Network auto-correlation for desire lines between NTAs in Brooklyn

Building off the network of the desire-lines from the previous section, the graph is transformed to describe the relationship between the desire lines. After this transformation, a node is defined as a trip. The nodes are named for the two NTAs that are part of the trip. Because the base graph is undirected, the order of the codes in the trip name are not related to the direction of the trips. An edge exists between two trips if the trips have an NTA in common. This graph is also undirected.

Network auto-correlation is performed with the graph in this form. The value of each node is the number of commutes that are made along that trip.

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

Define each trip as a node

```r
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 a utility function that builds the edges between trips

```r
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_nt
      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

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

Subgraph

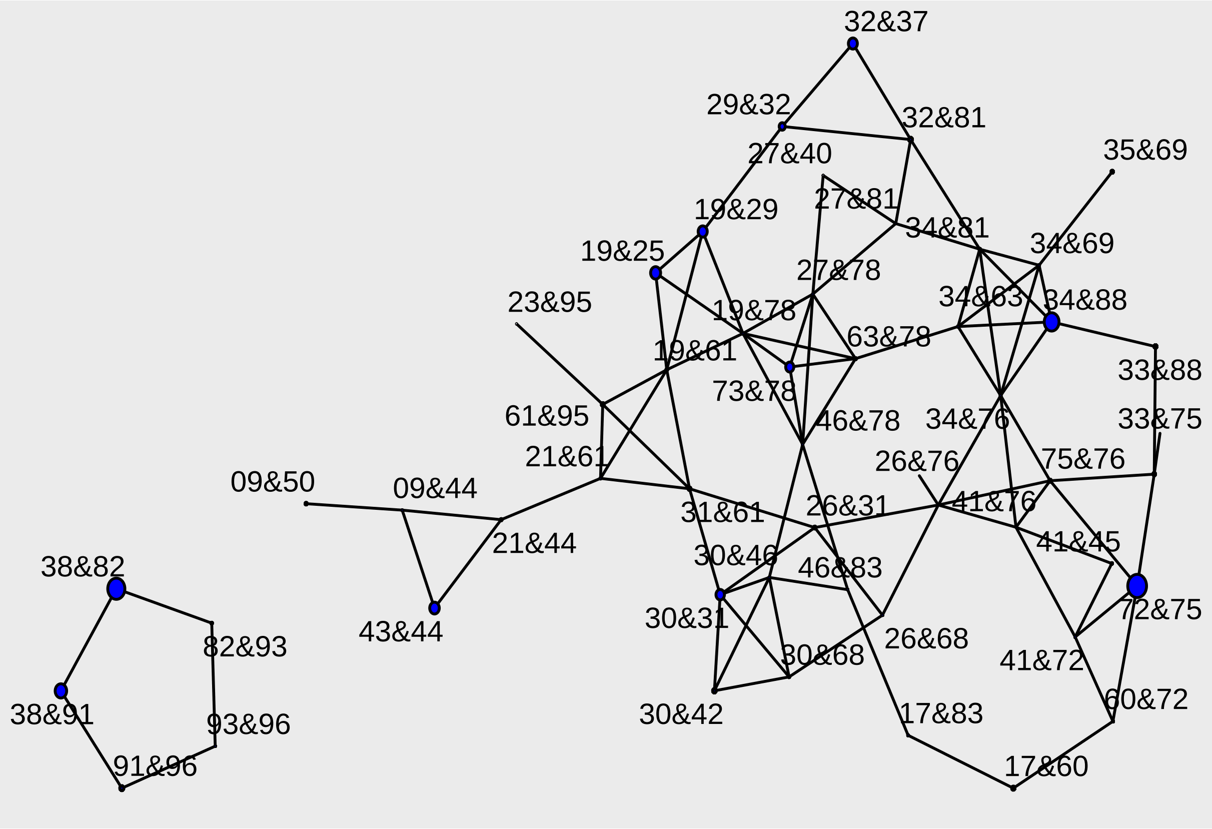
Use a sub section of the network to make it easier to visualize

Random sample

Construct a subgraph with a random sample of nodes

```
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
```

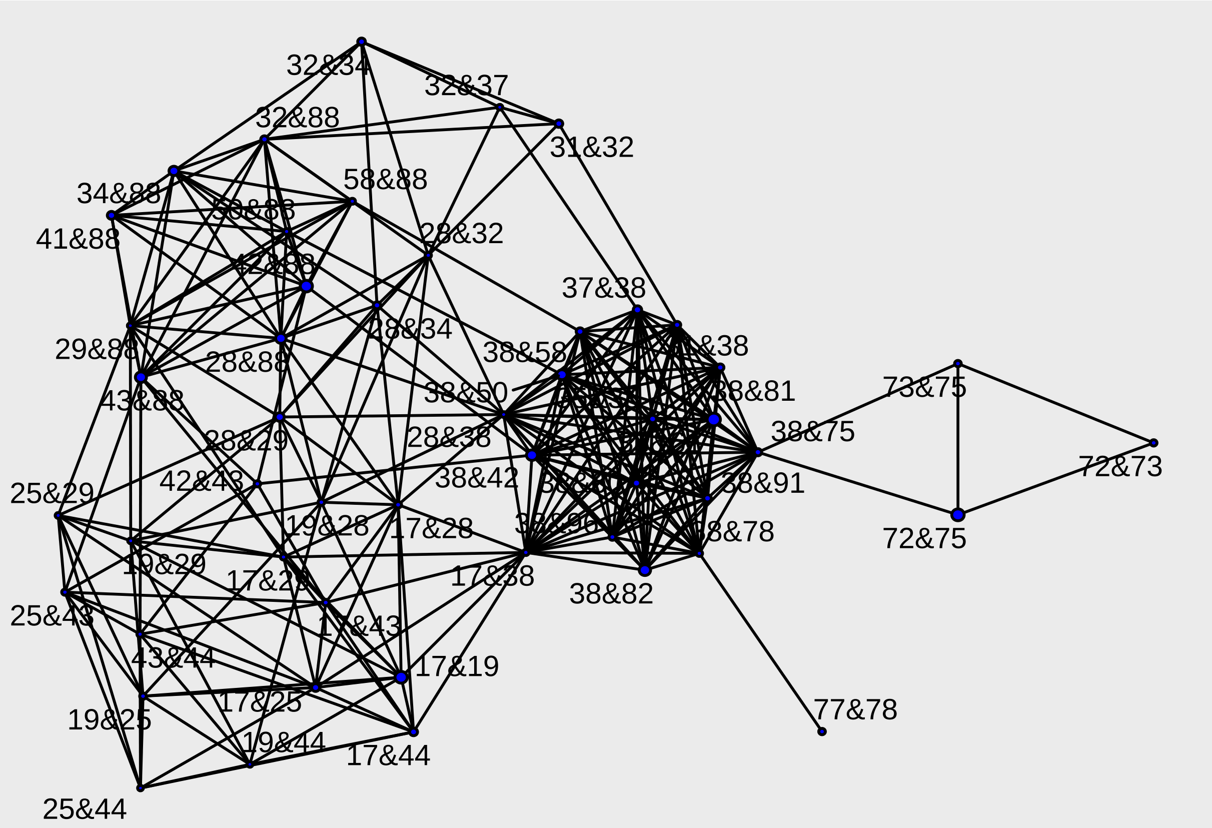
```
## Warning: Using the `size` aesthetic in this geom was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` in the `default_aes` field and elsewhere instead.
```



Most popular trips

Construct a subgraph using the nodes with the top 50 trip counts

```
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
```



Global Moran's I

Calculate the total Global Moran's I for the whole network.

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

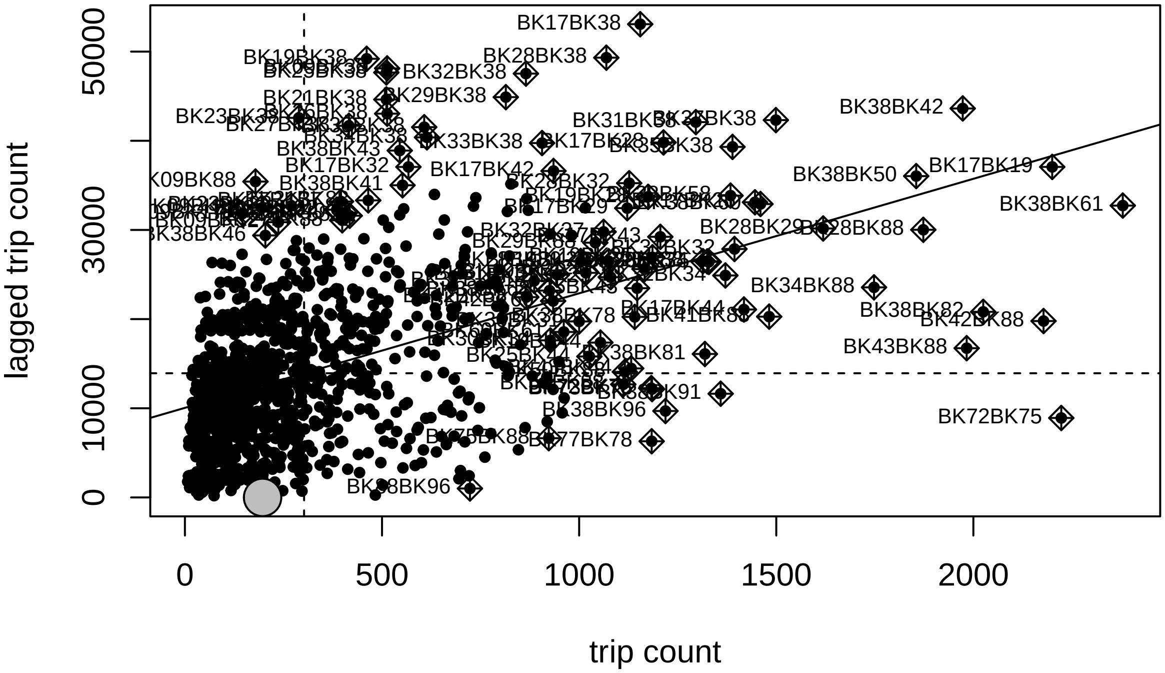
```
## Warning in sn2listw(df): BK95BK96 is not an origin
```

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

Plot the Global Moran's I

```
spdep::moran.plot(
  nta_trip_nodes$trip_count,
  nta_trip_network_weights,
  zero.policy = TRUE,
  xlab = "trip count",
  ylab = "lagged trip count",
  pch = 20,
)
```

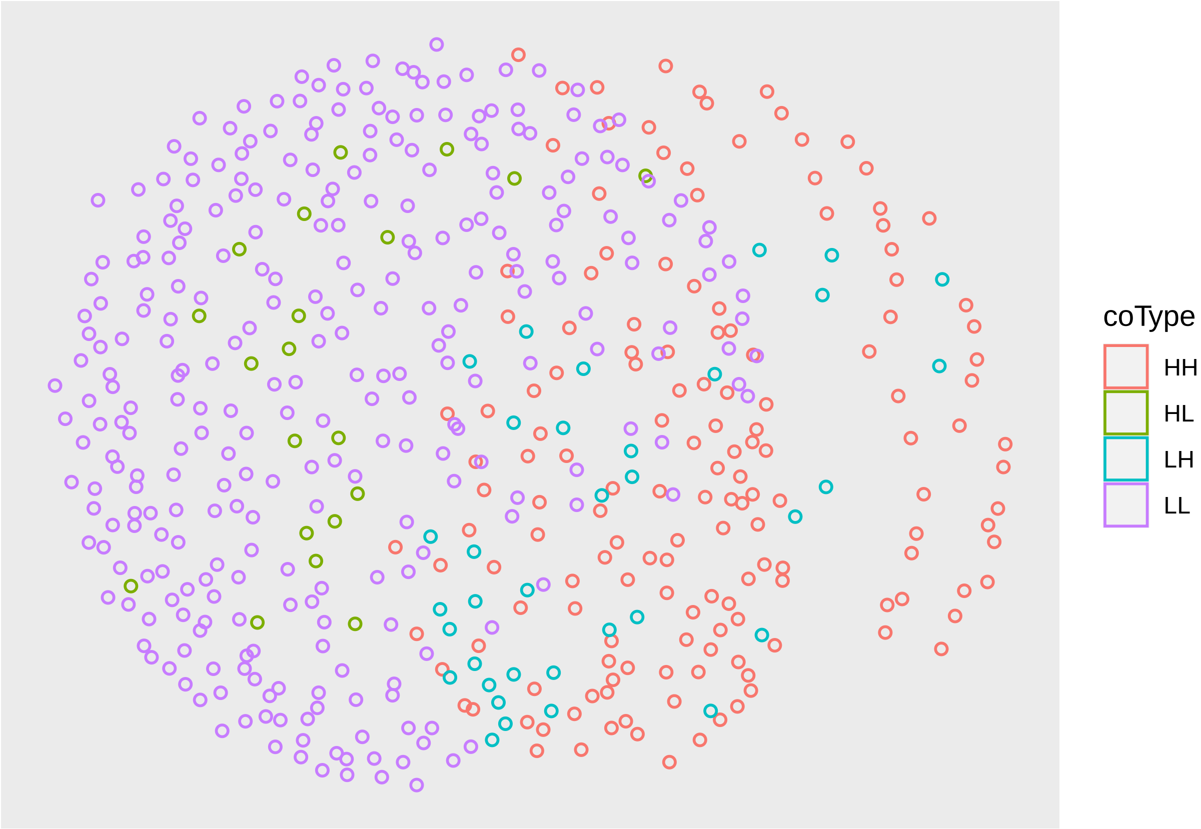


Local Indicators of spatial autocorrelation

```
local_moran <- spdep::localmoran(
  nta_trip_nodes$trip_count,
  nta_trip_network_weights,
  zero.policy = TRUE,
  na.action = na.omit,
)

sig_lev <- 0.05
avg_trip_count <- mean(nta_trip_nodes$trip_count)

lisa_classes <- local_moran %>%
  tibble::as_tibble() %>%
  magrittr::set_colnames(
    c("Ii","E.Ii","Var.Ii","Z.Ii","Pr(z > 0)")
  ) %>%
  dplyr::mutate(
    coType = dplyr::case_when(
      `Pr(z > 0)` > 0.05 ~ "Insignificant",
      `Pr(z > 0)` <= 0.05 & Ii >= 0 & nta_trip_nodes$trip_count >= avg_trip_count ~ "HH",
      `Pr(z > 0)` <= 0.05 & Ii >= 0 & nta_trip_nodes$trip_count <  avg_trip_count ~ "LL",
      `Pr(z > 0)` <= 0.05 & Ii < 0 & nta_trip_nodes$trip_count >= avg_trip_count ~ "HL",
      `Pr(z > 0)` <= 0.05 & Ii < 0 & nta_trip_nodes$trip_count < avg_trip_count ~ "LH"
    )
  )

nta_trip_network_cluster <- nta_trip_network %>%
  tidygraph::activate(nodes) %>%
  dplyr::mutate(coType = lisa_classes$coType %>% tidyr::replace_na("Insignificant"))

nta_sig <- nta_trip_network_cluster %>%
  dplyr::filter(coType != "Insignificant")
```

Visualize the confidence type for significant nodes

```
ggraph::ggraph(nta_sig, layout="stress") +
  ggraph::geom_node_circle(aes(r = 0.025, color = coType))
```


coType
HH
HL
LH
LL