optimization work sample

Jonny Mills

The following 2 questions are a sample of work that I did in my Optimization class earlier this fall.

n the first question, I show code to create a sparse matrix. In the second question, I use Prim's algorithm to calculate a minimum spanning tree froma an adjacency matrix containing the Euclidian distance between every pair of nodes.

Question 1:

Write code which will produce, from a matrix d, the sparse matrix ds in that it is a list of edges, where each row contains a source node, destination node, and edge weight.

```
######################
###### Question 1 ####
######################
rm(list=ls())

n <- 100
d <- runif(n*n)
d[d < 0.80] <- NA
d <- matrix(d,nrow=n,ncol=n) #reshape the vector

diag(d) <- NA # no self-loops
d[upper.tri(d)] = t(d)[upper.tri(d)] # undirected graphs are symmetric

AdjMatrix2List <- function(d) {
      ds = matrix(nrow=n^2, ncol = 3)
      k=1

      for (i in 1:nrow(d)){
            for (j in 1:nrow(d)){
                  if (!is.na(d[i,j])){
                        ds[k,] <- c(i,j,d[i,j])
                        k = k + 1
                  }
            }
      }
      return(ds)
}
print(head(AdjMatrix2List(d)))
```

```
##      [,1] [,2]      [,3]
## [1,]    1   13 0.9070716
## [2,]    1   16 0.9833869
## [3,]    1   17 0.9927738
## [4,]    1   22 0.9494699
## [5,]    1   30 0.8597420
## [6,]    1   36 0.8864711
```

Question 2

1) Create an adjacency matrix d by calculating the Euclidean distance between every pair of points (xi,yi),(xj,yj).
2) Calculate the minimum spanning tree using Kruskal or Prim. Store the result in the variable ds.mst.

3)The last step is to produce a plot visualizing your minimum spanning tree, using ds.mst$tree.arcs.

```
#########################
##### QUESTION 2 ########
#########################

library(optrees)
```

```
## Warning: package 'optrees' was built under R version 3.4.2
```

```
## Loading required package: igraph
```

```
## Warning: package 'igraph' was built under R version 3.4.2
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

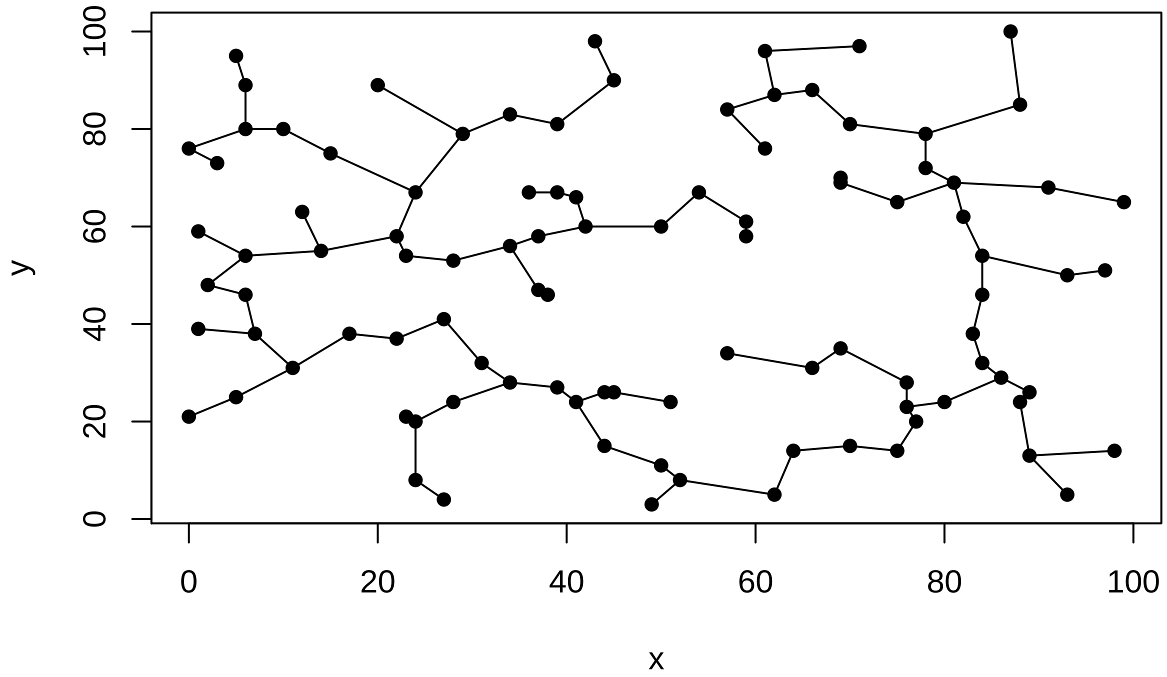
```
n <- 100
x <- round(runif(n)*100)
y <- round(runif(n)*100)
plot(x,y,pch=16)

#1. .
d <- matrix(x,y,nrow = n, ncol = n)

for (i in 1:n){
   for (j in 1:n){
      d[i,j] <- sqrt((y[j]-y[i])^2 + (x[j] - x[i])^2)
   }
}

#2.
ds.mst <- msTreePrim(1:n, AdjMatrix2List(d)) # nodes, arcs str(ds.mst)
plot.mst <- function (arcList) {
      start = arcList[,1]
      end = arcList[,2]
   for(i in 1:length(arcList[,1])){
      segments(x[start][i], y[start][i], x1 = x[end][i], y1 = y[end][i])
   }
}

#3.
plot(x,y,pch=16)
plot.mst (ds.mst$tree.arcs)
```


0
20
40
60
80
100
y
0
20
40
60
80
100
x