ANSWERS - Location-based analysis exercises

This module puts the skills learned in previous modules into practice. Using a public dataset from Kaggle, a number of location-based analysis questions will be posed to answer using Constellation. The questions have been separated into easy, intermediate, advanced, and experimental (the hardest). They are designed to be completed in order.

Note for moderators: Most of these questions have been designed to test the skills of the participants, rather than fitting into any broader 'game' or 'theme' running through other parts of ORCS. I'm happy to add additional questions to fit with this, they would mostly be around:

- *Where has X car visited, or where was X car on Y date?*
- *Which cars were seen near Z location on Y date?*

Additional attributes can be added to cars or locations that can link to other parts of the ORCS games if required.

Details on the dataset: Israeli car-sharing company locations
https://www.kaggle.com/gidutz/autotel-shared-car-locations
"In order to reduce the number of owned cars, the city of Tel Aviv launched a shared-car project, called AutoTel. Users of the service are able to reserve a car using a mobile app, and pay for it by the minute. The project that was launched in October 2017 attracted over 7500 users, with more than 50% of them using the service at least once a week. From the AutoTel website we extracted the location of the parked cars, every two minutes for several months."

The data has been cleaned and imported into Constellation already. Open 'CarlocsFeb2020.star'. You'll see car nodes connected to location nodes, with lat/longs includes so they can be plotted in the Map View.

Easy Questions:

- How many cars are there on the graph?
  - *Histogram on node type, count cars.*
  - *Answer: 260*
- How many unique locations are there on the graph?
  - *Histogram on node type, count locations.*
  - *Answer: 960*
- Which location has been visited by the most cars? Can you find a real world place near the location on the map?
  - *Histogram on node type, filter on location. Histogram the graph property of neighbour count.*
  - *Answer: loc_7 has been visited by 14 cars. Google maps layer indicates this is close to cheaper car audio shop and a restaurant called Suduch.*
- Which location has had the most visits in total? Is it the same location? Can you find a real world place near the location on the map?
  - *Histogram on node type, filter on location. Histogram the graph property of transaction count.*

    - *Answer: location is the same: loc_7*
- Which two cars have visited the most locations?
    - *Histogram on node type, filter on cars. Histogram the graph property of neighbour count.*
    - *Answer: Car 203 and Car 3 both visited 12 unique locations.*

Intermediate Questions:

- Which date has had the most locations recorded?
    - *Histogram on transaction DateTime. Change format to Date.*
    - *Answer: 12 February 2020 with 123 transactions*
- Find cars 189 and 88. Have they visited any of the same locations?
    - *Select the cars using the Histogram, or using Ctrl+i to search and select. Use the one-hop induced subgraph tool to select common locations.*
    - *Answer: Yes, location 993*
- Find a car that has visited near Cafe Zorik on Yehuda HaMakkabbi St, next to Milano Square
    - *Using the internet, search for Cafe Zorik. In Constellation, search for the location (you can use the 'zoom to location' lat/longs, or just look for it), then select a location recorded nearby. Use the one-hop tool to find a car connected to this location.*
    - *Answer: locations 749, 230, and 1358 could all be said to be near the location. Any cars connected to those locations are correct answers, but Car 184 is probably the closest.*

Advanced Questions:

- Find a car that has visited an area in the North (above 32.115 latitude) and the South (below 32.060 latitude)
    - *Using the map view, select locations in the north. Hop out one to select cars seen at these locations. Histogram on node type car, and blaze the cars. Using map view, select locations in the south. Hop out one to select cars seen at these locations. Histogram on node type blaze, select all cars with a blaze.*
    - *Answer: There are multiple answers to this question depending on where exactly you draw the boxes. As long as you can take a car from your answer, hop out to see it's locations, and prove that it has been in both the north and the south, then that is correct.*
- Find a car that has visited less than three locations overall, with one location in the North (above 32.115 latitude) and the South (below 32.060 latitude).
    - *Using the map view, select locations in the north. Hop out one to select cars seen at these locations. Histogram on node type car, and blaze the cars. Using map view, select locations in the south. Hop out one to select cars seen at these locations. Histogram on node type blaze, select all cars with a blaze. Histogram on graph property neighbour count, find cars with less than three neighbours.*

    - *Answer: Cars 43 and 77 should be in the results, but it's possible there are more based on where the boxes were drawn. If you can prove you have a car that meets the criteria, it will be correct.*

Challenge:

- Which date has had the most unique cars registering their location?
    - *There isn't really a great way to answer this question in Constellation that I'm aware of. You can break DateTime down into Date, but this shows the number of transactions, not unique cars. You can then select a date one at a time, hop out half, then histogram on node type to count the cars, but this doesn't scale well. If this was a recurring problem to answer, you might need to transform the data to create summarised Edges instead of transactions. You could do this in the scripting view, or transform the data before importing.*
- Find a car that has visited the same location as another car within 2 hours of each other, but not at exactly the same time.
    - *First, Histogram on transaction DateTime, and sort by total count. Create a new graph with only those events with a count of one.*
    - *Second, use the arrange, Layer by Time function, separating by hours with an interval of at least 3. On the new graph, Histogram node type Location, then Histogram graph property of neighbour count and look for greater than one to find two cars connected to a single location, with an event occuring within two hours of each other, but not at the same time.*
    - *This exercise is limited by the use of intervals, because separating by an interval of 2 hours will start at 0 and iterate upwards. Thus you can miss two events happening at 0159 and 0201, as they will appear in separate layers. There isn't a good way around this that I'm aware of.*
    - *Answer: Cars 35 and 165 visited loc_1333 within 2 hours of each other.*