Necessary libraries

```
library(tidyverse)
library(geojsonsf)
library(sf)
library(tmap)
options(scipen = 999)
```

Outline

- Changes to the topic
  - Shift from trying to predict changes in dynamics to instead focusing on how much of an impact each travel mode makes
- Changes to the timeline
  - Need to create a timeline
  - retroactively mark when progress has been made. Look forward to what progress still needs to be done.
- Review of the original timeline:
  - more detailed information on what has changed for the project
- A more detailed version of the methods to be applied.

Questions

- corrections for changes in census tracts ### Updated approach
- Network auto-correlation of OD desire lines for 2019
- Network auto-correlation of MTA system
- Network auto-correlation of Highway system
- Network correlation of contiguous NTAs
- Regression to compare the MTA and Highway systems, to see which has the greatest effect
- Is there are way to measure the shift in the networks?

Started with interest in Triboro line ridership - Interest in the four boroughs that are severed by the subway system - Manhattan overwhelms the travel patterns throughout the boroughs of interest - Insufficient data to track the total ridership and the means - Travel surveys did not offer the level of detail or confidence - Shifted to looking specifically at work commuting patterns within a single borough - Focused on what correlation each transporation system has with the OD patterns

Data sources

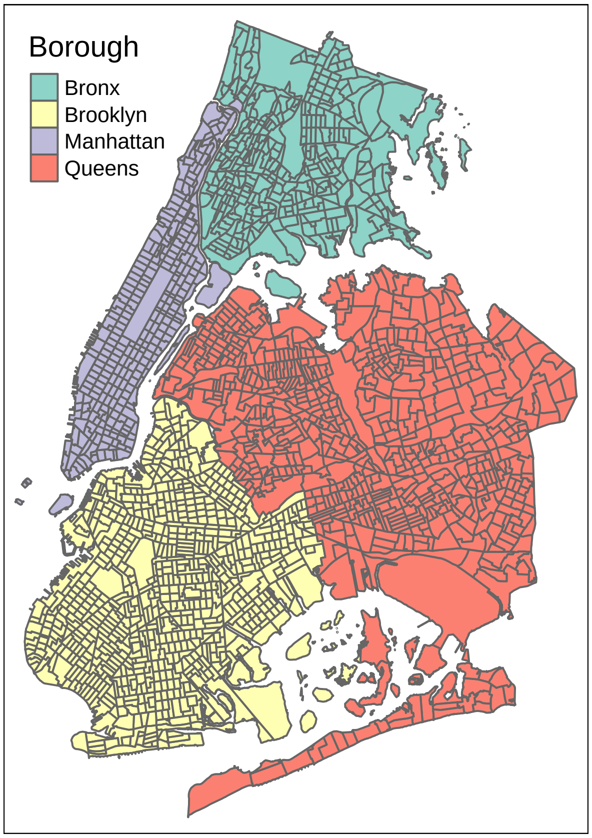
- Origin and Destination data for NYS in 2019 at the block level
- NYC Census tract borders
- Borders of NYC boroughs
- Equivalency of Neighborhood Tabulation Areas
- Subway routes
  - Good for data exploration
  - Will need to be suplemented with a routing service
- Arterials and Major Roads
  - Good for data exploration
  - Will need to be supplemented with a routing service

Visualize the census tracts and ntas, providing context for their size

```
bois_names <- c("Manhattan", "Bronx", "Brooklyn", "Queens")
bois_census_tract_borders <- geojson_sf('./data/nyc_2010_census_tract_borders.geojson') %>%
```

```
  dplyr::filter(BoroName %in% bois_names)

tmap::tm_shape(bois_census_tract_borders) +
  tmap::tm_polygons(
    col = "BoroName",
    title = "Borough"
  )
```



Estimate the average distance from the center to the edge of tract

```
avg_tract_area <- sum(bois_census_tract_borders$Shape__Area)/ length(bois_census_tract_borders$Shape__Ar
avg_tract_radius <- sqrt(avg_tract_area/pi)
print(avg_tract_radius)
```

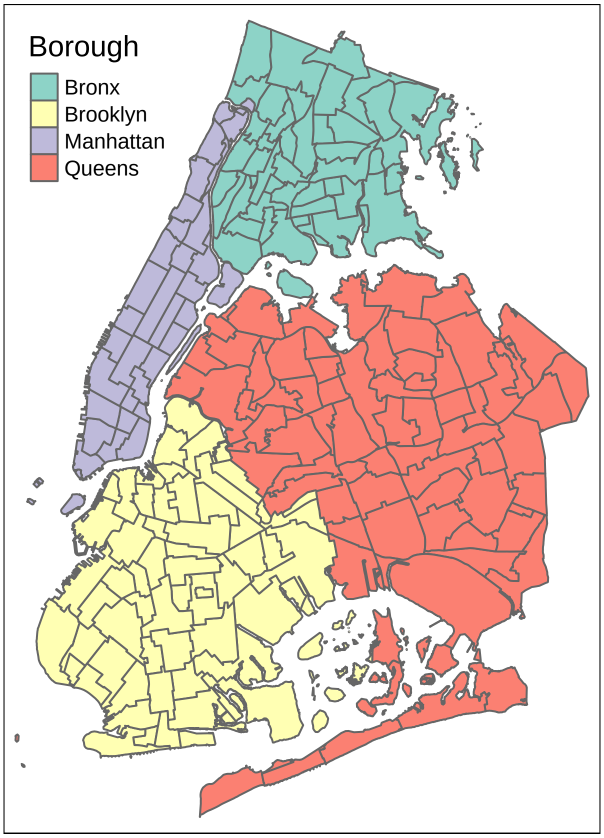
```
## [1] 1026.25
```

Visualize the ntas

```
bois_nta_borders <- geojson_sf('./data/nyc_2010_nta_borders.geojson') %>%
  dplyr::filter(BoroName %in% bois_names)

tmap::tm_shape(bois_nta_borders) +
  tmap::tm_polygons(
    col = "BoroName",
    title = "Borough"
```

```
  )
```




```
avg_nta_area <- sum(bois_nta_borders$Shape__Area) / length(bois_nta_borders$Shape__Area)
avg_nta_radius <- sqrt(avg_nta_area/pi)
print(avg_nta_radius)
```

```
## [1] 3506.898
```

Demonstrate the disparity between Manhattan and the other boroughs of interest

Create map between census tracts and ntas

Specify county because census tracts may not be unique across counties

```
bois_county_tract_nta_equiv <- readxl::read_xlsx('./data/nyc_2010_census_tract_nta_equiv.xlsx') %>%
  filter(borough_name %in% bois_names) %>%
  mutate(county_tract = str_c(`county_code`, `census_tract`)) %>%
  select("county_tract", "nta_code")
```

Reduce NYS origin destination data to only ntas of interest

```
bois_county_codes = c("061", "005", "047", "081") # Manhattan, Bronx, Brooklyn, Queens
bois_park_ntas <- c("BX10", "BX99", "BK99", "MN99", "QN99")
bois_nta_ods <- read_csv('./data/ny_od_main_JT00_2019.csv') %>%
  # Select only tracts within the boroughs of interest
  dplyr::filter(
    stringr::str_sub(as.character(w_geocode), 3, 5) %in% bois_county_codes &
      stringr::str_sub(as.character(h_geocode), 3, 5) %in% bois_county_codes
```

```r
    ) %>%
  # Create fields specifically for home counties and tracts
  dplyr::mutate(w_county_tract = stringr::str_sub(as.character(w_geocode), 3, 11)) %>%
  dplyr::mutate(h_county_tract = stringr::str_sub(as.character(h_geocode), 3, 11)) %>%
  # Narrow table down to tracts and all jobs
  dplyr::select(h_county_tract, w_county_tract, S000) %>%
  # Relate tracts with ntas
  dplyr::left_join(bois_county_tract_nta_equiv, c("h_county_tract" = "county_tract")) %>%
  dplyr::rename(h_nta_code = nta_code) %>%
  dplyr::left_join(bois_county_tract_nta_equiv, c("w_county_tract" = "county_tract")) %>%
  dplyr::rename(w_nta_code = nta_code) %>%
  # Remove trips within the same nta, only inter-nta trips are of interest
  dplyr::filter(w_nta_code != h_nta_code) %>%
  # Remove trips involving park NTAs, they are not true neighborhoods
  dplyr::filter(!(w_nta_code %in% bois_park_ntas) & !(h_nta_code %in% bois_park_ntas)) %>%
  # Label trips based on home and work ntas
  dplyr::mutate(od = str_c(h_nta_code, w_nta_code)) %>%
  # Count the number of trips made between these ntas and in this direction
  dplyr::group_by(od) %>%
  dplyr::summarise(
    h_nta_code,
    w_nta_code,
    S000 = sum(S000),
    ) %>%
  # Remove duplicate entries
  unique()
```

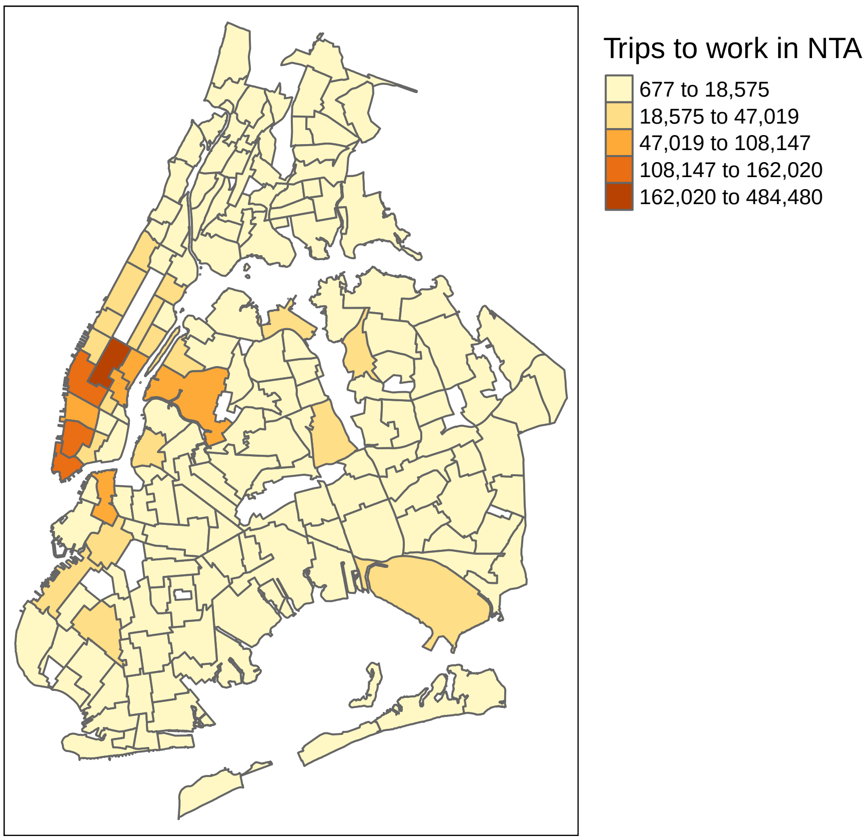
Define the most popular neighborhoods to work for those who live in the boroughs of interest

```r
get_nta_dest <- function(ods_of_interest) {
  ods_of_interest %>%
    dplyr::group_by(w_nta_code) %>%
    dplyr::summarise(
      w_nta_code,
      S000 = sum(S000)
    ) %>%
    unique() %>%
    left_join(bois_nta_borders, c("w_nta_code" = "NTACode")) %>%
    st_as_sf()
}

bois_nta_dest <- bois_nta_ods %>%
  dplyr::group_by(w_nta_code) %>%
  dplyr::summarise(
    w_nta_code,
    S000 = sum(S000)
  ) %>%
  unique() %>%
  left_join(bois_nta_borders, c("w_nta_code" = "NTACode")) %>%
  st_as_sf()

tmap::tm_shape(bois_nta_dest) +
  tmap::tm_polygons(
    col = "S000",
```

```
    style = "jenks",
    title = "Trips to work in NTA",
    legend.outside.width = 0.6
  ) +
  tmap::tm_layout(
    legend.outside = TRUE,
  )
```



Define the desire lines for work trips throughout the boroughs of interest

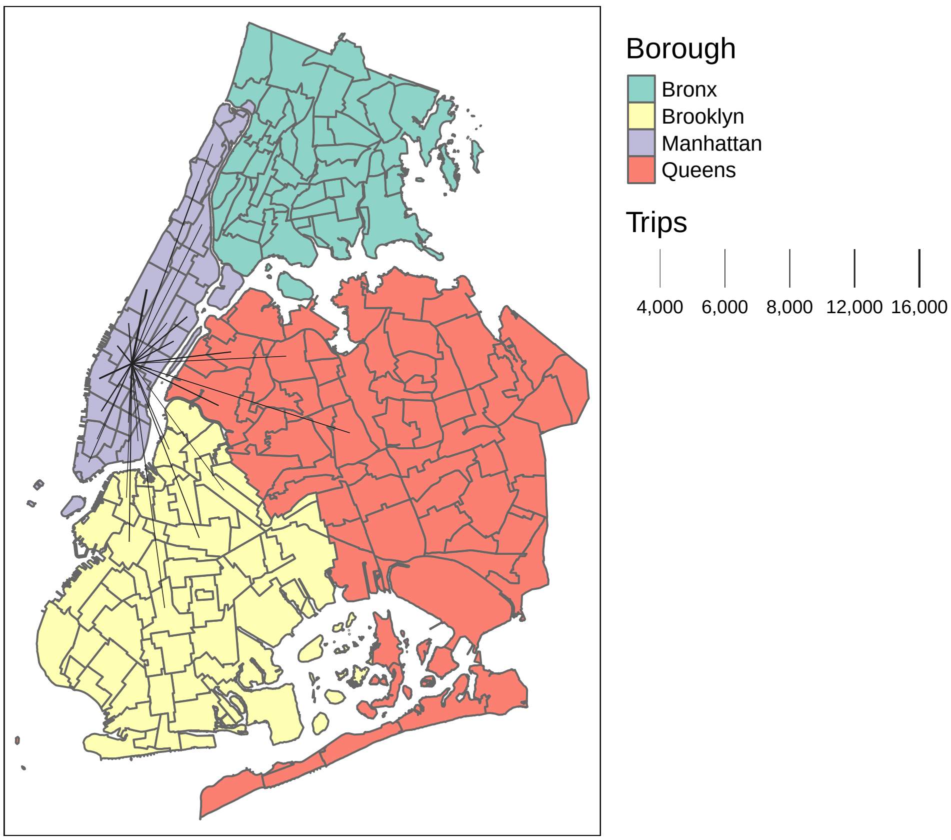
```
# Points on surface
bois_nta_pos <- bois_nta_borders %>%
  dplyr::mutate(geometry = sf::st_point_on_surface(geometry))
```

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

```
bois_nta_od_lines <- bois_nta_ods %>%
  dplyr::left_join(bois_nta_pos, c("h_nta_code" = "NTACode")) %>%
  dplyr::rename(h_geometry = geometry) %>%
  dplyr::left_join(bois_nta_pos, c("w_nta_code" = "NTACode")) %>%
  dplyr::rename(w_geometry = geometry) %>%
  dplyr::mutate(geometry = sf::st_union(h_geometry, w_geometry))%>%
  dplyr::mutate(geometry = sf::st_cast(geometry, "LINESTRING")) %>%
  dplyr::select("od", "S000", "geometry") %>%
  sf::st_as_sf()
```

```r
tmap::tm_shape(bois_nta_borders) +
  tmap::tm_polygons(
    col = "BoroName",
    title = "Borough",
  ) + tmap::tm_shape(dplyr::filter(bois_nta_od_lines, S000 > 5000)) +
  tmap::tm_lines(
    col = "#212121",
    lwd = "S000",
    title.lwd = "Trips",
  ) + tmap::tm_layout(
    legend.outside = TRUE
  )
```

```
## Legend labels were too wide. Therefore, legend.text.size has been set to 0.6. Increase legend.width
```



Define the most origin destination profiles for trips within the same borough

```r
## NTA borders for each borough
mn_nta_borders <- bois_nta_borders %>%
  dplyr::filter(BoroName == "Manhattan")
bx_nta_borders <- bois_nta_borders %>%
  dplyr::filter(BoroName == "Bronx")
bk_nta_borders <- bois_nta_borders %>%
  dplyr::filter(BoroName == "Brooklyn")
qn_nta_borders <- bois_nta_borders %>%
  dplyr::filter(BoroName == "Queens")
```

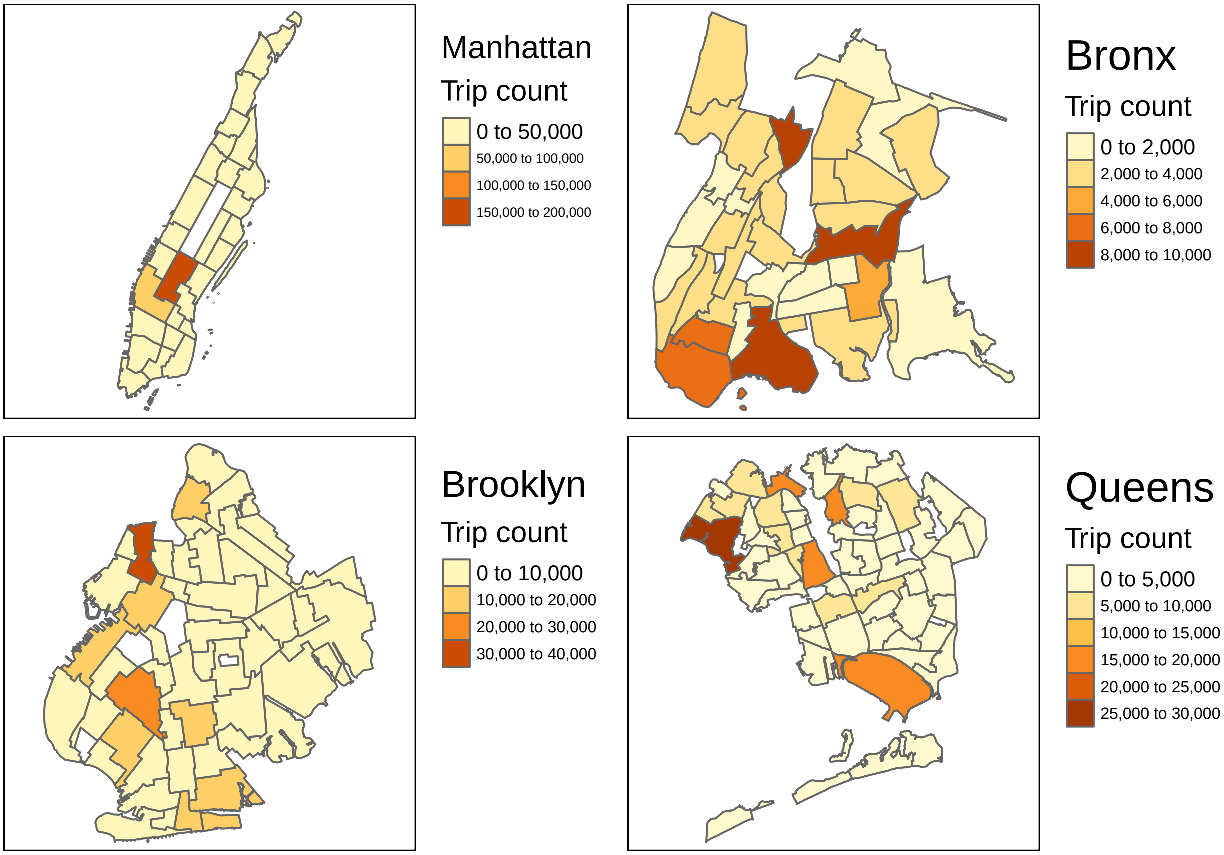
```r
## Intra Borough trips
intra_bois_nta_ods <- function(boro_abrv) {
  bois_nta_ods %>%
    dplyr::filter(
      stringr::str_sub(h_nta_code, 1,2) == boro_abrv &
      stringr::str_sub(w_nta_code, 1, 2) == boro_abrv
    )
}
mn_nta_ods <- intra_bois_nta_ods("MN")
bx_nta_ods <- intra_bois_nta_ods("BX")
bk_nta_ods <- intra_bois_nta_ods("BK")
qn_nta_ods <- intra_bois_nta_ods("QN")


## Most popular destinations for trips within a Borough

mn_nta_dest <- get_nta_dest(mn_nta_ods)
bx_nta_dest <- get_nta_dest(bx_nta_ods)
bk_nta_dest <- get_nta_dest(bk_nta_ods)
qn_nta_dest <- get_nta_dest(qn_nta_ods)

mn_dest_map <- tmap::tm_shape(mn_nta_dest) +
  tmap::tm_polygons(
    col = "S000",
    title = "Trip count"
  ) + tmap::tm_layout(
    legend.outside = TRUE,
    title = "Manhattan"
  )
bx_dest_map <- tmap::tm_shape(bx_nta_dest) +
  tmap::tm_polygons(
    col = "S000",
    title = "Trip count"
  ) + tmap::tm_layout(
    legend.outside = TRUE,
    title = "Bronx"
  )
bk_dest_map <- tmap::tm_shape(bk_nta_dest) +
  tmap::tm_polygons(
    col = "S000",
    title = "Trip count"
  ) + tmap::tm_layout(
    legend.outside = TRUE,
    title = "Brooklyn"
  )
qn_dest_map <- tmap::tm_shape(qn_nta_dest) +
  tmap::tm_polygons(
    col = "S000",
    title = "Trip count"
  ) + tmap::tm_layout(
    legend.outside = TRUE,
    title = "Queens"
  )
```

```
tmap::tmap_arrange(mn_dest_map, bx_dest_map, bk_dest_map, qn_dest_map, nrow = 2, ncol = 2)
```



Total trips in each borough

```
print("Intra borough trips")
```

```
## [1] "Intra borough trips"
```

```
sprintf("Manhattan: % s", sum(mn_nta_ods$S000))
```

```
## [1] "Manhattan: 508873"
```

```
sprintf("Bronx: % s", sum(bx_nta_ods$S000))
```

```
## [1] "Bronx: 114719"
```

```
sprintf("Brooklyn: % s", sum(bk_nta_ods$S000))
```

```
## [1] "Brooklyn: 370256"
```

```
sprintf("Queens: % s", sum(qn_nta_ods$S000))
```

```
## [1] "Queens: 265386"
```