and variables of those attributes.

Our network consists of 211 countries and territories and 5832 undirected and unvalued ties. It has a network density of 0.263, meaning only 26.3% of all possible preferential trade agreements between country pairs exist.

This suggests that preferential trade agreements between countries are selective but still substantial. It is substantial in that 26.3% is still a non-trival percentage, which could be attributed to the wave of globalization driving countries to trade between each other and enter agreements to facilitate such trade. It is selective in that countries do not form such agreements with all other countries. Intuitively, countries likely find value in negotiating such agreements with major trade partners, and not with very minor trade partners. They may also avoid forming such agreements with geopolitically 'adversarial' countries, while preferring to form agreements with allies.

```
# Create a graph from the adjacency matrix
graph <- graph_from_adjacency_matrix(trade_matrix_unvalued, mode = "undirected", diag = FALSE)

# Number of nodes (countries)
num_nodes <- vcount(graph)

# Number of edges (unique country pairs with agreements)
num_edges <- ecount(graph)

# Network density
density <- edge_density(graph)

# Print the topography of the network
cat("Number of Nodes:", num_nodes, "\n")
```

```
## Number of Nodes: 211
```

```
cat("Number of Edges:", num_edges, "\n")
```

```
## Number of Edges: 5832
```

```
cat("Network Density:", round(density, 3), "\n")
```

```
## Network Density: 0.263
```

# 2- Run the Girvan-Newman community detection algorithm. Then run the random walk community detection algorithm.

For both algorithms, our code only runs it if 'gn_communities.rds' and 'walk_communities.rds' files are not available in the directory. Then, it will run the algorithm and save the results in two separate files with these names. In future runs of this script, the script will simply load in data from the RDS files, so the algorithm need not be run again.

## 2a- Girvan-Newman community detection algorithm

The Girvan–Newman algorithm starts by calculating the betweenness of each edge (connection between nodes) in the network, which is the number of shortest paths between node pairs that run through it. Then it identifies and removes that edge with the highest betweenness. Betweenness is then recalculated again and the process repeats until no edges remain. Through this method, the algorithm detects communities in the network.

We run the algorithm below using the `cluster_edge_betweenness()` function from the `igraph` package and give some examples of identified communities.

```r
# Run the Girvan-Newman community detection algorithm
# if gn_communities.rds not available
if (!file.exists("gn_communities.rds")) {
  gn <- cluster_edge_betweenness(graph)
  saveRDS(gn, "gn_communities.rds")
} else {
  gn <- readRDS("gn_communities.rds")
}

# View the first few groups of the community detection result
head(gn)
```

```
## $`1`
##  [1] "ABW" "AIA" "ALB" "AND" "ARM" "ATG" "AUT" "BEL" "BGR" "BHS" "BIH" "BLZ"
## [13] "BMU" "BRB" "CAN" "CHE" "CHL" "COL" "CRI" "CYM" "CYP" "CZE" "DEU" "DMA"
## [25] "DNK" "DOM" "ECU" "ESP" "EST" "FIN" "FJI" "FRA" "FRO" "GBR" "GEO" "GRC"
## [37] "GRD" "GRL" "GTM" "GUY" "HND" "HRV" "HTI" "HUN" "IRL" "ISL" "ISR" "ITA"
## [49] "JAM" "JOR" "JPN" "KNA" "KOR" "LBN" "LCA" "LIE" "LTU" "LUX" "LVA" "MDA"
## [61] "MEX" "MKD" "MLT" "MNE" "MSR" "NCL" "NIC" "NLD" "NOR" "PAN" "PCN" "PER"
## [73] "PNG" "POL" "PRT" "PSE" "PYF" "ROU" "SGP" "SLB" "SLV" "SMR" "SPM" "SRB"
## [85] "SUR" "SVK" "SVN" "SWE" "SYR" "TCA" "TTO" "TUR" "UKR" "VCT" "VNM" "WLF"
## [97] "WSM"
##
## $`2`
## [1] "AFG" "BTN" "MDV" "NPL"
##
## $`3`
##  [1] "AGO" "ARG" "BDI" "BEN" "BFA" "BGD" "BOL" "BRA" "BWA" "CAF" "CIV" "CMR"
## [13] "COD" "COG" "COM" "CPV" "CUB" "DJI" "DZA" "EGY" "ETH" "GAB" "GHA" "GIN"
## [25] "GNB" "GNQ" "IDN" "IND" "IRN" "IRQ" "KEN" "LBR" "LBY" "LKA" "LSO" "MAR"
## [37] "MDG" "MLI" "MMR" "MOZ" "MRT" "MUS" "MWI" "MYS" "NAM" "NER" "NGA" "PAK"
## [49] "PHL" "PRK" "RWA" "SDN" "SEN" "SLE" "SOM" "SSD" "STP" "SWZ" "SYC" "TCD"
## [61] "TGO" "THA" "TUN" "TZA" "UGA" "VEN" "ZAF" "ZMB" "ZWE"
##
## $`4`
## [1] "ARE"
##
## $`5`
##  [1] "AUS" "COK" "FSM" "KIR" "MHL" "NIU" "NRU" "NZL" "TON" "TUV" "VUT"
##
## $`6`
## [1] "AZE" "RUS" "TKM" "UZB"
```

# 2b- Random walk community detection algorithm

Next, we consider another community detection algorithm based on random walks. A random walk on a graph starts from a node and moves to a neighboring node at random. This process continues for a specified number of steps. The longer the random walk remains within a set of nodes, the more likely those nodes are part of a community. The algorithm groups nodes into communities that maximize modularity. These are groups of nodes that the random walk stays within for longer periods.

We run the algorithm below using the `cluster_walktrap()` function from the `igraph` package and give some examples of identified communities.

```r
# Apply Random Walk (Walktrap) community detection
# if walk_communities.rds not available
if (!file.exists("walk_communities.rds")) {
  walk <- cluster_walktrap(graph)
  saveRDS(walk, "walk_communities.rds")
} else {
  walk <- readRDS("walk_communities.rds")
}

# View the first few groups of the community detection result
head(walk)
```

```
## $`1`
##  [1] "ARG" "BEN" "BGD" "BOL" "BRA" "BRN" "CHN" "CUB" "HKG" "IDN" "IND" "IRN"
## [13] "IRQ" "KHM" "LAO" "LKA" "MAC" "MMR" "MYS" "PAK" "PHL" "PRK" "PRY" "THA"
## [25] "URY" "VEN" "YUG"
##
## $`2`
## [1] "AZE" "BLR" "KAZ" "KGZ" "RUS" "TJK" "TKM" "UZB"
##
## $`3`
##   [1] "ABW" "AIA" "ALB" "AND" "ARE" "ARM" "ATG" "AUT" "BEL" "BGR" "BHR" "BHS"
##  [13] "BIH" "BLZ" "BMU" "BRB" "CAN" "CHE" "CHL" "COL" "CRI" "CYM" "CYP" "CZE"
##  [25] "DEU" "DMA" "DNK" "DOM" "ECU" "ESP" "EST" "FIN" "FRA" "FRO" "GBR" "GEO"
##  [37] "GRC" "GRD" "GRL" "GTM" "GUY" "HND" "HRV" "HTI" "HUN" "IRL" "ISL" "ISR"
##  [49] "ITA" "JAM" "JOR" "JPN" "KNA" "KOR" "KWT" "LBN" "LCA" "LIE" "LTU" "LUX"
##  [61] "LVA" "MDA" "MEX" "MKD" "MLT" "MNE" "MSR" "NCL" "NIC" "NLD" "NOR" "OMN"
##  [73] "PAN" "PCN" "PER" "POL" "PRT" "PSE" "PYF" "QAT" "ROU" "SAU" "SGP" "SLV"
##  [85] "SMR" "SPM" "SRB" "SUR" "SVK" "SVN" "SWE" "SYR" "TCA" "TTO" "TUR" "TWN"
##  [97] "UKR" "USA" "VCT" "VNM" "WLF" "XKX" "YEM"
##
## $`4`
##  [1] "AGO" "BDI" "BFA" "BWA" "CAF" "CIV" "CMR" "COD" "COG" "COM" "CPV" "DJI"
## [13] "DZA" "EGY" "ERI" "ETH" "GAB" "GHA" "GIN" "GMB" "GNB" "GNQ" "KEN" "LBR"
## [25] "LBY" "LSO" "MAR" "MDG" "MLI" "MOZ" "MRT" "MUS" "MWI" "NAM" "NER" "NGA"
## [37] "RWA" "SDN" "SEN" "SLE" "SOM" "SSD" "STP" "SWZ" "SYC" "TCD" "TGO" "TUN"
## [49] "TZA" "UGA" "ZAF" "ZMB" "ZWE"
##
## $`5`
## [1] "AUS" "FJI" "NZL" "PNG" "SLB" "WSM"
##
## $`6`
## [1] "COK" "FSM" "KIR" "MHL" "NIU" "NRU" "TON" "TUV" "VUT"
```

# 3- Tell me how many groups each algorithm finds. Analyze how similar the two partitioning algorithms are in terms of putting nodes into groups with each other.

## 3a- Number of groups found (Girvan-Newman)

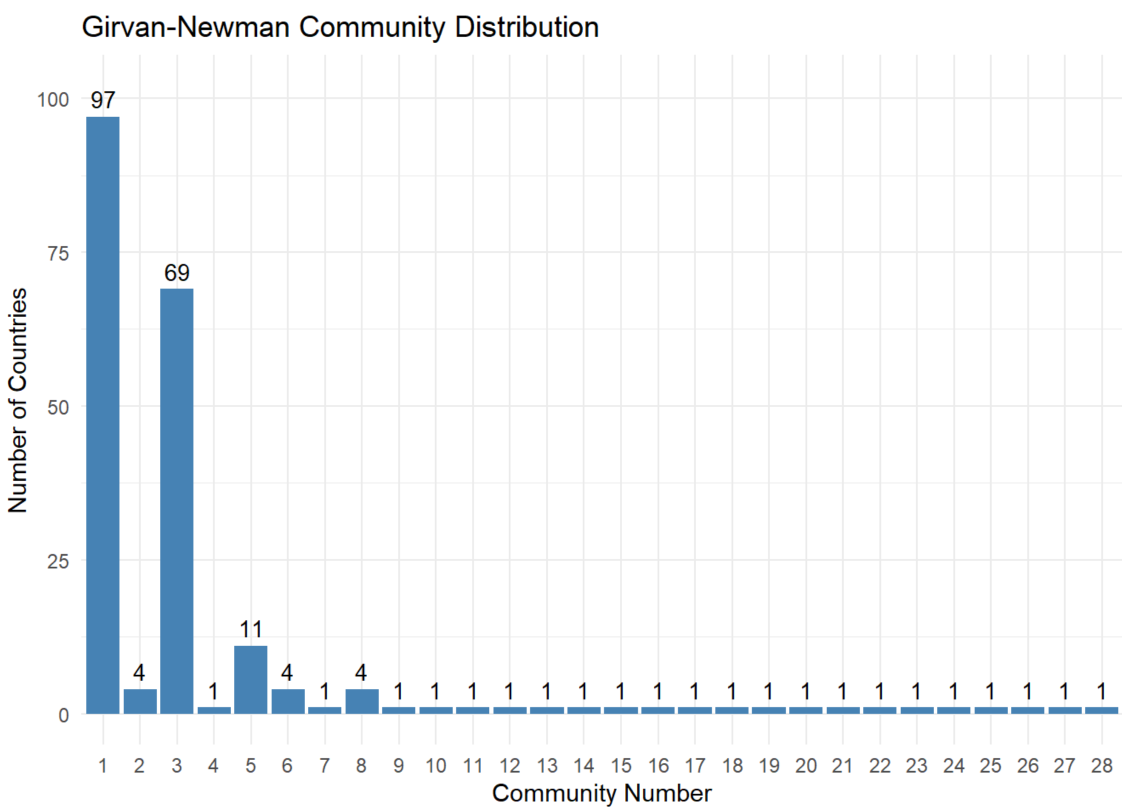
Let us get an overview of the communities detected. There are 28 communities detected. The three largest communities, in order, are the communities labelled 1, 3, and 5. They have 97, 69, and 11 countries respectively.

Besides these three communities, there are three other small communities of 4 countries. The rest of the “communities” are individual countries, which suggest that they are isolated from other countries in terms of their lack of PTAs.

```r
# Create membership dataframe
memb_df <- data.frame(
  community = gn$membership
)

# Count countries in each community
community_sizes <- memb_df %>%
  count(community, name = "size")

# Bar chart with labels
ggplot(community_sizes, aes(x = factor(community), y = size)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = size), vjust = -0.5, size = 3.5) +
  labs(
    title = "Girvan-Newman Community Distribution",
    x = "Community Number",
    y = "Number of Countries"
  ) +
  ylim(0, max(community_sizes$size) + 5) +
  theme_minimal()
```



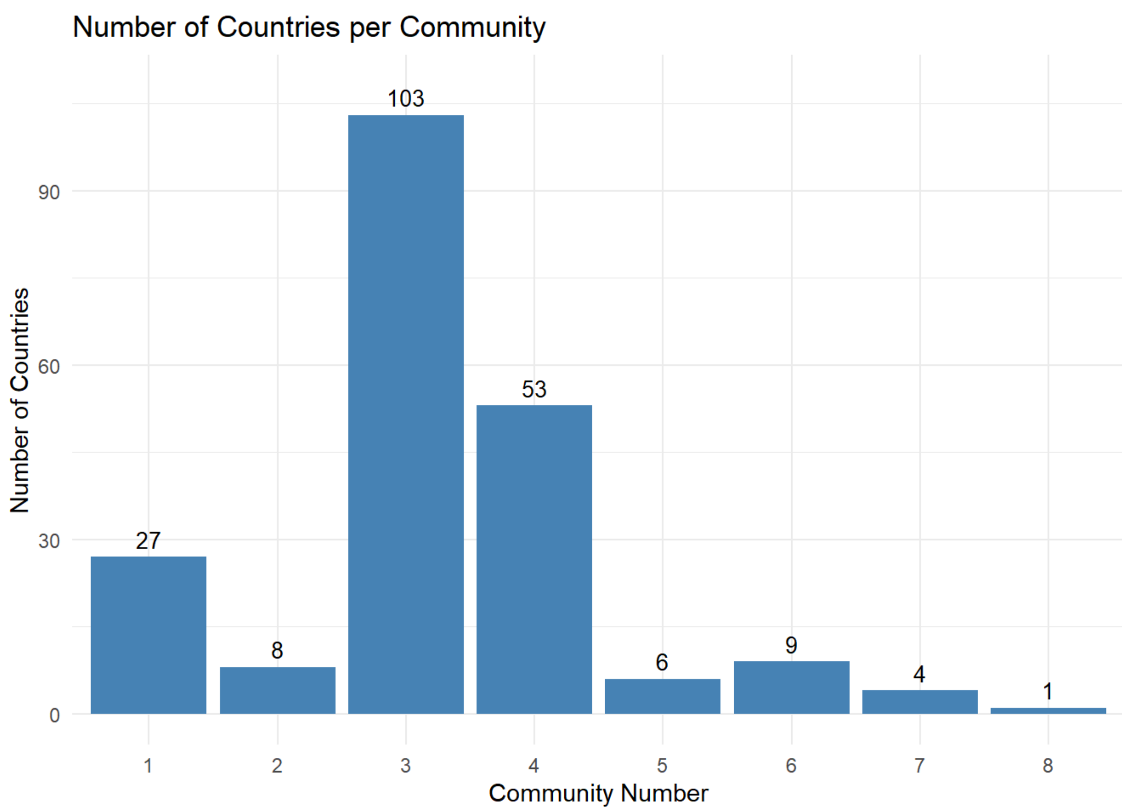

## 3b- Number of groups found (Random Walk)

Let us get an overview of the communities detected. There are 8 communities detected. The three largest communities, in order, are the communities labelled 3, 4, and 1. They have 103, 53, and 27 countries respectively. There is one “community” with just one country, suggesting that the country is isolated from other countries in terms of its lack of PTAs.

```
# Create membership dataframe
rw_memb_df <- data.frame(
  community = walk$membership
)

# Count countries in each community
community_sizes <- rw_memb_df %>%
  count(community, name = "size")

# Bar chart with labels
ggplot(community_sizes, aes(x = factor(community), y = size)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = size), vjust = -0.5, size = 3.5) +
  labs(
    title = "Number of Countries per Community",
    x = "Community Number",
    y = "Number of Countries"
  ) +
  ylim(0, max(community_sizes$size) + 5) +
  theme_minimal()
```




#### Interpretation

- The random walk walktrap algorithm produces much fewer groupings compared to the Girvan-Newman algorithm, suggesting that it produces broader, and possibly less dense communities whereas the Girvan-Newman algorithm defaults to finer subdivisions.
- In both algorithms, there are three communities which are significantly larger than all the other identified communities. It is possible that these communities identified between the two algorithms are similar, though we would have to check this.

## 3c- Similarity of groups produced (metrics)

Let’s first organize the results of the Girvan-Newman and Random Walk community detection algorithms into a single dataframe that associates each node (country) with its community membership from both algorithms. Each row corresponds to a country, and there is one column showing its community membership in the Girvan-Newman algorithm, and another column showing its community membership in the Random Walk algorithm.

```
# Extract vertices/ node names
countries <- data.frame(country = V(graph)$name)

# Rename columns
colnames(memb_df) <- "gn_community"
colnames(rw_memb_df) <- "rw_community"

# Merge the data frames
memb_combined <- cbind(countries, memb_df, rw_memb_df)

# Display results
head(memb_combined)
```

```
##   country gn_community rw_community
## 1     ABW            1            3
## 2     AFG            2            7
## 3     AGO            3            4
## 4     AIA            1            3
## 5     ALB            1            3
## 6     AND            1            3
```

Now, we use several metrics to evaluate the similarity between communities identified by the two different algorithms:

- Rand Index (RI): ranges from 0 to 1, where 1 indicates perfect agreement in the identified clusters, and 0 indicates no agreement. It includes agreement by chance.
- Adjusted Rand Index (ARI): takes values between -1 and 1. A value of 0 indicates chance agreement of the identified communities, while 1 indicates perfect agreement. Negative values suggest the clusterings are less similar than random chance. It improves on the raw Rand Index by correcting for chance.
- Normalized Mutual Information (NMI): measures how much information is shared between two approaches toward clustering countries into communities. It ranges from 0 to 1, with 0 being totally independent groupings (no overlap) and 1 being identical groupings (perfect agreement).
- Variation of Information (VI): measures how different two clusterings are, where 0 means the two partitions are identical. Higher VI means the partitions are more different, and it can theoretically grow as high as log(n) where n is the number of items being clustered.

```
# Convert to class 'cl_partition' (needed by clue metrics)
gn_cl <- as.cl_partition(memb_combined$gn_community)
walk_cl <- as.cl_partition(memb_combined$rw_community)

# Calculate raw Rand Index (RI)
ri <- cl_agreement(gn_cl, walk_cl, method = "rand")

# Calculate the Adjusted Rand Index (ARI)
ari <- adjustedRandIndex(memb_combined$gn_community,
                  memb_combined$rw_community)

# Calculate the Normalized Mutual Information (NMI)
nmi <- NMI(memb_combined$gn_community, memb_combined$rw_community)

# Calculate variation of Information (VI)
vi <- cl_dissimilarity(gn_cl, walk_cl, method = "vi")

# Display results
cat("Raw Rand Index (RI):", round(ri, 2), "\n")
```

```
## Raw Rand Index (RI): 0.88
```

```
cat("Adjusted Rand Index (ARI):", round(ari, 2), "\n")
```

```
## Adjusted Rand Index (ARI): 0.73
```

```
cat("Normalized Mutual Information (NMI):", round(nmi, 2), "\n")
```

```
## Normalized Mutual Information (NMI): 0.68
```

```
cat("Variation of Information (VI):", round(vi, 2))
```

```
## Variation of Information (VI): 0.82
```

**Interpretation**:

- The RI is 0.88, suggesting that 88% of all node pairs are either placed in the same group by both methods or placed in different groups by both, but this includes agreement by chance.
- The ARI is 0.73, taking on a lower value than the RI after correcting for chance agreements between the two community sets the Girvan-Newman and Walktrap algorithms detect. The relatively high value suggests that the communities detected by the two algorithms have considerable similarity, though some differences exist. This level of agreement is considered high, especially given how differently these algorithms work (one based on edge betweenness, the other on random walks).
- The NMI is 0.68. Hence, the two algorithms share about 68% of their grouping information. That's a fairly high overlap, although not as strong as the ARI. This suggests the methods agree on many communities but may differ on borderline countries and finer subdivisions etc.
- The VI is 0.82, out of a possible maximum of around log(211) ≈ 5.35 for 211 countries. This is relatively small compared to the maximum possible VI. It suggests that Girvan-Newman and Walktrap produce fairly similar, but not identical, community assignments.
- All metrics tell a consistent story: the community structures are broadly similar, with some interesting differences.

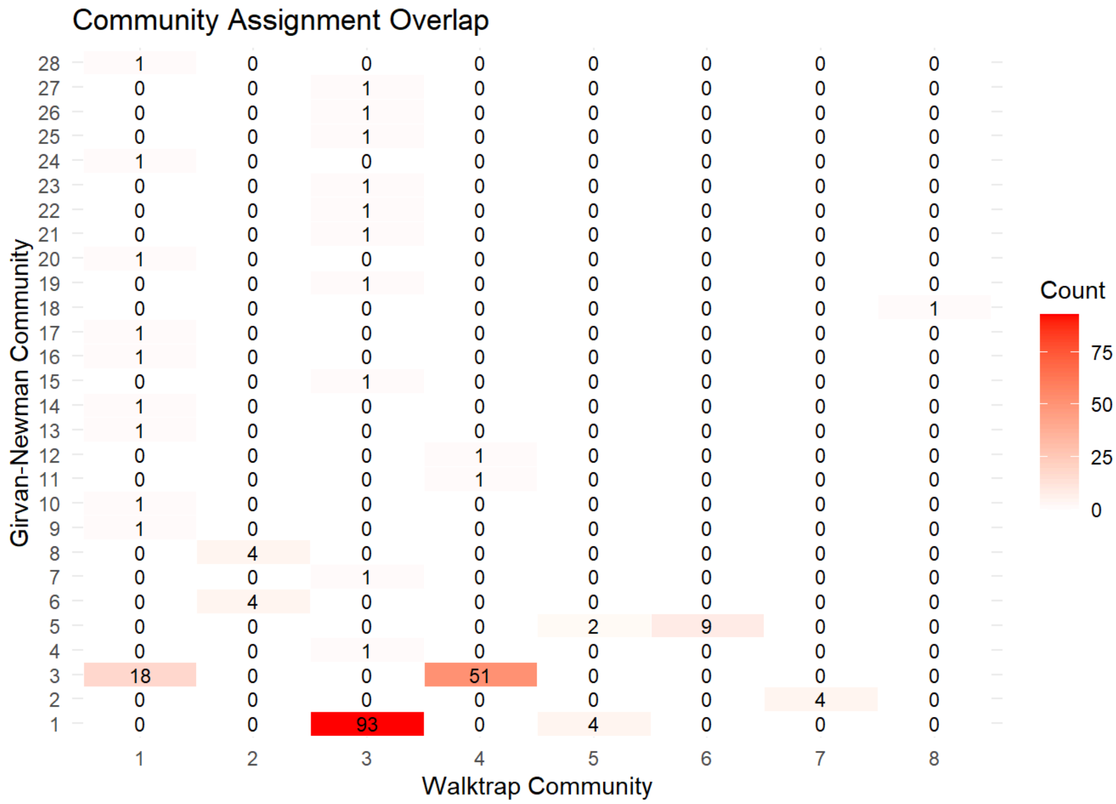
# 3d- Similarity of groups produced (matrix visualization)

Besides this metrics, let us visualize the overlap between the communities produced by the two sets of algorithms.

```
# Create the contingency table
comm_table <- table(GirvanNewman = memb_combined$gn_community,
                    Walktrap = memb_combined$rw_community)

# Convert to long format for ggplot
comm_df <- as.data.frame(comm_table)
colnames(comm_df) <- c("GirvanNewman", "Walktrap", "Count")

# Plot the heatmap
ggplot(comm_df, aes(x = factor(Walktrap), y = factor(GirvanNewman), fill = Count)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "white", high = "red") +
  labs(
    title = "Community Assignment Overlap",
    x = "Walktrap Community",
    y = "Girvan-Newman Community"
  ) +
  geom_text(aes(label = Count), color = "black", size = 3) +
  theme_minimal()
```




**Interpretation**

- Walktrap's (WT) community 3 is largely similar to Girvan-Newman's (GN) community 1, as they have 93 countries in common. However, there are some differences. There are 10 other countries in WT community 3 and 4 other countries in GN community 1 and 10 that do not overlap with each other.
- GN's community 3 is made entirely out of countries from WT community 1 and 4. Note however, that WT 1 and 4 have 9 and 2 other countries respectively that are not found in GN community 3.
- WT's community 6 (9 total countries) and GN's community 5 (11 total countries) are largely similar, as they share 9 countries. In other words, WT's community 6 is a subset of GN's community 5.
- WT's community 7 is identical to GN's community 2. Both contain the same 4 countries.
- WT's community 2 is made up of GN's community 6 and 8 combined together.
- These findings show that the GN and WT algorithms produce very similar groupings, although there are some notable differences.

# 4- Visualize the network (either in R or Gephi), coloring the nodes by either Girvan-Newman grouping or the random walk grouping.

## 4a- Girvan-Newman Community Visualization

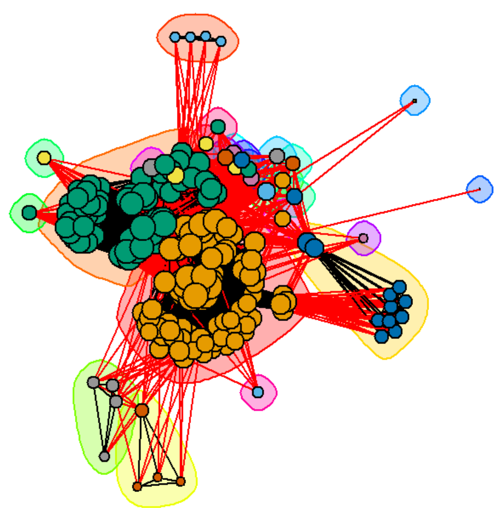
Now, visualize the network with Girvan-Newman community detection. We use the Fruchterman-Reingold layout for better visualization of the network graph. Nodes of the same color belong the the same detected community. Nodes are sized proportionally to the natural logarithm of their degrees, given the large range of degree values.

```
# Compute node degree
deg <- degree(graph)

# Fruchterman-Reingold layout
layout <- layout_with_fr(graph)

# Plot the graph
plot(gn, graph, layout = layout,
     vertex.size = log(deg) * 2.5,
     vertex.label = NA,
     edge.arrow.size = 0.3,
     main = "Girvan-Newman Community Detection")
```

**Girvan-Newman Community Detection**



**Interpretation**

- The Girvan-Newman partition breaks the network into two large groups, shown by the central orange and green clusters, and many other small groups. The third largest group is substantially smaller, and is shown as a dark blue cluster on the outer part of the network.
- The algorithm thus splits the network in a very fine-grained way, given the large number of groups the network is split into. The Girvan-Newman algorithm tends to produce many small groups, particularly when the network has loose connections between parts, which may be the case for our network given a network density of 26.3%.
- Nodes in the two large groups are highly connected relative to other groups, as they have larger node sizes indicating higher degrees. In other words, countries in these two main groups are connected to many other countries by PTAs. This is expected as we expect nodes in larger clusters to be connected to more nodes on average.

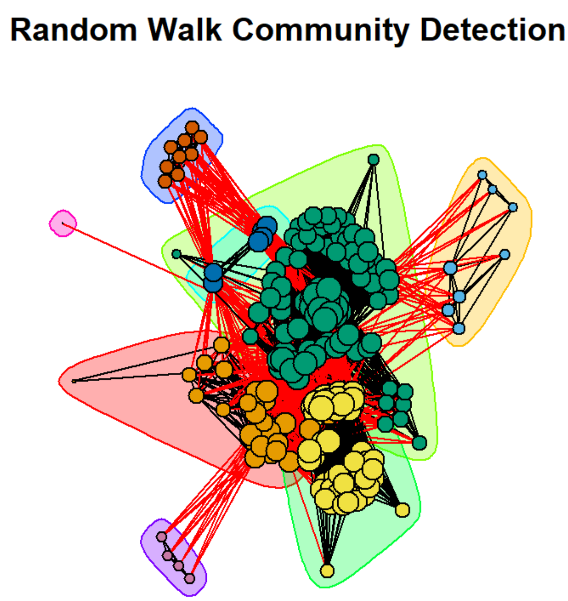
# 4b- Random Walk Community Visualization

Now, visualize the network with random walk community detection. Nodes of the same color belong the the same detected community. Nodes are sized proportionally to the natural logarithm of their degrees, given the large range of degree values.

```
# Compute node degree
deg <- degree(graph)

# Fruchterman-Reingold layout
layout <- layout_with_fr(graph)

# Plot the graph
plot(walk, graph, layout = layout,
     vertex.size = log(deg) * 2.5,
     vertex.label = NA,
     edge.arrow.size = 0.3,
     main = "Random Walk Community Detection")
```




**Interpretation**

- The Walktrap partition breaks the network into three large groups, and a 5 other small groups. The large groups are shown by the central green, orange, and yellow clusters.
- The Walktrap algorithm thus splits the network to a much lesser extent compared to the Girvan-Newman algorithm.
- Nodes in the three large groups are highly connected relative to other groups, as they have larger node sizes indicating higher degrees. In other words, countries in these three main groups are connected to many other countries by PTAs. This is expected as we expect nodes in larger clusters to be connected to more nodes on average.

# 5- Tell me anything else about whether the partitioning makes sense, based on attributes or who the nodes are, and so on.

## 5a- Sense-checking Girvan-Newman communities

To see if the partitioning makes sense, let us print all the countries in each community identified by the Girvan-Newman algorithm and manually identify if there are any logical patterns in the grouping.

```r
# Loop through each Girvan-Newman community
for (i in seq_along(gn)) {
  cat("\n--- Community", i, "---\n")
  # Convert 3-letter codes to full country names
  full_names <- countrycode(gn[[i]], origin = "iso3c", destination = "country.name")
  print(full_names)
}
```

```
##
## --- Community 1 ---
##  [1] "Aruba"                   "Anguilla"
##  [3] "Albania"                 "Andorra"
##  [5] "Armenia"                 "Antigua & Barbuda"
##  [7] "Austria"                 "Belgium"
##  [9] "Bulgaria"                "Bahamas"
## [11] "Bosnia & Herzegovina"    "Belize"
## [13] "Bermuda"                 "Barbados"
## [15] "Canada"                  "Switzerland"
## [17] "Chile"                   "Colombia"
## [19] "Costa Rica"              "Cayman Islands"
## [21] "Cyprus"                  "Czechia"
## [23] "Germany"                 "Dominica"
## [25] "Denmark"                 "Dominican Republic"
## [27] "Ecuador"                 "Spain"
## [29] "Estonia"                 "Finland"
## [31] "Fiji"                    "France"
## [33] "Faroe Islands"           "United Kingdom"
## [35] "Georgia"                 "Greece"
## [37] "Grenada"                 "Greenland"
## [39] "Guatemala"               "Guyana"
## [41] "Honduras"                "Croatia"
## [43] "Haiti"                   "Hungary"
## [45] "Ireland"                 "Iceland"
## [47] "Israel"                  "Italy"
## [49] "Jamaica"                 "Jordan"
## [51] "Japan"                   "St. Kitts & Nevis"
## [53] "South Korea"             "Lebanon"
## [55] "St. Lucia"               "Liechtenstein"
## [57] "Lithuania"               "Luxembourg"
## [59] "Latvia"                  "Moldova"
## [61] "Mexico"                  "North Macedonia"
## [63] "Malta"                   "Montenegro"
## [65] "Montserrat"              "New Caledonia"
## [67] "Nicaragua"               "Netherlands"
## [69] "Norway"                  "Panama"
## [71] "Pitcairn Islands"        "Peru"
## [73] "Papua New Guinea"        "Poland"
## [75] "Portugal"                "Palestinian Territories"
## [77] "French Polynesia"        "Romania"
## [79] "Singapore"               "Solomon Islands"
## [81] "El Salvador"             "San Marino"
## [83] "St. Pierre & Miquelon"   "Serbia"
## [85] "Suriname"                "Slovakia"
## [87] "Slovenia"                "Sweden"
## [89] "Syria"                   "Turks & Caicos Islands"
## [91] "Trinidad & Tobago"       "Turkey"
## [93] "Ukraine"                 "St. Vincent & Grenadines"
## [95] "Vietnam"                 "Wallis & Futuna"
## [97] "Samoa"
##
## --- Community 2 ---
## [1] "Afghanistan" "Bhutan"      "Maldives"    "Nepal"
##
## --- Community 3 ---
##  [1] "Angola"                  "Argentina"
##  [3] "Burundi"                 "Benin"
##  [5] "Burkina Faso"            "Bangladesh"
##  [7] "Bolivia"                 "Brazil"
##  [9] "Botswana"                "Central African Republic"
```

```
## [11] "Côte d’Ivoire"            "Cameroon"
## [13] "Congo - Kinshasa"         "Congo - Brazzaville"
## [15] "Comoros"                  "Cape Verde"
## [17] "Cuba"                     "Djibouti"
## [19] "Algeria"                  "Egypt"
## [21] "Ethiopia"                 "Gabon"
## [23] "Ghana"                    "Guinea"
## [25] "Guinea-Bissau"            "Equatorial Guinea"
## [27] "Indonesia"                "India"
## [29] "Iran"                     "Iraq"
## [31] "Kenya"                    "Liberia"
## [33] "Libya"                    "Sri Lanka"
## [35] "Lesotho"                  "Morocco"
## [37] "Madagascar"               "Mali"
## [39] "Myanmar (Burma)"          "Mozambique"
## [41] "Mauritania"               "Mauritius"
## [43] "Malawi"                   "Malaysia"
## [45] "Namibia"                  "Niger"
## [47] "Nigeria"                  "Pakistan"
## [49] "Philippines"              "North Korea"
## [51] "Rwanda"                   "Sudan"
## [53] "Senegal"                  "Sierra Leone"
## [55] "Somalia"                  "South Sudan"
## [57] "São Tomé & Príncipe"      "Eswatini"
## [59] "Seychelles"               "Chad"
## [61] "Togo"                     "Thailand"
## [63] "Tunisia"                  "Tanzania"
## [65] "Uganda"                   "Venezuela"
## [67] "South Africa"             "Zambia"
## [69] "Zimbabwe"
## 
## --- Community 4 ---
## [1] "United Arab Emirates"
## 
## --- Community 5 ---
##  [1] "Australia"                        "Cook Islands"
##  [3] "Micronesia (Federated States of)" "Kiribati"
##  [5] "Marshall Islands"                 "Niue"
##  [7] "Nauru"                            "New Zealand"
##  [9] "Tonga"                            "Tuvalu"
## [11] "Vanuatu"
## 
## --- Community 6 ---
## [1] "Azerbaijan"   "Russia"       "Turkmenistan" "Uzbekistan"
## 
## --- Community 7 ---
## [1] "Bahrain"
## 
## --- Community 8 ---
## [1] "Belarus"    "Kazakhstan" "Kyrgyzstan" "Tajikistan"
## 
## --- Community 9 ---
## [1] "Brunei"
## 
## --- Community 10 ---
## [1] "China"
## 
## --- Community 11 ---
## [1] "Eritrea"
## 
## --- Community 12 ---
## [1] "Gambia"
```

```
##
## --- Community 13 ---
## [1] "Hong Kong SAR China"
##
## --- Community 14 ---
## [1] "Cambodia"
##
## --- Community 15 ---
## [1] "Kuwait"
##
## --- Community 16 ---
## [1] "Laos"
##
## --- Community 17 ---
## [1] "Macao SAR China"
##
## --- Community 18 ---
## [1] "Mongolia"
##
## --- Community 19 ---
## [1] "Oman"
##
## --- Community 20 ---
## [1] "Paraguay"
##
## --- Community 21 ---
## [1] "Qatar"
##
## --- Community 22 ---
## [1] "Saudi Arabia"
##
## --- Community 23 ---
## [1] "Taiwan"
##
## --- Community 24 ---
## [1] "Uruguay"
##
## --- Community 25 ---
## [1] "United States"
##
## --- Community 26 ---
```

```
## Warning: Some values were not matched unambiguously: XKX
```

```
## [1] NA
##
## --- Community 27 ---
## [1] "Yemen"
##
## --- Community 28 ---
```

```
## Warning: Some values were not matched unambiguously: YUG
```

```
## [1] NA
```

#### Interpretation

- Community 1 and 3 are the largest communities containing countries all across the globe, and it is unclear how they are different or grouped in that manner.

- Community 5 contains Oceanian countries and territories which are geographically near each other. As they are geographically close, it is sensible that they have trade ties with each other.
- Community 6 and 8 contains Russia and ex-Soviet states which are geographically somewhat close to each other and historically tied. It is not suprising that they have been grouped together.
  - However, the split between 6 and 8 may be less intuitive, as we have two Central Asian countries in community 6 (Turkmenistan and Uzbekistan, and the three others in community 8 (Kazakhstan, Kyrgyzstan, Tajikistan). We would have expected them to be within the same community given their geographical proximity.
  - In particular, it is unexpected to see geographically further Belarus in community 8 when two other central Asian countries are not in the community.
- We would have expected a country like North Korea to be identified to be isolated in its own community, but it is found in Community 3. At the same time, we find countries like China and the United States which are highly connected in global trade in their own individual communities 10 and 25 respectively, which seems unintuitive.
- We find some Chinese territories isolated in their own communities such as Hong Kong SAR China (Community 13) and Macao SAR China (Community 17). This is sensible as countries are unlikely to have trade agreements directly with them. Trade agreements are likely formed through China instead.

# 5b- Sense-checking Walktrap communities

Let's do the same for communities identified by the random walk (walktrap) algorithm.

```
# Loop through each Walktrap community
for (i in seq_along(walk)) {
  cat("\n--- Community", i, "---\n")
  # Convert 3-letter codes to full country names
  full_names <- countrycode(walk[[i]], origin = "iso3c", destination = "country.name")
  print(full_names)
}
```

```
##
## --- Community 1 ---
```

```
## Warning: Some values were not matched unambiguously: YUG
```

```
##  [1] "Argentina"          "Benin"              "Bangladesh"
##  [4] "Bolivia"            "Brazil"             "Brunei"
##  [7] "China"              "Cuba"               "Hong Kong SAR China"
## [10] "Indonesia"          "India"              "Iran"
## [13] "Iraq"               "Cambodia"           "Laos"
## [16] "Sri Lanka"          "Macao SAR China"    "Myanmar (Burma)"
## [19] "Malaysia"           "Pakistan"           "Philippines"
## [22] "North Korea"        "Paraguay"           "Thailand"
## [25] "Uruguay"            "Venezuela"          NA
##
## --- Community 2 ---
## [1] "Azerbaijan"   "Belarus"      "Kazakhstan"   "Kyrgyzstan"   "Russia"
## [6] "Tajikistan"   "Turkmenistan" "Uzbekistan"
##
## --- Community 3 ---
```

```
## Warning: Some values were not matched unambiguously: XKX
```

```
##   [1] "Aruba"                    "Anguilla"
##   [3] "Albania"                  "Andorra"
##   [5] "United Arab Emirates"     "Armenia"
##   [7] "Antigua & Barbuda"        "Austria"
##   [9] "Belgium"                  "Bulgaria"
##  [11] "Bahrain"                  "Bahamas"
##  [13] "Bosnia & Herzegovina"     "Belize"
##  [15] "Bermuda"                  "Barbados"
##  [17] "Canada"                   "Switzerland"
##  [19] "Chile"                    "Colombia"
##  [21] "Costa Rica"               "Cayman Islands"
##  [23] "Cyprus"                   "Czechia"
##  [25] "Germany"                  "Dominica"
##  [27] "Denmark"                  "Dominican Republic"
##  [29] "Ecuador"                  "Spain"
##  [31] "Estonia"                  "Finland"
##  [33] "France"                   "Faroe Islands"
##  [35] "United Kingdom"           "Georgia"
##  [37] "Greece"                   "Grenada"
##  [39] "Greenland"                "Guatemala"
##  [41] "Guyana"                   "Honduras"
##  [43] "Croatia"                  "Haiti"
##  [45] "Hungary"                  "Ireland"
##  [47] "Iceland"                  "Israel"
##  [49] "Italy"                    "Jamaica"
##  [51] "Jordan"                   "Japan"
##  [53] "St. Kitts & Nevis"        "South Korea"
##  [55] "Kuwait"                   "Lebanon"
##  [57] "St. Lucia"                "Liechtenstein"
##  [59] "Lithuania"                "Luxembourg"
##  [61] "Latvia"                   "Moldova"
##  [63] "Mexico"                   "North Macedonia"
##  [65] "Malta"                    "Montenegro"
##  [67] "Montserrat"               "New Caledonia"
##  [69] "Nicaragua"                "Netherlands"
##  [71] "Norway"                   "Oman"
##  [73] "Panama"                   "Pitcairn Islands"
##  [75] "Peru"                     "Poland"
##  [77] "Portugal"                 "Palestinian Territories"
##  [79] "French Polynesia"         "Qatar"
##  [81] "Romania"                  "Saudi Arabia"
##  [83] "Singapore"                "El Salvador"
##  [85] "San Marino"               "St. Pierre & Miquelon"
##  [87] "Serbia"                   "Suriname"
##  [89] "Slovakia"                 "Slovenia"
##  [91] "Sweden"                   "Syria"
##  [93] "Turks & Caicos Islands"   "Trinidad & Tobago"
##  [95] "Turkey"                   "Taiwan"
##  [97] "Ukraine"                  "United States"
##  [99] "St. Vincent & Grenadines" "Vietnam"
## [101] "Wallis & Futuna"          NA
## [103] "Yemen"
## 
## --- Community 4 ---
##  [1] "Angola"                   "Burundi"
##  [3] "Burkina Faso"             "Botswana"
##  [5] "Central African Republic" "Côte d’Ivoire"
##  [7] "Cameroon"                 "Congo - Kinshasa"
##  [9] "Congo - Brazzaville"      "Comoros"
## [11] "Cape Verde"               "Djibouti"
## [13] "Algeria"                  "Egypt"
```

```
## [15] "Eritrea"                 "Ethiopia"
## [17] "Gabon"                   "Ghana"
## [19] "Guinea"                  "Gambia"
## [21] "Guinea-Bissau"           "Equatorial Guinea"
## [23] "Kenya"                   "Liberia"
## [25] "Libya"                   "Lesotho"
## [27] "Morocco"                 "Madagascar"
## [29] "Mali"                    "Mozambique"
## [31] "Mauritania"              "Mauritius"
## [33] "Malawi"                  "Namibia"
## [35] "Niger"                   "Nigeria"
## [37] "Rwanda"                  "Sudan"
## [39] "Senegal"                 "Sierra Leone"
## [41] "Somalia"                 "South Sudan"
## [43] "São Tomé & Príncipe"     "Eswatini"
## [45] "Seychelles"              "Chad"
## [47] "Togo"                    "Tunisia"
## [49] "Tanzania"                "Uganda"
## [51] "South Africa"            "Zambia"
## [53] "Zimbabwe"
##
## --- Community 5 ---
## [1] "Australia"        "Fiji"             "New Zealand"      "Papua New Guinea"
## [5] "Solomon Islands"  "Samoa"
##
## --- Community 6 ---
## [1] "Cook Islands"                   "Micronesia (Federated States of)"
## [3] "Kiribati"                       "Marshall Islands"
## [5] "Niue"                           "Nauru"
## [7] "Tonga"                          "Tuvalu"
## [9] "Vanuatu"
##
## --- Community 7 ---
## [1] "Afghanistan" "Bhutan"      "Maldives"    "Nepal"
##
## --- Community 8 ---
## [1] "Mongolia"
```

#### Interpretation

- Community 1 mainly contains countries from across Asia and South America.
  - This includes Southeast Asia (Indonesia, Cambodia, Laos, Myanmar), South Asia (Bangladesh Pakistan, India), East Asia (China, North Korea), and the Middle East (Iraq, Iran). South America countries include Argentina, Bolivia, Uruguay etc.
  - This covers a large swathe of area across Asia, alongside South America, which is geographically distant from Asia.
  - Weirdly, it also includes Benin, a west african country.
- Community 3 primarily contains European countries, North American (United States, Canada, Mexico), Central American (El Salvador, Belize, Panama), South American (Ecuador), some Asian (Singapore, Japan), and Middle Eastern (Yemen, Saudi Arabia, Kuwait, Qatar) countries.
- While Community 3 appears to be more ‘western-centric’ trade blocs in that it contains Europe and North America, it is not entirely clear why some countries are included in community 1 or 3.
  - Perhaps community 3 contains more western-aligned (trade-wise) countries as well.
  - For example, Iraq and Iran face western trade sanctions, and are found in community 1. Whereas other Middle Eastern countries like Saudi Arabia and Qatar are found in community 3, being more connected to the likes of Europe and North America than the geographically proximate Iraq and Iran.
  - Similarly, we find China and North Korea in community 1, and more ‘western-friendly’ Asian countries like Japan in community 3. Whilst China has significant trade with the west on the basis that it is the world’s second largest economy and largest expoorter, it may have been grouped in community 1 without western countries as there may be less trade deals given the geostrategic competition between China and the west.

- Community 2 contains ex-Soviet countries and Russia, and is made up of Girvan-Newman’s community 6 and 8 combined together.
- Community 4 primarily contains African countries. This makes sense as a grouping.
- Community 5 and 6 contains Oceanic countries and territories, much like Girvan-Newman’s Community 5, which is a subset of the Walktrap’s combined Community 5 and 6. It is sensible that they are grouped in this manner given their geographical proximity and thus likely trade relations between these countries.
- Community 7 is identical to Girvan-Newman’s Community 2. While Afghanistan, Bhutan, Maldives, and Nepal are somewhat geographically near each other, it seems like an unusual grouping, as it excludes the many countries in their region as well. While Nepal and Bhutan neighbor each other, Afghanistan is considerably further west, and Maldives much further South.

# Conclusion

Overall, both community detection algorithms produce communities with significant similarities, but also notable differences. We have demonstrated this through the RI, ARI, NMI, VI metrics, alongside a matrix visualization highlighting their similarities and differences.

By manually looking at the groupings, the communities produced by the random walk (walktrap) algorithm appear slightly more intelligible compared to Girvan-Newman’s community groupings. It is great to see that applying these algorithms produce communities that make some intuitive sense. We find communities based on geographic proximity, and possibly even geostrategic friendliness (extending to trade agreements), as discussed in the case of the random walk (walktrap) algorithm communities.