Visualization of desire line network between NTAs

This notebook provides a reference visualization for a network of desire lines. The nodes are the NTAs. They are labeled with the numeric portion of the NTA code. The edges are the trips between them. The edges are weighted based on the number of trips between the two NTAs. The graph is undirected. Consequently, a trip in either direction is counted towards the weight for the edge. If there are no trips between two NTAs, then they do not share an edge.

There is no statistical test performed on the network in this form. Instead, it serves as foundation for the next step of analysis.

```
library(tidyverse)
library(ggraph)
library(sf)
library(sfnetworks)
```

```
bk_name <- "Brooklyn"
bk_county_code <- "047"
bk_parks <- "BK99"
```

Graph desire lines

Construct the graph of all desire lines between all Brooklyn NTAs

Places nodes and edges in data frame

```
nta_trips <- readr::read_csv('./data/nta-trip-network.csv')
nta_trip_nodes <- sf::st_read('./data/nyc_2010_nta_borders.geojson') %>%
  dplyr::filter(BoroName == bk_name)  %>%
  dplyr::filter(NTACode != bk_parks) %>%
  select("NTACode") %>%
  mutate(geometry = sf::st_point_on_surface(geometry))
```

```
## Reading layer `nyc_2010_nta_borders' from data source
##   `/home/miller/GeoI/fall-2022/gtech_705-spatial_anlysis/triboro-line/brooklyn-lodes/data/nyc_2010_n
##   using driver `GeoJSON'
## Simple feature collection with 195 features and 8 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -74.25559 ymin: 40.49614 xmax: -73.70001 ymax: 40.91554
## Geodetic CRS:  WGS 84
```

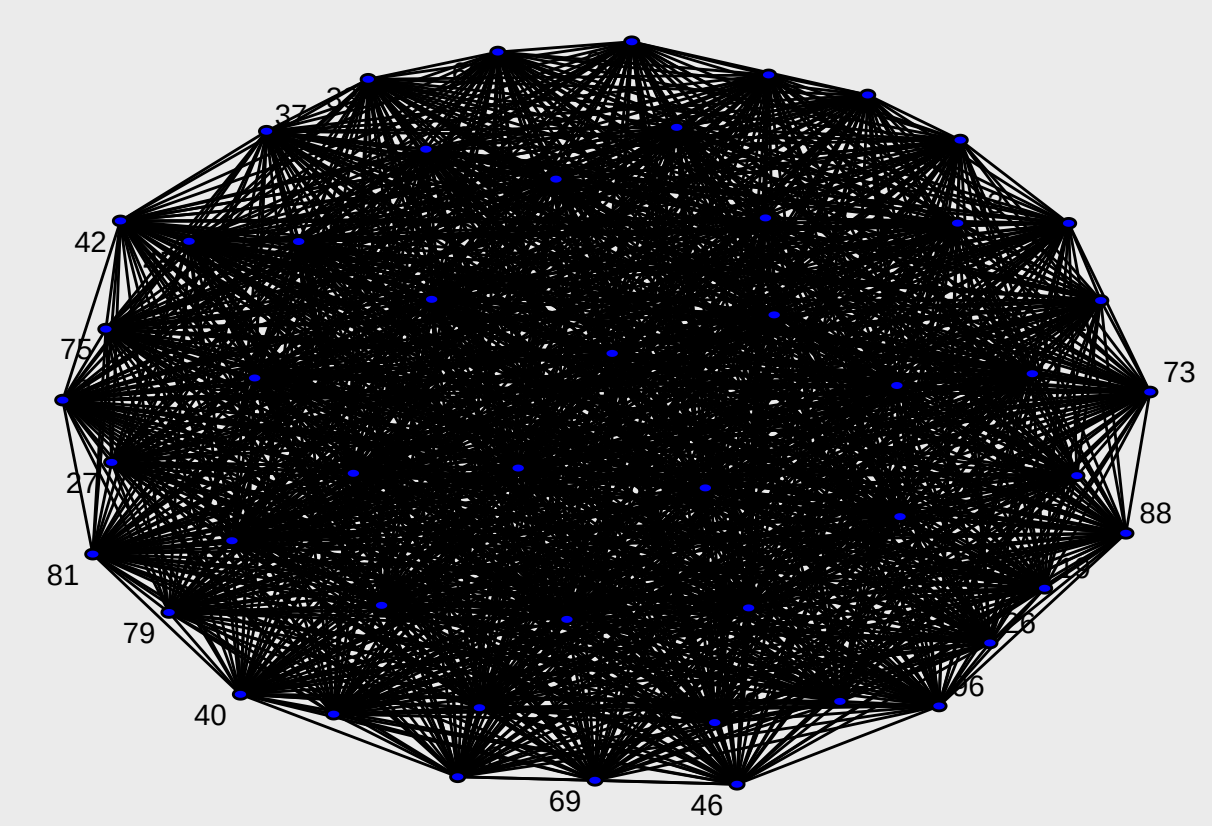
```
## Warning in st_point_on_surface.sfc(geometry): st_point_on_surface may not give
## correct results for longitude/latitude data
```

```
nta_trip_edges <- tibble::tibble(from = nta_trips$nta_code_one, to = nta_trips$nta_code_two, weight = n
nta_trip_graph <- sfnetworks::sfnetwork(nodes = nta_trip_nodes, edges = nta_trip_edges, node_key = 'NTA
```

Visualize the graph

```
ggraph::ggraph(nta_trip_graph, layout="stress") +
  geom_edge_link() +
  geom_node_circle(aes(r = 0.01), fill = "blue") +
  geom_node_text(aes(label = stringr::str_sub(NTACode, 3,4)), repel = TRUE)
```

```
## Warning: Using the `size` aesthetic in this geom was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` in the `default_aes` field and elsewhere instead.
```



Subgraph

The full graph is staturated with data. Here, I present subsections of the edges to help visualize the structure of the full graph.

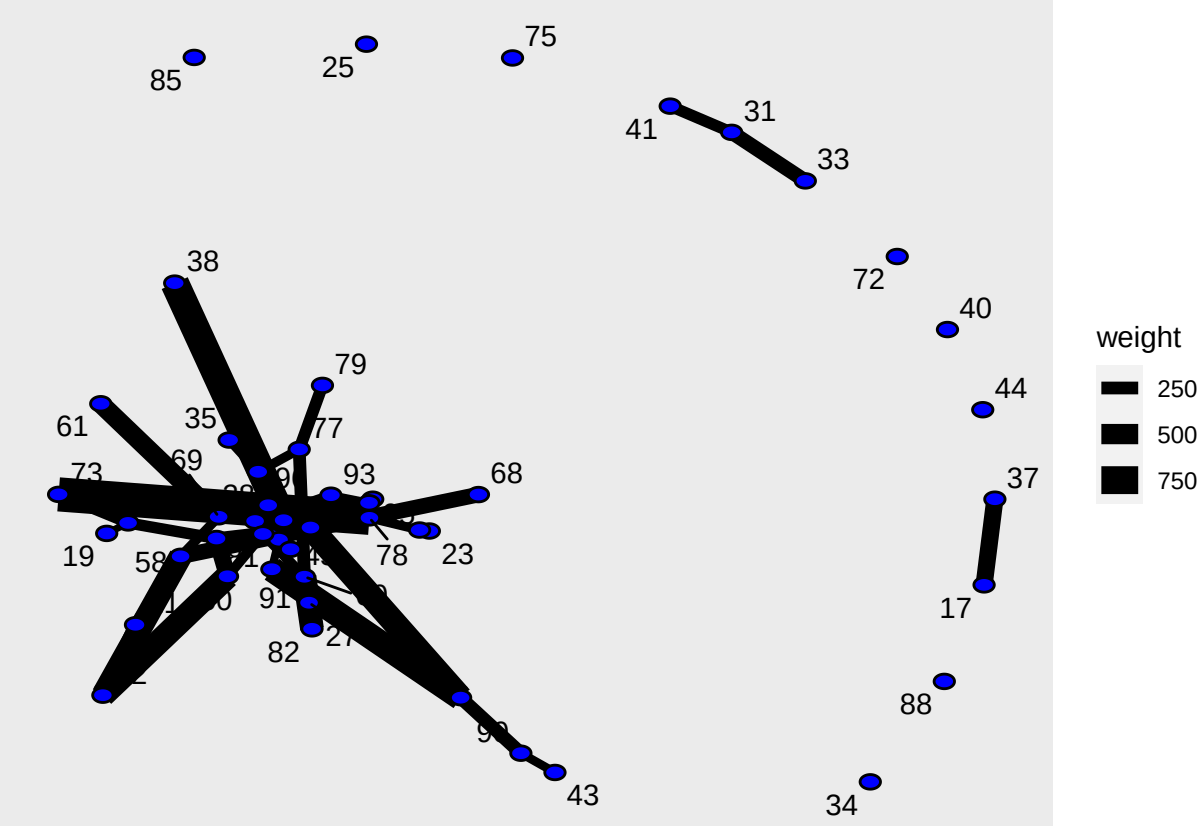
Random Sample

A random sampling of edges

```
nta_trip_edges_rand <- nta_trip_edges %>%
  dplyr::slice_sample(n = 50)
nta_trip_graph_rand <- sfnetworks::sfnetwork(nodes = nta_trip_nodes, edges = nta_trip_edges_rand, node_
```

```
ggraph::ggraph(nta_trip_graph_rand, layout="kk") +
  geom_edge_link(mapping = aes(edge_width = weight)) +
  geom_node_circle(aes(r = 0.1), fill = "blue") +
  geom_node_text(aes(label = stringr::str_sub(NTACode, 3,4)), repel = TRUE)
```

```
## Warning: ggrepel: 7 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

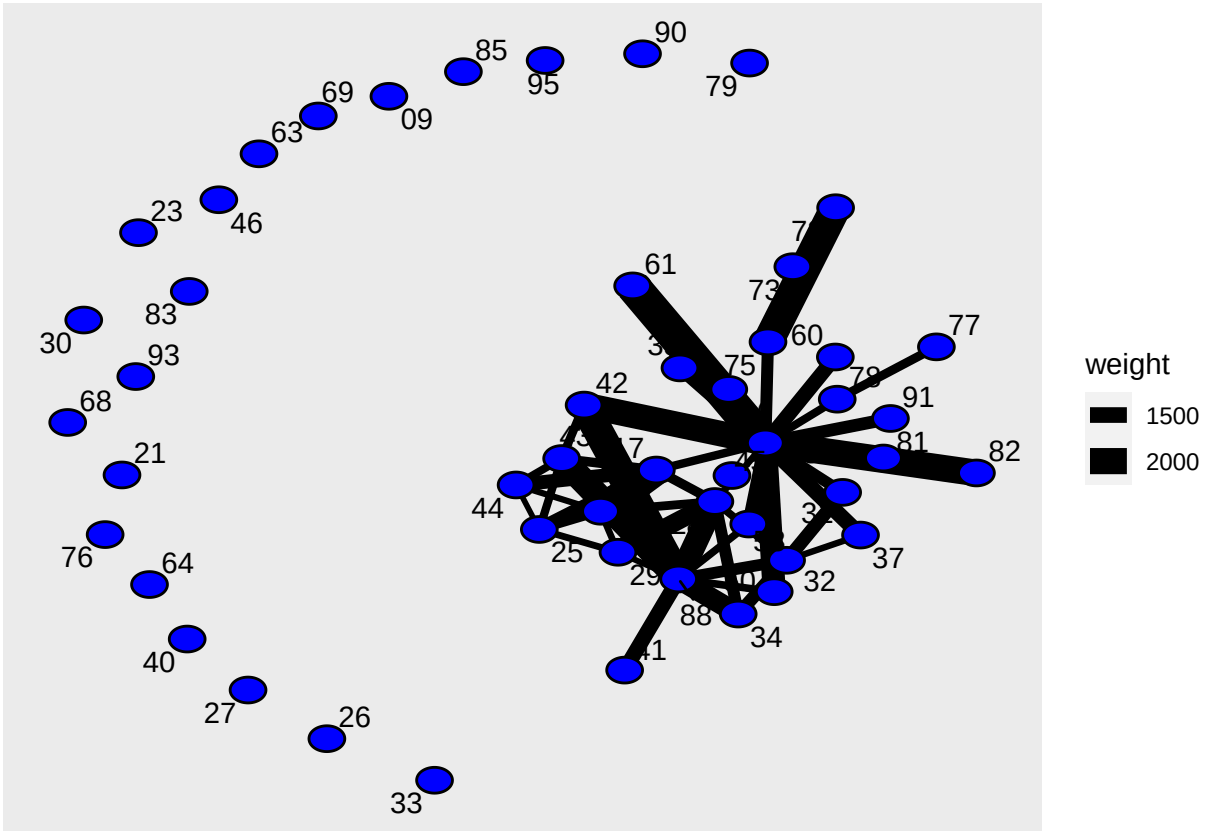


Most popular commutes

A sampling of the edges with the top 50 number of commutes

```
nta_trip_edges_top <- nta_trip_edges %>%
  dplyr::slice_max(order_by = weight, n = 50)
nta_trip_graph_top <- sfnetworks::sfnetwork(nodes = nta_trip_nodes, edges = nta_trip_edges_top, node_ke
```

```
ggraph::ggraph(nta_trip_graph_top, layout="kk") +
  geom_edge_link(mapping = aes(edge_width = weight)) +
  geom_node_circle(aes(r = 0.2), fill = "blue") +
  geom_node_text(aes(label = stringr::str_sub(NTACode, 3,4)), repel = TRUE)
```


90
85
95
79
69
09
63
23
46
61
73
30
83
60
77
93
75
42
91
68
81
82
21
44
25
37
32
76
64
88
34
40
27
26
33
weight
1500
2000