Network Data Analysis HW5

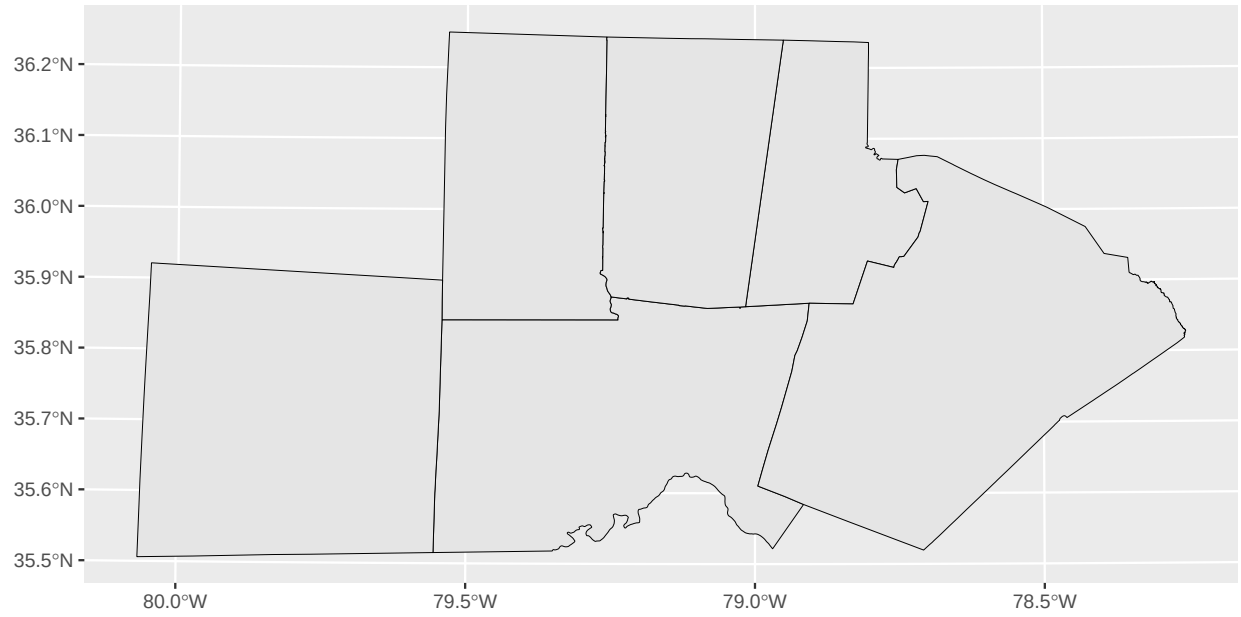
Julia Bright

2025-04-11

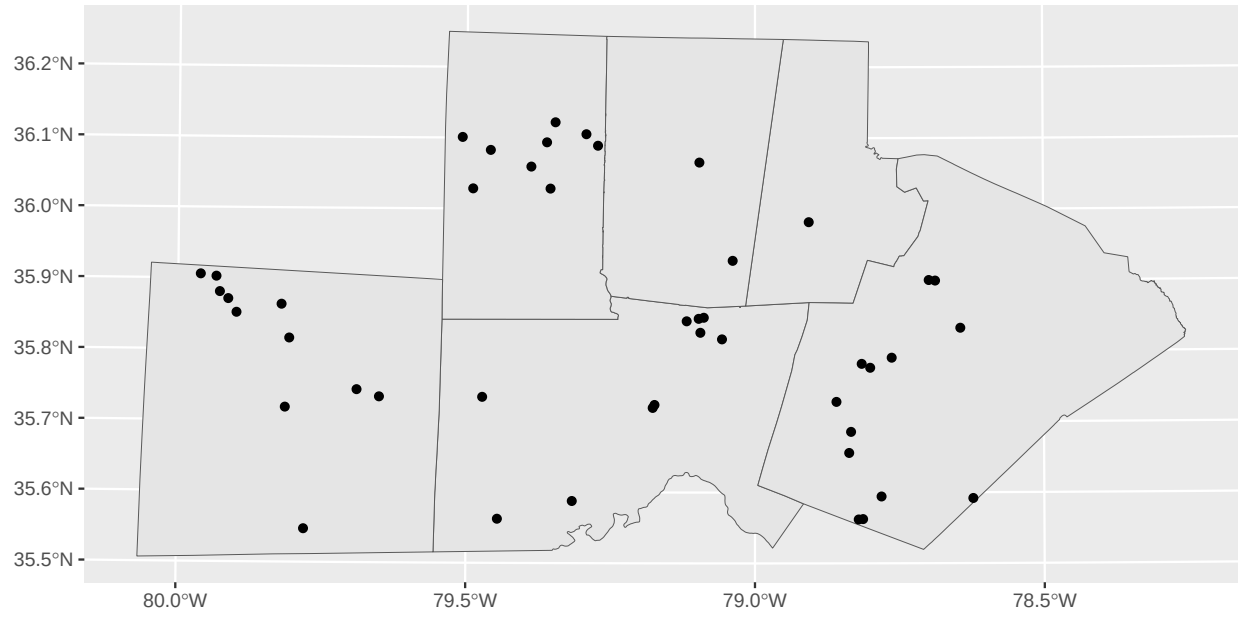
```
#read in the data
networks <- st_read("plan372_hmks/hw5/network_data/network.gpkg")
connected_points <- st_read("plan372_hmks/hw5/network_data/connected_points.gpkg")
unconnected_points <- st_read("plan372_hmks/hw5/network_data/unconnected_points.gpkg")
#load in map of NC counties
map <-
↪  st_read("/Users/julia/Library/CloudStorage/OneDrive-UniversityofNorthCarolinaatChapelHill/Documents/
```

Question 1

```
#filter counties in/around the triangle
map = map %>%
  filter(County == "Randolph" | County ==  "Alamance" | County == "Chatham" | County ==
  ↪  "Wake" | County ==  "Orange" | County == "Durham")

#background map
ggplot()+
  geom_sf(,map, color = "black")
```

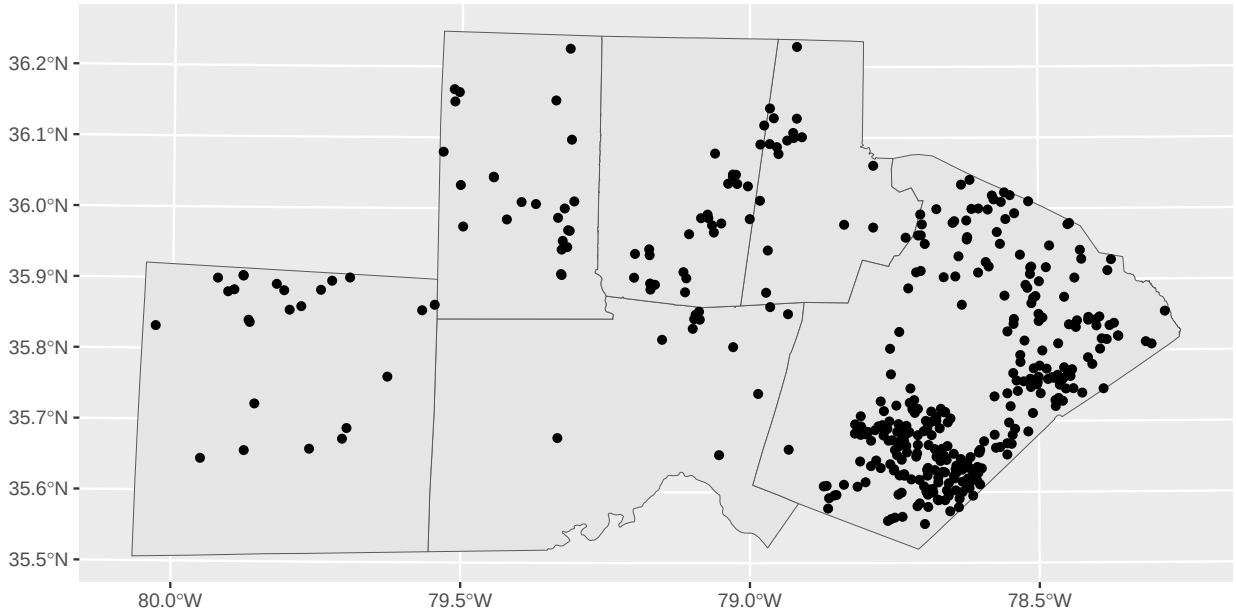



```
#map of connected water systems
ggplot() +
  geom_sf(data=map) +
   geom_sf(data=connected_points)
```

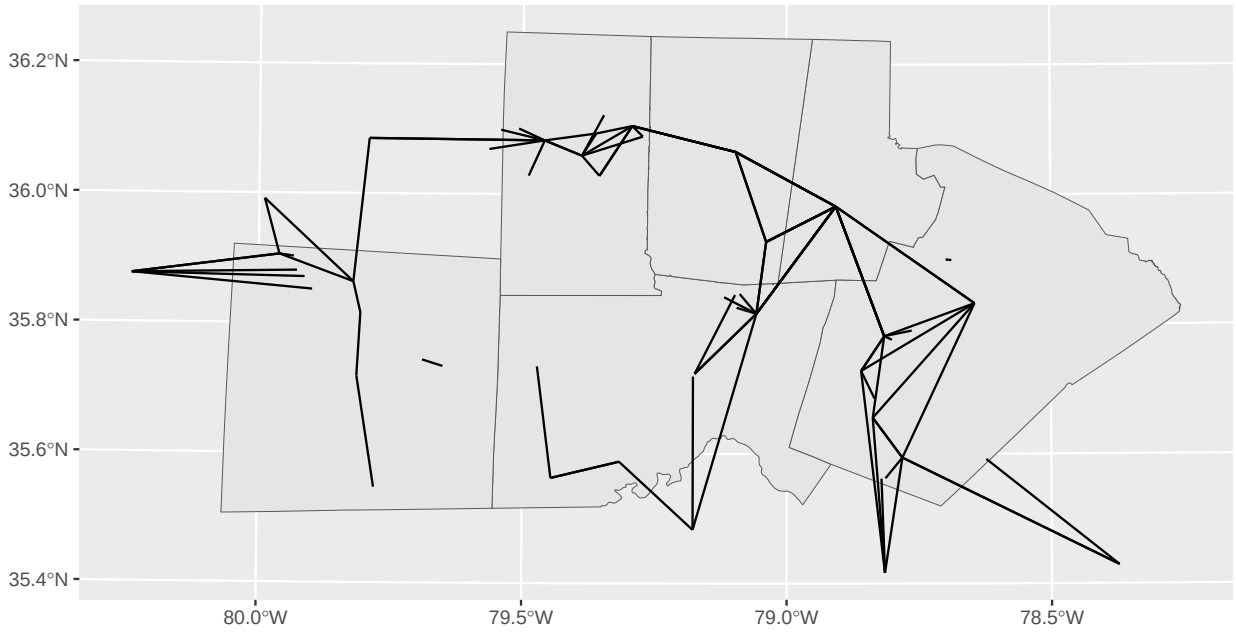



```
#map of unconnected water systems
ggplot() +
```

```
  geom_sf(data=map) +
   geom_sf(data=unconnected_points)
```

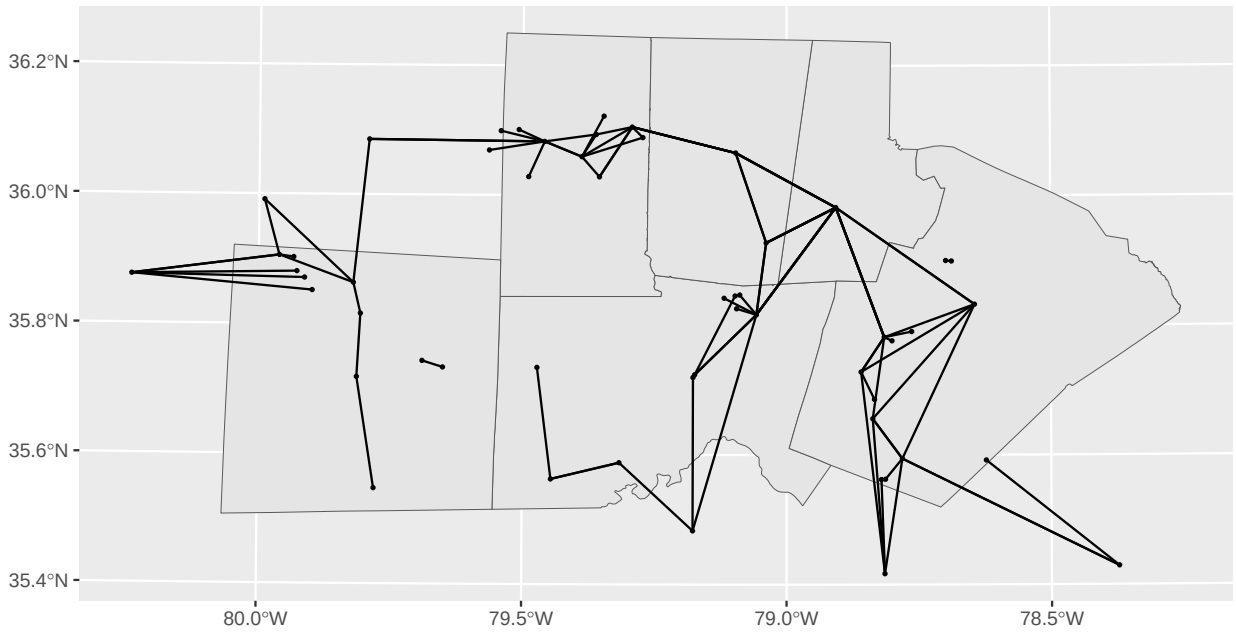



```
#map of network of interconnections
ggplot() +
  geom_sf(data=map) +
   geom_sf(data=networks)
```



Question 2

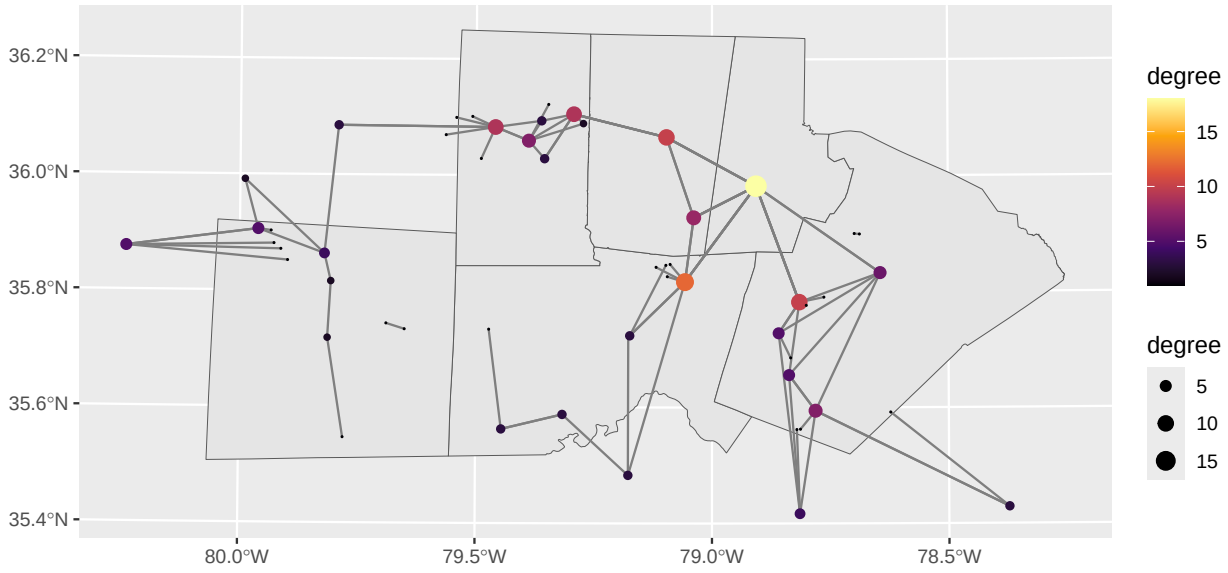
```
#give each edge a unique index
edges <- networks %>%
  mutate(edgeID = c(1:n()))

#create nodes at the start and end point of each edge
nodes <- edges %>%
  st_coordinates() %>%
  as_tibble() %>%
  rename(edgeID = L1) %>%
  group_by(edgeID) %>%
  slice(c(1, n())) %>%
  ungroup() %>%
  mutate(start_end = rep(c('start', 'end'), times = n()/2))

#give each node a unique index
nodes <- nodes %>%
  mutate(xy = paste(.$X, .$Y)) %>%
  mutate(nodeID = group_indices(., factor(xy, levels = unique(xy)))) %>%
  select(-xy)

#combine the node indices with the edges
source_nodes <- nodes %>%
  filter(start_end == 'start') %>%
  pull(nodeID)
target_nodes <- nodes %>%
  filter(start_end == 'end') %>%
  pull(nodeID)
edges = edges %>%
  mutate(from = source_nodes, to = target_nodes)

#remove duplicate nodes
nodes <- nodes %>%
  distinct(nodeID, .keep_all = TRUE) %>%
  select(-c(edgeID, start_end)) %>%
  st_as_sf(coords = c('X', 'Y')) %>%
  st_set_crs(st_crs(edges))

#convert to tbl_graph
graph = tbl_graph(nodes = nodes, edges = as_tibble(edges), directed = FALSE)

graph <- graph %>%
  activate(edges) %>%
  mutate(length = st_length(geom))

#map the network nodes and edges in a graph
map_network <- ggplot() +
  geom_sf(data=map) +
  geom_sf(data = graph %>% activate(edges) %>% as_tibble() %>% st_as_sf()) +
  geom_sf(data = graph %>% activate(nodes) %>% as_tibble() %>% st_as_sf(), size = 0.5)

map_network
```



Question 3

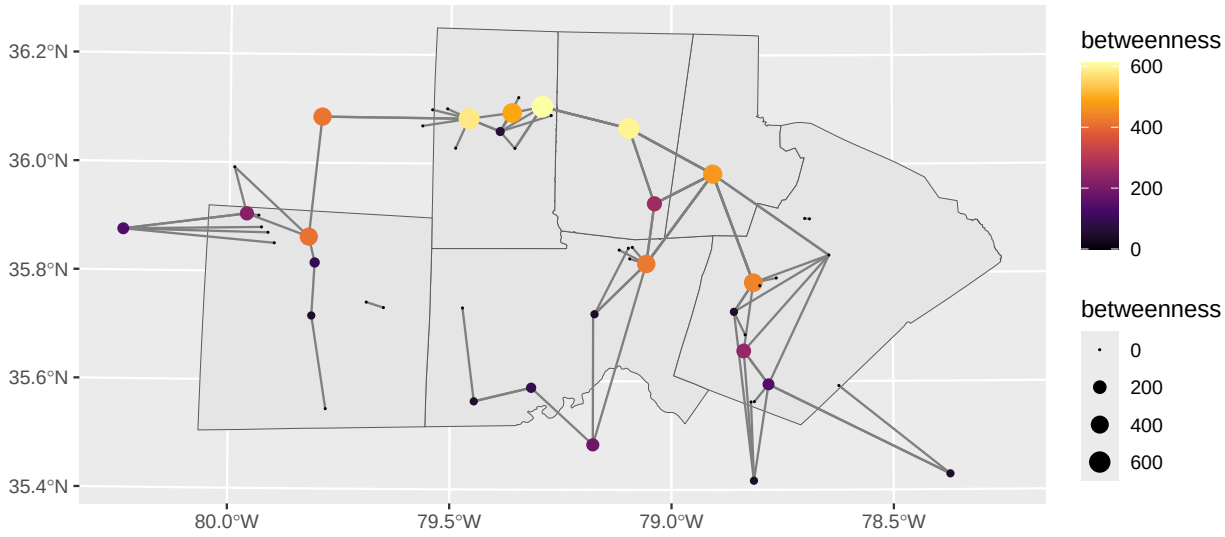
```
#compute centrality measures
graph <- graph %>%
  activate(nodes) %>%
  mutate(degree = centrality_degree()) %>%
  mutate(betweenness = centrality_betweenness(weights = length)) %>%
  activate(edges) %>%
  mutate(betweenness = centrality_edge_betweenness(weights = length))

#Graph of betweenness measures on network edges is difficult to interpret and seems less
↪  informative

#ggplot() +
  #geom_sf(data = map) +
  #geom_sf(data = graph %>% activate(edges) %>% as_tibble() %>% st_as_sf(), #aes(col =
  ↪  betweenness, size = betweenness)) +
 # scale_colour_viridis_c(option = 'inferno') +
  #scale_size_continuous(range = c(0,4))

#graph of degree centrality for nodes on the interconnections network
map_degree <- ggplot() +
  geom_sf(data = map) +
  geom_sf(data = graph %>% activate(edges) %>% as_tibble() %>% st_as_sf(), col =
  ↪  'grey50') +
  geom_sf(data = graph %>% activate(nodes) %>% as_tibble() %>% st_as_sf(), aes(col =
  ↪  degree, size = degree)) +
  scale_colour_viridis_c(option = 'inferno') +
  scale_size_continuous(range = c(0,4))
```

```
map_degree
```



The above map of degree centrality shows that the node for the water system in downtown Durham has the highest level of degree centrality. That is to say, it has the greatest number of edges connected to the node, or in this context, the greatest number of surrounding water systems connecting to it. It also shows that Chapel Hill has a lower degree centrality than some of its surrounding neighbors, such as Hillsborough, Raleigh, and what appears to be Fearrington. This may be due to Chapel Hill being relatively less connected to major interstate corridors like I-85 and I-40, which link areas like Burlington, Hillborough, Durham, and Raleigh.

```
#map of betweenness centrality for nodes on the interconnections network
map_betweenness <- ggplot() +
  geom_sf(data = map) +
  geom_sf(data = graph %>% activate(edges) %>% as_tibble() %>% st_as_sf(), col =
  ↪  'grey50') +
  geom_sf(data = graph %>% activate(nodes) %>% as_tibble() %>% st_as_sf(), aes(col =
  ↪  betweenness, size = betweenness)) +
  scale_colour_viridis_c(option = 'inferno') +
  scale_size_continuous(range = c(0,4))

map_betweenness
```



The map of betweenness centrality shown above shows a relatively similar story to that of degree centrality, with some exceptions. The nodes on the I-85 corridor near Burlington show a much greater betweenness centrality, as well as the node in Hillsborough. I believe this is due to that corridor being the only link between the Western nodes and Eastern nodes. The nodes in Wade county score relatively similarly on betweenness as they did on degree centrality. There is also a relatively greater betweenness centrality for those two nodes in the West which are the sole links to the rest of the Western nodes, for the same reason explained above regarding the nodes around Burlington.

Question 4

```
#create shortest distances matrix
distances <- distances(
  graph = graph,
  weights = graph %>% activate(edges) %>% pull(length)
)
#rename column to match connected points dataframe and join
#nodes <- nodes %>%
  #rename(geom = geometry)

#create points with nodes dataframe
connected_nodes <- st_join(connected_points, nodes)

#shortest path from Cary to Chapel Hill water system
from_node <- graph %>%
  activate(nodes) %>%
  filter(nodeID == 32) %>%
  pull(nodeID)
to_node <- graph %>%
  activate(nodes) %>%
  filter(nodeID == 31) %>%
```

```
  pull(nodeID)
path <- shortest_paths(
  graph = graph,
  from = from_node,
  to = to_node,
  output = 'both',
  weights = graph %>% activate(edges) %>% pull(length)
)

#create a subgraph with the nodes and edges of calculated path
path_graph <- graph %>%
    subgraph.edges(eids = path$epath %>% unlist()) %>%
    as_tbl_graph()

# summarize the length of the path. 37,055.4 meters from Cary node to OWASA node
path_graph %>%
  activate(edges) %>%
  as_tibble() %>%
  summarise(length = sum(length))
```

```
## # A tibble: 1 x 1
##   length
##      [m]
## 1 37055.
```

```
#Graph of shortest path. Water has to pass through Durham to get from Cary to Chapel Hill
ggplot() +
  geom_sf(data = map) +
  geom_sf(data = graph %>% activate(edges) %>% as_tibble() %>% st_as_sf(), col =
  ↪  'darkgrey') +
  geom_sf(data = graph %>% activate(nodes) %>% as_tibble() %>% st_as_sf(), col =
  ↪  'darkgrey', size = 0.5) +
  geom_sf(data = path_graph %>% activate(edges) %>% as_tibble() %>% st_as_sf(), lwd = 1,
  ↪  col = 'hotpink') +
  geom_sf(data = path_graph %>% activate(nodes) %>% filter(nodeID %in% c(from_node,
  ↪  to_node)) %>% as_tibble() %>% st_as_sf(), size = 2)
```

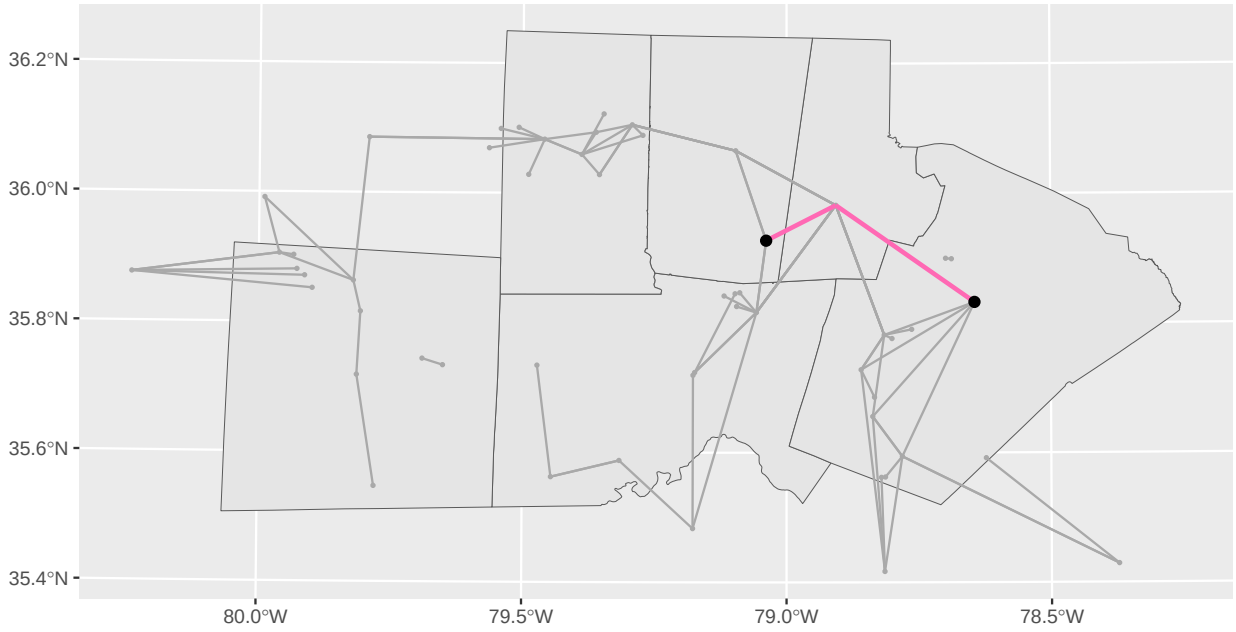
The shortest path for Cary to sell water to Chapel Hill goes through Durham, and is 37,055.4 meters.

```
from_node <- graph %>%
  activate(nodes) %>%
  filter(nodeID == 37) %>%
  pull(nodeID)
to_node <- graph %>%
  activate(nodes) %>%
  filter(nodeID == 31) %>%
  pull(nodeID)
path <- shortest_paths(
  graph = graph,
  from = from_node,
  to = to_node,
  output = 'both',
  weights = graph %>% activate(edges) %>% pull(length)
)

#create a subgraph with the nodes and edges of calculated path
path_graph <- graph %>%
    subgraph.edges(eids = path$epath %>% unlist()) %>%
    as_tbl_graph()

# summarize the length of the path. 42,232.69 meters from Raleigh node to OWASA node
path_graph %>%
  activate(edges) %>%
  as_tibble() %>%
  summarise(length = sum(length))
```

```
## # A tibble: 1 x 1
##   length
```

```
##      [m]
## 1 42233.
```

```
#Graph of shortest path. Water has to go through Durham to get from Raleigh to Chapel
↪  Hill.
ggplot() +
  geom_sf(data = map) +
  geom_sf(data = graph %>% activate(edges) %>% as_tibble() %>% st_as_sf(), col =
  ↪  'darkgrey') +
  geom_sf(data = graph %>% activate(nodes) %>% as_tibble() %>% st_as_sf(), col =
  ↪  'darkgrey', size = 0.5) +
  geom_sf(data = path_graph %>% activate(edges) %>% as_tibble() %>% st_as_sf(), lwd = 1,
  ↪  col = 'hotpink') +
  geom_sf(data = path_graph %>% activate(nodes) %>% filter(nodeID %in% c(from_node,
  ↪  to_node)) %>% as_tibble() %>% st_as_sf(), size = 2)
```



The shortest path for Raleigh to sell water to Chapel Hill goes through Durham, and is 42,232.69 meters.

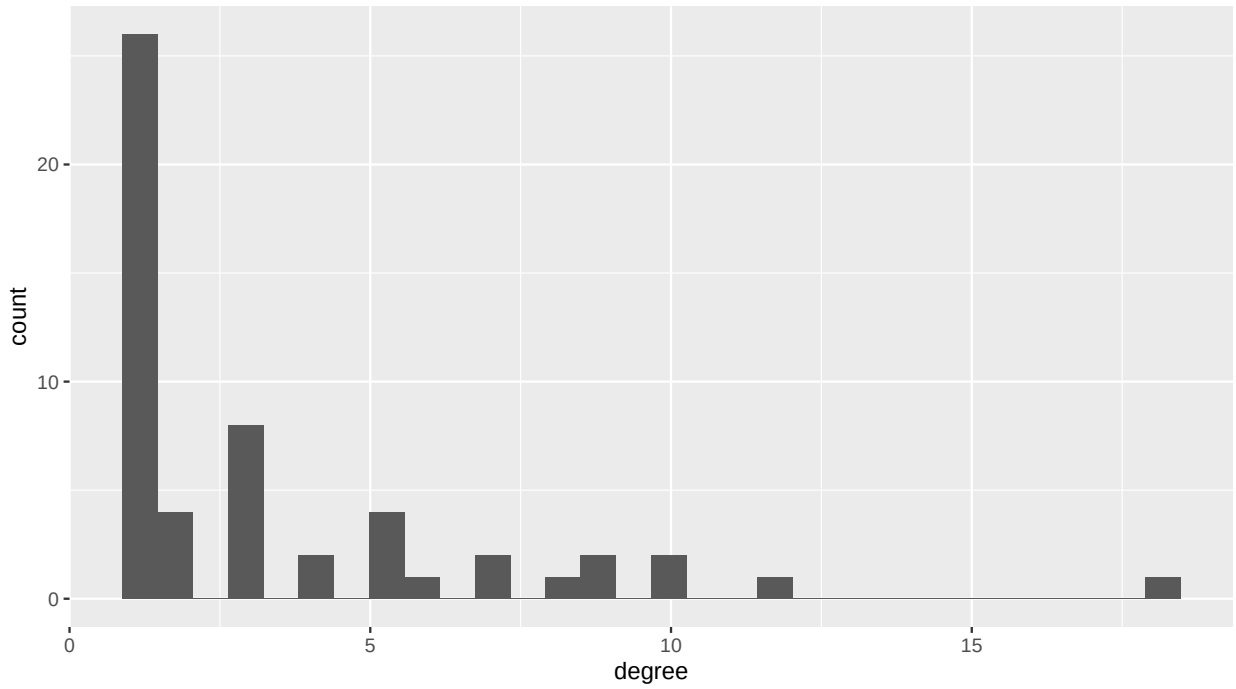
Question 5

```
unconnected_points[22,3] #town of liberty coordinates
# -79.5682 , 35.8557
```

```
#create geographic point for liberty's unconnected node
liberty <- st_point(c(-79.5682, 35.8557)) %>%
  st_sfc(crs=4236)

#coordinates of liberty
```

```r
coords_o <- liberty %>%
  st_coordinates() %>%
  matrix(ncol = 2)

#coordinates of all nodes in the network
nodes <- graph %>%
  activate(nodes) %>%
  as_tibble() %>%
  st_as_sf()
coords <- nodes %>%
  st_coordinates()

#index of neighbors
node_index_o <- knn(data = coords, query = coords_o, k = 1)
#find closest node neighboring, nodeID=45, Town of Ramseur
node_o <- nodes[node_index_o$nn.idx, ]
```

```r
#checking the distribution of degree centrality among connected points
ggplot(nodes, aes(x=degree)) + geom_histogram()
```




```r
#filtering nodes to those with 5 degrees or greater of degree centrality to explore
↪  potentially more beneficial options, given our closest neighbor has a degree
↪  centrality of 1.
coords_filtered <- nodes %>%
  filter(degree>=3) %>%
  st_coordinates()

#index of neighbors with filtered data
node_index_o2 <- knn(data = coords_filtered, query = coords_o, k = 1)
```

```r
#closest neighbor with degree centrality >=5, nodeID=2, Burlington.
node_o2 <- nodes[node_index_o2$nn.idx, ]
#Also performed this with >=2, which returned nodeID=11, Town of Elon. Elon and
↪  Burlington are very close, but Burlington's network is much more connected.
#Now that I'm reevaluating, even though the function runs with >=2, Elon's degree
↪  centrality is 1. Unsure why this result is given. When >=3, and >=5, it gives
↪  Burlington. I will exclude the finding about Elon in the writeup due to this issue.
```

To determine what options the Town of Liberty could explore to connect to an already interconnected water system, I first found the geographically closest neighboring node in the network. This is the Town of Ramseur, about a 15 minute drive south of Liberty. Ramseur's node has a degree centrality of 1 and a betweenness of 0, so I looked for a reasonable way to filter for nodes with a greater degree of centrality. Using the simple histogram above, I realized that a majority of systems score either a 0 or 1 in degree centrality, so I found the nearest neighbor when filtering the measurement to be >=3. Using this method, the nearest neighbor in the network is the City of Burlington, with a degree centrality of 9 and a very high betweenness score of 582, as was shown earlier in the graphing of centrality measures.

While there may be many benefits of developing a connection to the Burlington water system node, it would also require covering a bit more than double the distance than developing a connection to the Ramseur water system. The centrality of the Burlington water system likely translates to greater capability and reliability than the Ramseur water system. If the Town of Liberty has ample funding to develop such a connection, it may be the better decision. However, given the relatively small size of the town, it may not be at liberty to choose to connect with the Burlington system over the closer option of Ramseur.