Methodology

Overview of the Project

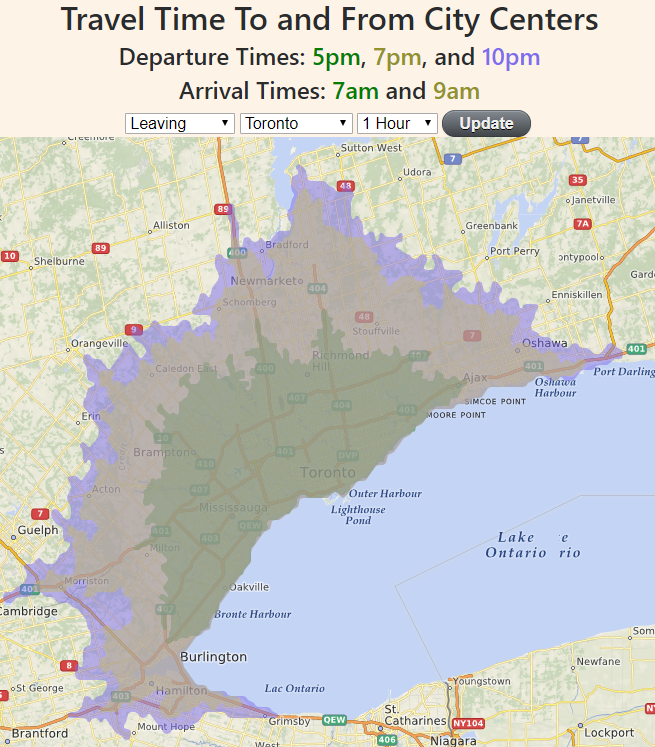
The project’s objective is to take GPS travel times to and from major city centers in Canada and to calculate the distance one can travel for given time periods. The purpose of this project is to assess the benefit of commuting during rush hours.

The time periods covered for coming into a city center are 7am and 9am. The time periods for leaving a city center are 5pm, 7pm, and 10pm. The travel time limit was selected as 30 and 60 minutes respectively.

The ten major Canadian cities selected for this project include Calgary, Edmonton, Halifax, Montreal, Mississauga, Ottawa, Quebec City, Toronto, Vancouver, and Winnipeg.

Steps

1. The first step was to find an online resource to pull distances based on GPS data. HERE Technologies’ 90 day free Automated Programming Interface (API) trial was selected to pull data and convert into a geospatial format.
2. Once an account was created for HERE, an App ID and App Code were generated to make calls to the API.
3. The geocoordinates were collected for each major city by Liam Donaldson.
4. Decision to use the HERE Isoline API to get shape files given parameters.
5. Python code was created to set parameters for:
   * Travel time limit (string)
   * Departure/arrival time (string – in ISO 8601 format)
   * City names (list)
   * City center geo coordinates (list)
   * Car/pedestrian transit (string)
   * Json request (list)
   * App ID (string)
   * App Code (string)
6. The parameters were entered into the API request to return a json object. The request was formulated using the [Routing API Documentation](https://developer.here.com/documentation/routing/topics/resource-calculate-isoline.html#resource-calculate-isoline).
7. The json object was then iterated over for each city to return a text file with a list of geocoordinates that created a polygon.
8. The files were converted to shape files using geojson.io for Murtaza Haider to analyze. Shape files were unzipped using python code. In addition, geojson files were generated by grouping travel times for each respective city in geojson.io.
9. A web interface was developed by using html inputs with D3.js to visualize the geojson files on top of a Leaflet.js map. The geojson files were called based on the html inputs and were then fed into the D3.js visualization to update the polygons on the map.



The Python and R version of Scrapper are available on the [Github](https://github.com/Shafquat/here). Credit to Sahil Chinoy for a framework on using HERE’s Isoline API and inspiration for this project.