Network Graphs in igraph

Sam Hillman

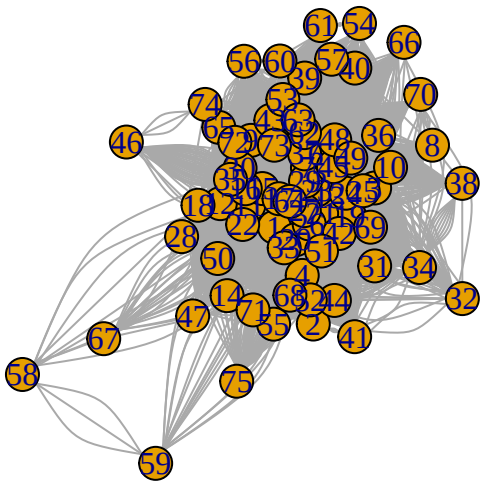
Assessment Questions:

Using a data set of your choosing from igraph. Do the following:

1. Plot a network with four different layouts of your choice.
2. Choose one layout to label/annotate in any way you see fit (be biologically meaningful)
3. Compute the degree, closeness, and betweenness centrality.
4. Plot the network, sizing the nodes based on degree, or produce a plot of degree distrubution.
5. Plot the results of one community detection algorithm applied to your network.
6. Add figure legends to each figure and a short paragraph at the end of the document interpeting the biology of what is occuring in your network.
7. Give the code used in an Appendix

We'll start by looking at the basic layout of our chosen dataset:

```
data(rfid)
plot(rfid,  main = "igraphdata RFID Dataset Network",
            sub = "Plotted with igraph Auto Layout")
```



Question One

1. Plot a network with four different layouts of your choice.

```
data(rfid)

plot(rfid, layout = layout_with_graphopt, main = "RFID Network, Graphopt Layout")
plot(rfid, layout = layout_with_lgl, main = "RFID Network, Large Graph Layout")
plot(rfid, layout = layout_with_fr, main = "RFID Network, FR Layout")
plot(rfid, layout = layout_with_gem, main = "RFID Network, GEM Layout")
```

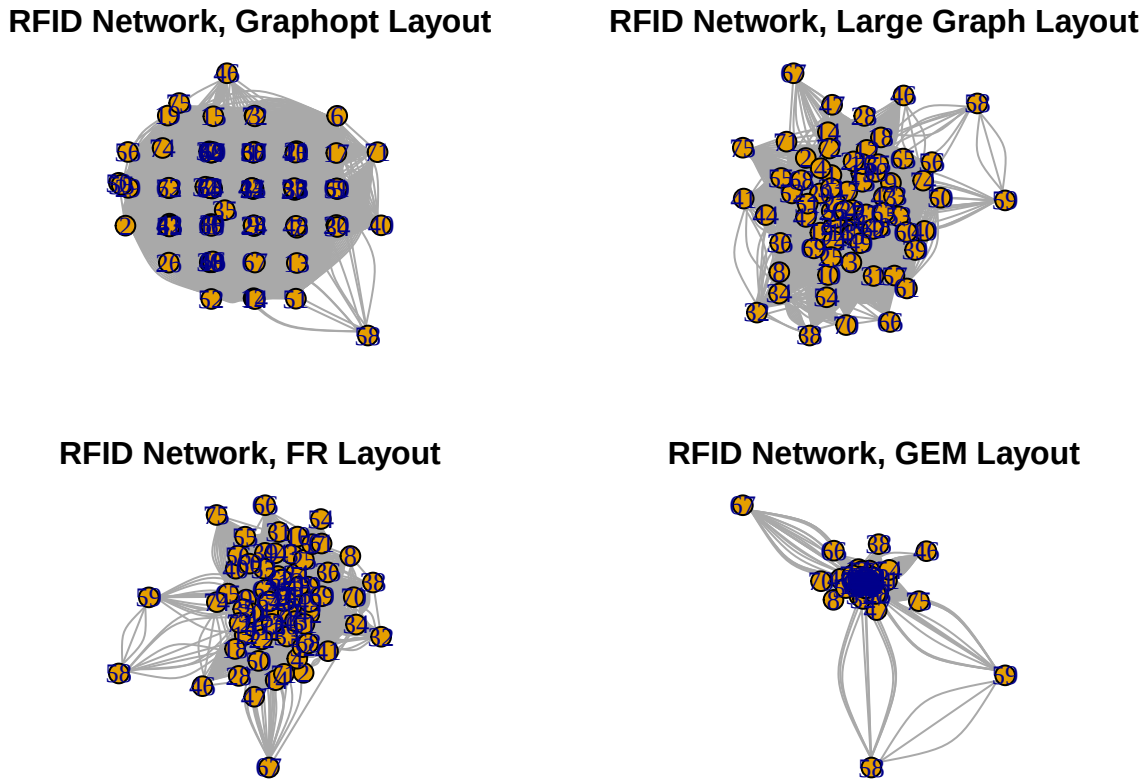


Fig. 1. Four layots of the RFID dataset from the igraphdata package. Layouts generated using different layout methods within the igraph package to demonstrate the difference in layout possiblities.

Question Two

2. Choose one layout to label/annotate in any way you see fit (be biologically meaningful)

To make our graph interpretable, we color by the status of the individuals (their job role), add edge weights to the edge width, and highlight key connections based on edge weight.

```
# To find edge weights, we first assign each edge a value of 1 and then use simplify()
E(rfid)$weight <- 1
rfid_simple <- simplify(rfid, remove.multiple = TRUE, remove.loops = TRUE,
                        edge.attr.comb=list(weight="sum", Time = "ignore"))

# We then set the vertex color using a palette from RColorBrewer
vertex_colour <- brewer.pal(4, "Set1")
E(rfid_simple)$color <- ifelse(E(rfid_simple)$weight > 10,"black", "grey")

plot(rfid_simple, layout = cords_fr, vertex.color = vertex_colour,
       edge.width = log(E(rfid_simple)$weight), edge.color = E(rfid_simple)$color,
     main = "RFID Network, Fruchterman-Reingold Layout")

legend("bottomleft", legend = levels(as.factor(V(rfid_simple)$Status)),
         col = coul , bty = "n", pch=20 , pt.cex = 2, cex = 1.25,
         text.col=coul , horiz = FALSE, inset = c(0.0, 0.0))
```

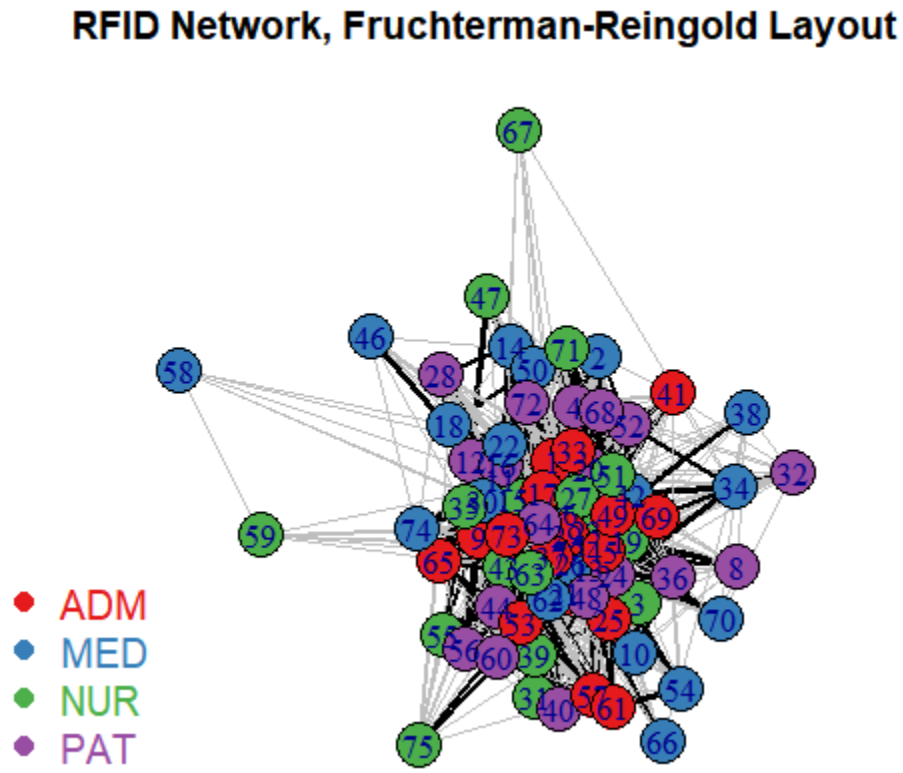


Fig. 2. Annotated RFID Network graph. Vertex colour denotes job role (Admin Staff [ADM], Medical Doctors [MED], Nurses [NUR], and Patients [PAT]) and edges with a weight greater than 10 are coloured black.

Question Three

3. Compute the degree, closeness, and betweenness centrality.

To compute the degree we use:

```
centr_degree(rfid, mode="all", normalized=T)
```

From this, our values for the centralization and the theoretical maximum values are:

```
centr_degree(rfid, mode="all", normalized=T)[2]
```

```
## $centralization
## [1] 46.23459
```

```
centr_degree(rfid, mode="all", normalized=T)[3]
```

```
## $theoretical_max
## [1] 5550
```

To compute closeness we use:

```
closeness(rfid, mode="all", weights=NA)
centr_clo(rfid, mode="all", normalized=T)
eigen_centrality(rfid, directed = FALSE, weights = NA)
```

From this, our values for closeness are:

```
centr_clo(rfid, mode="all", normalized=T)[2]
```

```
## $centralization
## [1] 0.4343957
```

```
centr_clo(rfid, mode="all", normalized=T)[3]
```

```
## $theoretical_max
## [1] 36.7483
```

```
eigen_centrality(rfid, directed = FALSE, weights = NA)["value"]
```

```
## $value
## [1] 2141.429
```

To calculate betweeness we use:

```
#as our graph is undirected (checked with is.directed(rfid)) we use directed = FALSE:
centr_betw(rfid, directed = FALSE, normalized = TRUE)
```

and our values are:

```
centr_betw(rfid, directed=FALSE, normalized=T)[2]
```

```
## $centralization
## [1] 0.05935457
```

```
centr_betw(rfid, directed=FALSE, normalized=T)[3]
```

```
## $theoretical_max
## [1] 199874
```

Question Four

4. Plot the network, sizing the nodes based on degree, or produce a plot of degree distrubution.

Sizing our nodes based on degree leads to unusable graphs due to the disparity in node degree (an example can be found in Section 7).
We can calculate the degree distributions using:

```
deg <- degree(rfid, mode="all")
```

and plot the distribution:

```
ggplot() +
  aes(deg) +
  geom_histogram(binwidth = 150) +
  labs(x = "Degree Value",
       y = "Count",
       title = "Histogram of Degree Values from the igraphdata RFID dataset") +
  theme_bw()
```

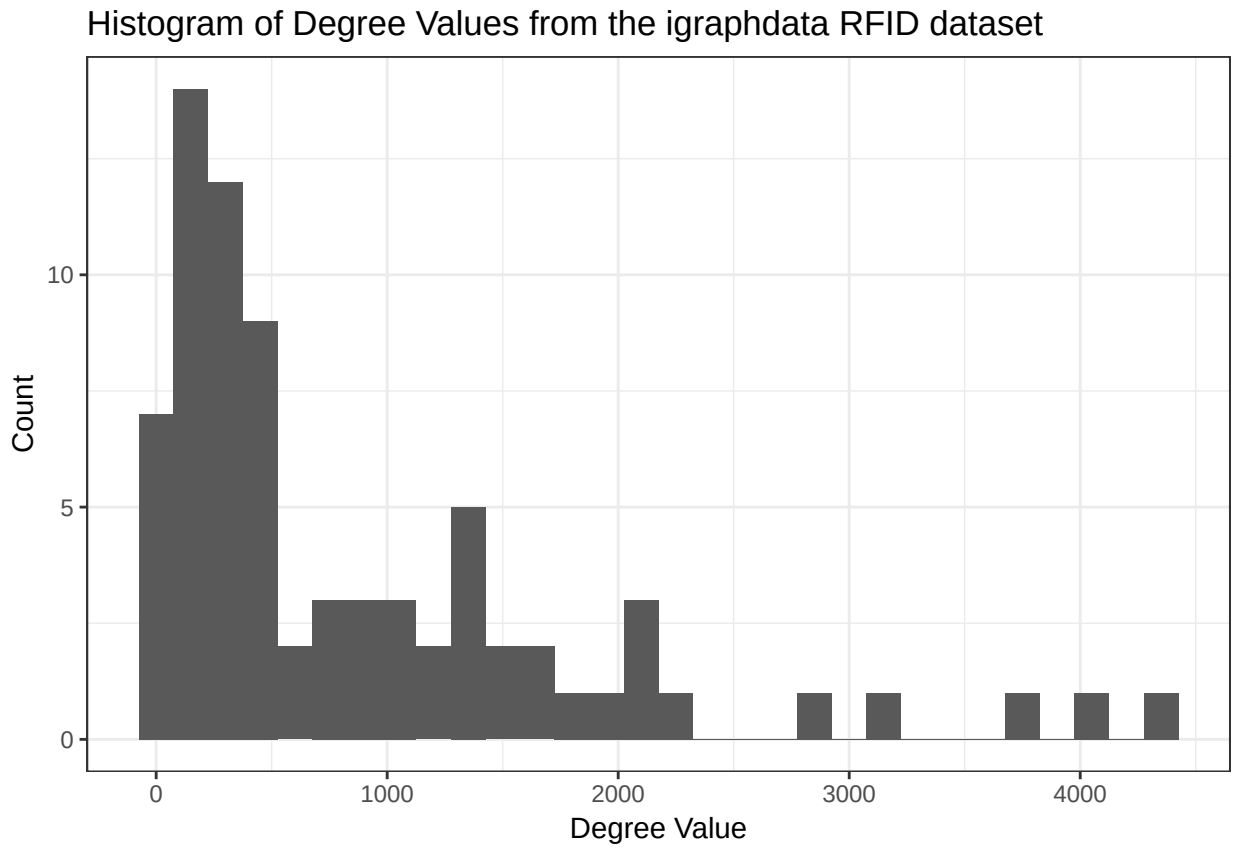


Fig. 3. Histogram of degree values from the RFID network graph. Histogram contains Degree values from all sub-groups within the RFID network.

As we have four groups of people (Admin Staff [ADM], Medical Doctors [MED], Nurses [NUR], and Patients [PAT]) we may also want to see how the distribution of degree may change between these groups. We can use density plots to see how they compare:

```
# A dataframe was created with the generated degree values, node ID (name) and
# status (role). The code for this can be found in Question 7
rfid_df %>%
  ggplot(aes(x = degree, fill = Status)) +
  geom_density(alpha=.5, position="identity") +
  labs(x = "Degree Value",
       y = "Density",
       title = "Density plot of Degree Values from the igraphdata RFID dataset") +
  scale_fill_brewer(palette="Set1") +
  theme_bw()
```

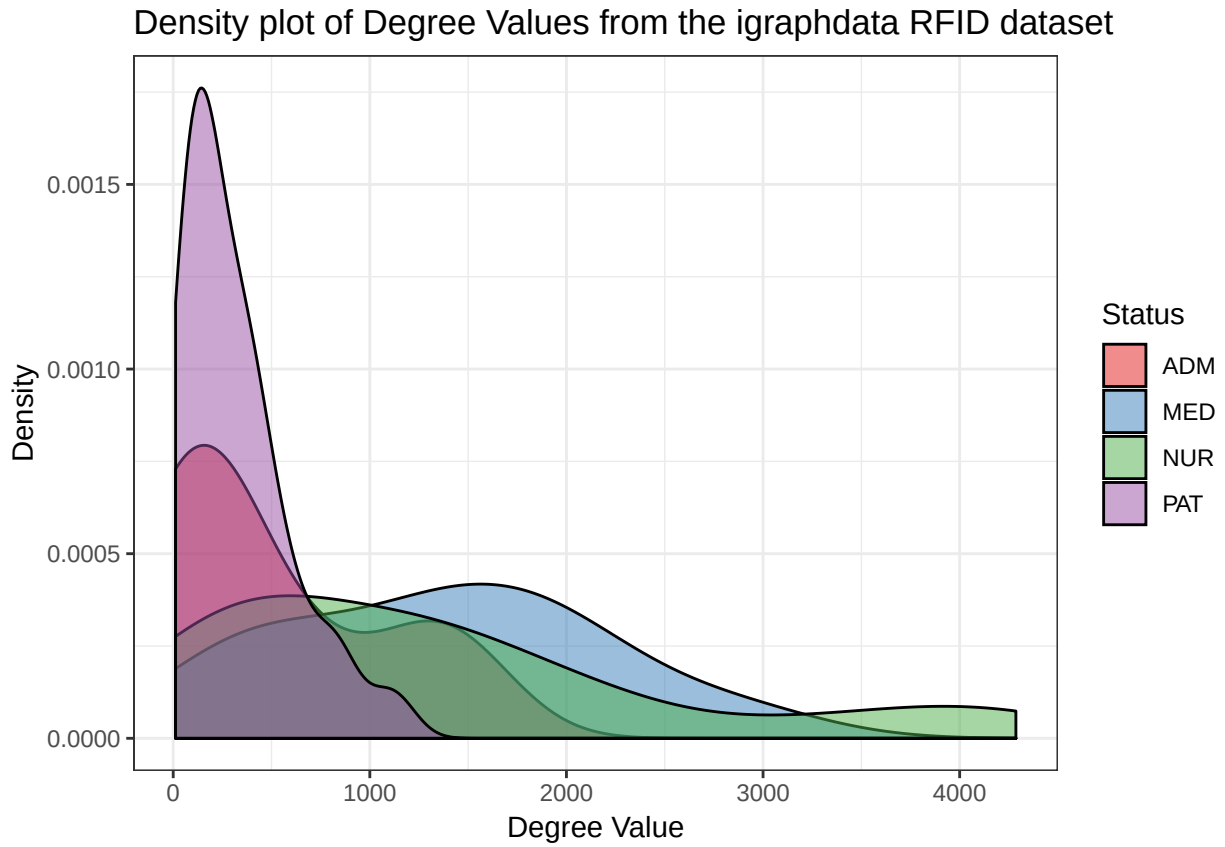


Fig. 4. Density Plots of degree value from the RFID network graph. Colours denote different job role (Admin Staff [ADM], Medical Doctors [MED], Nurses [NUR], and Patients [PAT]).

And plot the cumulative frequency of the degree distribution:

```
deg.dist <- degree_distribution(rfid, cumulative = TRUE, mode = "all")

plot(x = 0:max(deg), y = 1 - deg.dist,
     pch = 19, cex = 1.2, col = "orange",
      xlab = "Degree",
      ylab = "Cumulative Frequency",
      main = "Cumulative Frequency of Degree in the RFID dataset")
```

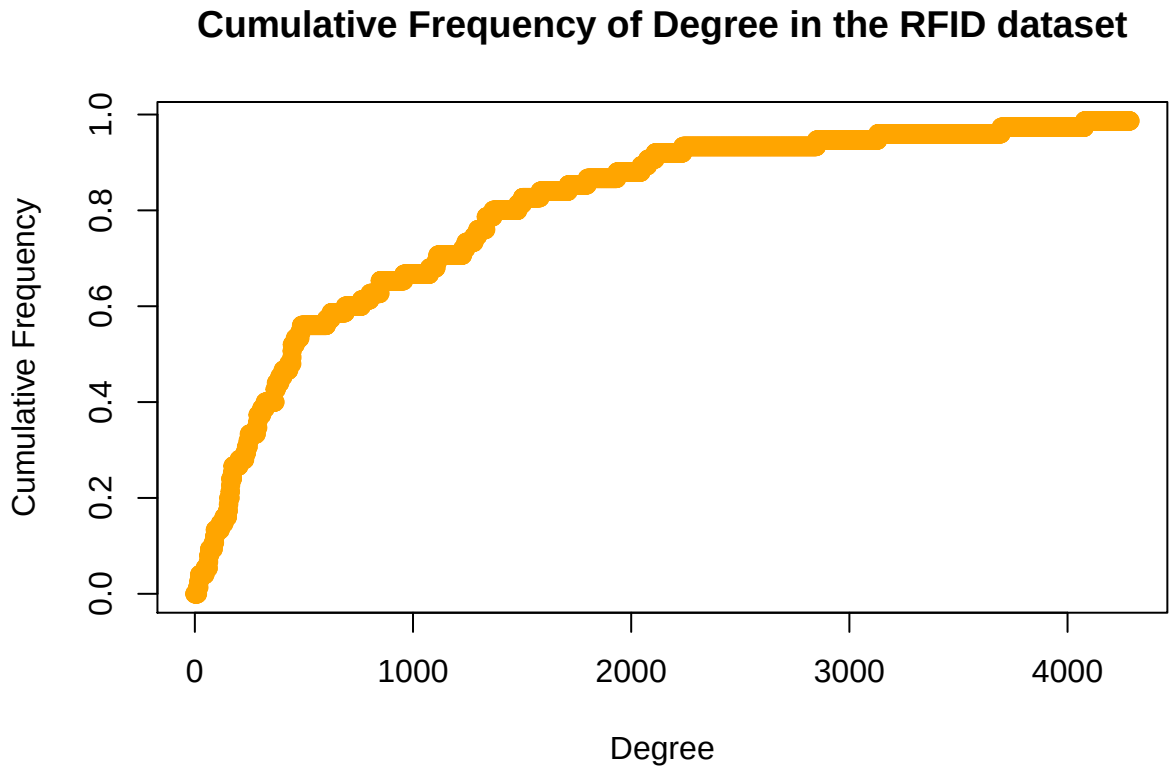


Fig. 5. Cumulative frequency of degree values from the RFID network graph. The degree of a vertex is defined as its number of adjacent edges. Cumulative frequency plot shows degree distibution for the RFID network.

Question Five

5. Plot the results of one community detection algorithm applied to your network.

```
# Using the greedy method (hiearchical)
rfid_greedy <-  cluster_leading_eigen(rfid)

# check the modularity
modularity(rfid_greedy)
```

```
## [1] 0.3472486
```

```
# plot communities with shaded regions
coords = layout_with_fr(rfid)
vertex_colour <- brewer.pal(4, "Set1")

plot(rfid_greedy, rfid, layout = layout_with_fr, vertex.color = vertex_colour,
     main = "RFID Network using the FR Layout")
```

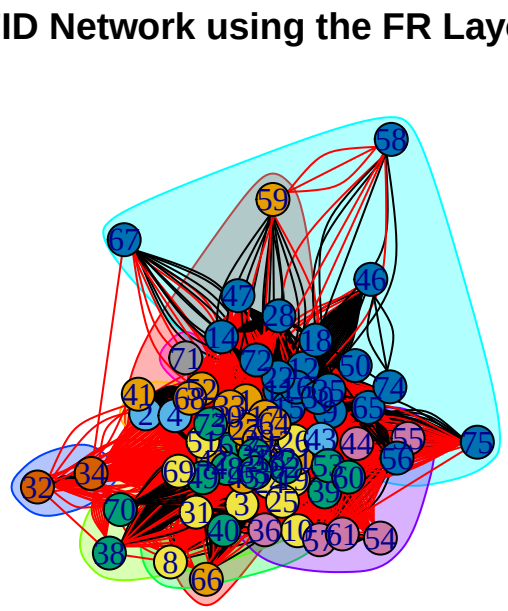


Fig. 6. Communities within the RFID network graph plotted with the FR layout algorithm. Shaded regions indicate densely connected subgraphs calculated using the leading non-negative eigenvector of the modularity matrix of the graph.

Question Six

6. Add figure legends to each figure and a short paragraph at the end of the document interpeting the biology of what is occuring in your network.

The RFID dataset comes from the interactions between people in a hospital over a five-day period and from the graphs alone we can see that our network is complicated, with many different interactions between all four groups of people (Admin Staff [ADM], Medical Doctors [MED], Nurses [NUR], and Patients [PAT]). This is probably as expected - in a hospital it would be expected that there are many interactions between different members of staff and patients. The different distributions of the number of interactions (degree) also show a similar tale. From our density plots in Question 4 we see that Admin staff and Patients have a right-skewed distribution, with nearly all Admin staff and Patients having a degree value of over 500. This is as expected when we consider the dynamics of hospitals - the admin staff and patients are highly stationary, with other staff coming to them. This is seen in the distribution of degree values for Medical staff and Nurses, with their average degree value being much higher, suggesting they are interacting with more people over the five-day RFID data collection period.

Our degree values, centralization values, betweenness centrality values and our subgroup/community structure also attest to the complicated interactions present within the network. All our different methods of calculating closeness and centrality show the values are low compared to the maximum possible values, and it is hard to find methods of trying to distinguish communities within the network that provide useful results (only one has been shown here as was asked). This, combined with the seen variation in contact rate (degree) dependent on type of person, suggests that the network is a 'small-world' type.

Hospital-acquired infections can be a serious public health hazard and knowledge of potential transmission routes can help inform control strategies. Data collection such as this allows for more accurate measurement of potential transmission routes, but the heterogeneity in contact rates and variability in contact numbers suggest that that using contanct network data such as this is important when incorporating data into models of hospital-acquired infections to accurately predict transmission dynamics.

Question Seven

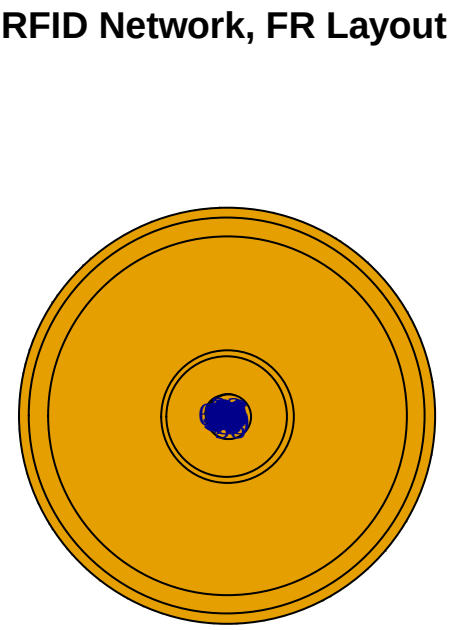
7. Give the code used in an Appendix

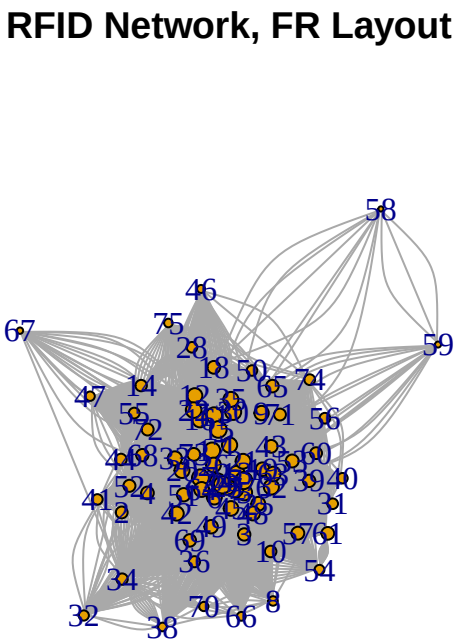
All code used, apart from in Question 4, is shown. A copy of the RMarkdown document and project can be found online at https://github.com/samhillman/sysbio-networks
The graphs mentioned as unsuitable for plotting in Question 4 look like:

```
par(mfrow=c(1,2), mar=c(0,0,0,0)+2)
deg_for_graphs <- igraph::degree(rfid, mode = "in")

plot(rfid, vertex.size = deg_for_graphs,
     layout = layout_with_fr, main = "RFID Network, FR Layout")

plot(rfid, vertex.size = log(deg_for_graphs),
     layout = layout_with_fr, main = "RFID Network, FR Layout")
```





The code for the dataframe in Question 4 (and used in Question Six) is:

```
# This code was also written for adding further vertex attributes if needed (hence the extra
# Columns in the dataframe rf_df)
rf_df <- as_long_data_frame(rfid) %>%
              dplyr::mutate(datetime =
                              lubridate::as_datetime(Time,
                                                     origin = ymd_hms("2010-12-6-13-00-00")),
                     date = as_date(datetime),
                     day_of_week = weekdays(date),
                     time = hour(datetime),
                     from_Status = as.character(`ver[el[, 1], ]`),
                     to_Status = as.character(`ver2[el[, 2], ]`),
                     weight = NA) %>%
              dplyr::select(-`ver[el[, 1], ]`, -`ver2[el[, 2], ]`)

rf_df_status <- tibble(name = c(rf_df$from, rf_df$to),
                   Status = c(rf_df$from_Status, rf_df$to_Status)) %>%
                   distinct()

rfid_df <- rf_df_status %>%
              dplyr::arrange(name) %>%
              dplyr::mutate(degree = deg)
```