Social Network Analysis of Monkey Grooming

Agnes Coaker

02/06/2022

Contents

Introduction

In order for a monkey sanctuary to better understand the social structure of squirrel monkeys, they observed the grooming behaviour of a large number of monkeys in the sanctuary for a week-long period. They recorded a large number of interactions including grooming, and in particular the direction of grooming (ie which monkey was grooming as opposed to being groomed).

Alongside this, they recorded some basic information about each monkey, specifically their gender, whether the monkey was juvenile or senior, and finally which of the two sleeping locations that the monkey slept at.

To better understand the monkey's social structure, we will consider the group of monkeys to be a 'graph'. This is the fundamental component of Statistical Network Analysis (also called Network Analysis or Social Network Analysis). A graph is composed of a set of nodes and and a set of edges that join these nodes.

We will consider each monkey to be a node and each edge to be a recorded grooming behaviour. As the grooming direction was also recorded, we can use 'directed edges' which only move in one direction. This concept is shown for an imaginary and simplified group of monkeys below.

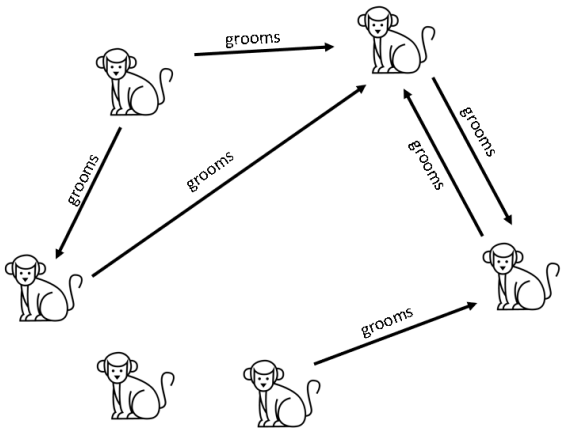


Figure 1: simplified graph of monkey grooming behaviour

This example purposefully demonstrates a few example behaviours that will be of interest for our Statistical Network Analysis. For example, one pair (or 'dyad') of monkeys in the top-right have a 'reciprocal' relationship where the directed edge points both ways. There is also a monkey for which we have no recorded grooming behaviour, this makes our graph 'disconnected', which two 'components' of connected nodes. However because we have directed edges, the component with most monkeys is only 'weakly connected' because we have to ignore edge direction to connect all monkeys (e.g. those who groomed others but were never groomed themselves).

There are other characteristics of graphs and ways to measure them, several of these will be used through the report and will be introduced where relevant. In this report, we will firstly do some exploratory analysis and inital plots/summaries of the data. Following this we will perform an Exponential Random Graph Model (ERGM) analysis of the network to understand the network statistically. We will then fit a Stochastic Block Model (SBM) to the data in order to get a better understanding of subgroups in the network. Finally we will make any final conclusions about the social structure of the monkeys and suggest next steps and ideas for future studies.

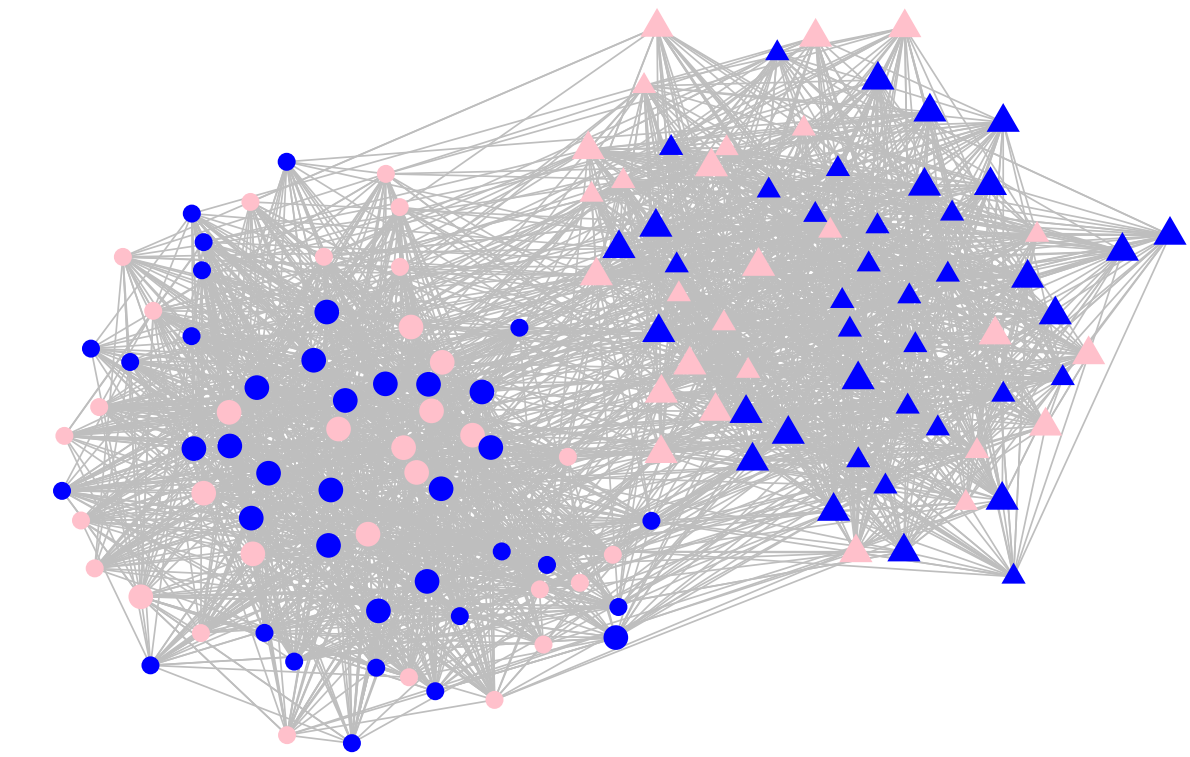
Initial Summaries and Plots

Exploratory Analysis

After reading in the data about each monkey and using the grooming data as a list of edges to create a network, we can begin some initial plots of our data. In the plot below we have coloured the nodes to represent the gender of the monkey they represent (pink for females, blue for males), sized the nodes to represent age (larger for senior, smaller for juvenile) and given each sleeping location a different shape.

```
mnky.net %v% "colour" = ifelse(mnky.net %v% "gender" == "Male",
                               "blue", "pink")
ggnet2(mnky.net,
       node.size = "age",
       size.palette = c("Senior" = 4, "Juvenile" = 2),
       max_size = 4,
```

```
      node.color = "colour",
      node.shape = "sleeploc",
      edge.size = 0.3,
      edge.color = "grey") +
  guides(color = "none", size = "none", shape = "none") +
  ggtitle("Monkey Grooming Behaviour Graph")
```



It is immediately obvious that there are two strongly connected subgraphs that align with the sleeping location of monkeys, simply put, it appears that monkeys are more likely to groom those who sleep in the same area. This is logical both in terms of opportunity and existing social bonds (i.e. friendship). It is not possible to deduce whether these social bonds and therefore also grooming behaviours were caused by the sleeping locations, or whether the sleeping locations were caused by the social bonds and grooming behaviour.

Of the other co-variates, age and gender, we can say that each sleeping location seems to have a fairly equal mix of both. However given the number of nodes, it is difficult to visually assess the edges within the graph and their direction. This makes further analysis using this visualisation difficult. Other plots will be used later in the report but we will now move to some more numeric initial analysis. Below is a frequency table of monkeys split by their co-variates.

```
mytable <- table(monkeys$Age, monkeys$Gender, monkeys$SleepLoc)
ftable(mytable)
```

```
##                  Loc1 Loc2
##
## Juvenile Female    20   12
```

```
##          Male      20   21
## Senior   Female    12   15
##          Male      18   19
```

We can see that there are broadly more juvenile monkeys than senior, and slightly more male than female monkeys. If we were interested in the demographics we could do a t-test to determine if there were significantly more than expected. There also seems to be a fairly even split between sleeping locations. We can get some of the same information using the summary command for our network.

```
summary(mnky.net)
```

The output has been hidden purely for size reasons as it generates a line per edge (ie grooming). It also reaffirms some knowledge about the graph (e.g. that edges are directed) and gives us some network attributes: there are 137 vertices (i.e. its order and the number of monkeys) in the data, there are 2468 total edges (i.e. its size and the number of recorded grooming behaviours), and the density is 0.13. Density is a network attribute which measures how dense the edges of the graph are, and is calculated as a proportion of the number of edges to the maximum possible number of edges. So on average, each monkey has groomed or been grooming by 13% of all other monkeys in the data.

Some other network measures which can be considered are the number of components, that is the number of connected subgraphs. For our directed graph, these can be strongly or weakly connected, dependent on whether their are directed edges that connect to and from nodes. In this case, and as suggested by the plot above, our graph is both strongly and weakly connected with only its one main component.

```
sna::components(mnky.net,connected="strong")
```

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

```
sna::components(mnky.net,connected="weak")
```

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

Another network measure is graph diameter, this is the maximum of shortest paths between all nodes. You can consider this for directed graphs, where paths must follow the direction of edges, and for symmetrised/undirected graphs where paths can use edges in any direction.

```
max(geodist(mnky.net)$gdist)
```

```
## [1] 3
```

```
max(geodist(symmetrize(mnky.net))$gdist)
```

```
## [1] 3
```

In this case the diameter is three for both cases, meaning that for the monkeys in the data, rather than 'six degrees' of separation, there are three or less between each monkey. Given the number of monkeys, this indicates a high level of connection across the network.

Within networks, we may also be interested in finding 'cliques' which are maximally connected subgraphs. For example a triad of monkeys with six edges, where each monkey has groomed and been groomed by both other monkeys.

```
igraph::clique.number(asIgraph(mnky.net))
```

```
## [1] 6
```

```
length(igraph::cliques(asIgraph(mnky.net), min = 6, max = 6))
```

```
## [1] 6
```

```
#igraph::cliques(asIgraph(mnky.net), min = 6, max = 6)
```

This shows that the largest size of clique is six, and that there are six of these groups of six. The final piece of code is commented out for the sake of size, but lists which monkeys are in these cliques. Several monkeys appear in more than one of these size six cliques, including monkey 2 which appears in three. It is possible than these well-connected monkeys may appear again when we begin to measure centrality.

Before centrality, we can look to detect communities within the data using modularity. This measures how well a graph clusters based on observed categorical covariates, it essentially measures the connections minus the level that would be expected with randomisation. The measures ranges from 0.5 to 1 with larger values suggesting stronger clustering. Note that this is only currently calculated in R for undirected graphs, hence as.undirected being used in the snipped below.

```
igraph::modularity(igraph::as.undirected(asIgraph(mnky.net)),
                   as.factor(monkeys$SleepLoc))
```

```
## [1] 0.3551971
```

```
igraph::modularity(igraph::as.undirected(asIgraph(mnky.net)),
                   as.factor(monkeys$Gender))
```

```
## [1] -0.005060095
```

```
igraph::modularity(igraph::as.undirected(asIgraph(mnky.net)),
                   as.factor(monkeys$Age))
```

```
## [1] -0.3409696
```

Calculating the modularity for each of our covariates, we have 0.36 for sleep location, indicating there is more grooming between monkeys sleeping in the same location than would be expected in a randomisation. This is as we would expect from the clustering we observed in the plot above. The figure is -0.01 for gender, which may well not be significant, but indicates very slightly fewer recorded grooming behaviours between monkeys of the same gender than expected. For age, the modularity is -0.34, this is interesting as it wasn't something we were able to observe in the plot above. Monkeys of the same age are less likely to groom each other than we would expect. One possible explanation for this, if we assume monkeys have limited time for grooming and are therefore busy grooming monkeys of different ages, is that they are grooming their parents/children, but there is no way to investigate this possibility with the current data.

Subgraph Exploratory Analysis

All of the above exploration is for all of the monkeys, however we have seen that there are two clearly defined subgraphs, split by sleeping location. We can do some further exploration by splitting into these subgroups and re-calculating some of the above measures. The output included below for brevity, but those with interesting results are discussed below.

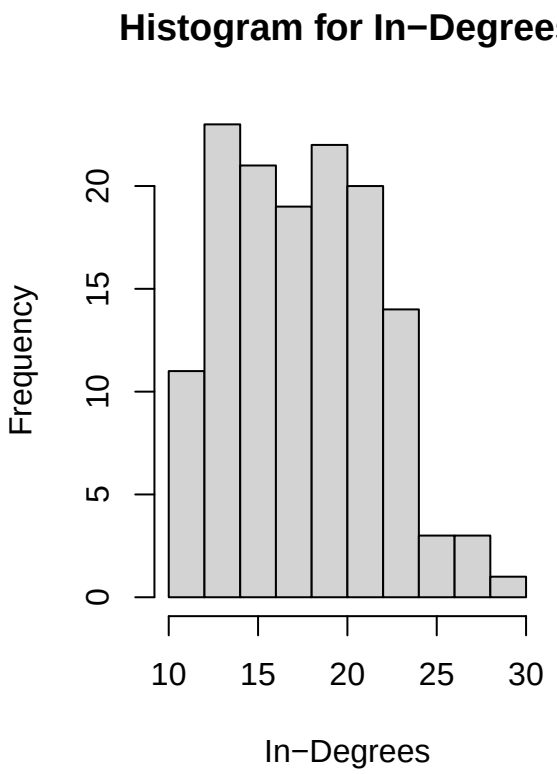
```
mnky.net.1<-get.inducedSubgraph(mnky.net,
                                which(mnky.net %v% "sleeploc" == "Loc1"))
mnky.net.2<-get.inducedSubgraph(mnky.net,
                                which(mnky.net %v% "sleeploc" == "Loc2"))
summary(mnky.net.1)
summary(mnky.net.2)
sna::components(mnky.net.1,connected="strong")
sna::components(mnky.net.1,connected="weak")
sna::components(mnky.net.2,connected="strong")
sna::components(mnky.net.2,connected="weak")
max(geodist(mnky.net.1)$gdist)
max(geodist(mnky.net.2)$gdist)
igraph::clique.number(asIgraph(mnky.net.1))
igraph::clique.number(asIgraph(mnky.net.2))
length(igraph::cliques(asIgraph(mnky.net.1), min = 6, max = 6))
length(igraph::cliques(asIgraph(mnky.net.2), min = 6, max = 6))
```

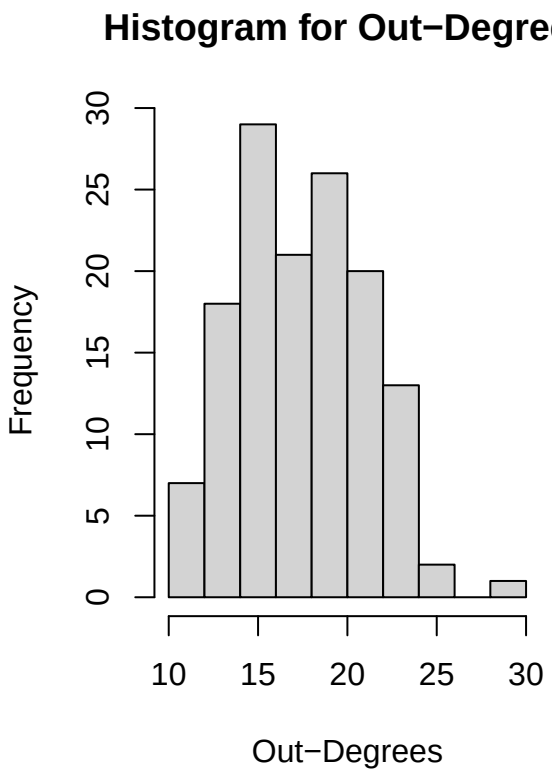
The summary functions give densities for the two subgraphs of 0.23 and 0.24 respectively. These are higher than the total graph as would be expected from the visibly more dense subgraphs. The components function does not identify any disconnected in either subgraph, the diameter of each subgraph is three, and the largest cliques have six nodes, with four and two in each subgraph respectively.

In- and Out- Degree Distributions

The degree of a node is the number of edges attached to it, for directed graphs we can consider in-degree and out-degree which count the edge into and out from the node respectively. We can calculate and view these figures in R as below.

```
indeg<-sna::degree(mnky.net, cmode="indegree")
outdeg<-sna::degree(mnky.net, cmode="outdegree")
par(mfrow=c(1,2))
hist(indeg, main="Histogram for In-Degrees", xlab="In-Degrees")
hist(outdeg, main="Histogram for Out-Degrees", xlab="Out-Degrees")
```





We can see that the distributions are very roughly similar, with similar minimums and maximums, although the out-degree distribution seems to be skewed somewhat more to the right. We can also use the summary function to look at these distributions numerically.

```
summary(indeg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   15.00   18.00   18.01   21.00   29.00
```

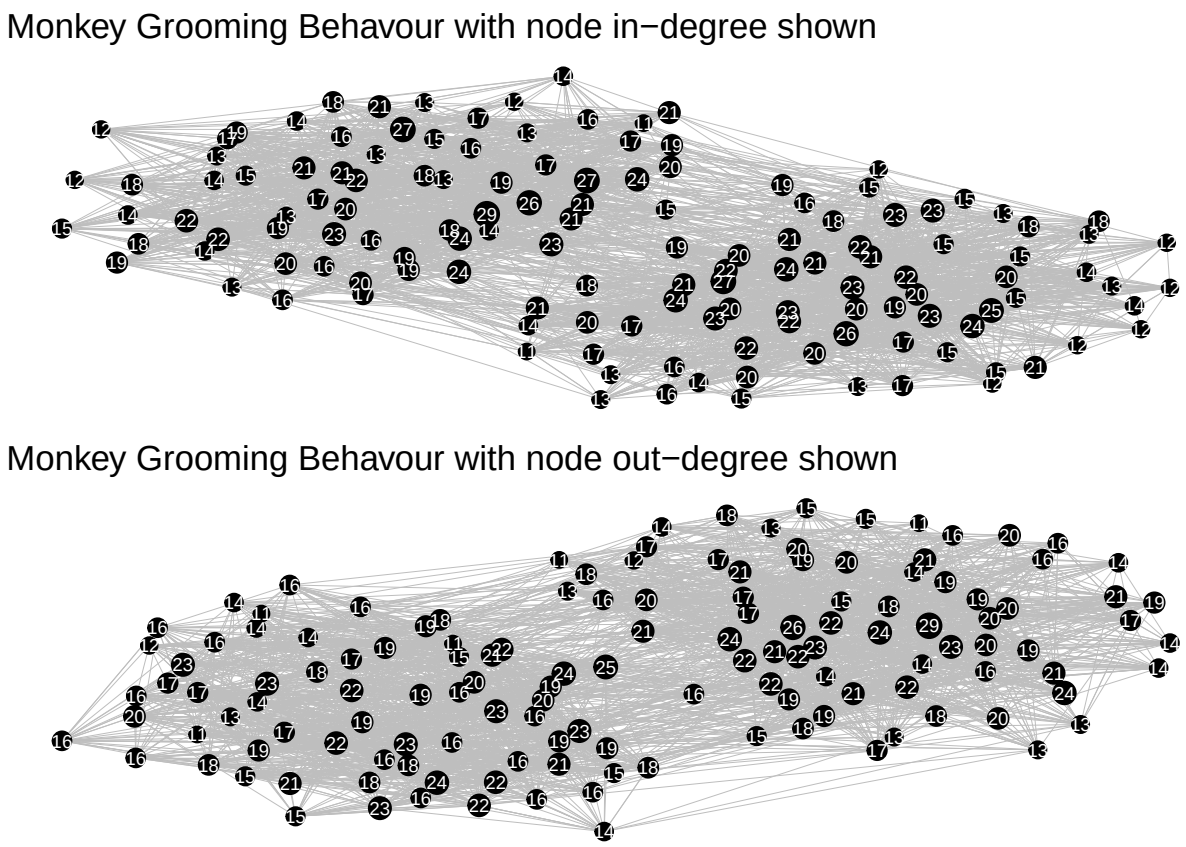
```
summary(outdeg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   16.00   18.00   18.01   21.00   29.00
```

For both in- and out-degrees the range is 11 to 29. So it seems that regardless of social ties, all monkeys are giving and receiving some grooming, and the range between the least and most grooming is the same for giving and receiving. We can also show the degree distributions within the graph.

```
in.plot <- ggnet2(mnky.net,
       node.size = indeg,
       max_size = 4,
       node.color = "black",
       edge.size = 0.01,
       edge.color = "grey") +
```

```
  guides(color = "none", size = "none", shape = "none") +
  geom_text(aes(label = indeg), color = "white", size = 2.5) +
  ggtitle("Monkey Grooming Behavour with node in-degree shown")
out.plot <- ggnet2(mnky.net,
       node.size = outdeg,
       max_size = 4,
       node.color = "black",
       edge.size = 0.01,
       edge.color = "grey") +
  guides(color = "none", size = "none", shape = "none") +
  geom_text(aes(label = outdeg), color = "white", size = 2.5) +
  ggtitle("Monkey Grooming Behavour with node out-degree shown")
grid.arrange(in.plot, out.plot, nrow=2)
```



Although a little small, we can see that the range of degree distributions is spread through both sleeping location groups for in-degree and out-degree.

Reciprocity and Transitivity

Reciprocity was mentioned in the introduction, it can be defined in a couple of ways but we will consider the proportion of reciprocated ties for pairs of nodes with at least one tie (the other definition includes other symmetric ties, including pairs with no ties).

```
grecip(mnky.net, measure = "dyadic.nonnull")
```

```
##       Mut
## 0.1791687
```

This means that approximately 18% of pairs of monkeys which have one grooming behaviour recorded, will have a reciprocal grooming behaviour also recorded. So although the default assumption may be that if one monkey is groomed by another, it will at some point groom its groomer, that is not by any means the standard. It therefore seems that grooming if far more complex than merely grooming and being groomed by friends.

Transitivity is a graph summary that returns a number ranging from 0 to 1 which records what proportion of triads with nodes which have at least two ties will have a third (e.g. forming a triangle). For directed graphs is can be calculated in the weak or strong sense, dependent on whether we consider the direction of edges, which we will not for this measure.

```
suppressWarnings(gtrans(mnky.net, mode="weak"))
```

```
## [1] 0.1200136
```

This returns 0.12, indicating that for all groups of three monkeys with at least two grooming behaviours recorded between them, 12% will have a third tie.

Network Centrality

There are many ways to measure the centre of a graph, one of the most basic is simply the node with the highest degree. For our directed graph we could select in-degree or out-degree, we will choose the former with the assumption that monkeys receiving more grooming are likely to have a higher status than monkeys doing more grooming.

```
max(sna::degree(mnky.net, cmode="indegree"))
```

```
## [1] 29
```

```
which.max(sna::degree(mnky.net, cmode="indegree"))
```

```
## [1] 106
```

So monkey 106 has the highest in-degree at 29. However given the two clear subgraphs within our data, it may not be logical to find a single central node, instead we may wish to find the centre of each of the subgraphs.

```
max(sna::degree(mnky.net.1, cmode="indegree"))
which.max(sna::degree(mnky.net.1, cmode="indegree"))
max(sna::degree(mnky.net.2, cmode="indegree"))
which.max(sna::degree(mnky.net.2, cmode="indegree"))
```

The highest in-degrees are 25 and 26 respectively, for monkeys 44 and 54.

Another measure of centrality is ‘closeness centrality’ which measures how far a node is from all other nodes. We use the inverse of the sum of all path lengths to find this. Therefore this figure is large for nodes that are close to all others (and therefore more central).

```
which.max(sna::closeness(mnky.net, gmode = "digraph"))
which.max(sna::closeness(mnky.net.1, gmode = "digraph"))
which.max(sna::closeness(mnky.net.2, gmode = "digraph"))
```

So for the full graph, the largest closeness centrality measure is for monkey 97, and for the subgraphs are for monkeys 25 and 32 respectively.

Another measure of centrality is 'betweenness centrality' which measures how crucial a node is for travelling between other nodes. This measure is found by considering the set of shortest paths between all of other nodes and then taking the proportion of those containing this node versus those which do not. Again, this figure is large for nodes which are in many of the shortest paths (and therefore more central).

```
which.max(sna::betweenness(mnky.net, cmode="directed"))
which.max(sna::betweenness(mnky.net.1, cmode="directed"))
which.max(sna::betweenness(mnky.net.2, cmode="directed"))
```

The highest betweenness centrality is for monkey 97 (the same as with closeness centrality), and for subgraphs is monkey 53 and monkey 19 respectively.

Another measure of centrality is 'eigenvector centrality' which is calculated using the first eigenvector of the graph adjacency matrix. Nodes with high eigenvector values are those which are connected to many other nodes which are then themselves also connected to many others. This helps to find tightly connected groups of nodes.

```
which.max(sna::evcent(mnky.net))
which.max(sna::evcent(mnky.net.1))
which.max(sna::evcent(mnky.net.2))
```

The highest eigenvector centrality is for monkey 51, and for subgraphs is monkey 25 and 66 respectively.

There are other possible measures of centrality however it is clear from the above that they do not always agree on which node is the most central. In the interest of brevity we will not investigate centrality further, but we would look at the top few monkeys for each measure and try to find more commonalities between them to identify the most central monkey in the data and in each subgraph.

Assortativity

Homophily, measured by assortativity, is the tendency of individuals to connect with those who have similar characteristics. So for graphs, we expect to see more edges between nodes who match (or who have similar) characteristics. The measure is between -1 and 1, with higher numbers being more assortative. For categorical variables, this is similar to modularity as discussed above.

```
assortativity_nominal(asIgraph(mnky.net), as.factor(monkeys$Gender))
```

```
## [1] -0.004794405
```

```
assortativity_nominal(asIgraph(mnky.net), as.factor(monkeys$Age))
```

```
## [1] -0.7206862
```

```
assortativity_nominal(asIgraph(mnky.net), as.factor(monkeys$SleepLoc))
```

```
## [1] 0.7469829
```

Our conclusions from these figures are similar to those we made when considering modularity. Monkeys are very slightly less likely to groom and be groomed by those of the same gender, but this figure may not be significant. Monkeys are less likely to groom and be groomed by those of the same age, which may tie into them spending more time with their parents/children. And finally monkeys are far more likely to groom and be groomed by those who sleep at the same location as them.

ERGM Analysis

Introduction and Null Model

An Exponential Random Graph Model (ERGM) is a statistical model for graphs which will allow us to perform statistical inference. They are flexible enough to handle many types of covariates and can cope with directed and undirected graphs. They extend Generalised Linear Models to networks and are fit using Bayesian inference, this means that each run can give small variation in the estimates.

We will first fit a null model which is only interested in graph density and therefor relies only on the number of edges. This will be helpful for comparison.

```
mnky.mod.0 <- ergm(mnky.net ~ edges, control=control.ergm(seed=42))
```

```
summary(mnky.mod.0)
```

```
## Call:
## ergm(formula = mnky.net ~ edges, control = control.ergm(seed = 42))
##
## Maximum Likelihood Results:
##
##       Estimate Std. Error MCMC % z value Pr(>|z|)
## edges -1.87938    0.02161      0  -86.97   <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 25829  on 18632  degrees of freedom
##  Residual Deviance: 14572  on 18631  degrees of freedom
##
## AIC: 14574  BIC: 14581  (Smaller is better. MC Std. Err. = 0)
```

We can see from the p-value that edges as a prediction term is significant. We can interpret the coefficient by considering that if we take the inverse logit transformation of it, we would have the probability of any individual tie existing, which for the null model is equal to the density (0.13) that we found for our graph.

Model Fitting

We will now fit a more full model but in order to do that we must consider that using the ergm package as we have above, there are many many different types of term that we can include in a model. One of these is structural type terms, which predict the probability of edges based on how the entire graph is configured.

One of these, introduced with 'mutual' is how likely a directed tie is to be reciprocated. There are many other types but we will also consider dyadic and node type terms. Dyadic terms are so named because they relate to dyads of nodes, and relate to whether or node the covariate values of dyads effect their probability of being tied. Node type terms are so named because they only consider a single node's covariates and whether or not they effect the probability of that node being tied to any others. In this next model we will only consider dyadic terms, and will consider them for all covariates.

```
mnky.mod.1 <- ergm(mnky.net ~ edges + mutual + nodematch("gender") +
                     nodematch("age") + nodematch("sleeploc"),
                   control=control.ergm(seed=42))
```

```
summary(mnky.mod.1)
```

```
## Call:
## ergm(formula = mnky.net ~ edges + mutual + nodematch("gender") +
##     nodematch("age") + nodematch("sleeploc"), control = control.ergm(seed = 42))
##
## Monte Carlo Maximum Likelihood Results:
##
##                      Estimate Std. Error MCMC % z value Pr(>|z|)
## edges               -2.757934   0.063946      0 -43.129   <1e-04 ***
## mutual              -0.071425   0.084201      0  -0.848    0.396
## nodematch.gender     0.003039   0.050648      0   0.060    0.952
## nodematch.age       -2.234744   0.067445      0 -33.134   <1e-04 ***
## nodematch.sleeploc   2.368390   0.068146      0  34.755   <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 25829  on 18632  degrees of freedom
##  Residual Deviance: 11022  on 18627  degrees of freedom
##
## AIC: 11032  BIC: 11071  (Smaller is better. MC Std. Err. = 0.1005)
```

This model predicts the probability of edges by looking at the overall density of the graph, the reciprocity of the graph, and whether or not pairs of nodes have the same or different values for each of our three covariates. The coefficient estimates for our dyadic terms suggest that monkeys are more likely to have groomed or been groomed is they have the same gender, less likely is they have the same age and more likely if they have the same sleep location, however the sizes of these effects vary.

The edges term as used in the null model is still significant according to it's p-value however some of the newly introduced terms are not. The mutual term is one of these, given the relatively low level of reciprocity we have previously examined, this is perhaps not surprising. Of the dyadic terms, that of gender is not significant but age and sleep location are. Again, this matches what we discovered when examining the modularity and assortativity above.

To improve this model we will remove the non-significant terms, mutual and nodematch(gender). For the dyadic sleep location term which has a significant positive effect, we can investigate to see if there is any evidence of different effects for different levels (i.e. for different sleep locations). We could consider adding other structural terms, for example 'triangles' which measures transitivity, however given the low value of this measure we will not for this model. We can also consider whether or not to include any node type terms by considering whether being a specific age/gender/sleep location makes the monkeys generally more likely to groom or be groomed by all other monkeys.

We can asses this using the following calculations which find the mean in-degree and out-degree figures for each of the different values. In this case the values have been hidden to preserve space but the only covariate with any significant and consistent difference in degree size is age, which we include in our improved model.

```
mean(indeg[monkeys$Age=="Senior"])
mean(indeg[monkeys$Age=="Juvenile"])
mean(outdeg[monkeys$Age=="Senior"])
mean(outdeg[monkeys$Age=="Juvenile"])
mean(indeg[monkeys$Gender=="Male"])
mean(indeg[monkeys$Gender=="Female"])
mean(outdeg[monkeys$Gender=="Male"])
mean(outdeg[monkeys$Gender=="Female"])
mean(indeg[monkeys$SleepLoc=="Loc1"])
mean(indeg[monkeys$SleepLoc=="Loc2"])
mean(outdeg[monkeys$SleepLoc=="Loc1"])
mean(outdeg[monkeys$SleepLoc=="Loc2"])
```

We include this in out model using nodefactor, and make the other improvements discussed above to obtain the model shown below.

```
mnky.mod.2 <- ergm(mnky.net ~ edges + nodefactor("age") + nodematch("age") +
               nodematch("sleeploc", diff = TRUE), control=control.ergm(seed=404))
```

```
summary(mnky.mod.2)
```

```
## Call:
## ergm(formula = mnky.net ~ edges + nodefactor("age") + nodematch("age") +
##     nodematch("sleeploc", diff = TRUE), control = control.ergm(seed = 404))
##
## Maximum Likelihood Results:
##
##                          Estimate Std. Error MCMC % z value Pr(>|z|)
## edges                    -2.83213    0.08113      0 -34.909   <1e-04 ***
## nodefactor.age.Senior     0.07195    0.05603      0   1.284    0.199
## nodematch.age            -2.20273    0.06234      0 -35.335   <1e-04 ***
## nodematch.sleeploc.Loc1   2.33696    0.06936      0  33.695   <1e-04 ***
## nodematch.sleeploc.Loc2   2.35080    0.07003      0  33.569   <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 25829  on 18632  degrees of freedom
##  Residual Deviance: 11021  on 18627  degrees of freedom
##
## AIC: 11031  BIC: 11071  (Smaller is better. MC Std. Err. = 0)
```

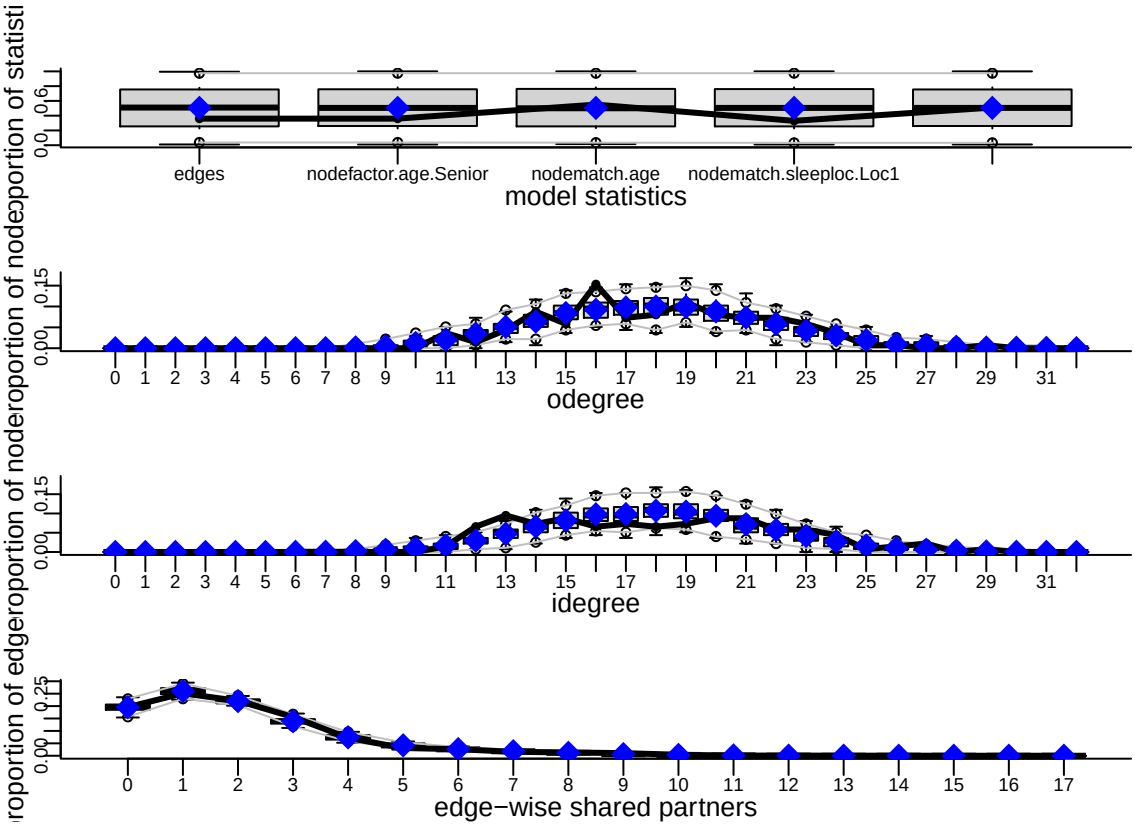
Despite the differences in mean degree for monkeys of different ages, it seems this this term is not significant in our model. Alongside this all other terms are significant, including the now separate dyadic terms for the two sleeping locations, which do indeed have two different coefficient estimates. The AIC criteria has improved slightly over the previous model. Broadly it seems that whether or not a monkey grooms or is groomed by another monkey is made more likely by being of different ages and being in the same sleeping location. In this data, there is also a non-significant positive effect whereby senior monkeys are more likely to groom and be groomed.

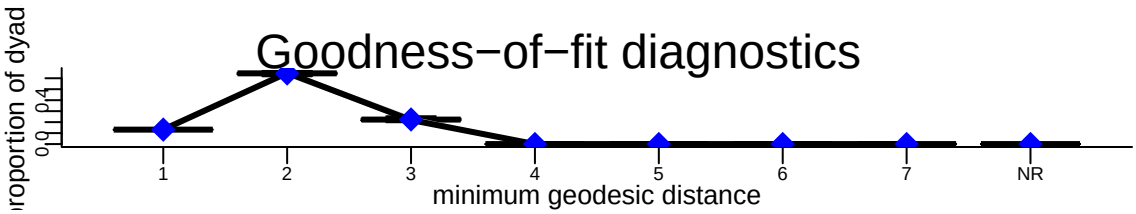
Although we are only instructed to fit one set of improvements to the suggested model, some future improvement could involve examining more modern and advance structural terms like geometrically weighted edgewise shared partnerships, examining whether we can find evidence of different effects at different levels for dyadic components with negative effects and using nodemix to examine all combinations of categorical covariates for different effects.

Goodness of Fit

For the ergm package in R, there is a gof command which gives the necessary summaries for testing goodness of fit.

```
gof.mnky.mod.2<-gof(mnky.mod.2)
par(mfrow=c(4,1),cex=0.85, mgp=c(0.95,0.2,0), mai=c(0.325,0.45,0.325,0.05), bty="L")
par(oma=c(0.5,2,1,0.5))
plot(gof.mnky.mod.2)
```





We test how well an ERGM model fits by considering how well it produces the observed properties of the graph. In order to do this we consider the value of statistics not part of our model and compare their values in the observed graph and in our simulated model. In the graphs above we can see that the edge-wise shared partners and minimum geodesic distance plots have the observed line very close to the centre of the boxplots which indicates that statistics are very similar. The model statistics plot is not so good, particularly for the dyadic term for sleep location 1. There are also issues in the in-degree plot specifically for 16 in-degrees which has a spike in the observed data.

Broadly this is not a perfectly fitting model, however the issues seem to be somewhat limited and could potentially be resolved with further investigation and the fitting of further improved models.

Stochastic Block Model

Introduction and Model Fit

A Stochastic Block Model (SBM) is a generative model (i.e. a model that could generate new data) which is used for community detection. Essentially we consider that each node in the graph belongs to an undetected community where their tendency to form edges is related to whether or not they are in the same community as any dyad. So we do not consider any covariates, only the edges connecting nodes. The results of this can imply that objects in a detected community are more similar than those not in it.

For example, were we to use this method with our data and discover two strong defined communities who did not correlate with any covariates, there may be another unrecorded factor that is affecting grooming, e.g. the monkeys are either red or black and red monkeys are more likely to groom and be groomed by red monkeys and vice-versa.

As we are trying to find undetected communities, we are also seeking to identify how many communities there are within the data. To do this, we fit the model for several numbers of communities and use the Intergrated Complete-Data Likelihood (ICL) criteria to determine the best fit. We will do this using the BM_bernoulli command from the blockmodels package to do this.

```
mnky.sbm <- BM_bernoulli("SBM", as.matrix(mnky.net))
mnky.sbm$estimate()
```

This output of this is hidden as it is several pages long but sets up and estimates the model in order for us to go on to interpret it. It considers between one and six communities in the data, we now use ICL to determine which is the best fit.

```
mnky.sbm$ICL
```

```
## [1] -7290.717 -6514.301 -6175.353 -5802.257 -5849.789 -5909.241
```

```
which.max(mnky.sbm$ICL)
```

```
## [1] 4
```

So the fourth model, with four communities in it is the best fit for the data. So there are four communities in the monkeys which may relate to existing covariates (perhaps a combination of sleep location and age) or which may relate to other unrecorded factors. We can look at the parameters for this model.

Model Interpretation

```
mnky.sbm$model_parameters[[4]]
```

```
## $pi
##             [,1]        [,2]        [,3]        [,4]
## [1,] 0.05673207 0.005973341 0.422521107 0.057006426
## [2,] 0.01122255 0.070460905 0.063146747 0.400373801
## [3,] 0.37108251 0.065102134 0.077514978 0.009626109
## [4,] 0.05404100 0.392380389 0.006732394 0.069889112
##
## $n_parameters
## [1] 16
```

The output of this relates to the formula used in the model which is not in our scope to cover, however the first matrix of figures, pi, gives the probability of an edge from one node to another given their respective community memberships. So the probability of a monkey in community one grooming a monkey in community two is 0.01.

We can also consider the membership probabilities for each community identified in the data. The below snippet shows these for the first few monkeys and also shows the proportion of monkey membership in each community.

```
head(mnky.sbm$memberships[[4]]$Z)
```

```
##             [,1]        [,2]        [,3]        [,4]
## [1,] 0.997813411 0.000728863 0.000728863 0.000728863
## [2,] 0.000728863 0.000728863 0.997813411 0.000728863
## [3,] 0.997813411 0.000728863 0.000728863 0.000728863
## [4,] 0.997813411 0.000728863 0.000728863 0.000728863
## [5,] 0.000728863 0.997813411 0.000728863 0.000728863
## [6,] 0.000728863 0.000728863 0.000728863 0.997813411
```

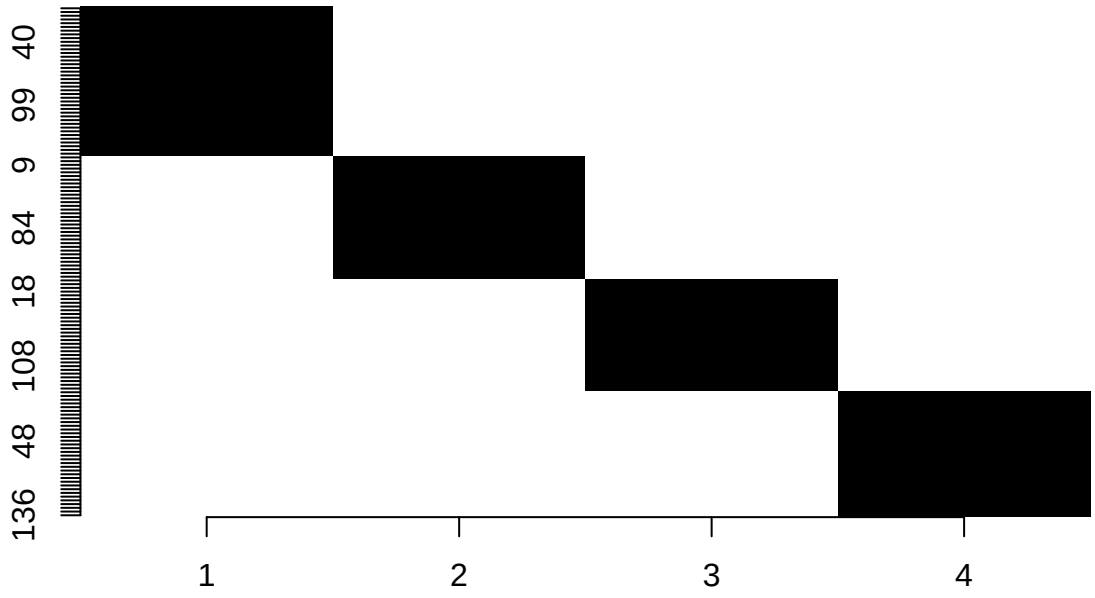
```
class.member<-apply(mnky.sbm$memberships[[4]]$Z,1,which.max)
prop.table(table(class.member))
```

```
## class.member
##         1         2         3         4
## 0.2919708 0.2408759 0.2189781 0.2481752
```

From this we can see that there is roughly an equal split of monkeys between each of the communities and a clear link to one community for each monkey.

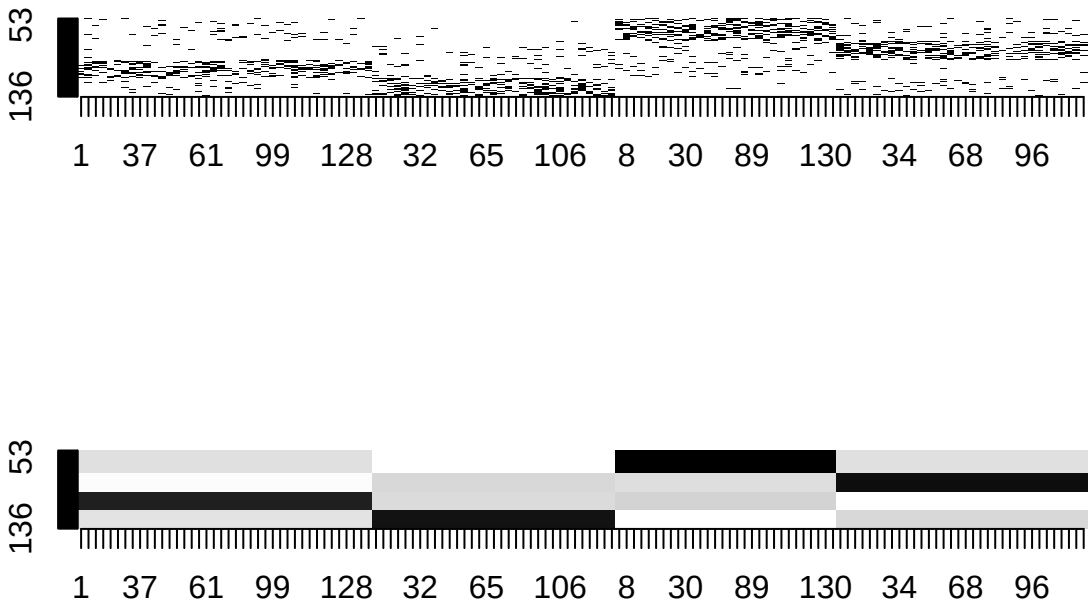
Our next step is to visualise these membership probabilities to see how certain the model is, although from the above figures we can judge that it is likely to be quite certain.

```
mnky.sbm$memberships[[4]]$plot()
```



So the model is indeed pretty certain. We can also visualise the observed graph versus the predicted graph, the first plot below is of observed edges and the second is of the probability of connection under the model.

```
mnky.sbm$plot_obs_pred(4)
```



We can see from this that the model really does seem to fit the underlying data very well, which gives rise to the question: how closey do these communities relate to our recorded covariates?

Covariate Comparison

We can initially examine this by considering the frequency tables below which show how many monkeys are in, for example class one and are Senior.

```
table(class.member, monkeys$Age)
```

```
##
## class.member Juvenile Senior
##            1       40      0
##            2       33      0
##            3        0     30
##            4        0     34
```

```
table(class.member, monkeys$Gender)
```

```
##
## class.member Female Male
```

```
##            1     20   20
##            2     12   21
##            3     12   18
##            4     15   19
```

```
table(class.member, monkeys$SleepLoc)
```

```
##
## class.member Loc1 Loc2
##            1   40    0
##            2    0   33
##            3   30    0
##            4    0   34
```

From these we can see that although there seems to be a fairly even spread of monkeys of different genders in the communities, the sleep location and age of monkeys seems to have a strong effect on which communities our SBM has placed them in. We can test if there is a statistical similarity between the communities and these two covariates using a chi-squared test.

```
chisq.test(table(class.member, monkeys$Age))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(class.member, monkeys$Age)
## X-squared = 137, df = 3, p-value < 2.2e-16
```

```
chisq.test(table(class.member, monkeys$SleepLoc))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(class.member, monkeys$SleepLoc)
## X-squared = 137, df = 3, p-value < 2.2e-16
```

The small p-values for both of these tests clearly indicates that there is no statistically significant different between the communities we have found using SBM and the age and sleep location covariates. So in this case we have not found any latent clustering of monkeys around some unrecorded covariate, but rather reconfirmed our earlier findings, that age and sleep location are significant predictors for how likely a pair of monkeys are to groom or be groomed by each other.

Conclusions

This is a limited dataset, we are only considering one group of monkeys, and they live in a sanctuary, which may effect their social structure. We probably could not extrapolate our results to wild monkeys, but perhaps could use them to predict the social structure of these monkeys in the future, or of other groups of monkeys in sanctuaries.

Our data is also somewhat limited because the only behaviour it records is directed grooming, without doing further research, we cannot know how important grooming is to the social structure of the monkeys. That said, there seem to be clear patterns in grooming behaviour. Monkeys are more likely to groom and be

groomed by monkeys who sleep in the same location, and who are of a different age to themselves. One suggested reason for the latter factor is parent-child relationships however we would need these relationships to be recorded in order to investigate this further.

Some other changes to the data that would allow for further analytical study include studying other groups of monkeys, including those in the wild; recording the number of times a grooming behaviour is observed to use as an edge covariate; record other social behaviours beyond grooming and recording further demographic information about the monkeys, for example their exact age, their size, and relevant data about their past (e.g. whether or not they may have been kept in captivity in the past).