Analysing the TfL tube network

Introduction

For my final project I analysed and visualised TfL's open data on the London Underground. I combined data from their APIs and csv files to create data frames that I used to generate text and scatter maps.

Initial exploration

Initially I wanted to use the Spotify API to compare two user's listening data but with the recent changes I had trouble working with Spotipy, so I looked into other available data.

I started by finding out what data TfL had available. I found two main sources. The first was a collection of APIs with data for different categories, and a link to a zip file which contained csv files. I also found a page listing their open data which linked back to the APIs and had more downloadable files.

I read the APIs descriptions and created a jupyter notebook to explore the data in more detail using requests and pandas. I also downloaded the zip file of csvs and did discovery on those too. This is shown in my files TfL_API_Lifts.ipynb, TfL_API_analysis.ipynb and TfLcsvs.ipynb.

While I was doing this I started brainstorming ideas for what I could visualise with the data available. These ideas included analysing crowding data, lift disruptions, line statuses and bike point occupancy, then visualising this using bar graphs, heat maps and scatter maps.

Ultimately based on the time constraints and the data available I decided I wanted to create a map visualisation of different stations in London and their amenities.

The process and challenges

Now that I had an idea of what I wanted to do, I started looking into python libraries that would allow me to visualise maps. I looked at matplotlib, plotly, folium, ipyleaflet and bokeh. I decided on plotly because the maps are interactive, there was customisation available and I liked the aesthetics of the example visuals.

I read the documentation and practiced plotting locations from the data frames I explored earlier using latitude and longitude values. I also did some basic styling. This is shown in my file PlotlyTests.ipynb.

While plotting this data I noticed that some stations in the data were outside of London. This lead to me finding out how to plot within a polygon boundary. Initially I wanted to break down data of all stations by mode, borough boundary and display their amenities. But as time was going on I realised this was a bigger scope of work than I anticipated with the time available, and decided to focus on one mode of transport - the tube.

My plan was to combine data from one API, with data from the stations.csv and toilets.csv to list selected amenities for each tube station. I started writing the code to analyse one tube line, which I would later turn into a function to be used on all tube lines.

After I merged the API data frame with the stations.csv file some columns had empty values which were not present in the individual data frames. After further analysis I discovered that not all of the tube stations were included in the csv. This meant my data on amenities would be incomplete.

I looked online to see if I could get the information from another source. After another review of the API web pages, I realised there was information on amenities nested inside a list of dictionaries in a column of the tube API data frame which potentially included the missing information I would need and more. I didn't realise earlier because the analysis methods I used to get an idea of what was in the data frame didn't display the entirety of the column. I was unsure of how to access this information to create a new data frame for my visualisation, as shown in my file Tube_API_amenities.ipynb, so with the remaining time I decided to continue with the data I had.

I successfully wrote some code that created a scatter map of one tube line, and also printed a couple of sentences about the London underground. This is shown in my jupyter notebook file ProjectTrial.ipynb. Initially I struggled with the data shown when hovering over markers. The empty values were displaying as code so I wanted only the available amenity information for that station to be displayed. However, after looking at forums I realised this was not possible with plotly, and there were other limitations with the hover date. Because of this, I replaced some values in the data frame so they would display in a way that made sense for the hover state.

I then figured out how to turn this code into a function that would display the scatter map and write more sentences about the tube line. I used a for loop to iterate over the different lines in a list I created from information in an API, then print them sequentially. I also created a dictionary using this list and colour values I found in a pdf online for the different tube lines. I used this to customise the markers on the map to reflect the tube line colour.

Reflection

Working on this project was a good opportunity to refresh my memory on the python knowledge I had gained on the course and learn how to do more. I understand more about what is and isn't possible with python. I know more about the plotly, pandas and shapely libraries, and how to work with data frames and APIs.

If I were to work on a similar challenge again I would do more investigation into what was available in every column of a data frame to avoid having empty values. If I were to try and compare the different tube lines or come to any conclusions about them based on the current data, there wouldn't be much credibility because there were many unknown values. I would have liked to have been able to make comparisons between the tube lines and visualise them, and also do a more in depth analysis of more modes of transport in the TfL network.

All data used in this project was factual open data with no biases.

Links

Final Project

Final project file

FinalProject.ipynb

Referenced jupyter notebooks

TfL_API_Lifts.ipynb
TfL_API_analysis.ipynb
TfLcsvs.ipynb
PlotlyTests.ipynb
Tube_API_amenities.ipynb
ProjectTrial.ipynb

Exercises & homework

Session 1

Tasks 1 & 2
Task 3
Task 4
Task 5
Task 6

Session 2

Labs all tasks

Session 3

Labs

Session 5

Openweather task
Spotify task
Youtube task