**Part 1: Setting up, converting, and formatting the data**

1. Extract geographical boundaries of 2016 census tracts from Statistics Canada in the form of shapefiles and accompanying data.
2. Filter shapefile data to produce a map of only Toronto’s census tracts using Statistics Canada’s official geographic designation. This will produce a map of the greater Toronto metropolitan area.
3. Convert the geometric polygon vertices stored in shapefile to lat/long coordinates in EPSG 4326 to allow for direct querying of the Google Maps API.
4. Make a CSV file to hold the following columns: TractID, centroid, area, walk\_time, bike\_time, transit\_time, driving\_time, quickest\_method. The times will be recorded in seconds as they are pulled from the Maps Directions API, and the centroid will be the set of coordinates that is used to request data from Google Maps.
5. Convert the shapefile data to CSV format and populate the relevant columns of the CSV file produced earlier.
6. Find the centroids of all census tracts and store them in their respective rows of the CSV file.
7. Format out-of-place or incorrectly converted data. This process will likely have to be done manually.

*Note: I am considering CSV format as opposed to SQL for example, because it is easier to read from and write to in Python and synergizes well with the libraries* ***geopandas*** *and* ***matplotlib*** *specifically.*

**Part 2: Pulling routes/requests from Google Maps**

1. Decide upon the nodes to use for route calculation. So far, we have considered: UofT, Union Station, and Pearson Airport. Find the relevant coordinates/placeIDs from Google Maps for each of these nodes once finalized.
2. Store the relevant placeIDs/coordinates of the selected nodes as ‘origin’ parameters and use the centroids of each census tract as ‘destination’ parameters.
3. Pull routes between **all origins and destinations** for all 4 methods of travel: driving, transit, biking, and walking, in JSON format.
4. Convert JSON output from Google Maps to CSV or Python dictionaries.

**Part 3: Filtering the data and producing the chloropleth map:**

1. Record shortest route duration for each method of travel for each census tract in CSV file.
2. Record which of the 4 methods has the shortest travel duration in CSV file for each census tract.
3. Merge the CSV with the shapefile on TractID column to produce a masterfile (as a pandas.DataFrame object) which can be used to produce the final map.
4. Assert formatting and presentation decisions like the legend, output file resolution, and colour palette.
5. Generate the final, coloured map(s) **for each of the selected origin nodes**.
6. Deliver map(s) to Professor Hall.
7. Deliver formatted code with instructions for future use to Professor Hall.