

```
library(tidyverse)
library(tidygraph)
library(ggraph)
library(spData)
library(spdep)
library(igraph)
```

Define each node as a trip

```
nta_trips <- readr::read_csv('./data/nta-trip-network.csv')
nta_trip_nodes <- tibble::tibble(name = nta_trips$trip, trip_count = nta_trips$S000)
```

Define each edge as a shared NTA between two trips

```
build_edges <- function(nodes){
  edges_from <- vector()
  edges_to <- vector()
  nodes_count <- length(nodes)
  for(i in 1:(nodes_count - 1)) {
    offset <- i + 1
    from_node <- nodes[i]
    from_nta_one <- stringr::str_sub(from_node, 1, 4)
    from_nta_two <- stringr::str_sub(from_node, 5, 8)
    for(j in offset:nodes_count){
      to_node <- nodes[j]
      are_neighbors <- stringr::str_detect(to_node, from_nta_one) | stringr::str_detect(to_node, from_nta_two)
      if (are_neighbors) {
        edges_from <- append(edges_from, from_node)
        edges_to <- append(edges_to, to_node)
      }
    }
  }
  return (tibble::tibble(from = edges_from, to = edges_to))
}
```

Construct the graph

```
nta_trip_edges <- build_edges(nta_trip_nodes$name)
nta_trip_network <- tidygraph::tbl_graph(nodes = nta_trip_nodes, edges = nta_trip_edges)
```

Subgraph

Use a sub section of the network to make it easier to visualize

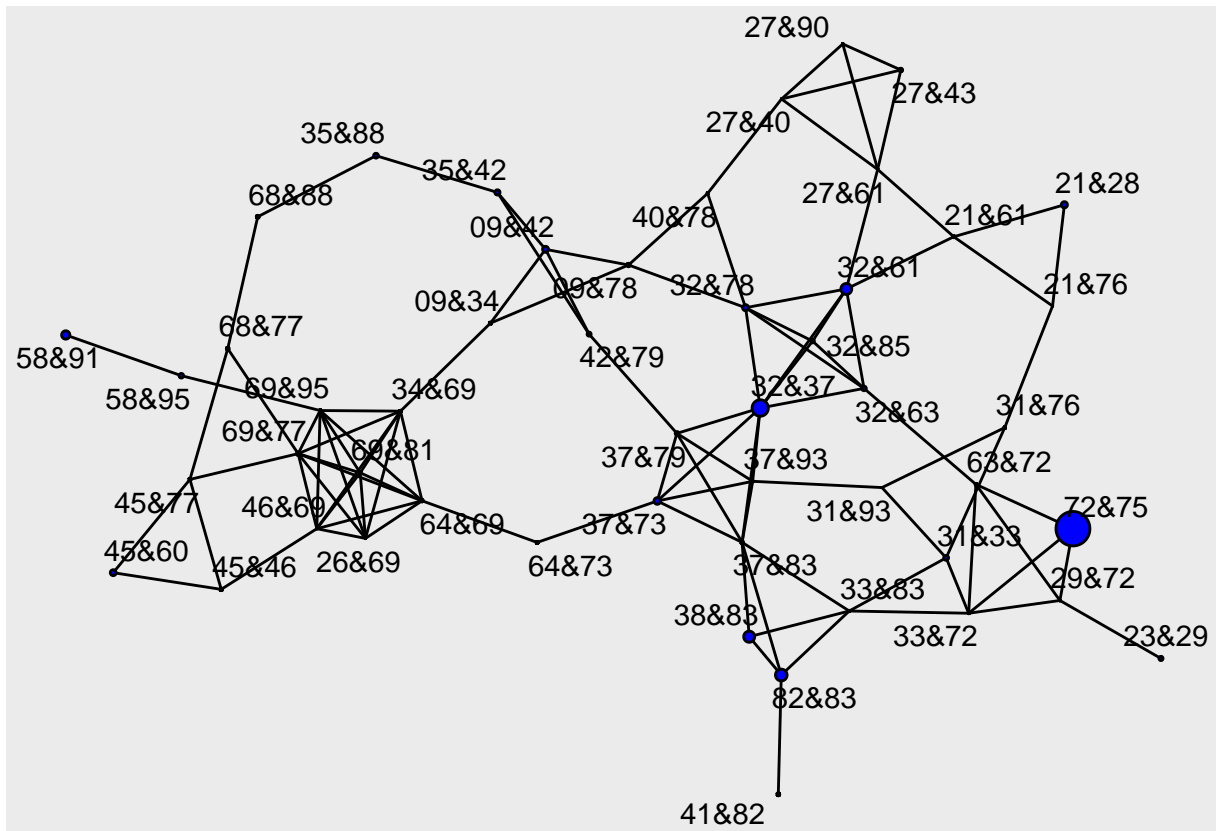
Random sample

```
nta_trip_nodes_rand <- nta_trip_nodes %>%
  dplyr::slice_sample(n = 50)
nta_trip_edges_rand <- build_edges(nta_trip_nodes_rand$name)
nta_trip_network_rand <- tidygraph::tbl_graph(nodes = nta_trip_nodes_rand, edges = nta_trip_edges_rand)

total_trips_rand <- sum(nta_trip_nodes_rand$trip_count)
ggraph::ggraph(nta_trip_network_rand, layout="stress") +
  geom_edge_link() +
  geom_node_circle(aes(r = (nta_trip_nodes_rand$trip_count / total_trips_rand)), fill = "blue") +
```

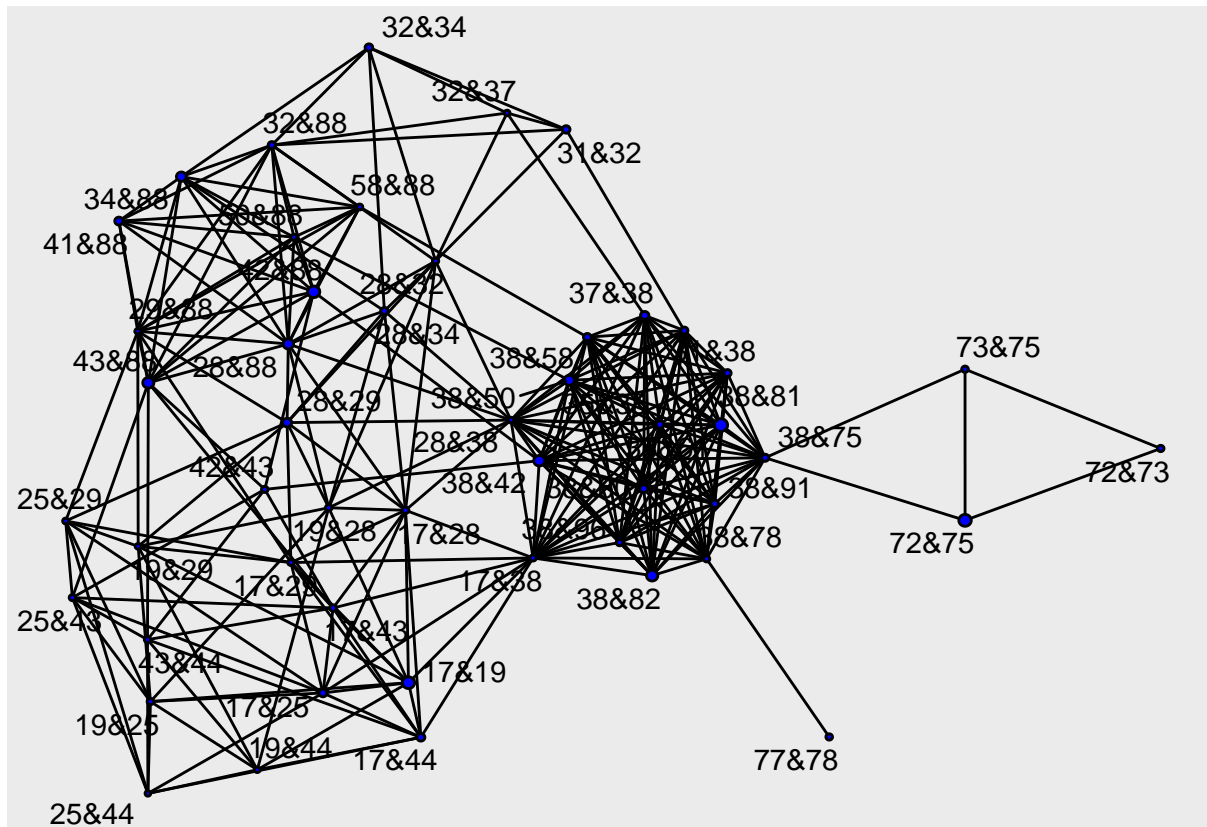
```
geom_node_text(aes(label = stringr::str_c(stringr::str_sub(name, 3,4), '&', stringr::str_sub(name, 7,8))
```

```
## 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.
```



Most popular trips

```
nta_trip_nodes_top <- nta_trip_nodes %>%  
  dplyr::slice_max(order_by = trip_count, n = 50)  
nta_trip_edges_top <- build_edges(nta_trip_nodes_top$name)  
nta_trip_network_top <- tidygraph::tbl_graph(nodes = nta_trip_nodes_top, edges = nta_trip_edges_top)  
  
total_trips_top <- sum(nta_trip_nodes_top$trip_count)  
ggraph::ggraph(nta_trip_network_top, layout="stress") +  
  geom_edge_link() +  
  geom_node_circle(aes(r = (nta_trip_nodes_top$trip_count / total_trips_top)), fill = "blue") +  
  geom_node_text(aes(label = stringr::str_c(stringr::str_sub(name, 3,4), '&', stringr::str_sub(name, 7,8))
```



```
nta_trip_network_weights <- nta_trip_network %>%
  igraph::as_adj() %>%
  spdep::mat2listw()
```

```
## Warning in sn2listw(df): BK95BK96 is not an origin
```

```
global_morans <- spdep::moran.test(nta_trip_nodes$trip_count, nta_trip_network_weights, zero.policy = TRUE)
global_morans
```

```
##
## Moran I test under randomisation
##
## data: nta_trip_nodes$trip_count
## weights: nta_trip_network_weights n reduced by no-neighbour observations
##
##
## Moran I statistic standard deviate = 58.918, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      2.316450e-01      -8.176615e-04      1.556722e-05
```