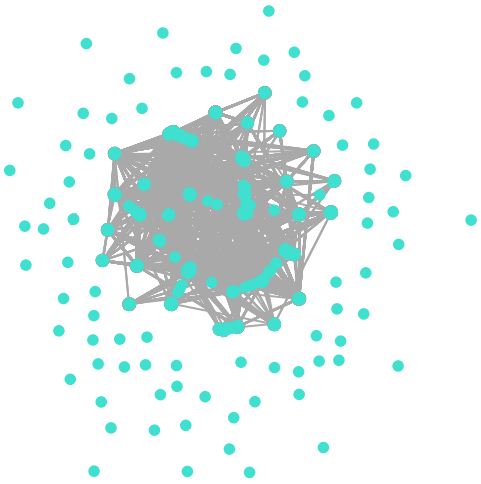
HW 6

2024-04-28

Question 1

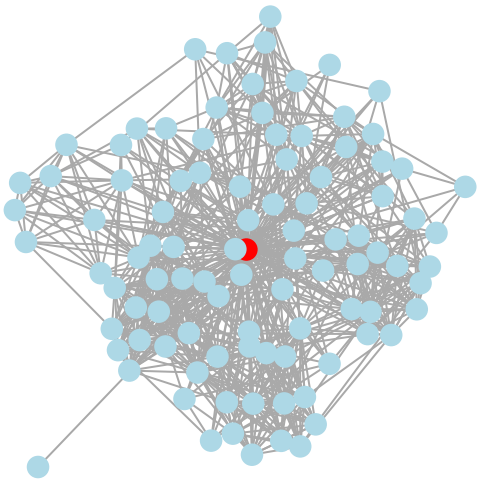
```
plot(actnet, vertex.size = 5, vertex.label = NA)
```



Question 2

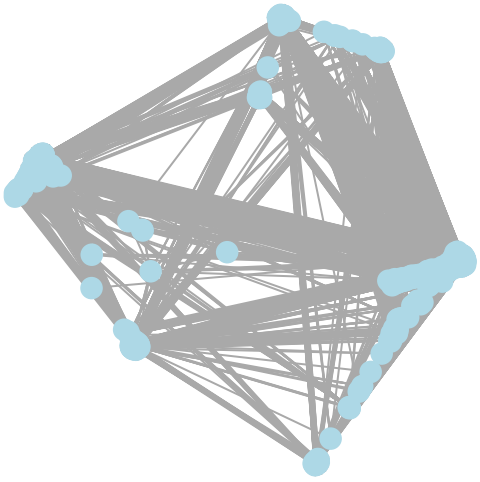
```
kevin_bacon <- "Bacon, Kevin"

for (order in 1:3) {
  bacon_neighborhood <- graph.neighborhood(actnet, order,
                                           V(actnet)[kevin_bacon], mode = "all")[[1]]
  V(bacon_neighborhood)$color <- "lightblue"
  V(bacon_neighborhood)[kevin_bacon]$color <- "red"
  plot(bacon_neighborhood, vertex.label.dist = 0, vertex.size = 10,
       vertex.label = NA, main = paste("Neighborhood of", kevin_bacon,
                                       "at order", order))
    cat("Order:", order, "Network Size:", vcount(bacon_neighborhood), "\n")
}
```

Neighborhood of Bacon, Kevin at order 1



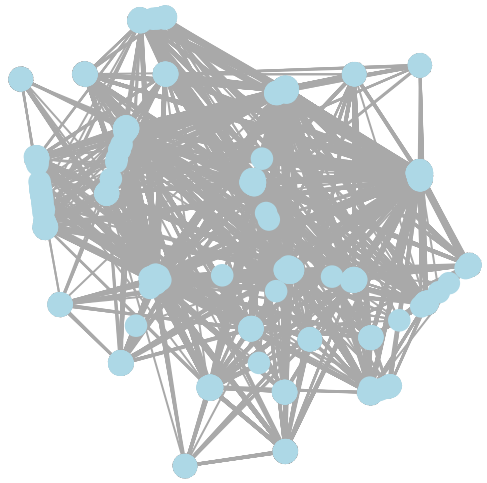
```
## Order: 1 Network Size: 97
```

Neighborhood of Bacon, Kevin at order 2



```
## Order: 2 Network Size: 2129
```

Neighborhood of Bacon, Kevin at order 3



```
## Order: 3 Network Size: 5981
```

Order 1: The neighborhood of "Bacon, Kevin" includes all the vertices that are directly connected to him. In other words, it includes all the actors who have appeared in the same movie as "Bacon, Kevin." The network size at this order is 97, indicating that there are 97 actors who have directly worked with "Bacon, Kevin" in at least one movie.

Order 2: The neighborhood expands to include not only the actors who have directly worked with "Bacon, Kevin" but also the actors who have worked with those actors. In other words, it includes the actors who have appeared in movies with the actors from the order 1 neighborhood. This expansion significantly increases the network size to 2129, indicating that there are many more actors who are indirectly connected to "Bacon, Kevin" through shared movie appearances.

Order 3: The neighborhood expands further to include the actors who have worked with the actors from the order 2 neighborhood. This results in an even larger network size of 5981, indicating that the neighborhood has expanded to include actors who are connected to "Bacon, Kevin" through multiple degrees of separation.

Question 3

```
mdegree <- degree(actnet)

most_common_actors <- names(sort(table(unlist(casts)), decreasing = TRUE)[1:5])
cat("Most common actors:", most_common_actors, "\n")
```

```
## Most common actors: Zivojinovic, Velimir 'Bata' Jeremy, Ron Dobtcheff, Vernon Doll, Dora Berléand, F
```

```
most_connected_actors <- names(sort(mdegree, decreasing = TRUE)[1:5])
cat("Most connected actors:", most_connected_actors, "\n")
```

```
## Most connected actors: Dobtcheff, Vernon Stévenin, Jean-François Muel, Jean-Paul Blanche, Roland Gar
```

```
actor1 <- "Jeremy, Ron"
actor2 <- "Bacon, Kevin"
```

```
shortest_path <- get.shortest.paths(actnet, from = actor1, to = actor2)
shortest_path_names <- V(actnet)$name[shortest_path$vpath[[1]]]
cat("Shortest path between", actor1, "and", actor2, ":", shortest_path_names, "\n")
```

```
## Shortest path between Jeremy, Ron and Bacon, Kevin : Jeremy, Ron Devon, Tony Bacon, Kevin
```

The top 5 most common actors: "Zivojinovic, Velimir 'Bata' ", "Jeremy, Ron", "Dobtcheff, Vernon", "Doll, Dora", and "Berléand, François". For betweenness centrality, the actor with the highest betweenness centrality was identified as "Neill, Sam". For degree centrality, the actor with the highest degree centrality was identified as "Dobtcheff, Vernon". The shortest path between "Jeremy, Ron" and "Bacon, Kevin" consists of the following actors: Jeremy, Ron -> Devon, Tony -> Bacon, Kevin.

Question 4

```
transactions <- as(casttrans, "transactions")
min_support <- 0.0001
min_confidence <- 0.1
max_length <- 3

actor_cast_rules <- apriori(transactions,
                            parameter = list(support = min_support, confidence = min_confidence,
                                             maxlen = max_length))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5   1e-04      1
##  maxlen target  ext
##       3  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[6953 item(s), 14326 transaction(s)] done [0.02s].
## sorting and recoding items ... [6874 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3
```

```
## Warning in apriori(transactions, parameter = list(support = min_support, :
## Mining stopped (maxlen reached). Only patterns up to a length of 3 returned!
```

```
##  done [0.43s].
## writing ... [387929 rule(s)] done [0.18s].
## creating S4 object  ... done [0.09s].
```

```
inspect(head(actor_cast_rules, 10))
```

```
##      lhs                       rhs                    support      confidence
## [1]  {Mizutani, Kei}      => {Jô, Asami}           0.0001396063 1.0000000
## [2]  {Jô, Asami}          => {Mizutani, Kei}       0.0001396063 0.5000000
## [3]  {Magdalena, Rita}    => {Garcia, Emilio}      0.0001396063 1.0000000
## [4]  {Garcia, Emilio}     => {Magdalena, Rita}     0.0001396063 0.3333333
## [5]  {Illiopoulos, Dinos} => {Logothetis, Ilias}   0.0001396063 1.0000000
## [6]  {Logothetis, Ilias}  => {Illiopoulos, Dinos}  0.0001396063 0.2500000
## [7]  {Murali (II)}        => {Mammootty}           0.0001396063 1.0000000
## [8]  {Mammootty}          => {Murali (II)}         0.0001396063 0.1111111
## [9]  {Anisa}              => {Ashley, Brooke}      0.0001396063 1.0000000
## [10] {Ashley, Brooke}     => {Anisa}               0.0001396063 0.2857143
##      coverage     lift      count
## [1]  0.0001396063 3581.5000 2
## [2]  0.0002792126 3581.5000 2
## [3]  0.0001396063 2387.6667 2
## [4]  0.0004188189 2387.6667 2
## [5]  0.0001396063 1790.7500 2
## [6]  0.0005584252 1790.7500 2
## [7]  0.0001396063  795.8889 2
## [8]  0.0012564568  795.8889 2
## [9]  0.0001396063 2046.5714 2
## [10] 0.0004886221 2046.5714 2
```

These association rules provide insights into the relationships between actors and casts. For example, rule 1 indicates that whenever actors Mizutani Kei appear together, they are always accompanied by Jô Asami in the cast. Similarly, rule 9 suggests a strong association between Anisa and Ashley Brooke in the cast. Overall, these association rules highlight interesting patterns of co-occurrence between actors and casts in the dataset.