Lesson 3: Employee Trip Tracker Discussion

This lesson covered a few topics I have come across in previous learning on my own, and was quite beneficial on reviewing some of the basics of filepath searches, high-level functions, Regex, and manipulating data in Pandas. As we progress in this class with more visualization techniques and using Widgets within Jupyter to make use of the ESRI API, the underlying work of managing files, and getting the data validated becomes clearly important to the entire workflow.

In this exercise, building a shapefile from a much more familiar object, the pandas dataframe, really helped lock in my understanding of how to filter data into the layers, and build up each Field. Going through the tutorials to see how to get the shapefiles to actually show up through the ESRI API was a big step, though I am still quite a novice in manipulating the widgets any further than adding an entire new layer.

**For the Above and Beyond:**

I tried to read more documentation on how to color lines per employee differently, as this seemed to be an obvious inclusion to improve my visualization. Unfortunately, I became a bit confused in whether this was more of a widget functionality, or possibly a requirement for a separate module in python, as I did not seem to find exactly the right set of documentation within GDAL.

Instead, I stuck to the theme of the lesson with data science, and ran a number of basic statistics on the resulting employee list, and included the recommended duration filter max and min, as this was a very simple feature to implement.

The analysis includes a first run through of the results csv, iterating per each employee.

['Name', 'total\_trips', 'total\_travel\_days',

'total\_cities', 'cities\_visited', 'total\_time\_span',\

'longest\_trip\_days','shortest\_trip\_days',\

'most\_stops\_in\_single\_trip', 'total\_distance\_mi']

Most of these are quite simple max/min built in functions, but the final “Distance” calculator is much more interesting. In this calculation I re-do the steps needed to split the route, in order to add the first city to the end of the list, for a return to home inclusion in the calculation. I then go back through the process using GDAL to create the line string and have a distance counter that utilizes the osgeo.Length() method.

To be honest, I had no idea what the result value was telling me, so I simply opened up a google maps tab, to punch in directions for estimated distances between cities, and looked up some haversine functions to see how close this value was to a more palatable distance in miles/km.

After a few quick tests, I realized the value just simply needed to be converted to radians, and multiplied by the Radius of the Earth (in Miles/Km based on preference), and proceeded to add this field into my analysis to help calculate more interesting statistics in the next step.

Once the quick sums were finished, I made some other intuitive averages, percentages, and ratio calculations. These are what I would assume to be important KPI’s if I were a manager looking at my team’s travel history to detect if someone may be exhausted, underutilized, or has the potential to travel to a new city (not included in their current list of “cities visited”) to help balance the trip allocation between the team for the next set of trips coming up.

The never ending scroll bar with the addition of more columns certainly reminds me of the ad-hoc reports I would review daily during my time as a wireless network system performance engineer in the telecom industry, so this was a bit of the inspiration behind the formulation of these “KPIs”.