Social Network Analysis

Andreas Opedal, Nicolò Grometto & Charlotte Out

23/05/2020

Loading Data and Packages

```
# installing the networkdata package

# install.packages("drat")
# drat::addRepo("schochastics")
# install.packages("networkdata")

# data(package = "networkdata")#

# extract chosen data set from package
# gang = networkdata::covert_28

# loading libraries
library(ggforce)
library(igraph)
library(sna)
library(ggplot2)
library(ggraph)
library(readr)
library(concaveman)
library(Rfast)
library(purrr)

# loading data
attributes <- read_csv("attributes.csv")
gang <- read_csv("london_gang.csv")
gang = gang[,-1]

# View(gang)
# View(attributes)
```

General Information - London Gang Data

DESCRIPTION: Data is on co-offending in a London-based inner-city street gang, 2005-2009, operating from a social housing estate. Data comes from anonymized police arrest and conviction data for all confirmed members of the gang.

DATA FORMAT: UCINET, .csv

DATA: 1-Mode matrix 54 x 54 persons by persons, undirected, valued.

Network tie values: = 1 (hang out together) = 2 (co-offend together) = 3 (co-offend together, serious crime) = 4 (co-offend together, serious crime, kin)

Attributes: Age, Birthplace (1 = West Africa, 2= Caribbean, 3= UK, 4= East Africa), Residence, Arrests, Convictions, Prison, Music.

We note the following of the network:
1. The network data contains a large collection of attributes relative to the subjects who are part of the gang under consideration, e.g. age, birthplace, residence, number of arrests, number of convictions and more. In the following analysis we focus particularly on age and birthplace;
2. The network data relative to the criminal gang under analysis is organised in layers, from members who only hang out together all the way up to members who have kin relationships and collaborate in serious crime acts, where upper layers are subsets of the lower. This particular feature allows to study the layered structure of this community and understand how interpersonal ties interact with crime acts in this kind of community;

In what follows, we concentrate specifically on the hang-out network (ties coded as 1 and above) and on the Serious Crime Network (ties coded as 3 and above). Limiting our focus to a subset of the data, we prioritise the depth of our analysis, rather than breadth:

We now proceed with the description, analysis and visual representation of the chosen data set.

Data Exploration

In this section, we proceed by calculating basic descriptives on the hang-out and co-offenders (serious crime) networks, with ties encoded by 1, 2, 3, and 4 as well as 3 and 4 in the network data, respectively. In particular, we are interested in analysing the data in the context provided by the associated attributes. We also provide appropriate visualisations of the different networks as well as comments and insights into our analysis, based on the obtained results.

General Traits

```
# Number of missing values
number_of_missings <- sum(is.na(gang))
print(paste0("Number of missings: ", round(number_of_missings, digits = 4)))
```

```
## [1] "Number of missings: 0"
```

```
# Network size
network_size <- nrow(gang)
print(paste0("Network(s) size: ", round(network_size, digits = 4)))
```

```
## [1] "Network(s) size: 54"
```

```
# Ethnic composition of gang
west_africa <- nrow(attributes[attributes$Birthplace == 1,]) / nrow(attributes)
caribbean <- nrow(attributes[attributes$Birthplace == 2,]) / nrow(attributes)
uk <- nrow(attributes[attributes$Birthplace == 3,]) / nrow(attributes)
east_africa <- nrow(attributes[attributes$Birthplace == 4,]) / nrow(attributes)
print(paste0("Proportion born in West Africa: ", round(west_africa, digits = 3)))
```

```
## [1] "Proportion born in West Africa: 0.222"
```

```
print(paste0("Proportion born in the Caribbeans: ", round(caribbean, digits = 3)))
```

```
## [1] "Proportion born in the Caribbeans: 0.222"
```

```
print(paste0("Proportion born in the UK: ", round(uk, digits = 3)))
```

```
## [1] "Proportion born in the UK: 0.444"
```

```
print(paste0("Proportion born in East Africa: ", round(east_africa, digits = 3)))
```

```
## [1] "Proportion born in East Africa: 0.111"
```

```
# Age composition of gang
Age_group = c()
for (i in 1:nrow(attributes)) {
  if (attributes$Age[i] < 20) {Age_group[i] = 1}
  else if (attributes$Age[i] < 25) {Age_group[i] = 2}
  else {Age_group[i] = 3}
}
attributes <- cbind(attributes, Age_group)
print(table(attributes$Age_group))
```

```
##
##  1  2  3
## 28 21  5
```

From the above R output, we make the following observations:

1. The data under consideration does not contain any missing values, whilst as already mentioned in the description, 54 individuals make up the whole gang population (these correspond to nodes in both examined networks);
2. Based on the descriptives above, we associate this particular data set with a snowball sampling procedure, which was carried out by

making use of the data provided by the authorities. This explains the lack of missing values;

3. The gang in uneven in terms of ethnicity of the members, with more than 44% of its members being British. The remaining individuals are from West Africa (22%), Caribbean (22%), and East Africa (11%). This prompts us to investigate further the degree of ethnic segregation both in the Hang-out and Serious Crime Networks, by conducting an appropriate cluster analysis;
4. The age break-down of the gang is given as follows: 16-19 years old - 28 individuals (Age group 1), 20-24 years old - 21 individuals (Age group 2) and above 25 years old - 5 individuals (Age group 3). The uneven age distribution will also be object of investigation later on in the analysis.

After having gathered basic descriptives on the dataset and on the gang composition of the gang, we now move on to a more advanced network analysis (including visual representations) of the chosen network layers.

Hang-Out Network

Advanced Descriptives

```
# Hang-Out Network:

# re-naming for convenience
gang_1 = gang

# re-coding edges labels
gang_1[gang_1 > 0] <- 1

off_1 <- sapply(gang_1,as.numeric)

off_1.graph <- graph_from_adjacency_matrix(off_1,
                          mode = "undirected",
                          diag = FALSE
                          )

# number of edges
number_of_edges_1 <- length(off_1[off_1 == 1]) / 2
print(paste0("Number of edges: ", round(number_of_edges_1, digits = 4)))
```

```
## [1] "Number of edges: 315"
```

```
# number of isolated nodes
isolates_1 <- sum(rowSums(off_1) == 0)
print(paste0("Number of isolates: ", round(isolates_1, digits = 4)))
```

```
## [1] "Number of isolates: 0"
```

```
# Density (number of ties over maximum)
density_1 <- number_of_edges_1 / ((network_size * (network_size - 1))/2)
print(paste0("Density:", round(density_1, digits = 4)))
```

```
## [1] "Density:0.2201"
```

```
# Average degree
attributes$degree_1 <- sna::degree(off_1, gmode = "graph")
average_degree_1 <- mean(attributes$degree_1)
print(paste0("Average degree: ", round(average_degree_1, digits = 4)))
```

```
## [1] "Average degree: 11.6667"
```

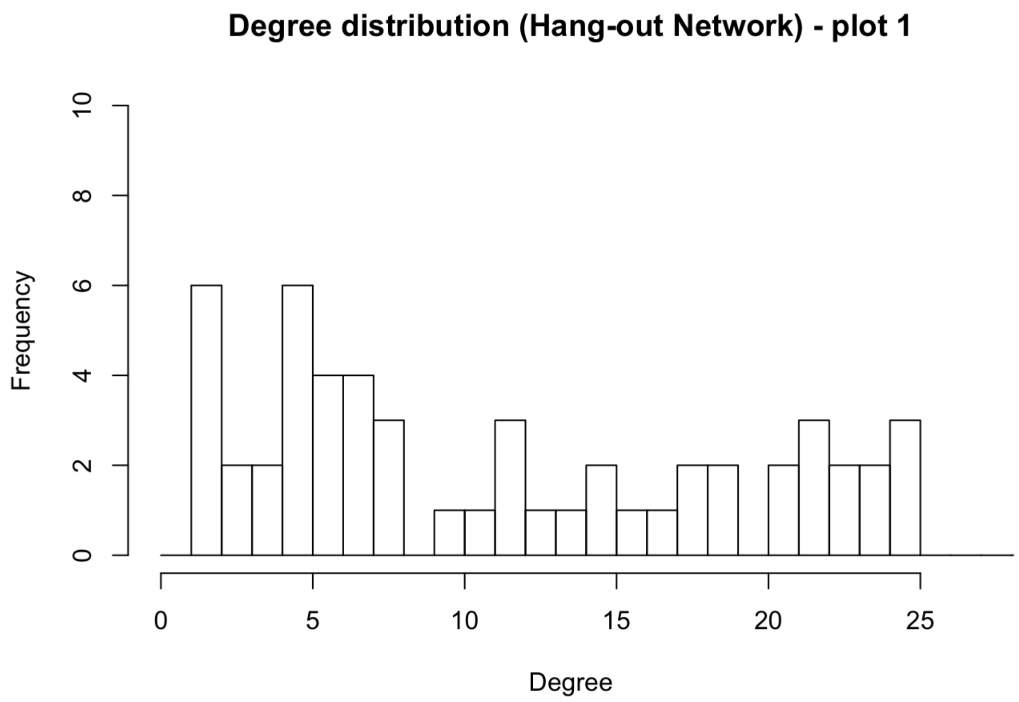
```
# degree standard deviation
deg_sd <- sd(attributes$degree_1)
print(paste0("Degree standard deviation: ", round(deg_sd, digits = 4)))
```

```
## [1] "Degree standard deviation: 7.78"
```

```
# degree distribution
degree.distrib_1 <- colSums(off_1, na.rm = T)
table(degree.distrib_1)
```

```
## degree.distrib_1
##  2  3  4  5  6  7  8 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 
##  6  2  2  6  4  4  3  1  1  3  1  1  2  1  1  2  2  2  3  2  2  3 
```

```
hist(degree.distrib_1, breaks = (0:54), xlim = c(0,27), ylim = c(0,10), main = 'Degree distribution (Hang-o
ut Network) - plot 1', xlab = 'Degree')
```



```
# same-ethnicity ties
west_africa_tie_net_1 <- off_1[attributes$Birthplace == 1, attributes$Birthplace == 1]
if (sum(west_africa_tie_net_1) > 0)
  {
  west_africa_ties_1 <- length(west_africa_tie_net_1[west_africa_tie_net_1 == 1])/2
west_africa_ratio_1 <- west_africa_ties_1 / number_of_edges_1
  } else {
  west_africa_ratio_1 <-0
}

caribbean_tie_net_1 <- off_1[attributes$Birthplace == 2, attributes$Birthplace == 2]
if (sum(caribbean_tie_net_1) > 0) {
  caribbean_ties_1 <- length(caribbean_tie_net_1[caribbean_tie_net_1 == 1])/2
caribbean_ratio_1 <- caribbean_ties_1 / number_of_edges_1
  } else {
    caribbean_ratio_1 <- 0
}

uk_tie_net_1 <- off_1[attributes$Birthplace == 3, attributes$Birthplace == 3]
if (sum(uk_tie_net_1) > 0) {
  uk_ties_1 <- length(uk_tie_net_1[uk_tie_net_1 == 1]) /2
uk_ratio_1 <- uk_ties_1 / number_of_edges_1
  } else {
  uk_ratio_1 <- 0
}

east_africa_tie_net_1 <- off_1[attributes$Birthplace == 4, attributes$Birthplace == 4]
if (sum(east_africa_tie_net_1) > 0) {
  east_africa_ties_1 <- length(east_africa_tie_net_1[east_africa_tie_net_1 == 1])/2
east_africa_ratio_1 <- east_africa_ties_1 / number_of_edges_1
} else {
    east_africa_ratio_1 <- 0
}

observed_et_1 <- west_africa_ratio_1 + caribbean_ratio_1 + uk_ratio_1 + east_africa_ratio_1

print(paste0("Same-ethnicity ties (proportion) - West Africa: ", round(west_africa_ratio_1, digits = 4)))
```

```
## [1] "Same-ethnicity ties (proportion) - West Africa: 0.1111"
```

```
print(paste0("Same-ethnicity ties (proportion) - Caribbean: ", round(caribbean_ratio_1, digits = 4)))
```

```
## [1] "Same-ethnicity ties (proportion) - Caribbean: 0.073"
```

```
print(paste0("Same-ethnicity ties (proportion) - UK: ", round(uk_ratio_1, digits = 4)))
```

```
## [1] "Same-ethnicity ties (proportion) - UK: 0.146"
```

```
print(paste0("Same-ethnicity ties (proportion) - East Africa: ", round(east_africa_ratio_1, digits = 4)))
```

```
## [1] "Same-ethnicity ties (proportion) - East Africa: 0.0254"
```

```
print(paste0("Same-ethnicity ties - TOT (proportion): ", round(observed_et_1, digits = 4)))
```

```
## [1] "Same-ethnicity ties - TOT (proportion): 0.3556"
```

```
# same-age ties
age_group_1_tie_net_1 <- off_1[attributes$Age_group == 1, attributes$Age_group == 1]
if (sum(age_group_1_tie_net_1) > 0) {
  age_group_1_ties_1 <-  length(age_group_1_tie_net_1[age_group_1_tie_net_1 == 1])/2
age_group_1_ratio_1 <- age_group_1_ties_1 / number_of_edges_1
} else {
    age_group_1_ratio_1 <- 0
}

age_group_2_tie_net_1 <- off_1[attributes$Age_group == 2, attributes$Age_group == 2]
if (sum(age_group_2_tie_net_1) > 0) {
  age_group_2_ties_1 <-  length(age_group_2_tie_net_1[age_group_2_tie_net_1 == 1])/2
age_group_2_ratio_1 <- age_group_2_ties_1 / number_of_edges_1
} else {
    age_group_2_ratio_1 <- 0
}

age_group_3_tie_net_1 <- off_1[attributes$Age_group == 3, attributes$Age_group == 3]
if (sum(age_group_3_tie_net_1) > 0) {
  age_group_3_ties_1 <-  length(age_group_3_tie_net_1[age_group_3_tie_net_1 == 1])/2
age_group_3_ratio_1 <- age_group_3_ties_1 / number_of_edges_1
} else {
    age_group_3_ratio_1 <- 0
}
observed_age_1 <- age_group_1_ratio_1 + age_group_2_ratio_1 + age_group_3_ratio_1

print(paste0("Same-age ties (proportion) - 16 to 19: ", round(age_group_1_ratio_1, digits = 4)))
```

```
## [1] "Same-age ties (proportion) - 16 to 19: 0.1873"
```

```
print(paste0("Same-age ties (proportion) - 20 to 24: ", round(age_group_2_ratio_1, digits = 4)))
```

```
## [1] "Same-age ties (proportion) - 20 to 24: 0.2603"
```

```
print(paste0("Same-age ties (proportion) - above 24: ", round(age_group_3_ratio_1, digits = 4)))
```

```
## [1] "Same-age ties (proportion) - above 24: 0.019"
```

```
print(paste0("Same-age ties - TOT (proportion): ", round(observed_age_1, digits = 4)))
```

```
## [1] "Same-age ties - TOT (proportion): 0.4667"
```

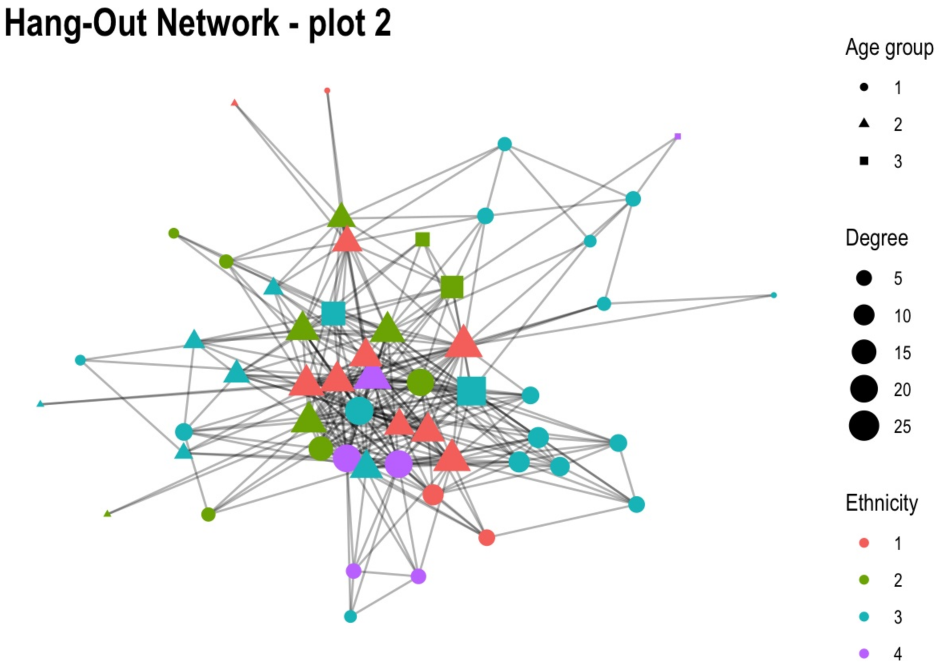
From the R output above, we make the following observations:

1. The Hang-Out Network consists of 315 edges;
2. The average degree (Note: the network is undirected) is 11.67, with a degree standard deviation being approximately 7.78. This indicates that on average each member of the gang has ties with approximately 12 other members, with an average variation of approximately 8 ties;
3. The graph is fairly sparse, with approximately 22% of the total possible number of edges appearing, as indicated by graph density;
4. The number of isolates is 0, indicating some degree of social cohesion based on interactions, rather than mere crime-based purpose within the gang members;
5. Same-ethnicity ties make up approximately 36% of the total number of ties. This suggests some level of ethnic segregation might exist in this network; however, we expect this structural property to be somewhat weak since almost two thirds of the total number of ties tend to be across ethnicities.
6. As expected from the ethnic composition of the gang, the number of ties between British members is the highest as a percentage of the total number of ties, since this is the majority ehtnic group in the gang;
7. It is interesting to point out that the degree distribution across the nodes is not highly skewed, and appears fairly uniform across the range of observed degrees;
8. Same-age group ties make up approximately 45% of the total number of ties in the network, also pointing towards some degree of segregation, here with respect to age; further details are enclosed in later sections.

We now proceed by plotting the above network, as displayed below.

Visual Representation

```r
off_1.graph <- graph_from_adjacency_matrix(off_1,
                          mode = "undirected",
                          diag = FALSE)

g_1 <- ggraph(off_1.graph,
        layout = 'nicely')+
        geom_edge_fan(alpha = .3) +
        geom_node_point(aes(shape = as.factor(attributes$Age_group),
                        colour = as.factor(attributes$Birthplace),
                        size = attributes$degree_1))+
        labs(colour = "Ethnicity",
        size = "Degree",
        shape = "Age group")+
        ggtitle("Hang-Out Network - plot 2")+
        theme_graph()

g_1
```



From the visualization of the above graph, we make the following observations:
1. As claimed above, we note that the data was arguably collected according to a snow-ball sampling procedure. Although the number of individuals in the gang is fairly small, we see a dense central nucleus of nodes with scattered branchings surrounding it. This structural property of the network is in line with our expectations, given the origin of the network data;
2. Some cluster structure emerges by looking at the figure above; in particular, one could argue that individuals of age group 2 form a community within the gang, given a strong concentration of triangles in the nucleuous of the network. It is important to note how individuals of ethnicity 1 (West Africa) mainly coincide with individuals of age group 2, hence implying some community structure with respect to ehtnicity as well (this will be analysed later);
3. The visual representation of the Hang-Out Network suggests a hierarchical structure of the gang under examination. In particular, there appears to be a restricted number of very central nodes with large degree in the network. These could arguably be associated with more influential members of the gang and are surrounded by a larger number of nodes with degree progressively decreasing as we move away from the core of the network;
4. Nodes with higher degree tend to be individuals belonging to age groups 2 and 3. This suggests that seniority plays a role in the gang, with older members having a larger number of ties, hence higher degree centrality in the network;

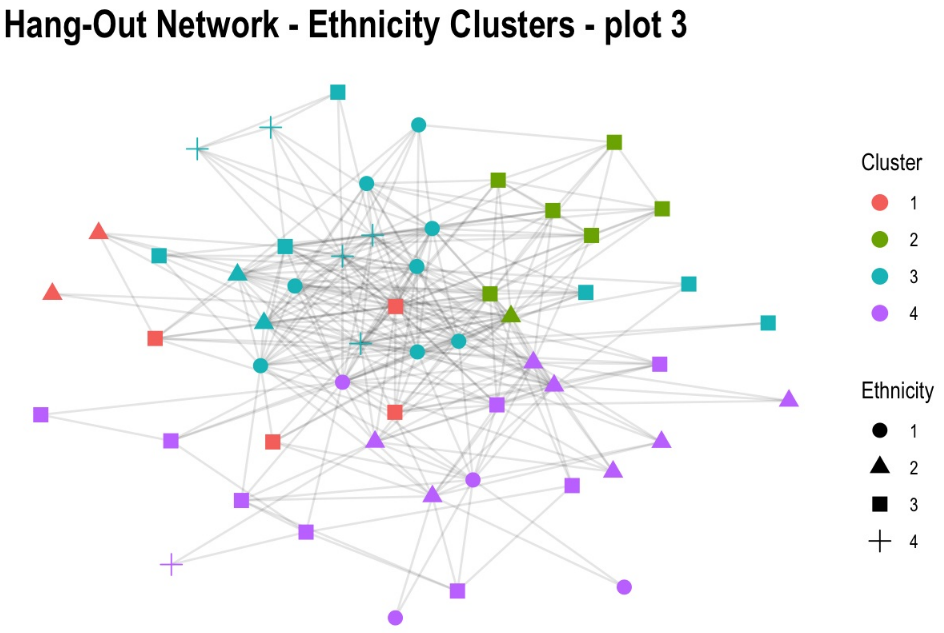
```r
par(mfrow=c(2,1))

off_1.fg <- cluster_fast_greedy(off_1.graph)
member.off_1 <- membership(off_1.fg)
print(paste0("Modularity based on Fast Greedy Algorithm: ", round(modularity(off_1.fg), digits = 4)))
```

```
## [1] "Modularity based on Fast Greedy Algorithm: 0.2559"
```

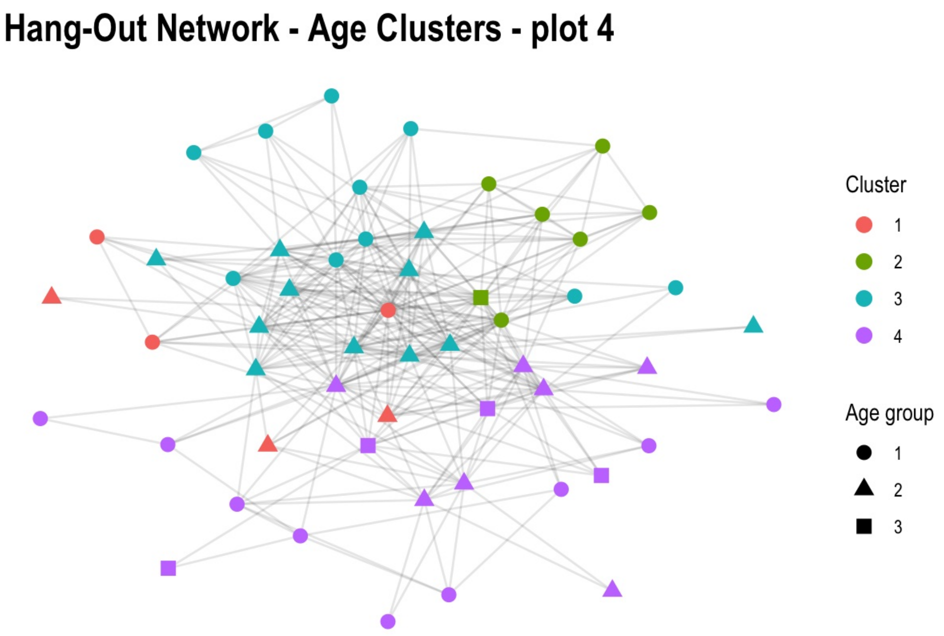
```
table(member.off_1)
```

```
## member.off_1
##  1  2  3  4
##  6  7 21 20
```

```
Cluster <- factor(member.off_1)

g1_cluster_A <- ggraph(off_1.graph,
     layout = 'kk')+

     geom_edge_fan(alpha = .1) +
     geom_node_point(aes(shape = as.factor(attributes$Birthplace),
                         color = Cluster),
                         show.legend = TRUE, size = 3) +
     labs(shape = "Ethnicity")+
     ggtitle("Hang-Out Network - Ethnicity Clusters - plot 3")+
     theme_graph()

g1_cluster_B <- ggraph(off_1.graph,
     layout = 'kk')+

     geom_edge_fan(alpha = .1) +
     geom_node_point(aes(shape = as.factor(attributes$Age_group),
                         color = Cluster),
                         show.legend = TRUE, size = 3) +
     labs(shape = "Age group")+
     ggtitle("Hang-Out Network - Age Clusters - plot 4")+
     theme_graph()

g1_cluster_A
```



```
g1_cluster_B
```



The above clusters were obtained via the fast greedy algorithm. From the obtained plots 3 and 4, we make the following observations:

1. There exist two major clusters (cluster 3 - 21 individuals; cluster 4 - 20 individuals) and two significantly smaller ones (cluster 1 - 6 individuals; cluster 2 - 7 individuals);
2. The modularity of the given network is approximately 0.26, pointing towards the existence of some community-like structure in the network analysed;
3. Some evidence of birthplace segregation appears by inspection, especially with respect to ethnic group 1, namely West Africa, as previously considered (see plot 2 and corresponding considerations);
4. Some community structure with respect to age emerges as well, in agreement with our previous considerations; in particular, one may notice how cluster 3 contains the majority of triangles (age group 2). We will later on test whether individuals in the gang are significantly more likely to hang out with people of the same age;

Serious Crime Network

In this section, we follow a similar procedure to the one adopted above in order to analyse and visualise the Serious Crime Network. We note that this consists of individuals who commit serious crime together and who might also have kin relationships, besides just hanging out together.

Advanced Descriptives

We start by providing the same descriptives as for the Hang-Out Network in order to compare the two.

```
gang_3 = gang

# re-coding edge labels
gang_3[gang_3 < 3] <- 0
gang_3[gang_3 > 2] <- 1

off_3 <- sapply(gang_3,as.numeric)

off_3.graph <- graph_from_adjacency_matrix(off_3,
                            mode = "undirected",
                            diag = FALSE
                            )

# number of edges
number_of_edges_3 <- length(off_3[off_3 == 1]) / 2
print(paste0("Number of edges: ", round(number_of_edges_3, digits = 4)))
```

```
## [1] "Number of edges: 41"
```

```
# percentage of serious crime ties
ratio_3_1 <- number_of_edges_3/number_of_edges_1
print(paste0("Ratio of crime to Hang-Out Networks: ", round(ratio_3_1, digits = 4)))
```

```
## [1] "Ratio of crime to Hang-Out Networks: 0.1302"
```

```
# number of isolated nodes
isolates_3 <- sum(rowSums(off_3) == 0)
print(paste0("Number of isolates: ", round(isolates_3, digits = 4)))
```

```
## [1] "Number of isolates: 22"
```

```
# Density (number of ties over maximum)
density_3 <- number_of_edges_3 / ((network_size * (network_size - 1))/2)
print(paste0("Density: ", round(density_3, digits = 4)))
```

```
## [1] "Density: 0.0287"
```

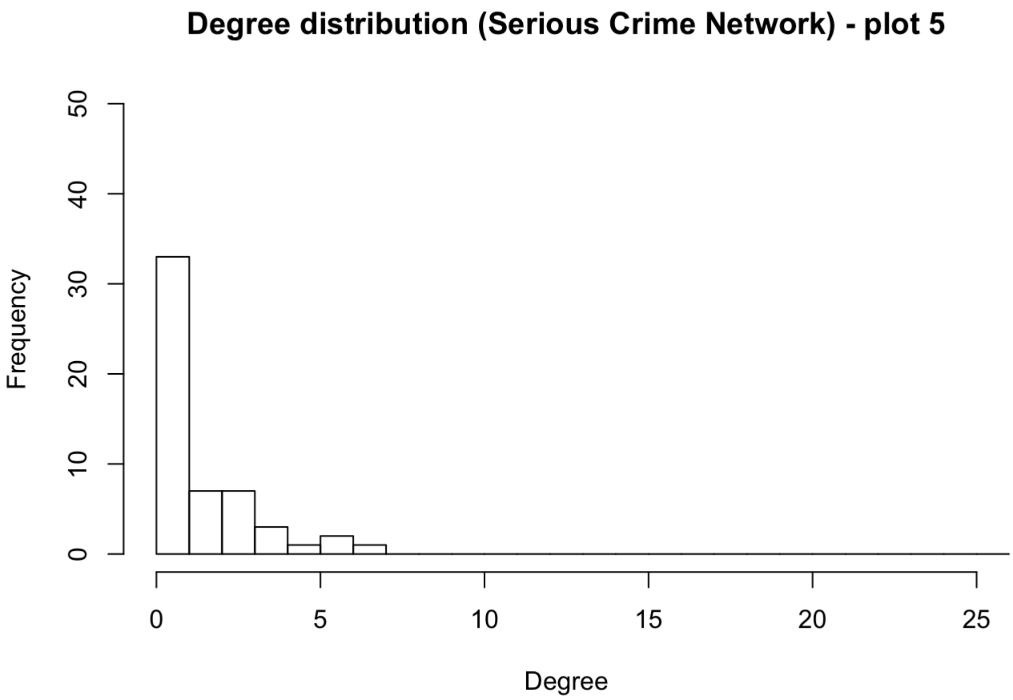
```
# Average degree
attributes$degree_3 <- sna::degree(off_3, gmode = "graph")
average_degree_3 <- mean(attributes$degree_3)
print(paste0("Average degree: ", round(average_degree_3, digits = 4)))
```

```
## [1] "Average degree: 1.5185"
```

```
# degree distribution
degree.distrib.3 <- colSums(off_3, na.rm = T)
table(degree.distrib.3)
```

```
## degree.distrib.3
##  0  1  2  3  4  5  6  7
## 22 11  7  7  3  1  2  1
```

```
hist(degree.distrib.3, breaks = (0:54), xlim = c(0,25), ylim = c(0,50), main = 'Degree distribution (Seriou
s Crime Network) - plot 5', xlab = 'Degree')
```



```
# same-ethnicity ties
west_africa_tie_net_3 <- off_3[attributes$Birthplace == 1, attributes$Birthplace == 1]
if (sum(west_africa_tie_net_3) > 0)
  {
  west_africa_ties_3 <- length(west_africa_tie_net_3[west_africa_tie_net_3 == 1])/2
west_africa_ratio_3 <- west_africa_ties_3 / number_of_edges_3
  } else {
  west_africa_ratio_3 <-0
}

caribbean_tie_net_3 <- off_3[attributes$Birthplace == 2, attributes$Birthplace == 2]
if (sum(caribbean_tie_net_3) > 0) {
  caribbean_ties_3 <- length(caribbean_tie_net_3[caribbean_tie_net_3 == 1])/2
caribbean_ratio_3 <- caribbean_ties_3 / number_of_edges_3
  } else {
    caribbean_ratio_3 <- 0
}

uk_tie_net_3 <- off_3[attributes$Birthplace == 3, attributes$Birthplace == 3]
if (sum(uk_tie_net_3) > 0) {
  uk_ties_3 <- length(uk_tie_net_3[uk_tie_net_3 == 1])/2
uk_ratio_3 <- uk_ties_3 / number_of_edges_3
  } else {
  uk_ratio_3 <- 0
}

east_africa_tie_net_3 <- off_3[attributes$Birthplace == 4, attributes$Birthplace == 4]
if (sum(east_africa_tie_net_3) > 0) {
  east_africa_ties_3 <- length(east_africa_tie_net_3[east_africa_tie_net_3 == 1])/2
east_africa_ratio_3 <- east_africa_ties_3 / number_of_edges_3
} else {
    east_africa_ratio_3 <- 0
}
observed_et_3 <- west_africa_ratio_3 + caribbean_ratio_3 + uk_ratio_3 + east_africa_ratio_3

print(paste0("Same-ethnicity ties (proportion) - West Africa: ", round(west_africa_ratio_3, digits = 4)))
```

```
## [1] "Same-ethnicity ties (proportion) - West Africa: 0.1463"
```

```
print(paste0("Same-ethnicity ties (proportion) - Caribbean: ", round(caribbean_ratio_3, digits = 4)))
```

```
## [1] "Same-ethnicity ties (proportion) - Caribbean: 0.1707"
```

```
print(paste0("Same-ethnicity ties (proportion) - UK: ", round(uk_ratio_3, digits = 4)))
```

```
## [1] "Same-ethnicity ties (proportion) - UK: 0.0976"
```

```
print(paste0("Same-ethnicity ties (proportion) - East Africa: ", round(east_africa_ratio_3, digits = 4)))
```

```
## [1] "Same-ethnicity ties (proportion) - East Africa: 0.1463"
```

```
print(paste0("Same-thnicity ties - TOT (proportion): ", round(observed_et_3, digits = 4)))
```

```
## [1] "Same-thnicity ties - TOT (proportion): 0.561"
```

```
# same-age ties
age_group_1_tie_net_3 <- off_3[attributes$Age_group == 1, attributes$Age_group == 1]
if (sum(age_group_1_tie_net_3) > 0) {
  age_group_1_ties_3 <-  length(age_group_1_tie_net_3[age_group_1_tie_net_3 == 1])/2
age_group_1_ratio_3 <- age_group_1_ties_3 / number_of_edges_3
} else {
    age_group_1_ratio_3 <- 0
}

age_group_2_tie_net_3 <- off_3[attributes$Age_group == 2, attributes$Age_group == 2]
if (sum(age_group_2_tie_net_3) > 0) {
  age_group_2_ties_3 <-  length(age_group_2_tie_net_3[age_group_2_tie_net_3 == 1])/2
age_group_2_ratio_3 <- age_group_2_ties_3 / number_of_edges_3
} else {
    age_group_2_ratio_3 <- 0
}

age_group_3_tie_net_3 <- off_3[attributes$Age_group == 3, attributes$Age_group == 3]
if (sum(age_group_3_tie_net_3) > 0) {
  age_group_3_ties_3 <-  length(age_group_3_tie_net_3[age_group_3_tie_net_3 == 1])/2
age_group_3_ratio_3 <- age_group_3_tie_net_3 / number_of_edges_3
} else {
    age_group_3_ratio_3 <- 0
}
observed_age_3 <- age_group_1_ratio_3 + age_group_2_ratio_3 + age_group_3_ratio_3

print(paste0("Same-age ties (proportion) - 16 to 19: ", round(age_group_1_ratio_1, digits = 4)))
```

```
## [1] "Same-age ties (proportion) - 16 to 19: 0.1873"
```

```
print(paste0("Same-age ties (proportion) - 20 to 24: ", round(age_group_2_ratio_1, digits = 4)))
```

```
## [1] "Same-age ties (proportion) - 20 to 24: 0.2603"
```

```
print(paste0("Same-age ties (proportion) - above 24: ", round(age_group_3_ratio_1, digits = 4)))
```

```
## [1] "Same-age ties (proportion) - above 24: 0.019"
```

```
print(paste0("Same-age ties - TOT (proportion): ", round(observed_age_3, digits = 4)))
```

```
## [1] "Same-age ties - TOT (proportion): 0.5122"
```

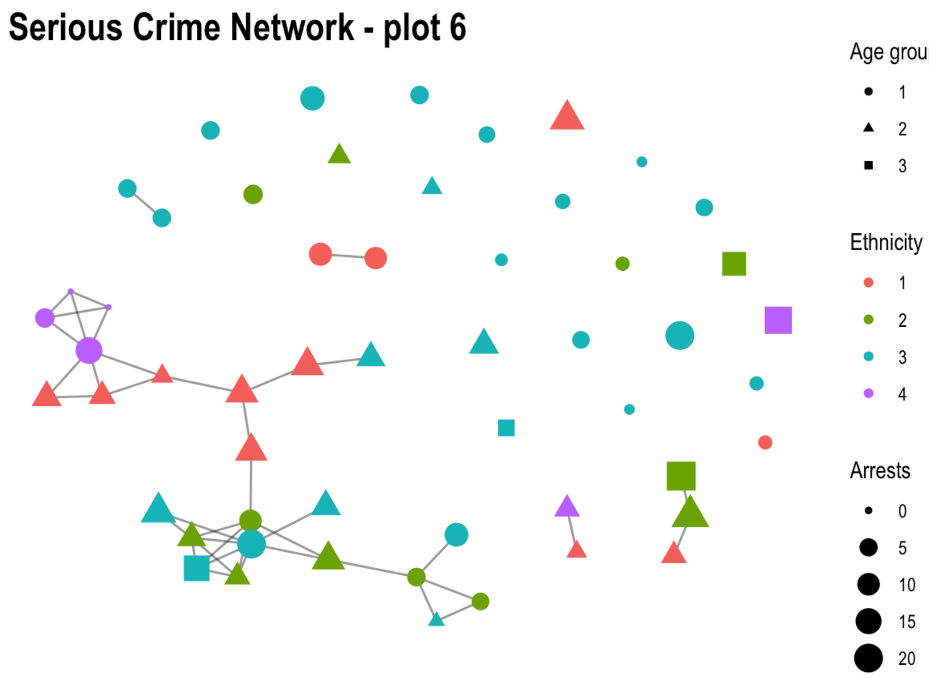
From the R output above, we make the following considerations:

1. Approximately 40% of the nodes are now isolates (22/54). This indicates that a restricted number of people who interact with one another within the gang actually collaborate in carrying out serious criminal acts. This is an interesting phenomenon as it highlights the fact that the gang acts as a social platform, on which its members develop social interactions in different forms, rather than being solely a criminal organisation. This result is also supported by the fact that the percentage of ties in the Hang-Out Network which are serious-crime ties is only approximately 13%, as calculated above;
2. Approximately 56% of the ties in the Serious Crime Network are between members belonging to the same ethnic groups. This figure is substantially higher compared to what was observed in the case of the Hang-Out Network. In particular, one may argue that this phenomenon stems from the fact that a stronger degree of trust is required for people to carry out serious crime acts together, which may involve high risks of getting caught by the authorities. Hence, ethnic groups are somewhat tighter with respect to joint criminal activity, compared to the weaker general social groups; in particular, we could identify mere hang-out ties as “weak ties” in the network, as opposed to “strong ties”, as given by ties in the Serious Crime Network. We will later perform an appropriate hypothesis test to better understand whether the above values for same-ethnicity ties is significantly larger than what one would expect by change in a similar network;
3. One might also observe a degree distribution very much concentrated around zero, due to high sparsity in the network.

We now continue our analysis with visual representations of the network.

Visual Representation

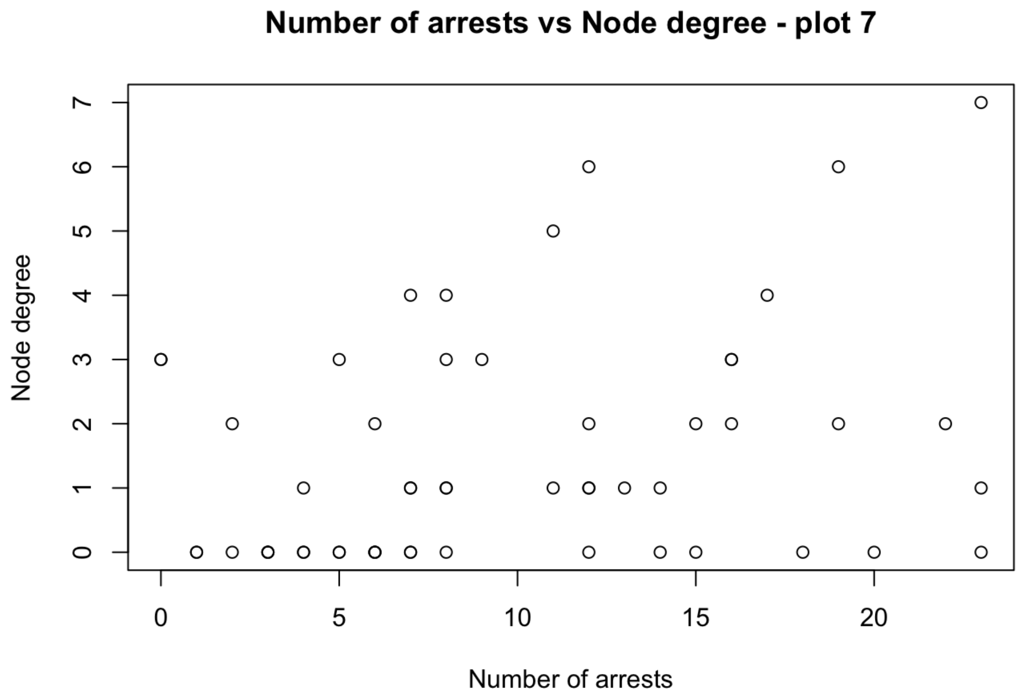
```
off_3.graph <- graph_from_adjacency_matrix(off_3,
                          mode = "undirected",
                          diag = FALSE
                          )

g_3 <- ggraph(off_3.graph,
         layout = 'nicely')+
         geom_edge_fan(alpha = .4) +
         geom_node_point(aes(shape = as.factor(attributes$Age_group),
                             colour = as.factor(attributes$Birthplace),
                             size = attributes$Arrests))+
         labs(colour = "Ethnicity",
         size = "Arrests",
         shape = "Age group")+
         ggtitle("Serious Crime Network - plot 6")+
         theme_graph()

g_3
```



By inspecting the above output, we make the following considerations:

1. Ties in the network are much sparser, as observed above, giving way to many more isolates;
2. Two major clusters appear; these are separated by an individual of West African origin, belonging to age group 2, who is in a structural hole position; in particular, we notice how the previous community of West African individuals of age group 2 is somewhat preserved in this network;
3. There seems to be a positive correlation between the number of arrests and nodes degree (thats is, individuals with fewer serious-crime ties with other individuals in the network tend to have fewer arrests as well). This can be observed by noticing that nodes in the main connnected component have in general larger size compared to isolated nodes. We verify this observations by examining the plot below (plot 7);

```
plot(attributes$Arrests, attributes$degree_3, xlab = 'Number of arrests', ylab = 'Node degree', main = 'Numb
er of arrests vs Node degree - plot 7')
```



```
print(paste0('Empirical correlation: ', round(cor(attributes$Arrests, attributes$degree_3), digits = 4)))
```

```
## [1] "Empirical correlation: 0.2975"
```

As observed above, indeed members who have more ties in the Serious Crime Network tend to exhibit a higher number of arrests. In particular, the empirical correlation between these two variables is approximately 0.30, indicating some degree of positive linear dependence between the above mentioned variables. This is also visible from the plot above, in which datapoints are located below the 45-degree line going through the origin. One may interpret this by arguing that serious crime acts are in general carried out in groups of more than one individual. Hence, individuals with high degree in the Serious Crime Network are more likely to be embedded in sub-communities devoted to carrying out such crime acts and therefore will expose themselves to higher risks of being arrested.

We now proceed with analysing the cluster structure of the Serious Crime Network.

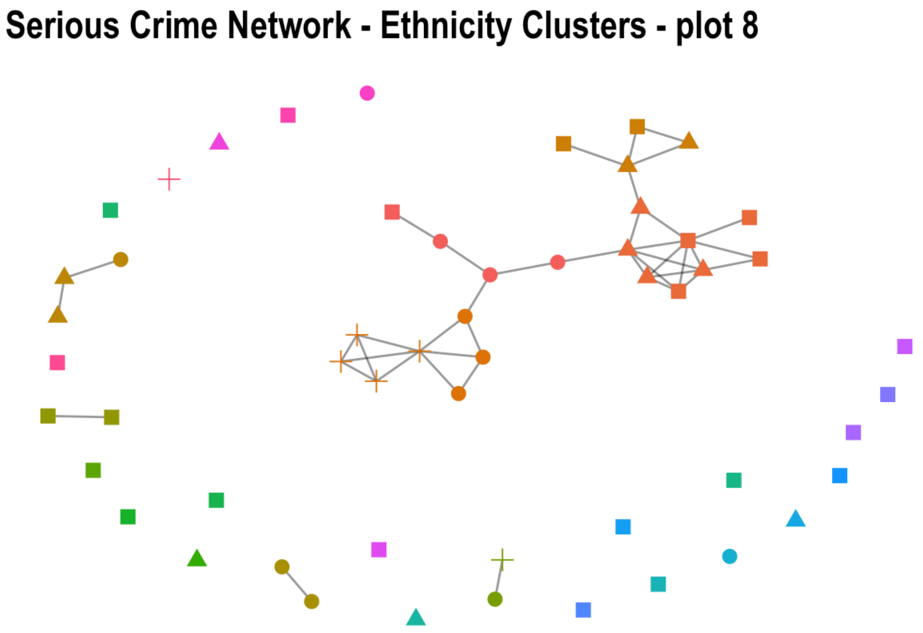
```
par(mfrow=c(2,1))

off_3.fg <- cluster_fast_greedy(off_3.graph)
member.off_3 <- membership(off_3.fg)

print(modularity(off_3.fg))
```

```
## [1] 0.6701368
```

```
table(member.off_3)
```

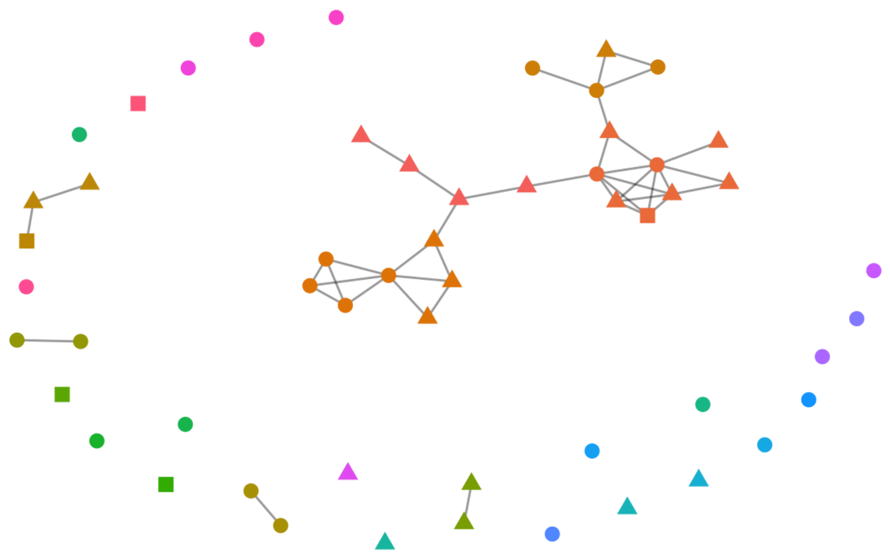
```
## member.off_3
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 
##  4  8  7  4  3  2  2  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 
## 26 27 28 29 30 
##  1  1  1  1  1
```

```
Cluster <- factor(member.off_3)

g3_cluster_A <- ggraph(off_3.graph,
     layout = 'kk')+

     geom_edge_fan(alpha = .4) +
     geom_node_point(aes(shape = as.factor(attributes$Birthplace),
                         color = Cluster),
                         show.legend = FALSE, size = 3) +
     labs(shape = "Ethnicity")+
     ggtitle("Serious Crime Network - Ethnicity Clusters - plot 8")+
     theme_graph()

g3_cluster_B <- ggraph(off_3.graph,
     layout = 'kk')+

     geom_edge_fan(alpha = .4) +
     geom_node_point(aes(shape = as.factor(attributes$Age_group),
                         color = Cluster),
                         show.legend = FALSE, size = 3) +
     labs(shape = "Age group")+
     ggtitle("Serious Crime Network - Age Clusters - plot 9")+
     theme_graph()

g3_cluster_A
```



```
g3_cluster_B
```

Serious Crime Network - Age Clusters - plot 9



Note: in plots 8 and 9, we use the same shape labels as in plots 3 and 4 for ethnicity and age, respectively, whilst node colours refer to cluster membership. From the plots above, we make the following considerations (ignoring the 1-node and 2-node clusters and focusing on the community structures obtained):

1. Four main clusters emerge;
2. Plot 8: we notice how 4 out of 6 individuals of East African ethnicity (plus labels) aggregate in a very densely connected community structure which includes also 3 individuals of West African origins. More specifically, the sub-community of these 4 East African individuals forms a complete graph of 4 vertices in the network. This points towards a high level of ethnic cohesion within the Serious Crime Network, with respect to members with East African orgins. This feature is very important for analysing the dynamics of how crime acts are carried out by these individuals; in fact, partial information about members of this clique allows to perfectly reconstruct the actions of all 4 gang members, with respect to serious crime. In addition, we notice that the majority of individuals of British orgigns (squares) are isolates in this network, and very few connections between individuals of British origins are observed; on the other hand, the majority of the ties in this network occur between individuals belonging to minority groups (West Africa, Caribbean, East Africa), indicating that serious crime acts are more likely to be carried out by individuals of the same ethnicity, particularly in the case of minority ethnic groups (this is aligned with our previous observations);
3. Plot 9: the majority of individuals belonging to age group 2 (triangles) are located on the main connected component of the network, indicating some degree of cohesion with repect to age as well. Also, we notice how the previously mentioned clique of individuals of East African origins is also preserved under age group structure. This indicates the existence of very strong interpersonal bonds between the 4 individuals in question;
4. We observe that modularity in the network is approximately 0.7, indicating rather solid community structures being present;
5. Two communities display an interesting feature: there exists a central node which is in a structural hole between symmelian ties (‘ties that torture’); these individuals could be arguably seen as heads of the respective communities they are found in within the Serious Crime Network, occupying a position of control with respect to the other nodes in the respective communities;
6. It is much easier to analyse cluster structure and extract meaningful information about the gang from the Serious Crime Network, compared to the Hang-Out Network. This is due to fewer ties appearing and more structure being present with respect to the considered network attributes.

In this section, we focus on the previously introduced networks (Hang-out and Serious Crime Networks) and test three Conditional Uniform Graph (CUG) tests using an Erdös-Rényi model with the same (expected) density.

```r
# random allocation of nodes to ethnic and age groups
nodes <- seq(1,54)

# sampling without replacement from nodes 12 1's (representing)
# the west african population, 12 2's (caribbean), 24 3's (UK)
# 6 4's (east africa).
et_nums <- sample(as.vector(rep(1:4,c(12,12,24,6))))

# sampling without replacement from nodes 28 1's representing the people
# in age group 1, 21 2's representing age group 2, 5 3's representing age group 3
age_nums <- sample(as.vector(rep(1:3,c(28,21,5))))

# grouping indices of nodes based on ethnicity and age ('p' stand for 'people')
west_africa_p <- which(et_nums == 1)
caribbean_p <- which(et_nums == 2)
uk_p <- which(et_nums == 3)
east_africa_p <- which(et_nums == 4)

AG1 <- which(age_nums == 1)
AG2 <- which(age_nums == 2)
AG3 <- which(age_nums == 3)
```

```r
# functions returning statistics in generated random graphs

# INPUT: number of nodes, number of edges
# in the Hang-Out Network (preserved expected density)
# OUTPUT: number of isolates in the generated random graph

# The function creates a random Erdös-Renyi graph with fixed density
# and extracts the adjacency matrix from it
# We then count the total number of existing isolates

numisolates <- function(n,m){
  graph1 <- sample_gnm(n, m, directed = FALSE, loops = FALSE)
  adjmat <- as_adjacency_matrix(graph1)
  adjmat <- as.matrix(adjmat)
  count = 0
  for (i in 1:n){
    if (sum(adjmat[i,]) == 0){
      count = count + 1
    }
  }
 return(count)
}

# INPUT: number of nodes, number of edges in Hang-Out Network,
# the set of nodes having ethnicity west african, set of nodes with
# ethnicity caribbean, ethnicity UK and ethnicity east african
# OUTPUT: ratio of same-ethnicity ties in the generated random graph

# The function creates a random Erdös-Renyi graph with fixed density
# and extracts the adjacency matrix from it
# We then check for each ethnicity how many same-ethnicity ties are encoded
# in this adjacency matrix and sum them up

same_et_ties <- function(n,m, west_africa_p, caribbean_p, uk_p, east_africa_p){
  graph1 <- sample_gnm(n, m, directed = FALSE, loops = FALSE)
  adjmat <- as_adjacency_matrix(graph1)
  adjmat <- as.matrix(adjmat)
  count = 0
  twototal = 2*m
  for (i in west_africa_p){
    for(j in west_africa_p){
      if (adjmat[i,j] == 1){
        count = count+1
      }
    }
  }
  for (i in caribbean_p){
    for (j in caribbean_p){
      if(adjmat[i,j] == 1){
        count = count+1
        count = count+1
      }
    }
  }
  for(i in uk_p){
    for(j in uk_p){
      if (adjmat[i,j] == 1){
        count = count+1
      }
    }
  }
  for (i in east_africa_p){
    for(j in east_africa_p){
      if (adjmat[i,j] == 1){
        count = count + 1
      }
    }
  }
  return(count/twototal)
}

# INPUT: number of nodes, number of edges in Hang-Out Network,
# the set of nodes having age group1, age group2 and age group3.
# OUTPUT: the ratio of same-age ties in the generated random graph

# The function creates a random Erdós-Reyni graph with fixed density
# and extracts the adjacency matrix from this.
# We then check for each age group how many same-age ties are encoded
# in this adjacency matrix and sum them up.

same_age_ties <- function(n,m, AG1, AG2, AG3){
  graph1 <- sample_gnm(n, m, directed = FALSE, loops = FALSE)
  adjmat <- as_adjacency_matrix(graph1)
  adjmat <- as.matrix(adjmat)
  count = 0
  twototal = 2*m
  for (i in AG1){
    for(j in AG1){
      if (adjmat[i,j] == 1){
        count = count + 1
      }
    }
  }
  for (i in AG2){
    for(j in AG2){
      if (adjmat[i,j] == 1){
        count = count + 1
      }
    }
  }
  for (i in AG3){
    for(j in AG3){
      if (adjmat[i,j] == 1){
        count = count + 1
      }
    }
  }
  return(count/twototal)
}
```

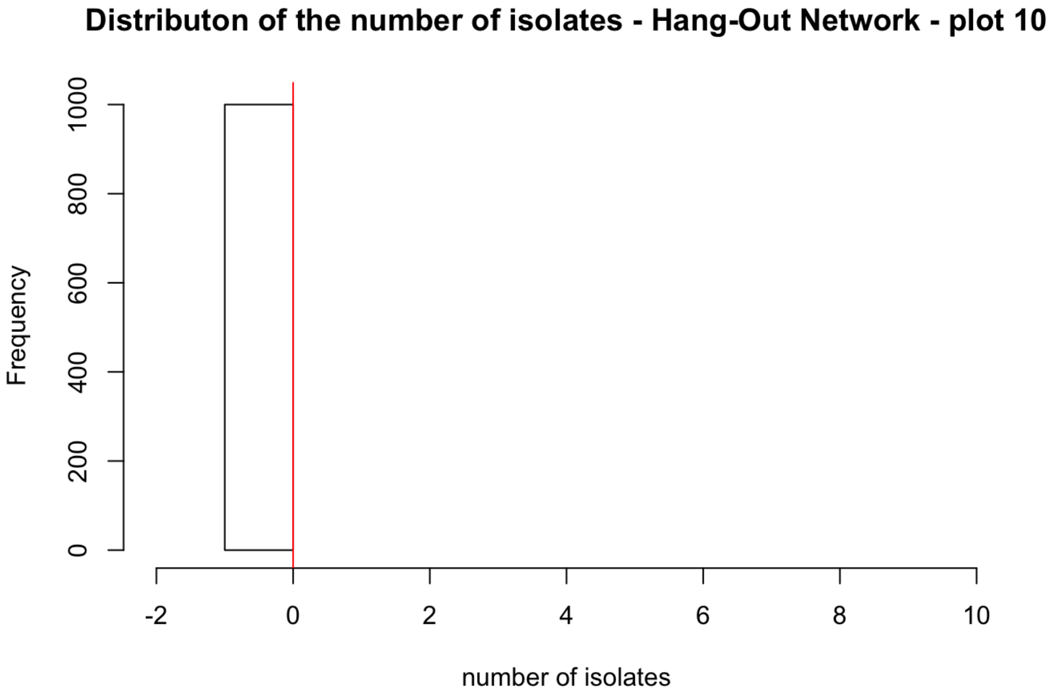
```r
# functions for performing the CUG's

# INPUT: number of nodes, number of edges in the Hang-Out Network,
# OUTPUT: a list of length 1000, where each entry corresponds
# to the number of isolates of 1 simulation.

# The function calls the above defined "numisolates" function 1000 times,
# generating 1000 random graphs, extracting their adjacency matrices,
# calculating the number of isolates and appending them to the list.

 num_iso_list <- function(n,m, n_sim){
  iso_list = c()
  for (i in 1:n_sim){
    iso_list <- append(iso_list, numisolates(n,m))
  }
  return(iso_list)
 }

# INPUT: number of nodes, number of edges in the Hang-Out Network,
# the set of nodes having ethnicity west african, set of nodes with ethnicity caribbean,
# ethnicity UK and ethnicity east african.
# OUTPUT: a list of length 1000, where each entry corresponds
# to the ratio of same-ethnicity ties of 1 simulation.

# The function calls the above defined "same_et_ties" function 1000 times,
# generating 1000 random graphs, extracting their adjacency matrices,
# calculating the ratio of same-ethnicity ties and appending them to the list.

ratio_same_et_tie_list <- function(n,m,west_africa_p, caribbean_p, uk_p, east_africa_p, n_sim){
  same_et_tie_list = c()
  for (i in 1:n_sim){
    same_et_tie_list <- append(same_et_tie_list, same_et_ties(n,m,west_africa_p, caribbean_p, uk_p, east_africa_p))
  }
  return(same_et_tie_list)
}

# INPUT: number of nodes, number of edges in the Hang-Out Network,
# the set of nodes having age group1, age group2 and age group3.
# OUTPUT: a list of length 1000, where each entry corresponds to the ratio
# of same-age ties of 1 simulation.

# The function calls the above defined "same_age_ties" function 1000 times,
# generating 1000 random graphs, extracting their adjacency matrices,
# calculating the ratio of same-age ties and appending them to the list.

ratio_same_age_tie_list <- function(n,m,AG1, AG2, AG3, n_sim){
  same_age_tie_list <- c()
  for (i in 1:n_sim){
    same_age_tie_list <- append(same_age_tie_list, same_age_ties(n,m,AG1, AG2, AG3))
  }
  return(same_age_tie_list)
}
```

We now proceed to perform the CUG tests on both networks analysed.

Hypothesis Test 1:
H0 : Not more/less isolates than expected by chance
H1 : More/less isolates than expected by chance

We start by testing the above on the Hang-out Network.

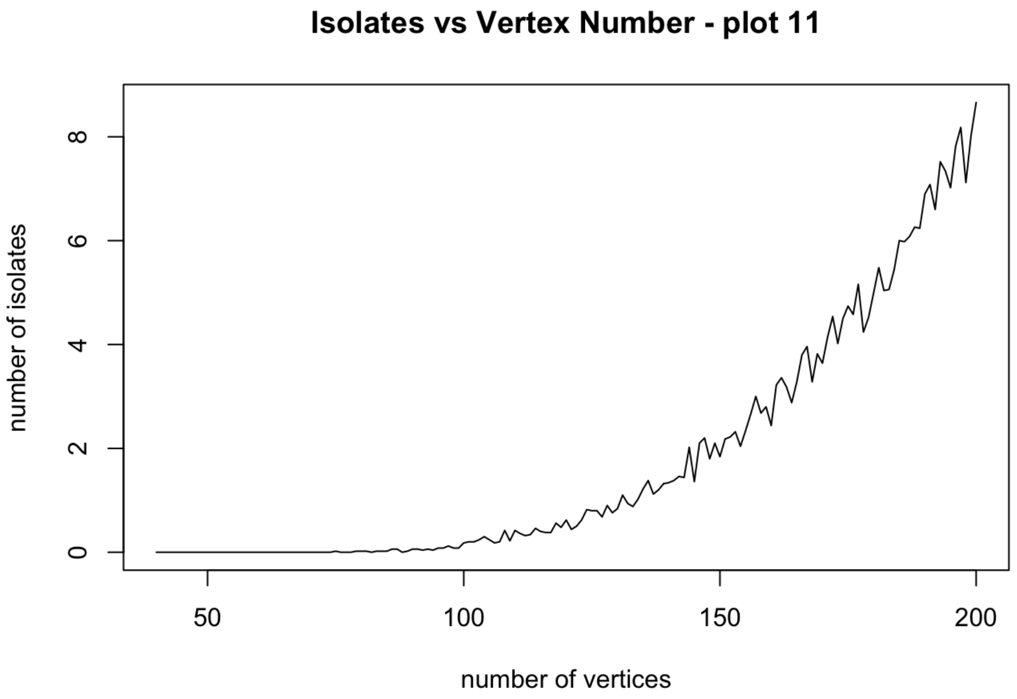
```
# simulate graphs and compute the number of isolates

n_sim_L_iso = 10^3
L_iso <- num_iso_list(network_size, number_of_edges_1, n_sim = n_sim_L_iso)

# Plotting the distribution over the number of isolates, with the red line representing the
# number of isolates in the Hang-out Network

hist(L_iso, main="Distributon of the number of isolates - Hang-Out Network - plot 10", xlab='number of isola
tes', xlim = c(-2,10), ylim = c(0,1010))
observed_iso <- isolates_1
abline(v = observed_iso, col='red')
```



```
# calculating p-value
# setting type I error = 0.05

# note: include observed value in p-value, resulting in +1 below
p_value_iso_1 <- (sum(L_iso >= observed_iso) + 1)/(n_sim_L_iso + 1)
print(paste0("p-value - isolates in Hang-Out Network: ", p_value_iso_1))
```

```
## [1] "p-value - isolates in Hang-Out Network: 1"
```

```
# plotting the number of isolates for number of nodes ranging
# from 40 to 500. n_sim set to 50 to limit computational time
# isolates start to appear in the generated graphs from around 70 nodes

isolates_num <- c()
vertices <- 40:200
for (n in vertices) {
 iso_mean <- mean(num_iso_list(n, m = number_of_edges_1, n_sim = 50))
 isolates_num <- append(isolates_num, iso_mean)
 }
plot(x = vertices, y = isolates_num, xlab = 'number of vertices', ylab = 'number of isolates', type = 'l', m
ain = 'Isolates vs Vertex Number - plot 11')
```

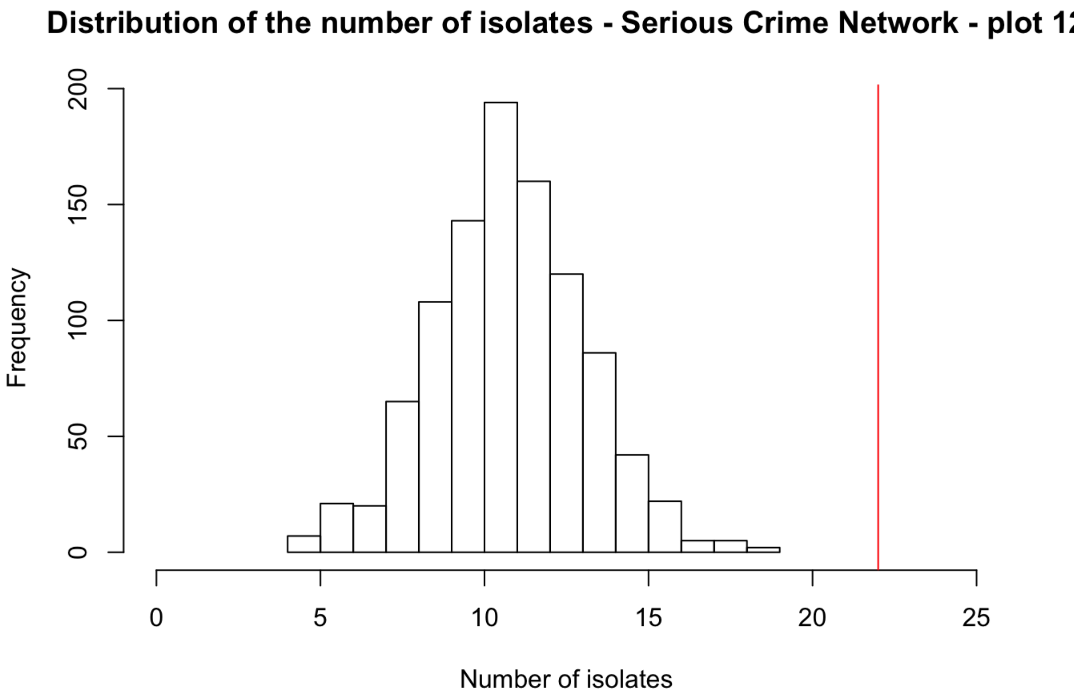


From plot 10, we make the following observations:

1. We see that the distribution of the number of isolates across simulations is entirely concentrated on zero. This suggests that for the expected graph density (graph density of Hang-Out Network), the number of edges is proportionally too large for allowing isolates in any of the above simulations, given the chosen number of vertices. We support this view with with plot 11, proposed above, where for the number of nodes ranging from 40 to 200, we average the results of 50 simulations with respect to the number of observed isolates (this number was constrained based on the computational time required for the simulations - we note that although the plot of the number of isolates as a function of the number of nodes shows as wiggly, this choice does undermine the overall interpretation of the obtained results). The plotted result suggests that isolates start appearing when the number of nodes is above 70, supporting our previous claim;
2. We obtain a p-value of 1, which under type I error being specified at 0.05, provides no evidence against the null hypothesis that not more/less isolates are expected by chance.

We complement the above with a CUG test on the Serious Crime Network, with identical null and alternative hypotheses.

```
L_iso_3 <- num_iso_list(network_size, number_of_edges_3, n_sim = n_sim_L_iso)

observed_iso_3 <- isolates_3
hist(L_iso_3, main="Distribution of the number of isolates - Serious Crime Network - plot 12", xlab='Number
of isolates', xlim = c(0,25))
abline(v = observed_iso_3, col='red')
```



```
# note: alternative hypothesis does not specify whether more or less,
# hence we compute the p-value as extreme or more as the observed

p_value_iso_3 <- (sum(L_iso_3 >= observed_iso_3) +
              sum(L_iso_3 <= (mean(L_iso_3) - (observed_iso_3 - mean(L_iso_3)))) + 1)/(n_sim_L_iso + 1)
print(paste0("p-value - isolates in Serious Crime Network: ", round(p_value_iso_3, digits = 4)))
```

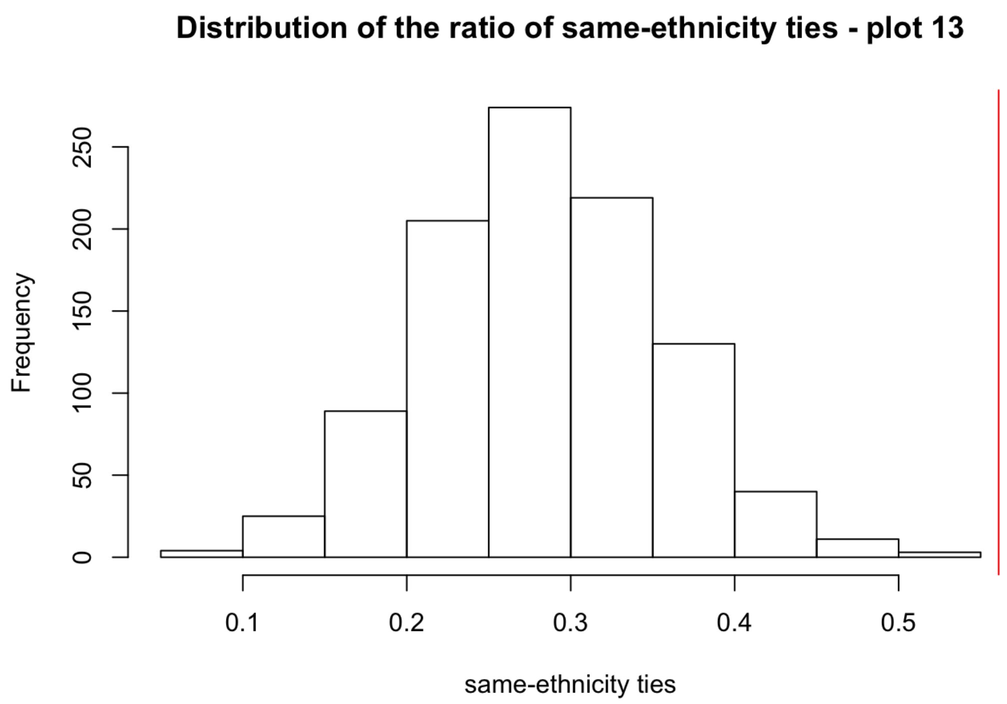
```
## [1] "p-value - isolates in Serious Crime Network: 0.001"
```

We now obtainm a p-value of 9.9900110^{-4}, which is highly significant at the specified type I error. This indicates that the number of isolates observed in the Serious Crime Network is significantly different compared to those of the simulated graphs. In particular, none of the values from the simulated graphs are even close to the observed value in the Serious Crime Network. We thus reject the null hypothesis and conclude that there are significantly more isolates in the Serious Crime Network than one would expect by chance.

We proceed to the second CUG test to compare same-ethnicity ties. For this test we choose the Serious Crime Network in light of what was noted above: there are considerably more same-ethnicity ties in the Serious Crime Network compared to the Hang-Out Network.

Hypothesis Test 2:
H0 : People of the same ethnicity are not more likely to commit serious crime together
H1 : People of the same ethnicity are more likely to commit serious crime together

```
# simulate graphs and compute ratio of same-ethnicity ties

n_sim_L_sameet = 10^3
L_sameet <- ratio_same_et_tie_list(n = network_size, m = number_of_edges_3,
              west_africa_p, caribbean_p, uk_p, east_africa_p, n_sim = n_sim_L_sameet)

#Plotting the distribution over the same-ethnicity ties,
# with the red line representing the number of same-age ties in the Hang-Out Network

hist(L_sameet, main = "Distribution of the ratio of same-ethnicity ties - plot 13", xlab = 'same-ethnicity t
ies')
abline(v = observed_et_3, col='red')
```



```
# calculating p-value
# setting type I error = 0.05

# note: here the test is one-sided, and the p-value is computed accordingly
p_value_et <- (sum(L_sameet >= observed_et_3) + 1)/(n_sim_L_sameet + 1)
print(paste0("p-value - same-ethnicity ties in Serious Crime Network: ", round(p_value_et, digits = 4)))
```

```
## [1] "p-value - same-ethnicity ties in Serious Crime Network: 0.001"
```

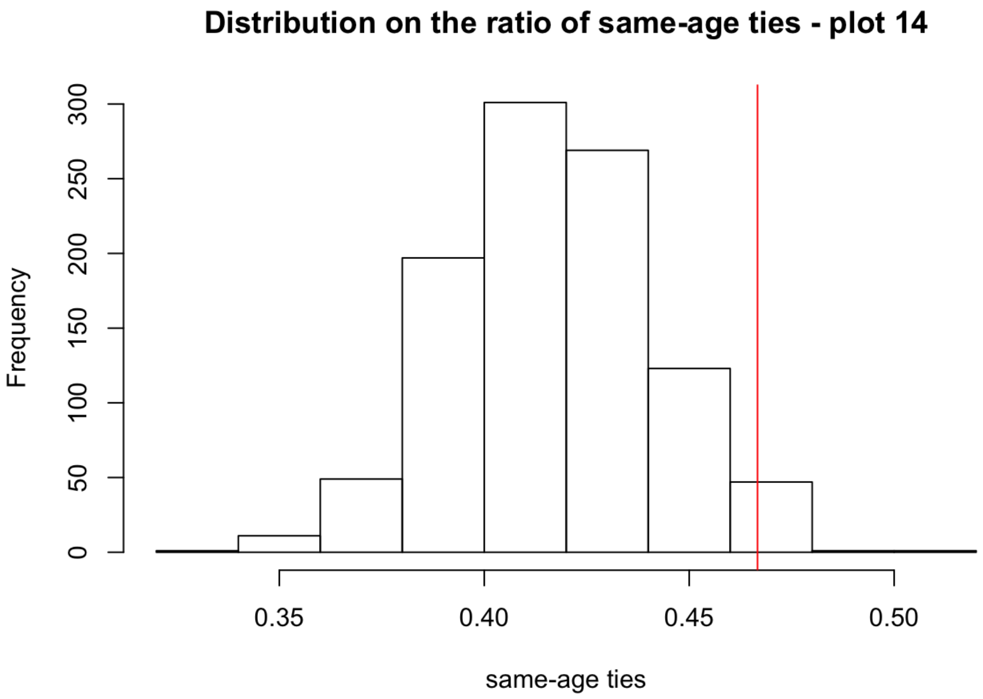
From plot 13 above, we make the following observations:

1. We see that the distribution is fairly symmetric, as expected by randomly generating the graphs;
2. The mean of the percentage of same-ethnicity ties over the total number of ties within the simulated graphs is slightly below 0.3, suggesting that on average approximately 30% of the expected number of edges are same-ethnicity ties (across the simulated networks); this is strictly lower than the number of same-ethnicity ties observed in the Serious Crime Network, which we calculated to be approximately 56% of the total number of ties in the Hang-out netwrok;
3. The number of same-ethnicity ties in the Serious Crime Network amounts to 0.5609756;
4. We obtain a p-value of 9.9900110^{-4}, which under type I error being specified at 0.05, provides strong evidence against the null hypothesis that people of the same ethnicity are not more likely to commit serious crime together.

We now proceed to the last CUG test which is conducted on the Hang-Out Network.

Hypothesis Test 3:
H0 : People of the same age group are not more likely to hang out together
H1 : People of the same age group are more likely to hang out together

```
# simulate graphs and compute ratio of same-age group ties

n_sim_L_sameage = 10^3
L_sameage <- ratio_same_age_tie_list(n = network_size, m = number_of_edges_1,
                                   AG1, AG2, AG3, n_sim = n_sim_L_sameage)

hist(L_sameage, main="Distribution on the ratio of same-age ties - plot 14", xlab='same-age ties')
abline(v = observed_age_1, col='red')
```



```
# calculating p-value
# setting type I error = 0.05

p_value_age <- (sum(L_sameage >= observed_age_1) + 1)/(n_sim_L_sameage + 1)
print(paste0("p-value - same-age ties in Hang-Out Network: ", round(p_value_age, digits = 4)))
```

```
## [1] "p-value - same-age ties in Hang-Out Network: 0.024"
```

```
#calculate the corresponding statistics from simulations
iso.mean <- mean(L_iso)
iso3.mean <- mean(L_iso_3)
sameettie.mean <- mean(L_sameet)
sameagetie.mean <- mean(L_sameage)

iso.max <- max(L_iso)
iso3.max <- max(L_iso_3)
sameettie.max <- max(L_sameet)
sameagetie.max <- max(L_sameage)

iso.min <- min(L_iso)
iso3.min <- min(L_iso_3)
sameettie.min <- min(L_sameet)
sameagetie.min <- min(L_sameage)

iso.sd <- sd(L_iso)
iso3.sd <- sd(L_iso_3)
sameettie.sd <- sd(L_sameet)
sameagetie.sd <- sd(L_sameage)
```

From plot 14 above, we make the following observations:

1. We again see that the distribution of the percentage of same-age ties across simulations is fairly symmetric, as expected;
2. The mean of the percentage of same-age ties over the total number of ties within the simulated graphs is 0.4171778, suggesting that on average approximately 40% of the expected number of edges are same-age ties (across the simulated networks). The corresponding standard deviation is 0.024355;
3. The number of same-age ties in the Hang-out Network amounts to 0.4666667;
4. We obtain a p-value of 0.023976, which under type I error being specified at 0.05, provides significant evidence against the null hypothesis that people of the same age are not more likely to hang out together. Thus, we see that the mean across the simulated graphs is significantly lower than the observed number of sameage ties in the Hang-out Network . This allows us to reject the hypothesis that people of the same age group are not more likely to hang out together.

These two latest tests indicate an increased level of homophily (overrepresentation of ties among similar people), as same-ethincity ties and same-age ties are overrepresented in comparison to the simulated networks, as shown above.

We finally report different statistics of the CUG tests conducted above, for completeness. These include mean and standard deviation, max and min values for the number of same-ehtnicity ties and the number of same-age ties, as for the above simulations.

```
print(paste0("Mean of isolated nodes - hang-out network: ", iso.mean))
```

```
## [1] "Mean of isolated nodes - hang-out network: 0"
```

```
print(paste0("Mean of isolated nodes - Serious Crime Network: ", round(iso3.mean, digits = 4)))
```

```
## [1] "Mean of isolated nodes - Serious Crime Network: 11.235"
```

```
print(paste0("Mean of same-ethnicity ties: ", round(sameettie.mean, digits = 4)))
```

```
## [1] "Mean of same-ethnicity ties: 0.2897"
```

```
print(paste0("Mean of same-age ties: ", round(sameagetie.mean, digits = 4)))
```

```
## [1] "Mean of same-age ties: 0.4172"
```

```
print(paste0("Max of isolated nodes - hang-out network: ", iso.max))
```

```
## [1] "Max of isolated nodes - hang-out network: 0"
```

```
print(paste0("Max of isolated nodes - Serious Crime Network: ", round(iso3.max, digits = 4)))
```

```
## [1] "Max of isolated nodes - Serious Crime Network: 19"
```

```
print(paste0("Max of same-ethnicity ties: ", round(sameettie.max, digits = 4)))
```

```
## [1] "Max of same-ethnicity ties: 0.5122"
```

```
print(paste0("Max of same-age ties: ", round(sameagetie.max, digits = 4)))
```

```
## [1] "Max of same-age ties: 0.5016"
```

```
print(paste0("Min of isolated nodes - hang-out network: ", iso.min))
```

```
## [1] "Min of isolated nodes - hang-out network: 0"
```

```
print(paste0("Min of isolated nodes - Serious Crime Network: ", round(iso3.min, digits = 4)))
```

```
## [1] "Min of isolated nodes - Serious Crime Network: 4"
```

```
print(paste0("Min of same-ethnicity ties: ", round(sameettie.min, digits = 4)))
```

```
## [1] "Min of same-ethnicity ties: 0.0976"
```

```
print(paste0("Min of same-age ties: ", round(sameagetie.min, digits = 4)))
```

```
## [1] "Min of same-age ties: 0.3365"
```

```
print(paste0("Standard deviation of isolated nodes - hang-out network: ", iso.sd))
```

```
## [1] "Standard deviation of isolated nodes - hang-out network: 0"
```

```
print(paste0("Standard deviation of isolated nodes - Serious Crime Network: ", round(iso3.sd, digits = 4)))
```

```
## [1] "Standard deviation of isolated nodes - Serious Crime Network: 2.3258"
```

```
print(paste0("Standard deviation of same-ethnicity ties: ", round(sameettie.sd, digits = 4)))
```

```
## [1] "Standard deviation of same-ethnicity ties: 0.0703"
```

```
print(paste0("Standard deviation of same-age ties: ", round(sameagetie.sd, digits = 4)))
```

```
## [1] "Standard deviation of same-age ties: 0.0244"
```

In this section, we propose a function that uniformly generates random networks with a prescribed number of nodes, number of ties and number of isolates. After introducing the function, we will conduct and document a number of tests. The motivation is to then perform CUG tests like above which also conditions on the number of isolates in the network.

```
# Inputs to the function:
# number of nodes, edges and isolates in the
# desired class of networks we want to uniformly sample from

sample_gnm_constant_isolates <- function(nodes, ties, isolates){
  n <- nodes
  m <- ties
  p <- isolates

  # impose conditions on input
  # assert that n >= 1 and p <= n as well as that p == n in the case that we have no edges
  if (n < 1 | p > n | (m == 0 & p != n)) {
    stop("Invalid input")
  }

  # case 1: generated graphs must must have 0 edges
  if (m == 0) {
    graph <- array(0L, dim = c(n,n))
    graph.igraph <- graph_from_adjacency_matrix(graph, mode = "undirected", diag = F,
                                              add.colnames = NA, add.rownames = NA)
    return(graph.igraph)
  }

  # impose conditions on inputs
  # assert that m <= (n-p)*(n-p-1)/2 && m >= (n-p)/2
  else if (m > (n-p)*(n-p-1)/2 | m < (n-p)/2) {
    stop("Invalid input")}

  # all other cases
  else {

    graph <- array(0L, dim = c(n,n))

    # case 2: generated graphs must have 0 isolated nodes
    if(p == 0){
      for (i in seq(m)) {
      not.finished <- T
        while (not.finished) {

          # randomly sample positions of edges to be inserted
          i <- rdunif(1, 1, n)
          j <- rdunif(1, 1, n)
          if (i != j & graph[i,j] != 1) {
            graph[i,j] <- 1
            graph[j,i] <- 1
            not.finished <- F
          }
        }
      }

      # correct if isolates are created in the step above
```

```r
  # correct if isolates are created in the step above
  # by making sure no row/column of adjacency matrix
  # contains only 0-entries
  while(min(colMaxs(graph, value = T)) < 1) {
   i <- rdunif(1, 1, n)
   j <- rdunif(1, 1, n)
   if (graph[i,j] == 1) {
    graph[i,j] <- 0
    graph[j,i] <- 0
    not.fixed <- T
    while(not.fixed) {
     k <- rdunif(1, 1, n)
     l <- rdunif(1, 1, n)
     if (k != l & max(graph[k,]) == 0) {
      graph[k,l] <- 1
      graph[l,k] <- 1
      not.fixed <- F
     }
    }
   }
  }
 }
 # case 3: generated graphs must have at least one isolate, at least one edge
 # choose isolates uniformly at random from nodes
 else {
  # generate isolate nodes
  isolates <- rep(0, p)
  for (i in seq(length(isolates))) {
   while (isolates[i] == 0) {
    iso <- rdunif(1, 1, n)
     if (!(iso %in% isolates)) {
      isolates[i] <- iso
     }
    }
   }
   # fill graph randomly except for isolate positions
   for (i in seq(m)) {
    not.finished <- T
    while (not.finished) {
     i <- rdunif(1, 1, n)
     j <- rdunif(1, 1, n)
     if (!(i %in% isolates) & !(j %in% isolates) & i != j & graph[i,j] != 1) {
      graph[i,j] <- 1
      graph[j,i] <- 1
      not.finished <- F
     }
    }
   }
  # fix if there are more isolates than necessary
   while(min(colMaxs(graph[-isolates,-isolates], value = T)) < 1) {
    i <- rdunif(1, 1, n)
    j <- rdunif(1, 1, n)
    if (graph[i,j] == 1) {
     graph[i,j] <- 0
     graph[j,i] <- 0
     not.fixed <- T
     while(not.fixed) {
      k <- rdunif(1, 1, n)
      l <- rdunif(1, 1, n)
      if (!(k %in% isolates) & !(l %in% isolates) & k != l & max(graph[k,]) == 0) {
       graph[k,l] <- 1
       graph[l,k] <- 1
       not.fixed <- F
      }
     }
    }
   }
  }
}
# produce and return igraph object as output
graph.igraph <- graph_from_adjacency_matrix(graph, mode = "undirected", diag = F,
                                            add.colnames = NA, add.rownames = NA)
return (graph.igraph)
```

```
}
```

We now proceed by testing the above function on a set of instances, in order to empirically evalute correct execution of the administered tasks.

We start by testing the function invalid inputs. We recall these cases below.

```
# # impose conditions on input
#   # assert that n >= 1 and p <= n as well as that p == n in the case that we have no edges
#   if (n < 1 | p > n | (m == 0 & p != n)) {
#     stop("Invalid input")
#   }
#
# # impose conditions on inputs
#   # assert that m <= (n-p)*(n-p-1)/2 && m >= (n-p)/2
#   else if (m > (n-p)*(n-p-1)/2 | m < (n-p)/2) {
#     stop("Invalid input")}
```

We set the number of nodes to be 10 and the number of isolates to be 2, arbitrarily and choose the number of ties explicitly, so that the conditions specified above are violated.

```
# test 1
# commented out to allow knitting ouput - uncomment to verify
# # test 1: n < 1
#
# n <- 0
# m <- 0
# p <- 0
# sample_gnm_constant_isolates(nodes = n, ties = m, isolates = p)
```

```
# test 2
# commented out to allow knitting ouput - uncomment to verify
# # test 2: p > n
#
# n <- 10
# p <- 11
# m <- 15
# sample_gnm_constant_isolates(nodes = n, ties = m, isolates = p)
```

```
# test 3
# commented out to allow knitting ouput - uncomment to verify
# # test 3: m == 0 & p < n
#
# n <- 10
# p <- 5
# m <- 0
# sample_gnm_constant_isolates(nodes = n, ties = m, isolates = p)
```

```
# test 4
# commented out to allow knitting ouput - uncomment to verify
# # test 4 - m > (n-p)*(n-p-1)/2
#
# n <- 10
# p <- 2
# m <- (n-p)*(n-p-1)/2 + 1
# sample_gnm_constant_isolates(nodes = n, ties = m, isolates = p)
```

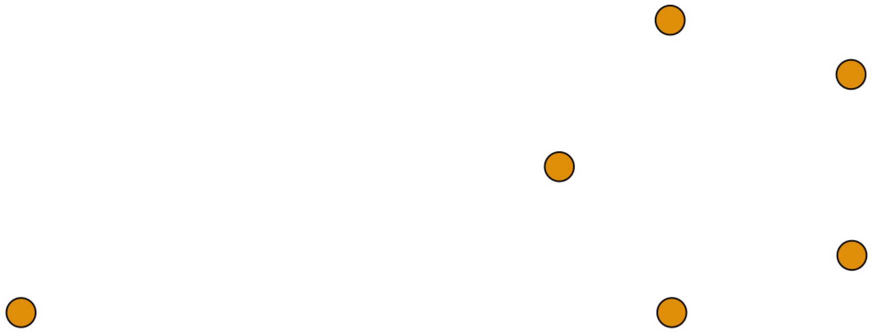
```
# test 5
# commented out to allow knitting ouput - uncomment to verify
# # test 5 - m < (n-p)/2
#
# n <- 10
# p <- 2
# m <- (n-p)/2 - 1
# sample_gnm_constant_isolates(nodes = n, ties = m, isolates = p)
```

One can verify the the printed output in all the instances above is given by the following.

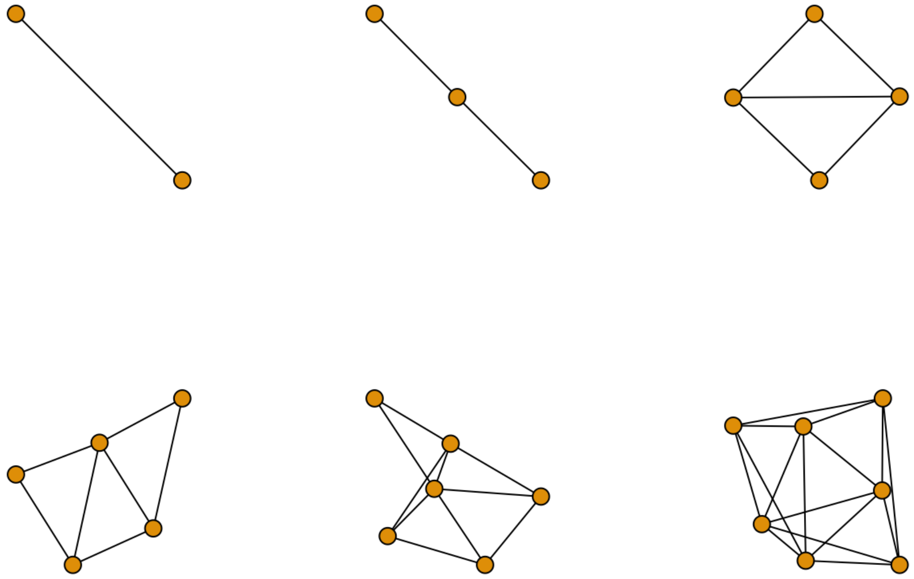
```
# Error in sample_gnm_constant_isolates(nodes = n, ties = m, isolates = p) : Invalid input
```

The code boxes above have all been commented out in order to allow RStudio to correctly knit the output file. We now focus on cases 1,2 and 3 as indicated in the script of the function, and for simple instances (easily computable by inspection), we verify that the above function returns correct outputs.

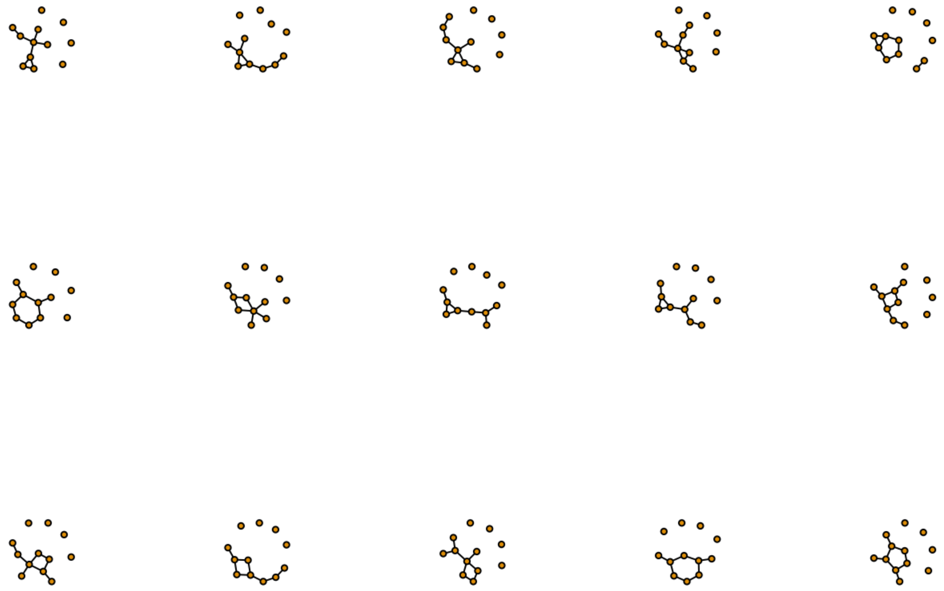
```
# test 6: generated graphs must have 0 edges

par(mfrow=c(1,2))

g1 <- sample_gnm_constant_isolates(nodes = 1, ties = 0, isolates = 1)
g2 <- sample_gnm_constant_isolates(nodes = 5, ties = 0, isolates = 5)

plot(g1, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g2, vertex.label = NA, vertex.size = 20, edge.color = 'black')
```



```
# test 7: generated graphs must have 0 isolated nodes

par(mfrow=c(2,3))

g1 <- sample_gnm_constant_isolates(nodes = 2, ties = 1, isolates = 0)
g2 <- sample_gnm_constant_isolates(nodes = 3, ties = 2, isolates = 0)
g3 <- sample_gnm_constant_isolates(nodes = 4, ties = 5, isolates = 0)
g4 <- sample_gnm_constant_isolates(nodes = 5, ties = 7, isolates = 0)
g5 <- sample_gnm_constant_isolates(nodes = 6, ties = 10, isolates = 0)
g6 <- sample_gnm_constant_isolates(nodes = 7, ties = 16, isolates = 0)

plot(g1, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g2, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g3, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g4, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g5, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g6, vertex.label = NA, vertex.size = 20, edge.color = 'black')
```



```
# test 8: generated graphs must have at least 1 isolate, at least 1 edge
# verify sampling is uniform with the chosen class of graphs
# random sampling from class of graphs with 12 nodes, 8 ties and 4 isolates

par(mfrow=c(3,5))

g1 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g2 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g3 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g4 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g5 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g6 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g7 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g8 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g9 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g10 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g11 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g12 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g13 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g14 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)
g15 <- sample_gnm_constant_isolates(nodes = 12, ties = 8, isolates = 4)

plot(g1, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g2, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g3, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g4, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g5, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g6, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g7, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g8, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g9, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g10, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g11, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g12, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g13, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g14, vertex.label = NA, vertex.size = 20, edge.color = 'black')
plot(g15, vertex.label = NA, vertex.size = 20, edge.color = 'black')
```



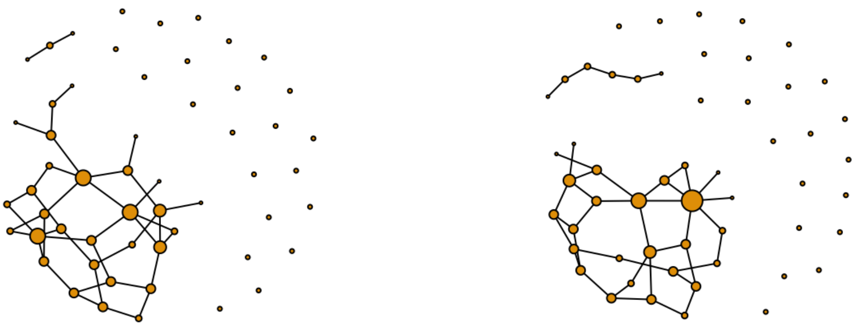
From the tests conducted above we make the following observations:

1. From tests 1 to 5, invalid inputs are correctly identified by the function, confirming its correctness with respect to such instances. We also highlight the fact that values for n and p were chosen arbitrarily and this choice does not in any way influence the validity of the above conclusions. We also note that the invalid input captured by the function is not exhaustive, it does not for instance capture non-integer values;
2. The cases for graphs consisting of only isolates are corrrectly identified by the function, as seen in test 6;
3. Cases 2 and 3 as given in the script are all correctly identified by the function in the above tests, where respectively, graphs with no isolates and graphs with at least one isolate and at least one edge are all correcly identified. As already mentioned, these tests were all conducted on easily computable cases, which can be inspected by eye;
4. We can see from test 8 that a large variety of possible graphs with the specified characteristics is sampled across 15 trials conducted. This confirms the validity of the above function with respect to uniform random sampling from graph classes.

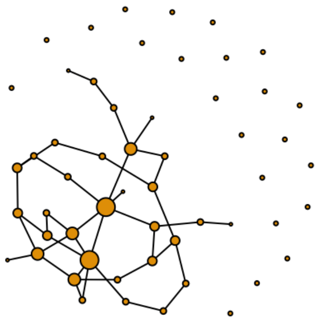
Examples from the original Serious Crime Network

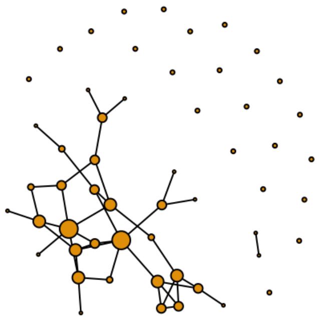
Here, we focus on plotting six examples with respect to the Serious Crime Network.

```r
# generating graphs according to the parameter values of the Serious Crime Network with 54 nodes, 41 ties and 22 isolate nodes
# plotting with nodes size given by degree
# from left to right:
# row 1: graphs 1, 2
# row 2: graphs 3, 4
# row 3: graphs 5, 6

par(mfrow=c(1,2))

orig_1 <- sample_gnm_constant_isolates(nodes = 54, ties = 41, isolates = 22)
orig_2 <- sample_gnm_constant_isolates(nodes = 54, ties = 41, isolates = 22)
orig_3 <- sample_gnm_constant_isolates(nodes = 54, ties = 41, isolates = 22)
orig_4 <- sample_gnm_constant_isolates(nodes = 54, ties = 41, isolates = 22)
orig_5 <- sample_gnm_constant_isolates(nodes = 54, ties = 41, isolates = 22)
orig_6 <- sample_gnm_constant_isolates(nodes = 54, ties = 41, isolates = 22)

mat_orig_1 <- as.matrix(as_adjacency_matrix(orig_1))
mat_orig_2 <- as.matrix(as_adjacency_matrix(orig_2))
mat_orig_3 <- as.matrix(as_adjacency_matrix(orig_3))
mat_orig_4 <- as.matrix(as_adjacency_matrix(orig_4))
mat_orig_5 <- as.matrix(as_adjacency_matrix(orig_5))
mat_orig_6 <- as.matrix(as_adjacency_matrix(orig_6))

degree_1 <- sna::degree(mat_orig_1, gmode = "graph")
degree_2 <- sna::degree(mat_orig_2, gmode = "graph")
degree_3 <- sna::degree(mat_orig_3, gmode = "graph")
degree_4 <- sna::degree(mat_orig_4, gmode = "graph")
degree_5 <- sna::degree(mat_orig_5, gmode = "graph")
degree_6 <- sna::degree(mat_orig_6, gmode = "graph")

plot(orig_1, vertex.label = NA, edge.color = 'black', vertex.size = degree_1*2)
plot(orig_2, vertex.label = NA, edge.color = 'black', vertex.size = degree_2*2)
```

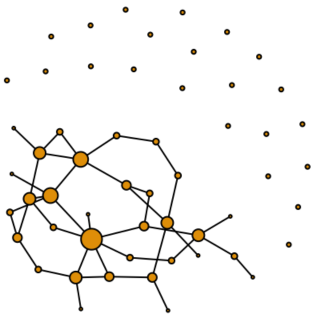


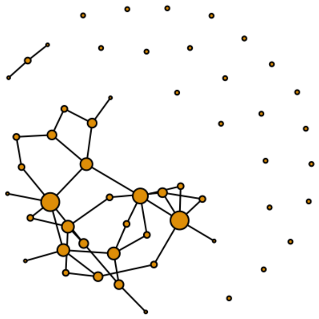
```r
plot(orig_3, vertex.label = NA, edge.color = 'black', vertex.size = degree_3*2)
plot(orig_4, vertex.label = NA, edge.color = 'black', vertex.size = degree_4*2)
```





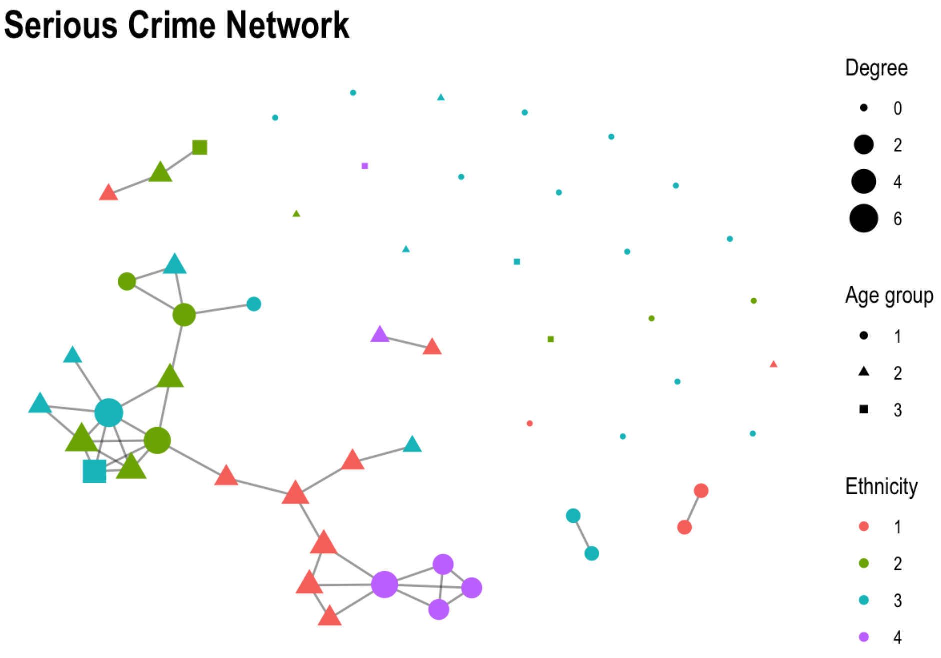
```
plot(orig_5, vertex.label = NA, edge.color = 'black', vertex.size = degree_5*2)
plot(orig_6, vertex.label = NA, edge.color = 'black', vertex.size = degree_6*2)
```





We also plot the Serious Crime Network again for ease of comparison (note: here the size of nodes is given by degree, rather than by arrests as in previous sections).

```
off_3.graph <- graph_from_adjacency_matrix(off_3,
                          mode = "undirected",
                          diag = FALSE
                          )

g_3 <- ggraph(off_3.graph,
          layout = 'nicely')+
          geom_edge_fan(alpha = .4) +
          geom_node_point(aes(shape = as.factor(attributes$Age_group),
                              colour = as.factor(attributes$Birthplace),
                              size = attributes$degree_3))+
          labs(colour = "Ethnicity",
          size = "Degree",
          shape = "Age group")+
          ggtitle("Serious Crime Network")+
          theme_graph()

g_3
```



By comparing the original Serious Crime Network with the above plots, we make the following observations:

1. The above generated networks overall mimic well the structural properties of the Serious Crime Network. In particular, we see how all such graphs present a main connected component where community-like structures are formed, surrounded by much smaller connected components, just as in the originial graph;
2. In both the original network and the simulated ones, there exist a restricted number of high degree nodes, situated in the core nucleus of community-like structures within the networks. This is a strongly desirable feature to have in a model used to simulate random graphs as the above function with respect to analysing data sets such as the Serious Crime Network;
3. A key difference between the randomly generated networks and the original Serious Crime Network is the perceived level of modularity. In the Serious Crime Network the nodes are quite clearly divisible in at least two clusters within the core nucleus. On the other hand, the randomly sampled examples above exhibit more sparsity within wider communities (e.g. one can observe that no cliques on more than 3 vertices and 'ties that torture' exist in these graphs);
4. Another similarity is the presence of tree-like structures in both the sampled graphs and the original graph;
5. We conclude by saying that on average, the graphs generated via the above function present a reasonable degree of similarity with the original network. However, we note that all the above considerations are made based on visually comparing the graphs. Ideally, a more in-depth analysis of the similarity between the randomly sampled graphs and the original one should be conducted, for theoretical guarantees on the above function.

In this section we conduct two CUG tests, mimicking those performed on same-ethnicity ties and same-age group ties performed above, but keeping the number of isolated nodes constant.

The first test is performed on the Serious Crime Network, as before. We re-iterate the hypotheses:

H0 : People of the same ethnicity are not more likely to commit serious crime together
H1 : People of the same ethnicity are more likely to commit serious crime together

```r
# First re-write function from above, to simulate networks
# conditioned on a constant number of isolates
# Subsequently, compute the proportion of same-ethnicity ties in simulations
# p denotes the prescribed number of isolates

same_et_ties_iso <- function(n, m, p, west_africa_p, caribbean_p, uk_p, east_africa_p){
  graph1 <- sample_gnm_constant_isolates(nodes = n, ties = m, isolates = p)
  adjmat <- as_adjacency_matrix(graph1)
  adjmat <- as.matrix(adjmat)
  count = 0
  twototal = 2*m
  for (i in west_africa_p){
    for(j in west_africa_p){
      if (adjmat[i,j] == 1){
        count = count+1
      }
    }
  }
  for (i in caribbean_p){
    for (j in caribbean_p){
      if(adjmat[i,j] == 1){
        count = count+1
      }
    }
  }
  for(i in uk_p){
    for(j in uk_p){
      if (adjmat[i,j] == 1){
        count = count+1
      }
    }
  }
  for (i in east_africa_p){
    for(j in east_africa_p){
      if (adjmat[i,j] == 1){
        count = count + 1
      }
    }
  }
  return(count/twototal)
}
```

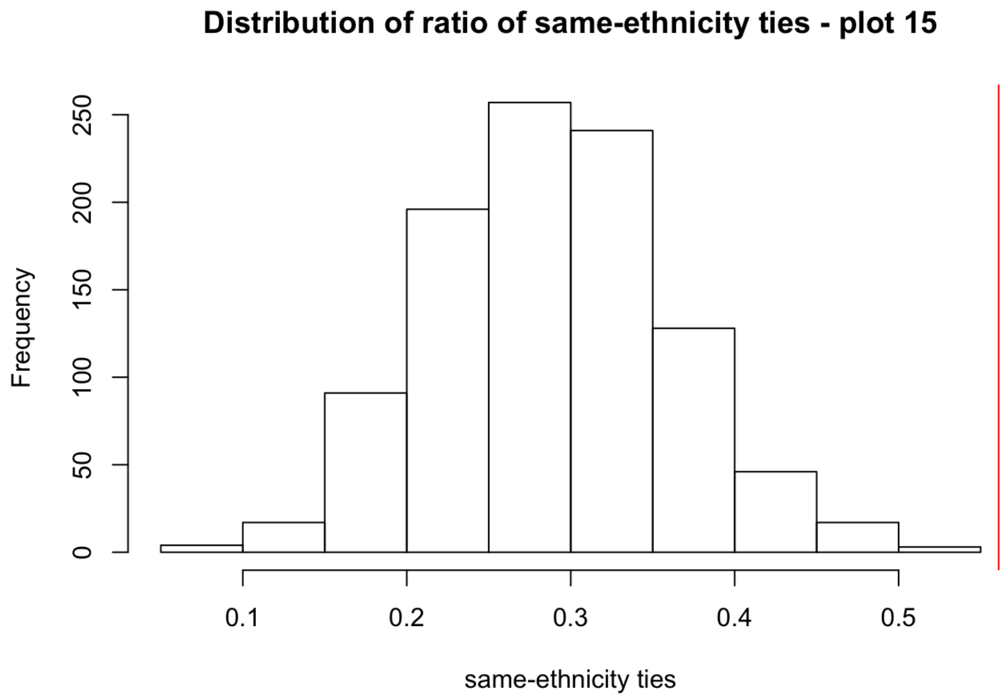
```r
# Recall the parameters of the Serious Crime Network: 54 nodes, 41 ties and 22 isolates
# Ethnicity compositions as computed previously
# running simulations and testing hypothesis

n_sim <- 1000
same_ethnicity_prop <- c()
for (i in seq(n_sim)) {
  et_p <- same_et_ties_iso(54, 41, 22, west_africa_p, caribbean_p, uk_p, east_africa_p)
  same_ethnicity_prop <- append(same_ethnicity_prop, et_p)
}

print(paste0("Average of proportion of same-ethnicity ties: ", round(mean(same_ethnicity_prop), digits = 4))
)
```

```
## [1] "Average of proportion of same-ethnicity ties: 0.2943"
```

```r
print(paste0("Standard deviation of proportion of same-ethnicity ties: ", round(sd(same_ethnicity_prop), dig
its = 4)))
```

```
## [1] "Standard deviation of proportion of same-ethnicity ties: 0.0719"
```

```r
hist(same_ethnicity_prop, main = "Distribution of ratio of same-ethnicity ties - plot 15", xlab='same-ethnic
ity ties')
abline(v = observed_et_3, col='red')
```



```r
# calculating p-value
# setting type I error = 0.05

# note: here the test is one-sided, and the p-value is computed accordingly
p_value_et_const_iso <- (sum(same_ethnicity_prop >= observed_et_3) + 1)/(n_sim + 1)
print(paste0("p-value - same-ethnicity ties for Serious Crime Network: ", round(p_value_et_const_iso, digits
=4)))
```

```
## [1] "p-value - same-ethnicity ties for Serious Crime Network: 0.001"
```

The observed value is 9.9900110^{-4}, which at the chosen type I error is highly significant. We thus reject the null hypothesis in favour of the alternative hypothesis and conclude that there is significant evidence that same ethinicity members are more likely to commit serious crime together.

The histogram of simulated same-ethnicity ties is very similar to the one from above without conditioning on isolated nodes, in terms of both location and scale. The added conditioning on number of isolated nodes (which as seen before differed significantly in the Serious Crime Network compared to random Erdös-Rényi networks with the same number of nodes and edges) does not seem to add any valuable information to test this particular hypothesis. This test does however increase the validity of results obtained above, since replicating more of the features of the original network in the CUG produces a very similar result as before.

We now proceed to the second CUG test which is performed on the Hang-Out Network. This CUG test is expected to produce nearly identical results to the counterpart obtained above, as the parameters of the Hang-Out Network are extremely unlikely to produce isolated nodes in randomly generated Erdös-Rényi networks (as was also seen above). Conditioning on isolated nodes being equal to 0 will then make no difference. This additional CUG test is thus mainly performed to confirm that this is in fact the case.

We recall the hypotheses:

Tested hypotheses:
H0 : People of the same age group are not more likely to hang out together
H1 : People of the same age group are more likely to hang out together

```
# First re-write the function from 2a) to compute the proportion of same-age group ties from a simulated net
work conditioned on number of isolates

same_age_ties_iso <- function(n, m, p, AG1, AG2, AG3){
  graph1 <- sample_gnm_constant_isolates(nodes = n, ties = m, isolates = p)
  adjmat <- as_adjacency_matrix(graph1)
  adjmat <- as.matrix(adjmat)
  count = 0
  twototal = 2*m
  for (i in AG1){
    for(j in AG1){
      if (adjmat[i,j] == 1){
        count = count + 1
      }
    }
  }
  for (i in AG2){
    for(j in AG2){
      if (adjmat[i,j] == 1){
        count = count + 1
      }
    }
  }
  for (i in AG3){
    for(j in AG3){
      if (adjmat[i,j] == 1){
        count = count + 1
      }
    }
  }
  return(count/twototal)
}
```

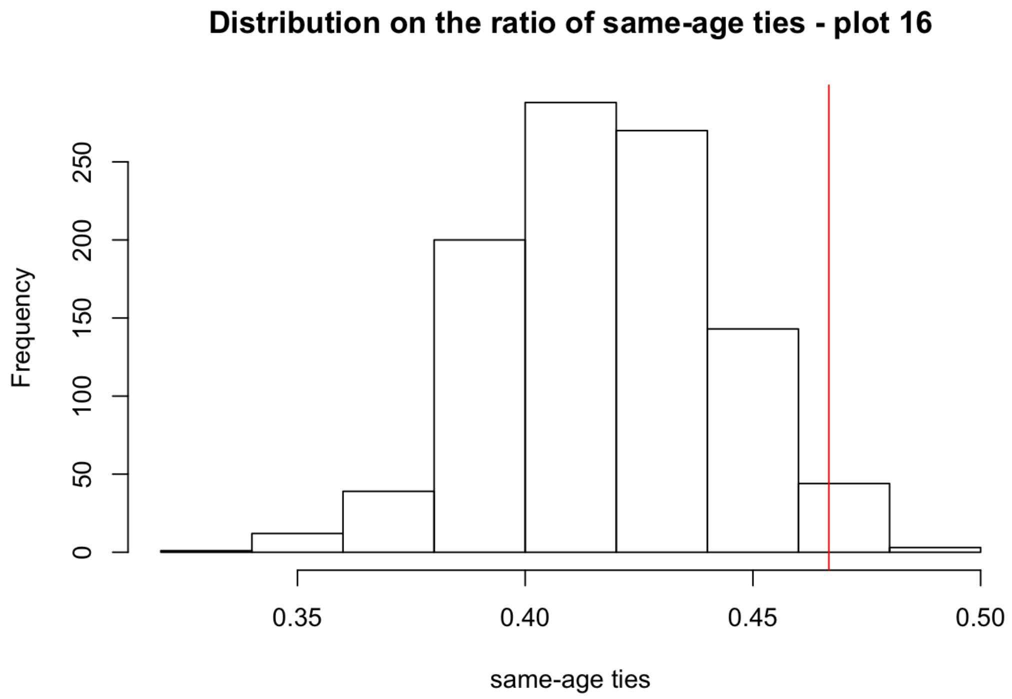
```
# Now perform CUG test on Hang-Out Network
# Parameters: 54 nodes, 315 edges and 0 isolates
# Age group compositions as computed previously

n_sim <- 1000
same_age_prop <- c()
for (i in seq(n_sim)){
  age_p <- same_age_ties_iso(54, 315, 0, AG1, AG2, AG3)
  same_age_prop <- append(same_age_prop, age_p)
}

print(paste0("Average value of proportion of same-age group ties: ", round(mean(same_age_prop), digits = 4))
)
```

```
## [1] "Average value of proportion of same-age group ties: 0.4176"
```

```
print(paste0("Standard deviation of proportion of same-age group ties: ", round(sd(same_age_prop), digits =
4)))
```

```
## [1] "Standard deviation of proportion of same-age group ties: 0.0244"
```

```
hist(same_age_prop, main="Distribution on the ratio of same-age ties - plot 16", xlab='same-age ties')
abline(v = observed_age_1, col='red')
```



```
# calculating p-value
# setting type I error = 0.05

p_value_age_const_iso <- (sum(same_age_prop >= observed_age_1) + 1)/(n_sim + 1)
print(paste0("p-value - same-age ties: ", round(p_value_age_const_iso, digits = 4)))
```

```
## [1] "p-value - same-age ties: 0.029"
```

As seen in the output above the obtained p-value is 0.029, which is highly significant at the chosen type I error. Thus, we reject the null hypothesis. This provides evidence for the claim that gang members of the same age group are more likely to hang out together. The p-value is not identical to the one obtained above however, which is due to the randomness associated with simulating 1000 networks. If we instead compare the mean and standard deviations of the empirical distributions we see very similar values. The mean and standard deviation in the standard CUG test were 0.4171778 and 0.024355 respectively. In the CUG test conditioned on number of isolates they are 0.4176 and 0.0244. We can then (qualitatively) conclude that both samples come from the same distribution, as was argued above. In conclusion, the outcomes of our hypothesis tests in this section match those of the above one, with slight variations in the numercal expressions for p-values, as expected by randomness. This validates our previous conclusions, as well as the quality of our function for generating random networks with prescribed structural properties.