Question 2:

Explanations:

To generate the MSTs and RNGs the euclidean distance measure was used to generate a distance matrix, this was achieved by looping through the matrixes comparing columns or rows and determining their distances from each other by that measure. The following code determines the euclidean distance of two objects.

```
euclidean_dist <- function(x, y) {
  sqrt(sum((x - y)^2))
}
```

Code for MSTs:

```
generate_distances <- function(datafile, filename1,filename2){
  row_matrix <- get_distance_rows(datafile)
  cols_matrix <- get_distance_cols(datafile)
  name1 = paste("Results/",filename1,"distancematrix.csv", sep = "")
  name2 = paste("Results/",filename2,"distancematrix.csv", sep = "")
  write.csv(row_matrix,file = name1)
  write.csv(cols_matrix,file = name2)
  relative_neighbourhoods(row_matrix,filename1)
  relative_neighbourhoods(cols_matrix,filename2)

  rows_graph <- mst(graph_from_adjacency_matrix(row_matrix, mode = "undirected",weighted = TRUE))
  cols_graph <- mst(graph_from_adjacency_matrix(cols_matrix, mode = "undirected",weighted = TRUE))

  V(rows_graph)$label <- colnames(row_matrix)
  E(rows_graph)$label <- round(E(rows_graph)$weight, 3)
  V(cols_graph)$label <- colnames(cols_matrix)
  E(cols_graph)$label <- round(E(cols_graph)$weight, 3)
  print(sum(E(rows_graph)$weight))
  print(sum(E(cols_graph)$weight))
  name1 = paste("Results/",filename1,"MST.gml", sep = "")
  name2 = paste("Results/",filename2,"MST.gml", sep = "")

  write_graph(rows_graph, name1,format = 'gml')
  write_graph(cols_graph,name2,format = 'gml')
}
```

The mst() function in R uses prims algorithm to find the path between nodes with the lowest total edge weight by chossing a random vertex, adding the lowest weighted edge to the minimum spanning tree and repeating the process, however if the lowest weighted edge forms a loop it is ignored.

Code for RNGs:

```
relative_neighbourhoods <- function(matrix,title)
{
  matrix1 <- as.matrix(matrix)
  rng_graph <- rng(dx = matrix1,r = 1, algorithm = 'cover_tree')
  print(rng_graph)
  rng_graph <- as.undirected(rng_graph)
  #adds labels
  V(rng_graph)$label <- colnames(matrix1)
  E(rng_graph)$weight <- apply(get.edges(rng_graph, 1:gsize(rng_graph)),1,function(x)matrix1[x[1],x[2]])
  E(rng_graph)$label <- round(E(rng_graph)$weight, 3)
  name <- paste("Results/",title,"RNG.gml", sep ="")
  plot(rng_graph)
  write_graph(rng_graph, name, format = 'gml')
}
```

The rng() function takes a distance matrix as input and uses the cover tree algorithm to generate a relative neighbourhood graph. The apply function is used to apply edge weights to the new graph by looping through the original matrix and finding the cell that had the edge as the current edge within the graph and applying the weight of that edge as a label within the graph.