Practice7

Sebastian Montesinos

Due by midnight, Friday, April 15

Reminder: Practice assignments may be completed working with other individuals.

Reading

The associated reading for the week is Chapter 17 on Spatial Data. In addition, this practice has one question from Chapter 20 on Networks based on previous material.

Practicing Academic Integrity

If you worked with others or used resources outside of provided course material (anything besides our textbook, course materials in the repo, labs, R help menu) to complete this assignment, please acknowledge them below using a bulleted list.

I acknowledge the following individuals with whom I worked on this assignment:

Name(s) and corresponding problem(s)

-

I used the following sources to help complete this assignment:

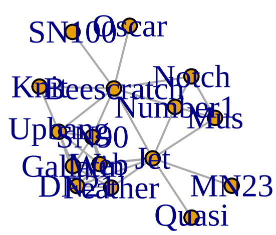
Source(s) and corresponding problem(s)

-

1 - Dolphins

A pod of dolphins has been studied for years in New Zealand to try to understand their social interactions. While the whole network is available (you can plot it if you like) it's hard to visualize. We took a "snowball sample" starting with the dolphin called "Notch". We took all the dolphins he was connected to, and then all the dolphins they were connected to, and found this induced subgraph. (This is a snowball sample with 3 waves, with one originating vertex).

```
dolphins <- read_graph("https://awagaman.people.amherst.edu/stat240/dolphins.gml", format = "gml")

# focus on Notch - vertex 27
# take snowball sample by finding neighbors sequentially
myvert1 <- neighbors(dolphins, 27)
myvert2 <- neighbors(dolphins, myvert1)
myvert3 <- neighbors(dolphins, myvert2)

# put set of all needed vertices together
# duplicates do not need removed
combined <- c(27, myvert1, myvert2, myvert3)

# create sub-graph
dolphin_Notch <- induced_subgraph(dolphins, combined, impl = "copy_and_delete")
plot(dolphin_Notch)
```

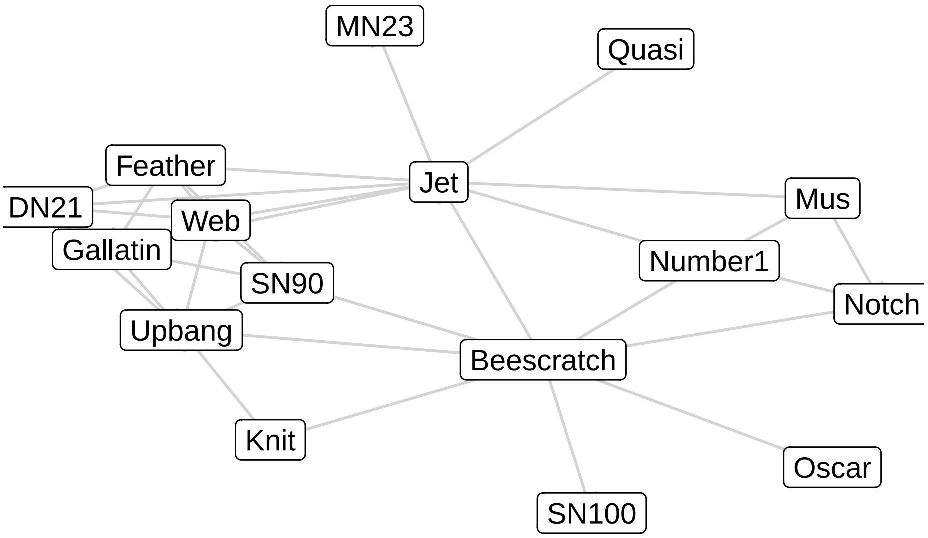


part a - Improve the visualization of the Notch-based subgraph by plotting it using ggplot2.

Solution:

```
dolphin_network <- ggnetwork(dolphin_Notch)
ggplot(data = dolphin_network, aes(x = x, y = y,
                                   xend = xend, yend = yend)) +
  geom_edges(arrow = arrow(type = "closed", length = unit(8, "pt")),
             color = "lightgray") +
```

```
geom_nodes() +
geom_nodelabel(aes(label = label)) +
theme_blank()
```



> part b - Notch was used to generate this graph, but is Notch the most central dolphin in the network? To help answer this question, compute the degree of each node. Which dolphins have the 3 largest degrees? Share their names and associated degree values in a nice table.

If there are ties for the top 3, include the tied dolphins.

Solution:

```
# Put id and label in a data set for later use
id <- get.vertex.attribute(dolphin_Notch)$id
label <- get.vertex.attribute(dolphin_Notch)$label
dolphin_data <- data.frame(id, label)

# find degrees
dolphin_data <- dolphin_data %>%
  mutate(degree = degree(dolphin_Notch)) %>%
  arrange(desc(degree))

# make a table
dolphin_data %>%
  head(5) %>%
  kable(booktabs = TRUE)
```

id	label	degree
17	Jet	9
1	Beescratch	8
13	Gallatin	6
54	Upbang	6
57	Web	6

part c - Based on the snowball sampling methodology, what is the furthest any dolphin could be from Notch in this network? What is the furthest apart any two dolphins could be as a result? No code is necessary to answer this.

Solution: The furthest any dolphin could be from notch is 3, since the connections from notch are done 3 times over. The furthest apart any two could be from Notch is 6, since on each side 3 is the farthest they could spread from notch (3 x 2).

part d - What is the diameter of this network? Interpret the value you obtain.

Solution: The diameter of this network is 3, meaning that 3 is the length of the longest geodesic between any pair of vertices in the network.

```
diameter(dolphin_Notch)
```

```
## [1] 3
```

part e - Notch was used to generate the network, but is Notch on many of the shortest paths between dolphins? Compute the betweenness centrality of each node. Use a nice table to show which dolphins have the top 3 betweenness centrality values.

If there are ties for the top 3, include the tied dolphins.

Solution:

```
dolphin_data <- dolphin_data %>%
  mutate(betweenness = betweenness(dolphin_Notch, weights = NA))

dolphin_data %>%
  arrange(desc(betweenness)) %>%
  select(id, label, betweenness) %>%
  head(4) %>%
  kable(booktabs = TRUE)
```

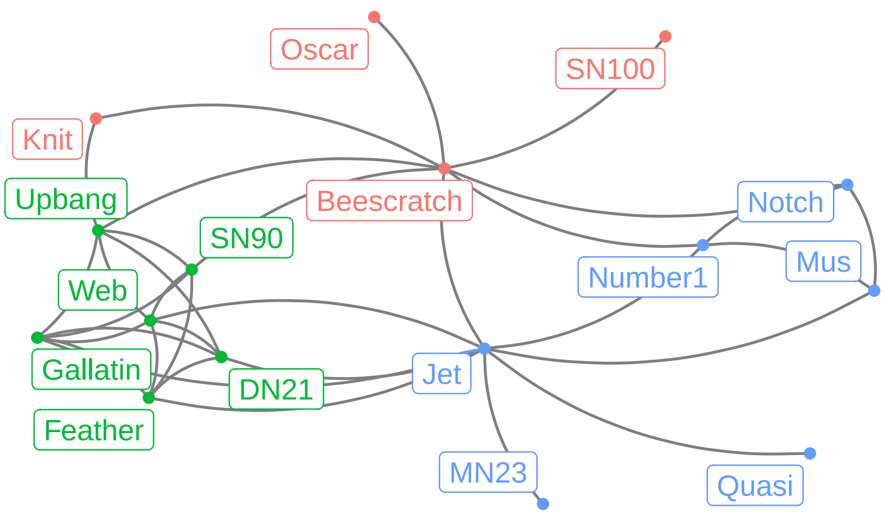
id	label	betweenness
57	Web	45.533333
17	Jet	43.533333
31	Quasi	8.566667
28	Oscar	4.733333

part f - Run a network clustering algorithm of your choice on the dolphin_Notch network. Plot the results including the cluster membership using ggplot2. How many clusters were found?

Solution:

```
dolphin_cl <- cluster_louvain(dolphin_Notch)
dolphin_membership <- membership(dolphin_cl)
dolphin_Notch <- set_vertex_attr(dolphin_Notch,
                                 name = "membership",
                                 value = dolphin_membership)
#Creating Network
```

```
dolphin_network <- ggnetwork(dolphin_Notch) %>%
  mutate(membership = factor(membership))

#Seed for reproducibility
set.seed(230)
#Creating graph
dolphin_network %>%
  ggplot(aes(x = x, y = y, xend = xend, yend = yend)) +
#Adding edges and nodes
geom_edges(color = "gray50", curvature = 0.2) +
geom_nodelabel_repel(aes(label = label, color = membership)) +
geom_nodes(aes(color = membership)) +
guides(color = "none") +
theme_blank()
```

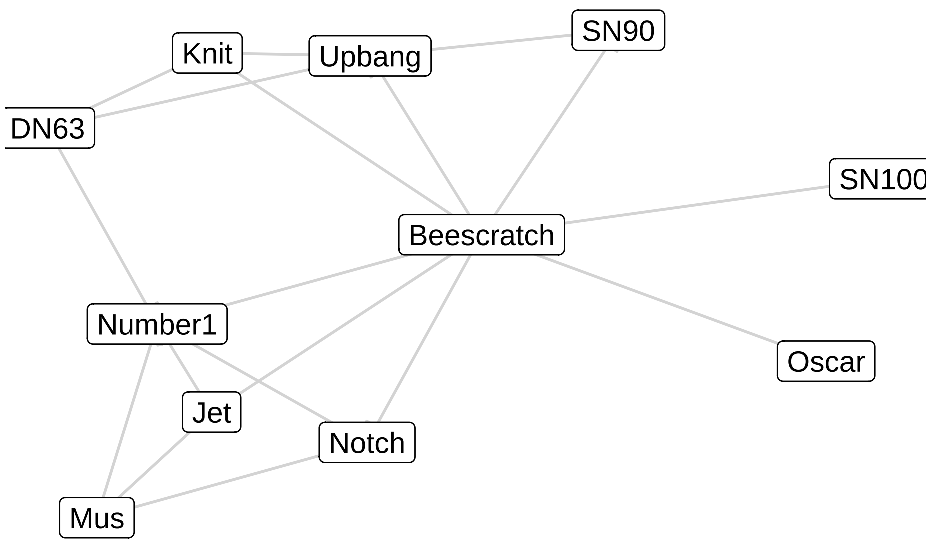


part g - Create an induced subgraph for a dolphin of your choice from the original network, using a snowball sample with TWO waves. Plot the network nicely using ggplot2.

Solution:

```
myvert4 <- neighbors(dolphins, 28)
myvert5 <- neighbors(dolphins, myvert4)

combined2 <- c(28, myvert4, myvert5)

# create sub-graph
dolphin_28 <- induced_subgraph(dolphins, combined2, impl = "copy_and_delete")


dolphin_network2 <- ggnetwork(dolphin_28)

#Plotting subgraph with ggplot
ggplot(data = dolphin_network2, aes(x = x, y = y,
```

```
                                    xend = xend, yend = yend)) +
  geom_edges(arrow = arrow(type = "closed", length = unit(8, "pt")),
             color = "lightgray") +
geom_nodes() +
geom_nodelabel(aes(label = label)) +
theme_blank()
```



2 - Mapping spatial data

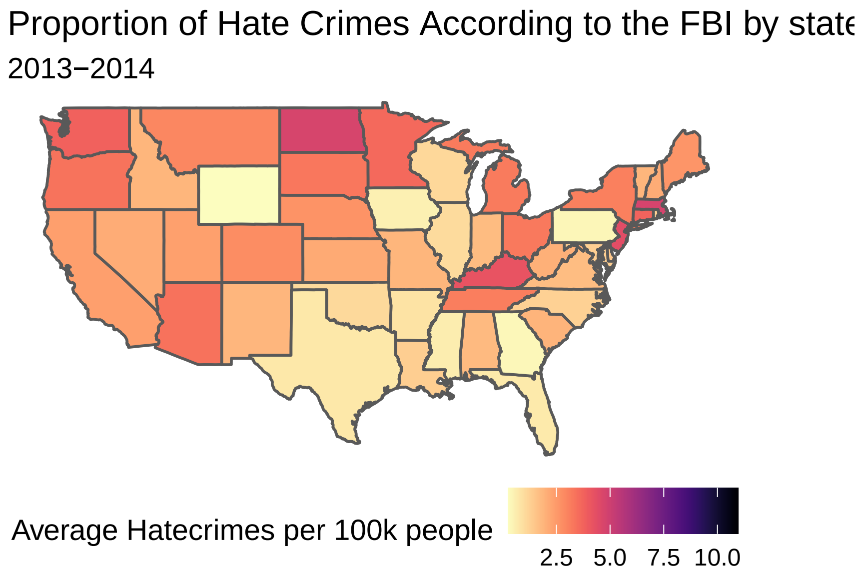
Reproduce the map you created in Lab 9's Part 3 - `Your turn` - where you made your own map.

In 2-4 sentences, interpret the visualization. What stands out as the central message? Note: You shouldn't say what colors are representing what feature; this is obvious to the viewer, assuming there's an appropriate legend and title. (Add one if you don't have one!) Rather, share what information you extract from the visualization.

Be sure to load all appropriate packages in the setup chunk above.

Solution: The choropleth below shows that hatecrimes per capita varies significantly across states in the US. Massachussetts, North Dakota, Washington, and Kentucky stand out as the states with the highest proportion of hate crimes. In contrast, states such as Wyoming, Pennsylvania, and Georgia have some of the lowest rates of hate crimes.

```r
#Getting state location data
us_states <- maps::map("state",
                    plot = FALSE, fill = TRUE) %>%
  st_as_sf()

#Getting hate crimes data
data(hate_crimes)
hate_crimes <- hate_crimes %>%
#Making sure state names match in both
  mutate(state = str_to_lower(state))

#Joining state data to hate crimes data
joined_map <- us_states %>%
  right_join(hate_crimes, by = c("ID" = "state"))



#Creating plot
ggplot(joined_map) +
  geom_sf(aes(fill = avg_hatecrimes_per_100k_fbi)) +
  scale_fill_viridis(option = "magma", direction = -1) +
  theme_void() +
  labs(fill = "Average Hatecrimes per 100k people",
       color = "Hate Crimes",
       title = "Proportion of Hate Crimes According to the FBI by state",
       subtitle = "2013-2014") +
  theme(legend.position = "bottom")
```


Proportion of Hate Crimes According to the FBI by state
2013−2014
Average Hatecrimes per 100k people
2.5
5.0
7.5
10.0

3 - Leaflet - Based on MDSR 17.2

In this final problem, we'll explore making an interactive map with leaflet. (You can revisit the last question on the lab for assistance with this as well.) We will use the package *pdxTrees* to access a data set which contains information on trees in Portland parks (be sure to install this package).

```
# loads data set
tree_data <- get_pdxTrees_parks()
```

```
# shows all common names for trees
unique(tree_data$Common_Name)
```

> part a - Pick three common names and make a new data set for those trees.

Note: Don't select Douglas-Fir, Norway Maple, or Western Redcedar (~ 1000 or more each). These have too many observations and we want the map to be readable.

Solution:

```
#Choosing 3 trees
my_trees <- tree_data %>%
  filter(Common_Name == "Silver Maple" | Common_Name == "Elm Hybrid" | Common_Name == "Giant Sequoia" )
```

We are going to create a leaflet map that plots the trees by Common_Name and allows you to pick between which ones you want to view (either, both, or neither), using circles to mark them on the map.

> part b - The example code below runs for the demo with 2 Common_Names. Adapt this for your Common_names, and be sure to add the third you chose! Feel free to change colors, etc. as you like.

Solution:

```
leaflet(my_trees) %>%
  addTiles() %>%
  addCircleMarkers(data = filter(my_trees, Common_Name == "Silver Maple"),
                   group = "Silver Maple",
                   lng = ~ Longitude,
                   lat = ~ Latitude,
                   color = "blue") %>%
  addCircleMarkers(data = filter(my_trees, Common_Name == "Elm Hybrid"),
                   group = "Elm Hybrid",
                   lng = ~ Longitude,
                   lat = ~ Latitude,
                   color = "white") %>%
    addCircleMarkers(data = filter(my_trees, Common_Name == "Giant Sequoia"),
                   group = "Giant Sequoia",
                   lng = ~ Longitude,
                   lat = ~ Latitude,
                   color = "yellow") %>%
  addLayersControl(overlayGroups = c("Silver Maple", "Elm Hybrid", "Giant Sequoia"),
                  options = layersControlOptions(collapsed = FALSE))
```

part c - Write a few sentences about what you learned from your map.

Solution: This map with 3 trees shows me that there are many more giant sequoias than silver maples and elm hybrids in Portland parks. Additionally, the range that the sequoias cover is great than that of the elm hybrids and silver maple. There ranges all over lap though, there are places you could see all three types of trees at once.