

# Final\_Project

Phyllis Chen

2024-03-13

2.1 genNodes() function & acceptance-rejection sampling with plot

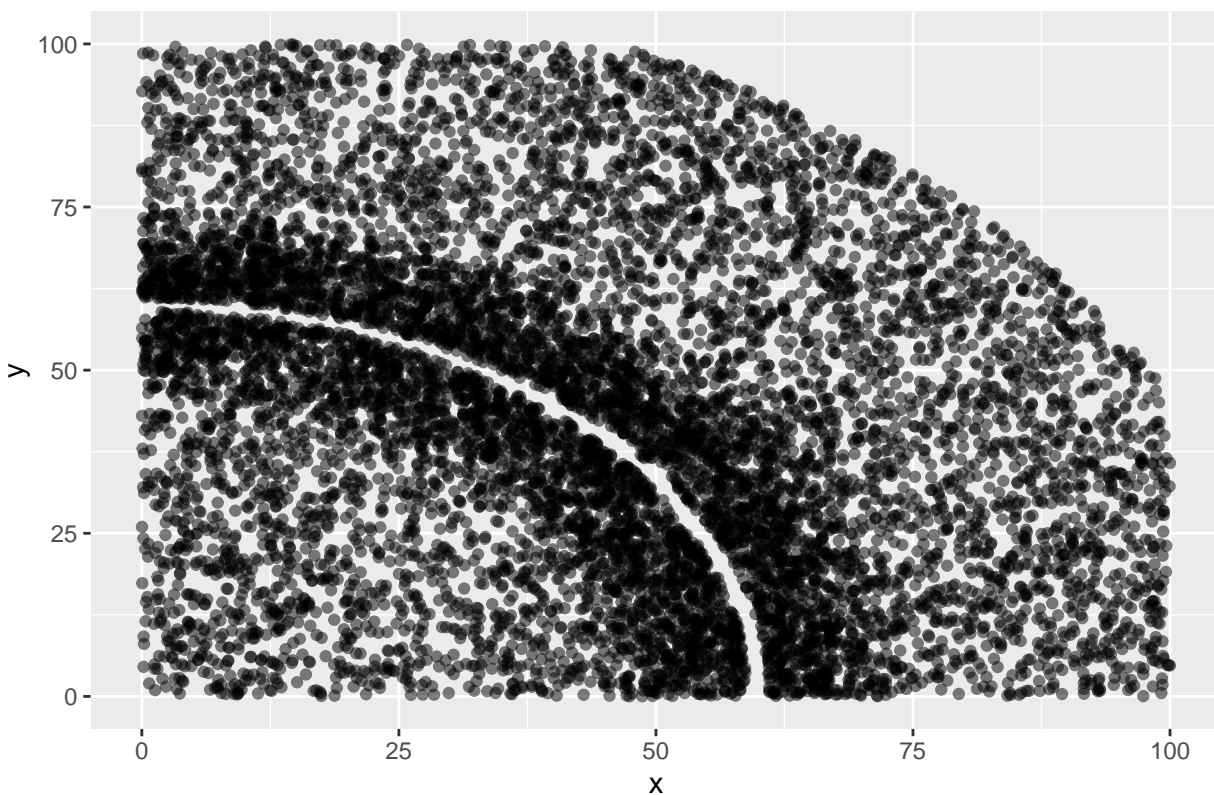
```
library(RSpectra)
library(MASS)
library(ggplot2)
#set.seed(123)

source("nodeDensity.R")
genNodes = function(n) {
  x<- numeric(0)
  y <- numeric(0)
  M = 4
  #define grid point, find max value of grid points, find max overall

  while (length(x) < n) {
    xn = runif(1, 0, 100)
    yn = runif(1, 0, 100)
    u = runif(1)
    if (u*M <= nodeDensity(xn, yn)) {
      x = c(x, xn)
      y = c(y, yn)
    }
  }
  if (length(x) > n) {
    x = x[-((n + 1) : length(x))]
    y = y[-((n + 1) : length(y))]
  }
  nodes <- matrix(c(x, y), nrow = n, ncol = 2)
  return(nodes)
}

nodes1 = genNodes(10000)
ggplot(data.frame(x = nodes1[, 1], y = nodes1[, 2])) + geom_point(mapping = aes(x, y), alpha = 0.5) + xlab("x") + ylab("y")
```

Scatterplot of 10000 generated Nodes



2.2 function findRc() & 3. function findTranMat(), function getEigen2(), function findRange(),

*#finds the range of possible values to search for Rc*

`findRange = function(mat) {`

*# minimum and maximum values of the mat*

`gr_row_min = max(apply(mat, 1, function(row) min(row[row > 0], na.rm = TRUE)))`

*# i think this is wrong*

*# gr\_row\_min = max(apply(mat, 1, min, na.rm = TRUE))*

`sm_row_max = min(apply(mat, 1, max, na.rm = TRUE))`

*# need each row of mat that computes mim value excluding 0, MAX OF MIN values. Has to be AT LEAST AS*

*#find the greatest row minimum in matrix*

`L = gr_row_min`

`U = sm_row_max`

*#returns interval min and max*

`return(c(L, U))`

`}`

*#finds transition matrix based on distance mat and R*

`findTranMat = function(mat, R) {`

`P = matrix(0, nrow = nrow(mat), ncol = ncol(mat))`

*#for each node*

`for (i in 1:nrow(mat)) {`

`Si = which(mat[i, ] <= R)`

`P[i, Si] = 1/length(Si)`

`}`

```

    # returns the transition matrix based on above two inputs
    return(P)
}

getEigen2 = function(mat) {
  # returns magnitude of second largest eig value
  eigenvalues = eigen(mat)$values
  sorted_eigenvalues = sort(eigenvalues, decreasing = TRUE)

  # Returns numeric value mag of second largest eig of mat
  return(Mod(sorted_eigenvalues[2]))
}

# find rc function
findRc = function(nodes, tol = 0.05) {
  mat = as.matrix(dist(nodes))
  range = findRange(mat)
  # Rmin and Rmax
  Rmin = range[1]
  Rmax = range[2]

  RO_bar = (Rmin + Rmax) / 2

  #bisection method
  while(abs(Rmin - Rmax) > tol) {
    # transition matrix for R
    P = findTranMat(mat, RO_bar)
    # if R gives connected network
    #this from rspectra??? not sure if correct
    if (abs(getEigen2(P) - 1) > 1e-7 & getEigen2(P) < 1) {
      Rmax = RO_bar
    } else {
      Rmin = RO_bar
    }

    RO_bar = (Rmin + Rmax) / 2
  }
  return(Rmax)
}

```

```

sample_size = c(50, 100, 150)
network = 1000

Rc_val = lapply(sample_size, function(n) {
  replicate(1000, {
    set.seed(n)
    nodes = genNodes(n)
    findRc(nodes, tol = 0.0001)
  })
})

```

density plot with all sample sizes

```

library(tidyr)
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##      select

## The following objects are masked from 'package:stats':
##
##      filter, lag

## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union

sample_size2 = c(50, 100, 150, 200, 250)
networks2 = 50

Rc_val2 = matrix(0, nrow = length(sample_size2), ncol = networks2)
seed_list2 = c(1:networks2)
for (i in 1:length(sample_size2)) {
  n = sample_size2[i]
  for (seed in seed_list2) {
    set.seed(seed)
    nodes = genNodes(n)
    Rc_val2[i, seed] = findRc(nodes)
  }
}

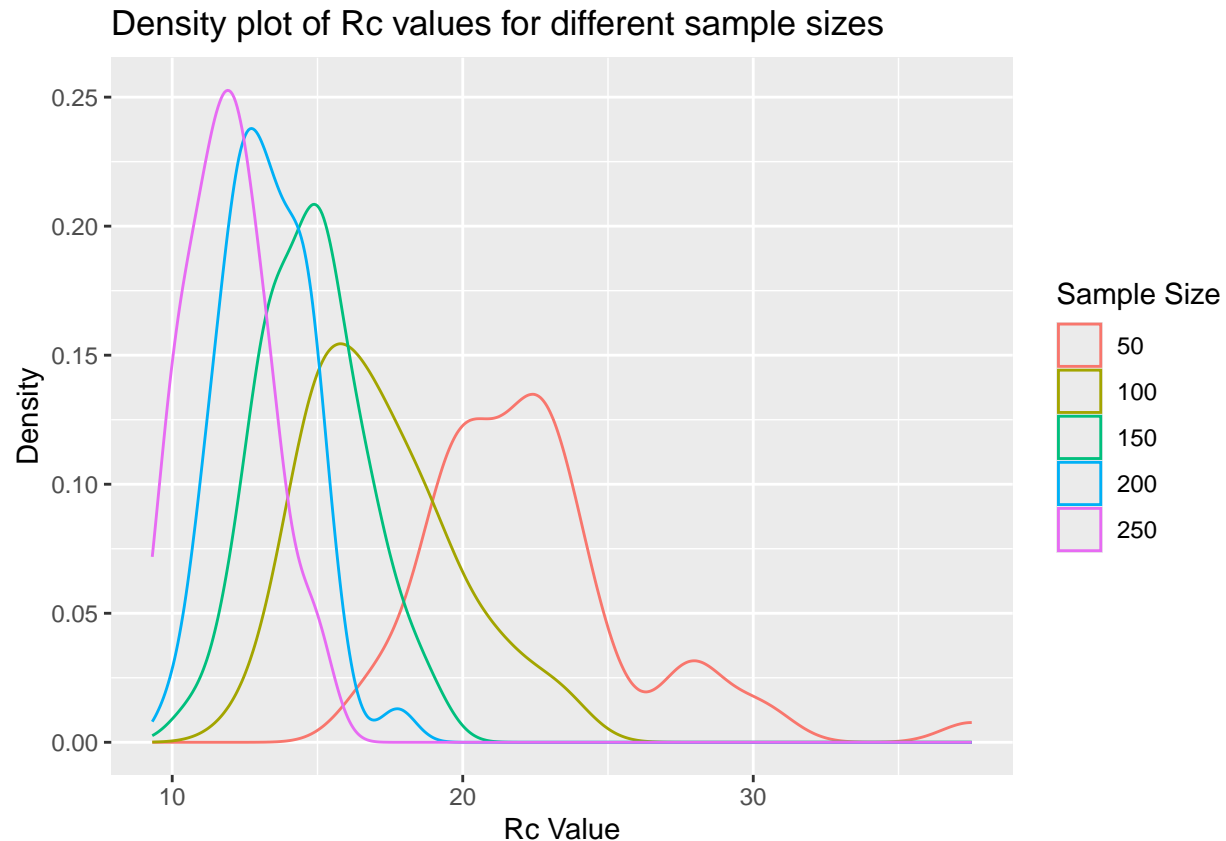
seed_list2 = c(1:networks2)

Rc_df2 = data.frame(Rc = as.vector(t(Rc_val2)),
                    sample_size2 = rep(sample_size2, each = ncol(Rc_val2)),
                    seed = rep(seed_list2, times = nrow(Rc_val2)))

Rc_df2$sample_size2 = as.factor(Rc_df2$sample_size2)

ggplot(Rc_df2, aes(x = Rc, color = sample_size2)) +
  geom_density() +
  labs(x = "Rc Value", y = "Density", color = "Sample Size",
       title = "Density plot of Rc values for different sample sizes")

```



3a The Rc with smaller radius changes with node configurations because they are more spread out than the ones that are higher. When i look at the node network with minimum values corresponding to Rc they are more spread out across the X and Y and have various small concentrations, rather than one concentration in the middle or many at once. 3b For all sample sizes, they are not symmetric mostly and most of htem are right skewed. For the smalles sample size of 50, it is more symmetric than the others, but a little bit has a longer right tail. All of them have a longer right tail. The biggest sample size is the most right skewed, and it is also the highest density and heavy-tailed. It is the most heavy tailed. Each one as the sample size decreases becomes a little bit more symmetric and less right-skewed, and also less heavy-tailed. They are not multimodal because they mostly all only have one peak.

```
seed_list = c(1:network)

Rc_val_list = lapply(seed_list, function(seed) {
  set.seed(seed)
  nodes = genNodes(50)
  findRc(nodes)
})

Rc_val50 = do.call(rbind, Rc_val_list)

mean_Rc = mean(Rc_val50)
median_Rc = median(Rc_val50)
max_Rc = max(Rc_val50)
min_Rc = min(Rc_val50)

median_seed <- which.min(abs(Rc_val50 - median_Rc))
```

```

mean_seed <- which.min(abs(Rc_val50 - mean_Rc))

max_seed <- which(Rc_val50 == max_Rc)
min_seed <- which(Rc_val50 == min_Rc)

set.seed(max_seed)

max_network = genNodes(50)

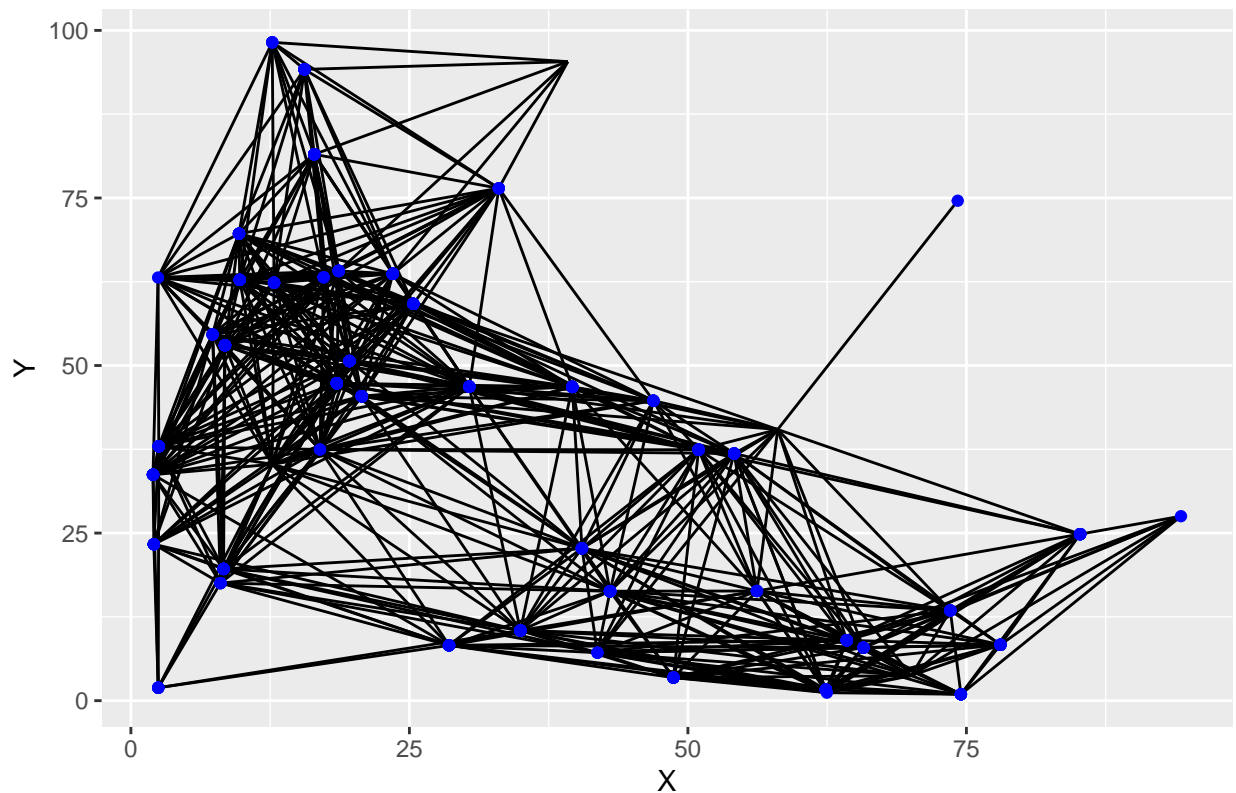
df_max = data.frame()
nodes = 50
for (i in 1:(nrow(max_network)-1)) {
  for (j in (i + 1):nrow(max_network)) {
    if(sqrt(sum((max_network[i, ] - max_network[j, ])^ 2)) <= max_Rc) {
      df1 = data.frame(x = max_network[i, 1], y = max_network[i, 2], xend = max_network[j, 1], yend = max_network[j, 2])
      df_max = rbind(df_max, df1)
    }
  }
}

plot_max <- ggplot(df_max, aes(x = x, y = y)) +
  geom_segment(aes(xend = xend, yend = yend)) +
  geom_point(color = "blue") +
  labs(x = "X", y = "Y", title = "Node Network Corresponding to maximum values of Rc: 37.80386")

plot_max

```

Node Network Corresponding to maximum values of Rc: 37.80386



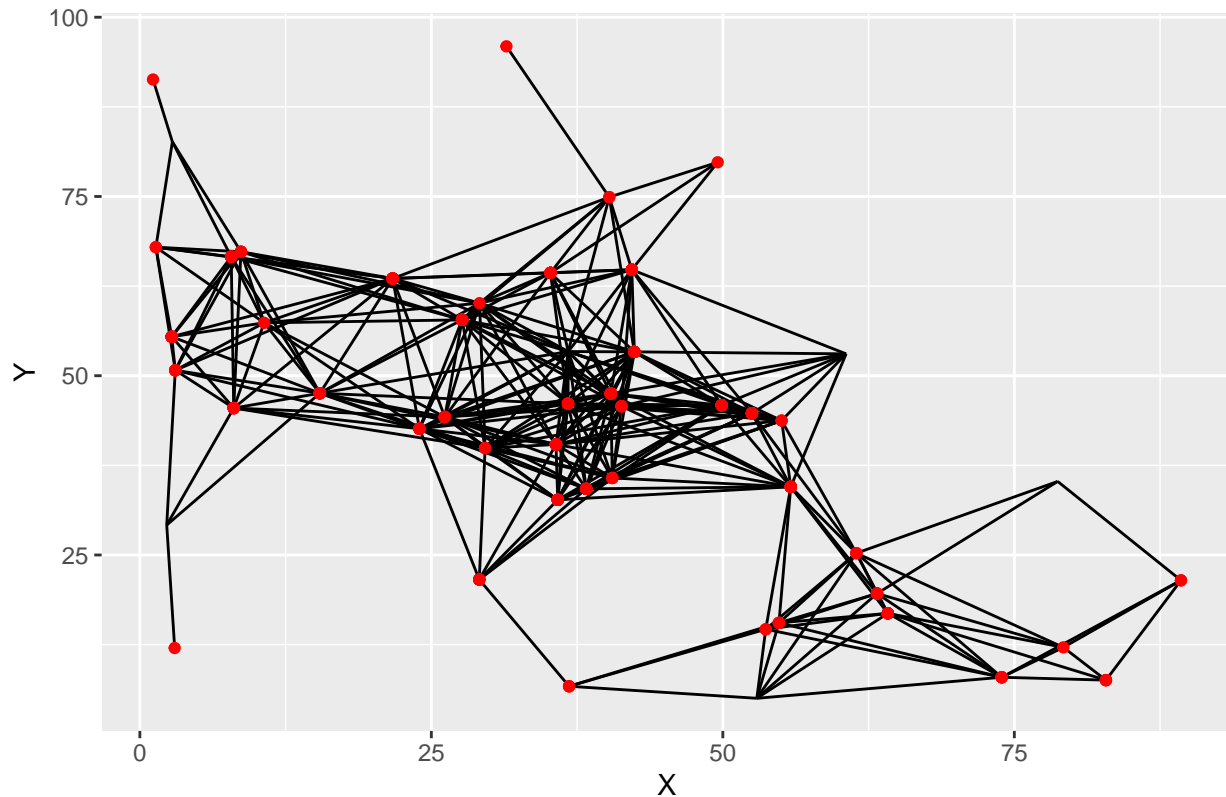
```
set.seed(mean_seed)
mean_network = genNodes(50)

df_mean = data.frame()
nodes = 50
for (i in 1:(nrow(mean_network)-1)) {
  for (j in (i + 1):nrow(mean_network)) {
    if(sqrt(sum((mean_network[i, ] - mean_network[j, ])^ 2)) <= mean_Rc) {
      df2 = data.frame(x = mean_network[i, 1], y = mean_network[i, 2], xend = mean_network[j, 1], yend = mean_network[j, 2])
      df_mean = rbind(df_mean, df2)
    }
  }
}

plot_mean <- ggplot(df_mean, aes(x = x, y = y)) +
  geom_segment(aes(xend = xend, yend = yend)) +
  geom_point(color = "red") +
  labs(x = "X", y = "Y", title = "Node Network Corresponding to mean values of Rc: 22.8528")

plot_mean
```

Node Network Corresponding to mean values of Rc: 22.8528



```
set.seed(median_seed)
median_network = genNodes(50)

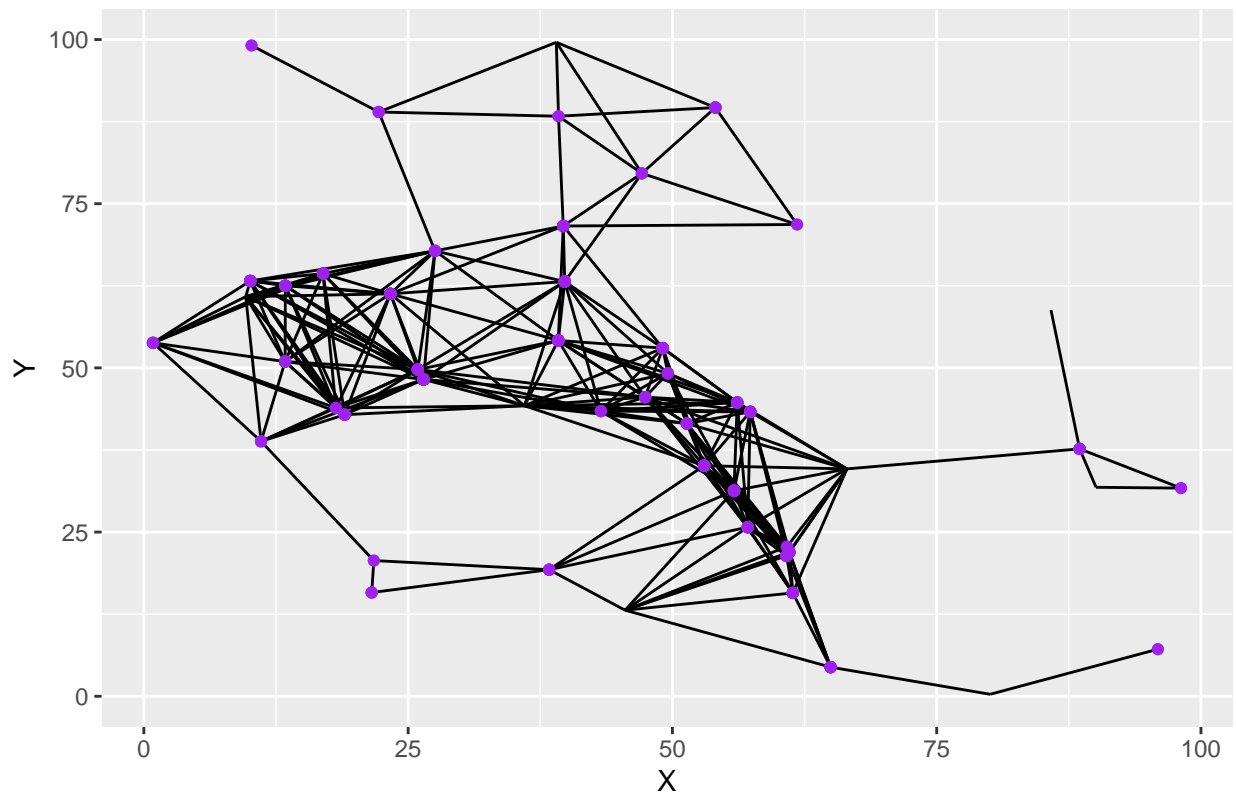
df_med = data.frame()
nodes = 50
for (i in 1:(nrow(median_network)-1)) {
  for (j in (i + 1):nrow(median_network)) {
    if(sqrt(sum((median_network[i, ] - median_network[j, ])^ 2)) <= median_Rc) {
      df3 = data.frame(x = median_network[i, 1], y = median_network[i, 2], xend = median_network[j, 1],
        yend = median_network[j, 2])
      df_med = rbind(df_med, df3)
    }
  }
}

plot_med <- ggplot(df_med, aes(x = x, y = y)) +
  geom_segment(aes(xend = xend, yend = yend)) +
  geom_point(color = "purple") +
  labs(x = "X", y = "Y", title = "Node Network Corresponding to median values of Rc:22.24529 ")

plot_med
```



Node Network Corresponding to median values of Rc:22.24529



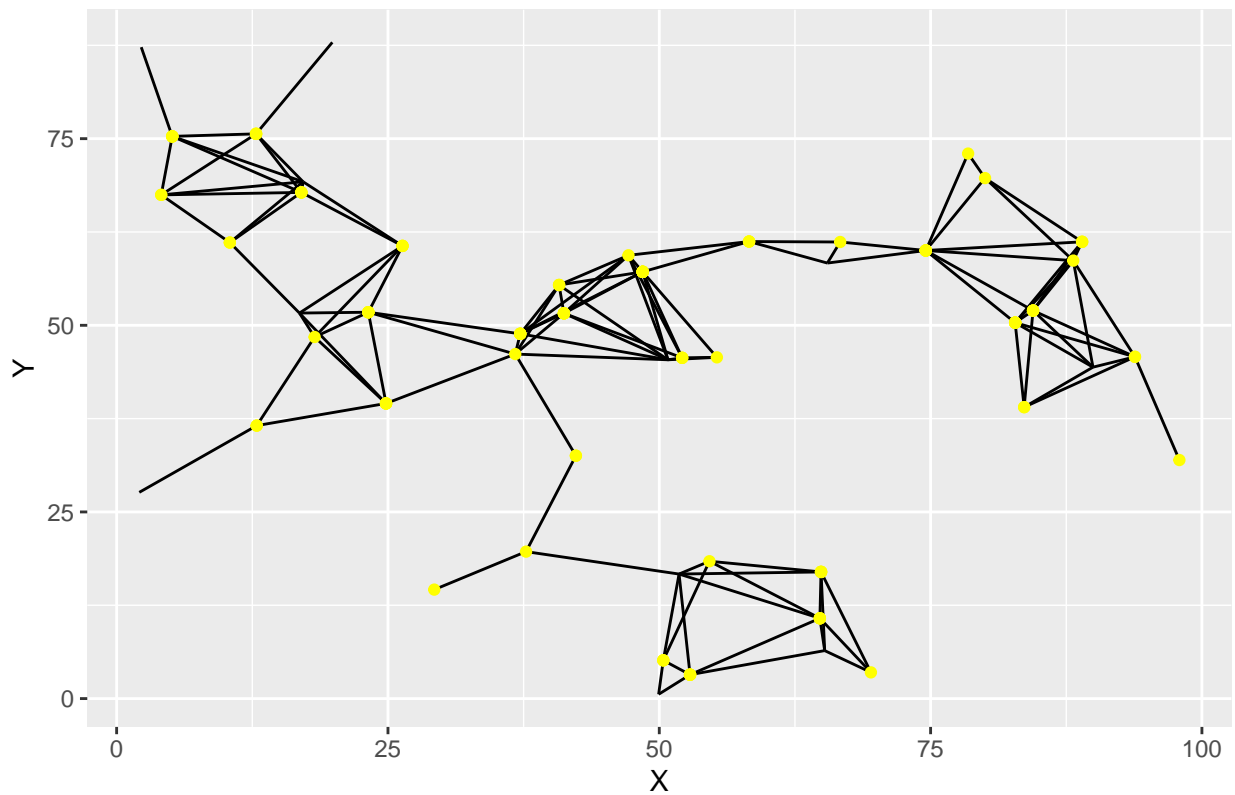
```
set.seed(min_seed)
min_network = genNodes(50)

df_min = data.frame()
nodes = 50
for (i in 1:(nrow(min_network)-1)) {
  for (j in (i + 1):nrow(min_network)) {
    if(sqrt(sum((min_network[i, ] - min_network[j, ])^ 2)) <= min_Rc) {
      df4 = data.frame(x = min_network[i, 1], y = min_network[i, 2], xend = min_network[j, 1], yend = min_network[j, 2])
      df_min = rbind(df_min, df4)
    }
  }
}

plot_min <- ggplot(df_min, aes(x = x, y = y)) +
  geom_segment(aes(xend = xend, yend = yend)) +
  geom_point(color = "yellow") +
  labs(x = "X", y = "Y", title = "Node Network Corresponding to minimum values of Rc: 14.75555")

plot_min
```

Node Network Corresponding to minimum values of Rc: 14.75555



```
df_max$category <- "Max"
df_mean$category <- "Mean"
df_med$category <- "Median"
df_min$category <- "Min"

df_all <- rbind(df_max, df_mean, df_med, df_min)

df_all$category <- as.factor(df_all$category)

ggplot(df_all, aes(x = x, y = y, color = category)) +
  geom_segment(aes(xend = xend, yend = yend)) +
  geom_point() +
  labs(x = "X", y = "Y", title = "Combined Node Network") +
  scale_color_discrete(name = "Category")
```

Combined Node Network

