Exploratory Data Analysis (EDA)

CRS: EPSG 2154 (Lambert 93 - France)

Hypothesis: (1) we can spatially identify transit desert clusters, and (2) these clusters will have lower socioeconomic outcomes like higher unemployment and lower median income.

Data Acquisition:

- RATP - Transit data
  - Stations
  - Route Networks
  - GTFS
- INSEE - Census Data
  - 2021 mode-de-vie: https://www.insee.fr/fr/statistiques/8268843 (2019 census)
  - 2021 Income: https://www.insee.fr/fr/statistiques/8229323 (2019 census)
- Esri France - Administrative Polygons
  - Region (Ile-de-France; use as clip layer)
  - Departement
  - Arrondissement
  - Commune
  - IRIS: https://esrifrance.hub.arcgis.com/datasets/7558906c371d46e883069f845d0161e7_0/about

QGIS Visualizations

- Transport station points
- Transport route networks
- Administrative polygons (Region > Department > Arrondissement > Commune > IRIS)

INSEE Census Data

- **Jupyter Notebook: 1_census_data_processing.ipynb**

Perform necessary calculations on census data to get % rates

- IRIS census calculations
  - Population
  - Median net/disposable income
  - % employed/unemployed/retired/studying
  - % working outside of area
  - % commute method

Spatial Processing

Administrative Boundaries

- **Jupyter Notebook: 2_admin_boundaries_merge.ipynb**

  Preparing the administrative boundaries for spatial analysis and choropleth maps
  - In communes layer, replace Paris polygon with 20 Paris arrondissements; and replace arr code with comm code
  - Generate new communes geojson [Communes_iledefrance_with_parisarrs.geojson]
  - Combine all administrative layers into single geopackage [admin_polygons_allayers.gpkg]
  - Spatial Join associated Region, Department, Arrondissement, and Commune for each IRIS; for Arrondissements will need to handle IRIS that are in multiple Arrondissements with overlay intersection
  - Compute centroid for each IRIS (EPSG:2154)
  - Output as final geopackage to use for future spatial joins and choropleths [IRIS_with_RegDepCommArr.gpkg]

Travel Time Data per station

- **Jupyter Notebook: 3_isochrone_traveltime.ipynb**

  Calculate travel times to transit stations in Paris metro area
  - Acquire 1min isochrone increments for travel time on public transit from central station [TravelTime API]
  - Process and save each isochrone as geojson entity
  - Generate combined geopackage with all isochrone geojson layers [Paris_3hr_isochrones_combinedlayers.gpkg]
  - **Assign travel time to each station within isochrone catchment areas**
  - Work from largest isochrone down to smallest to catch stations and assign travel times; catchment overwrites overwrites previous time entry based on station location in polygon hierarchy
  - Save stations data with travel time [Stations_with_traveltime.gpkg]

Transport Accessibility per IRIS

- **Jupyter notebook: 4_transport_accessibility.ipynb**

  Find which IRIS our stations are located in
  - Identify which IRIS each station belongs to / save IRIS code to each station
  - Record distance of closest station to IRIS centroid (calculate distance)
  - Calculate average walking time for each station to its IRIS centroid

  Get basic information on transport accessibility per IRIS
  - Count stations in each IRIS

- Calculate average walking time between IRIS centroid to all stations in IRIS
- Save data on transport accessibility per IRIS []

<u>Spatial Joins</u>

- **Jupyter Notebook: 5_IRIS_spatial_joins.ipynb**
  Spatial joins to add required information to each IRIS entity
    - Census data: each IRIS should have population, unemployment rate, income, etc.
    - Transit time: each IRIS should have the average travel time (to Paris Center) for its centroid station
    - Transport accessibility: each IRIS should have station count, distance of closest station to centroid, average walking time to stations
  - **Spatial joins to add required information to each Station**
    - Census data: each station should have its census data from its IRIS (population, unemployment rate, income, etc.)

<u>Aggregating calculations by administrative boundaries</u>

- Aggregate to quantify data by IRIS > Commune > Arrondissement > Department
  - Census data
  - Transit network
  - Transport Accessibility

Spatial Analysis

Identify "transport deserts"

- Visually inspect which IRIS clusters can be considered transport deserts
    - Based on travel times
    - Based on station accessibility
    - Based on population taking car versus public transit to work

ContiguityWeights - neighbor analysis

- Determine if the neighborhoods of a transport desert are themselves transport deserts
- Identify clusters of transport deserts

Hypothesis Plots

- The first iteration of analyzing transport desert clusters did not prove our hypothesis, as both unemployment and median income were better in transport deserts than non-transport-deserts. The parameters for what constituted a transit desert was the worst decile of the distribution of average walk time to stations in an IRIS, and the worst decile of travel time on public transportation to reach the center of Paris from the closest station to the centroid of an IRIS.
- The second iteration took into consideration that wealthier people were likely able to live in areas without public transport due to being able to afford cars. As a result, a new set of parameters was set for the contiguity neighbor analysis, for commuters needing over 30 minutes to reach the center of Paris, and by filtering on IRIS residents who relied on public transport as a means to get to work and commuters who work in a different department than their residence (over ⅓ of the IRIS population for both). This decision ended up proving our hypothesis as seen in the second set of plot distributions.