UCLouvain

ÉCOLE POLYTECHNIQUE DE LOUVAIN

LINMA 2472

Transport for London

December 2020

Group 23

Breno Tiburcio - 74042000
Nima Farnoodian - 6837200
Charles Rongione - 51841500

Professor : Jean-Charles Delvenne, Gautier Krings
Assistant : Alexey Medvedev, Rémi Delogne

Academic Year : 2020-2021

1 Introduction

Considerations

For our final project, we have opted for an urban question : Transportation. One of the most cosmopolitan cities, London is known for its sophisticated and highly developed transport system. Undoubtedly, it would be audacious of our part to point out problems or either make recommendations consistent enough. We are aware that we picked an example of a complex social system that requires deep insights.

However, we aim to grasp London's commuter journey by applying their journey information into a replica of Transport for London net. We created a directed and weighted network based on the part of the trips in November of 2009.

To exploit such a rich social study, we also implemented a current social theme in our network. As travelling safely on the bus, train and subway are the primary concern during this unprecedented period due to the current Covid-19 pandemic ; we opted for proposing a shortest-path recommendation framework in terms of safety in the London transportation network with the hope of reducing contamination risk for commuters. Indeed, we formalized and created a new term "safer shortest-path", which is meant to provide the commuters with low-risk and relatively short travel routes.

Objectives

The shortest-path is a common ground between transportation and network analysis. Our aim with this study is to provide different shortest-path taking into consideration different perspectives. Journey duration, journey cost, journey distance, journey safety are just examples of views that independently might not always converge to one shortest-path.

An important variable that, thanks to our data-set, we can emulate each trip segment's cost based on the number of people travelling —also known as stress.

In the followings chapter, you can encounter the detailed procedures that lead us to this work conclusion.

2 Data Analysis

For this project, we used different data sets to respond to our main questions such as the shortest-paths in terms of journey time length or distance, the journey patterns, how the people commute, and what could be the safer but also shorter paths. It is worth noting that, from now onward, the term "safety" will be referred to as only the safety concerning Covid-19 risk contamination. In what follows, we will go through our main data sets with details.

Data details

Our data-set initially contained over than one million entries reduced to seven hundred thousand after we discount all bus trips. The exact start and end station of each trip are essential to our study. Consequentially, we also discarded any other entry that was incomplete. Curiously, there is also a register of journeys that start and ends at the same station with time duration equals zero. Also excluded. Below the journey data-set metadata :

Index	Columns	Descrip.	# Entries	Type
1	downo	Day of week	729534	int64
2	daytype	Day of week	729534	object
3	SubSystem	Means of Journey	729534	object
4	StartStn	Journey start station	729534	object
5	EndStation	Journey end station	729534	object
6	EntTime	Entering Time (Min)	729534	int64
7	EntTimeHHMM	Entering Time (HHMM)	729534	object
8	ExTime	Exiting Time (Min)	729534	int64
9	EXTimeHHMM	Exiting Time (HHMM)	729534	object
10	ZVPPT	Ticket Zones	729534	object
11	JNYTYP	Ticket Type	729534	object
12	DailyCapping	Pays-As-Go Capping	729534	object
13	FFare	Pays-As-Go Before Discount	729534	int64
14	DFare	Pays-As-Go Discount	729534	int64
15	RouteID	Bus route ID	729534	object
16	FinalProduct	Ticket description	729534	object

Table 2.1 – TFL Journey's data-set

The core of our network is the London for Transport grid. We used the stations' locations (nodes) and the stations' connections (edge list) to replicate that. Into each node, we attributed the journey information and also COVID data from zip-code Geo-referenced study. To accomplish that we had to standardize all stations names.

This dataset was only providing the starting and ending point of the journeys. To do analytics, we needed to know how these stations were connected. That's why we used two other datasets : The tube lines data set that contains for each station, the stations it is associated with and the locations dataset that includes the geographical coordinates in the World Geodetic System (WGS). Combining

these two data sets, we could create this map where each point is a station, the connections are the edges, and the colours are the tube lines.

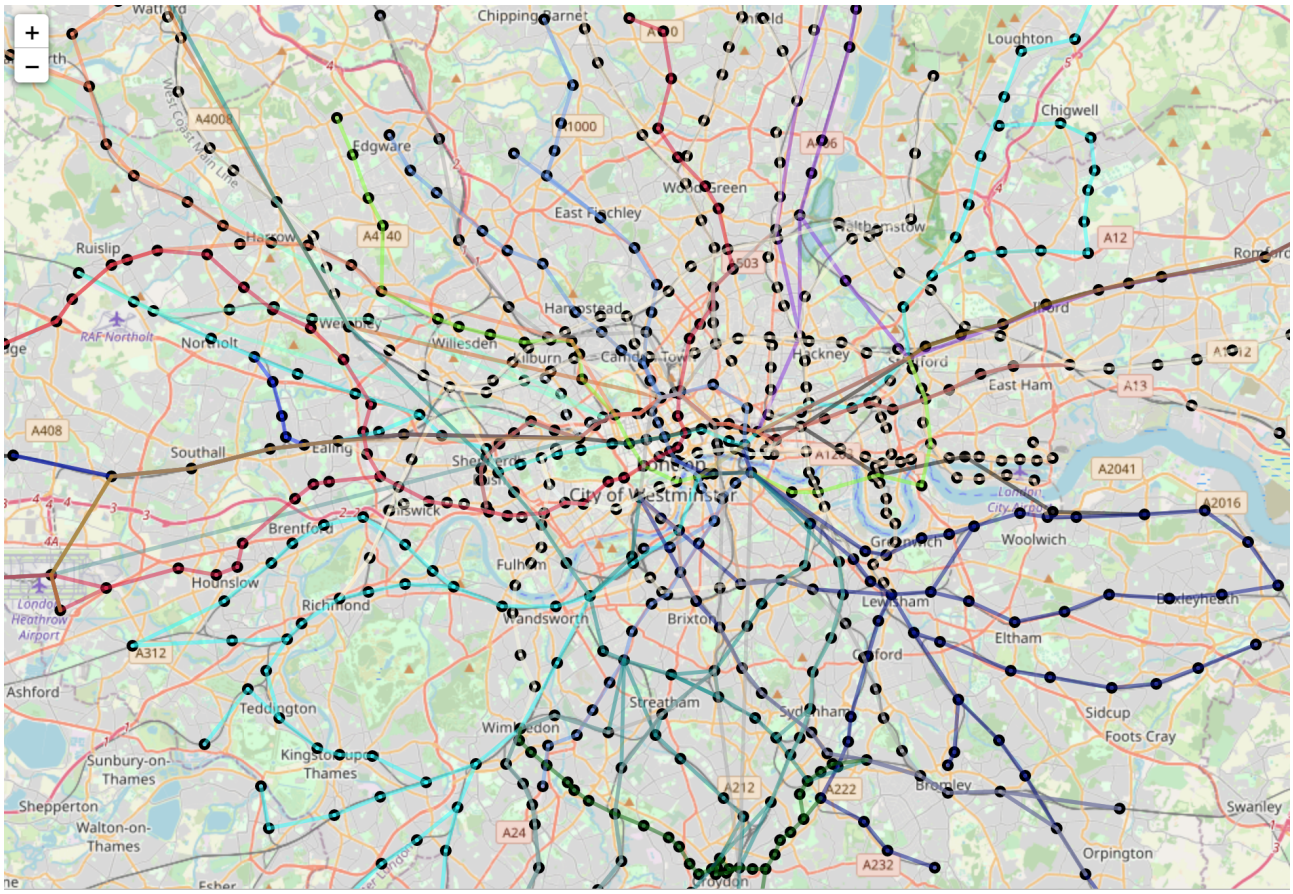

FIGURE 2.1 – Journeys boxplot

Data Treatment

Matching Station Names

The first challenge was to make our data sets compatible. Indeed, the stations' names in the journeys data set do not match the ones from the tube lines. It was challenging, but we manage to check all the names between the data sets (including Covid, tube line, Journey) to create our network. To match the names, we initially removed punctuation and space and lowered all names in all data-sets. Then we considered tube line data-set as our reference. Then, we took the difference of each data-set with tube line data-set to find unmatched names. Next, we found a close name for each odd name using "SequenceMatcher" method of "difflib" python package. SequenceMatcher gives a score between 0 and 1 for a pair of words where the more similar words get a score close to one. Using this technique, we could make all data sets compatible.

Transport For London Data-set

It was overriding that we find the distances between the stations ; otherwise, we could not find the shortest path. To do that, we converted the coordinates into radiance then processed the trigonometric distance, assuming that the radius of the earth is a constant of 8373 meters. These distances are not the real length of the transport path but a straight geometric line.

We could create the stations' physical network with the spaces between them as edges based on these distances. It is worth noting that we could find a real data-set —London stations gathered in 2017—with actual distances, but the problem was that there was no peer for many stations in our

journey network in that data-set. For example, for Stations Clapton and Acton Central, no peer exists in the London data-set. For this reason, we had to stick to the physical network we built for consistency.

Journeys Data-sets

Before inserting each node the journey attribute, we made our first assumption : we broke all the journey into stations steps considering that each passenger would take the least. With each journey distance (the sum of lengths of the 'step-stations') and start and end time, we could calculate the average trip speed.

Below the summary table about all the Journeys trip from our database :

	Duration (Hours)	Step-Stations (#)	Avg Speed (Kmh)
#obs	724434	724434	724434
minmax	(0.02 ;2.7)	(1 ;43)	(0.26 ;498.76)
mean	0.459	7.357	19.526
variance	0.065	19.791	85.960
skewness	1.104	0.915	6.245
kurtosis	1.793	0.9217	106.158

TABLE 2.2 – Statistics description after assumption one.

Below the box chart displaying the outliers and the distribution for each journey duration, the estimated number of step-stations and all passengers' average speed.

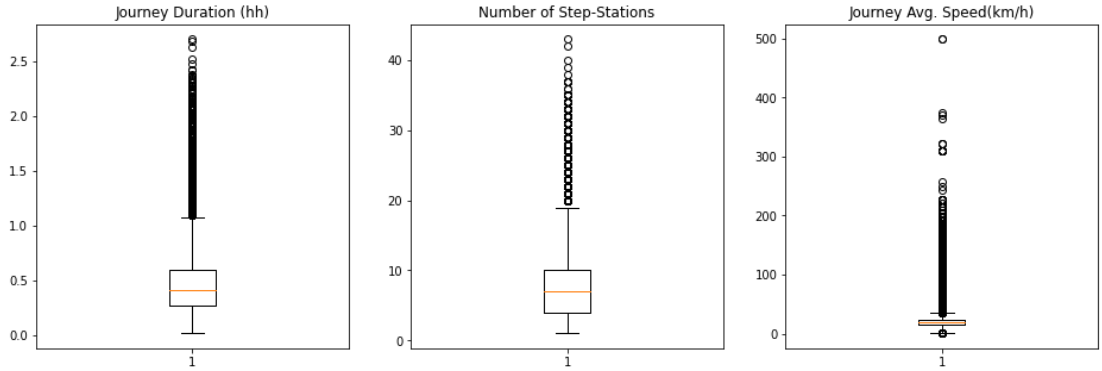

FIGURE 2.2 – Journeys boxplot

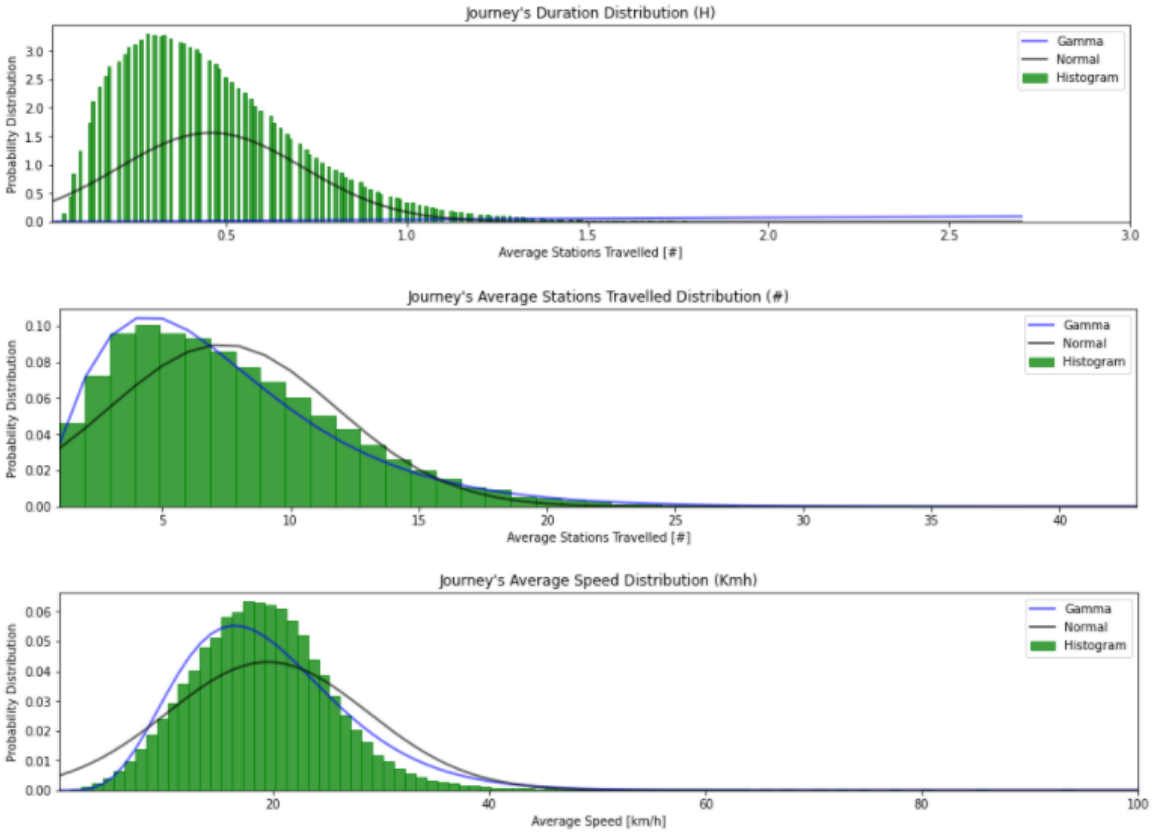

FIGURE 2.3 – Journeys distributions

The second assumption we made is that the average speed is of each journey's segment is equal to the average speed of the entire journey which it is part from. Later, after break our journey's into step journeys, we grouped them by segments averaging the speed-averages and counting the number of passenger for each "To Station".

Below you can find the map plot for the each station connection average speed (fig 2.4) and the number of passengers.

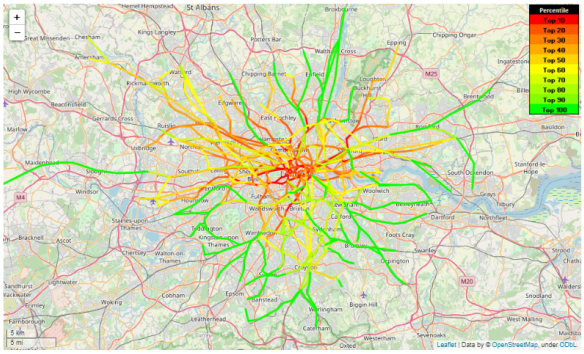

FIGURE 2.4 – Number of passengers plot by percentile.

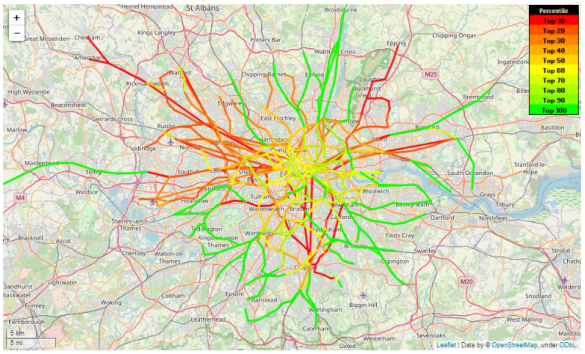

FIGURE 2.5 – Segments average speed plot by percentile.

We have opted by the average speed because the station has differences between each other, and time by itself could not account for this crucial physical factor.

As expected, the central area presents the stations with a more significant concentration of passengers. However, the average speed is intermediate due to the station's proximity, which means more

stop and more passengers what may lead to congestion.

From the maps and charts above, we can infer that London's transport is well balanced and aligned with its passenger demand, presenting no significant variance about the average journey speed (our primary efficiency metric).

The covid map

To create this map and compute the infection risk of an area, we used data from https ://www.covidlive.co.uk/. These data are the number of deaths per thousands inhabitant of a postal code. These postal codes were encoded in MSOA11CD, a kind of geographical code that can be translated into multi-polygons coordinates. After that, we could display them on the map, thanks to the Folium library. It is made on the leaflet API that is a JavaScript framework used to display data on interactive maps. We had to find which station was in which polygon could finally create a covid risk map.

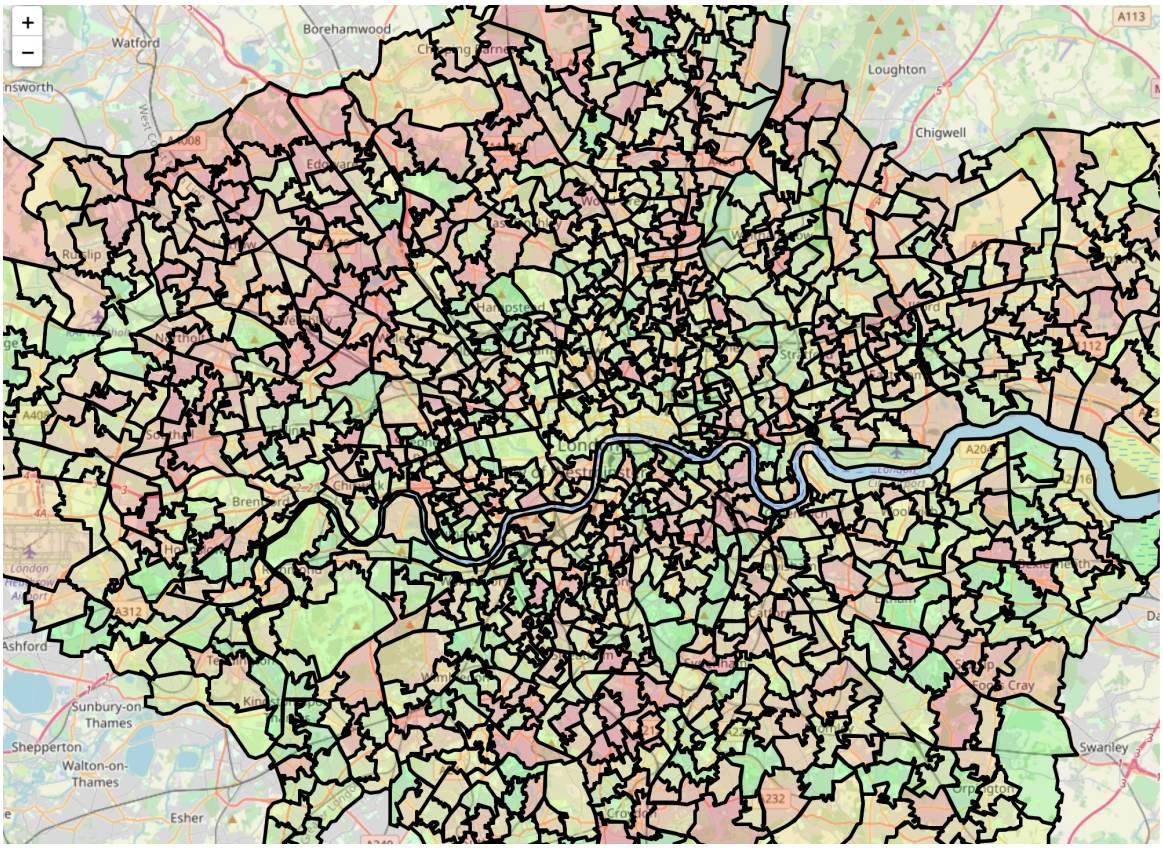

FIGURE 2.6 – Death of covid19/1000 inhabitants until July/2020.

3 The Network

Network Creation

For computing the optimal paths in terms of journey length or safety, we first need a network capturing all necessary information about each direct route, including journey length, journey distance, safety weight, etc. Indeed, the network, in turns, has a large contribution to realizing our goal. Furthermore, the network should hold information concerning different time frames. For example, the journey length or the stress of a route is more likely to be different at other time slices as the journey pattern of the commuters is varying ; the people go to work in the morning and come back home in the afternoon, or many people may travel to the centres on Saturday, but they prefer to stay home on Sunday. (see Figure 3.1) Therefore, the network can be deemed as a multi-weighted network such that each weight represents a cost at a specific time frame (e.g., parts of the day, days in the week, etc.).

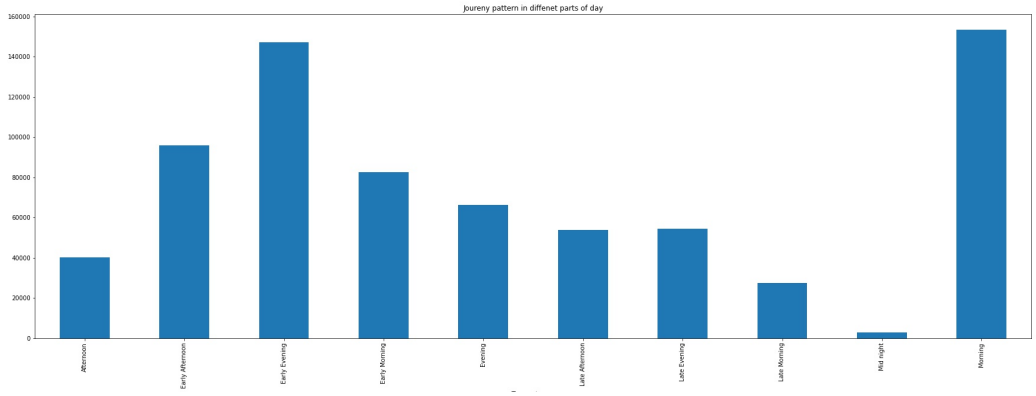

FIGURE 3.1 – Journeys pattern at different parts of day

To build such a network, we used all the data-sets we mentioned earlier. The network is a combination of all data-sets carefully matched to give us a closer look at the journeys and an understanding of how the real transportation network of London may look like. In one simple sentence, we should remark that all we performed was to assign different attributes to London's real railway transportation network according to our journey data-set gathered in January 2009.

The general framework of our approach to create the network is as follow. We first created a journey network based on our journey data-set. The network consists of the nodes —each represents a station—and the arcs —each represents a journey between two endpoints —with several arc attributes such as stress at different time frames (w.r.t. days and parts of the day), journey lengths at different time frames, journey distances and so on. This network allows us to obtain an insight into the journey flows structurally, and then generalize them to the physical network. In other words, we tried to make a more detailed picture of trips from a bigger picture. To create an attributed physical network, we relied on the classical assumption that is "the commuters take the shortest-path for their journeys". With this assumption, we have the following scenario. Before going further, let us specify the stations

at the endpoints by "stations" and the middle stations by "stops". Suppose, in the journey network ; we captured a journey route between two stations u and v as shown in Figure 3.2 (a) such that x commuters took this rout at a specific part of the day like Afternoon—Stress of the rout is x at Afternoon. In this case, we know that there might be many stops in the middle of the path, therefore, if we compute the shortest-path between u and v using the physical network we obtained, then we can approximately find the middle stations (stops) and assign attributes to them accordingly. For example, if x commuters travel from u to v, there must be at least x commuters traversing each stop since each link (arc) between two stops is an arc of the journey path. (Figure 3.2 (b) illustrates an example of attribute generalization.) Using this approach, we can approximately generalize all the attributes (e.g. stress, journey length, etc. at different time frame) of the journey network's journeys to the stops or stations that lie on their routs. Notice that a stop in the middle of a path may be an endpoint of many travels. Some questions may arise : What should we do if a stop is a station about which we have information ? Should we update the values of their attributes or we should ignore update ? Frankly, We decided to consider updating because the journey data-set only represents 5 per cent of the whole trajectory data in a month. Using updating, we could enhance the network with more information. For example, we lack sufficient knowledge about average journey length, stress, and early morning for many journeys. So, by updating the attributes, we could overcome this issue to some extend.

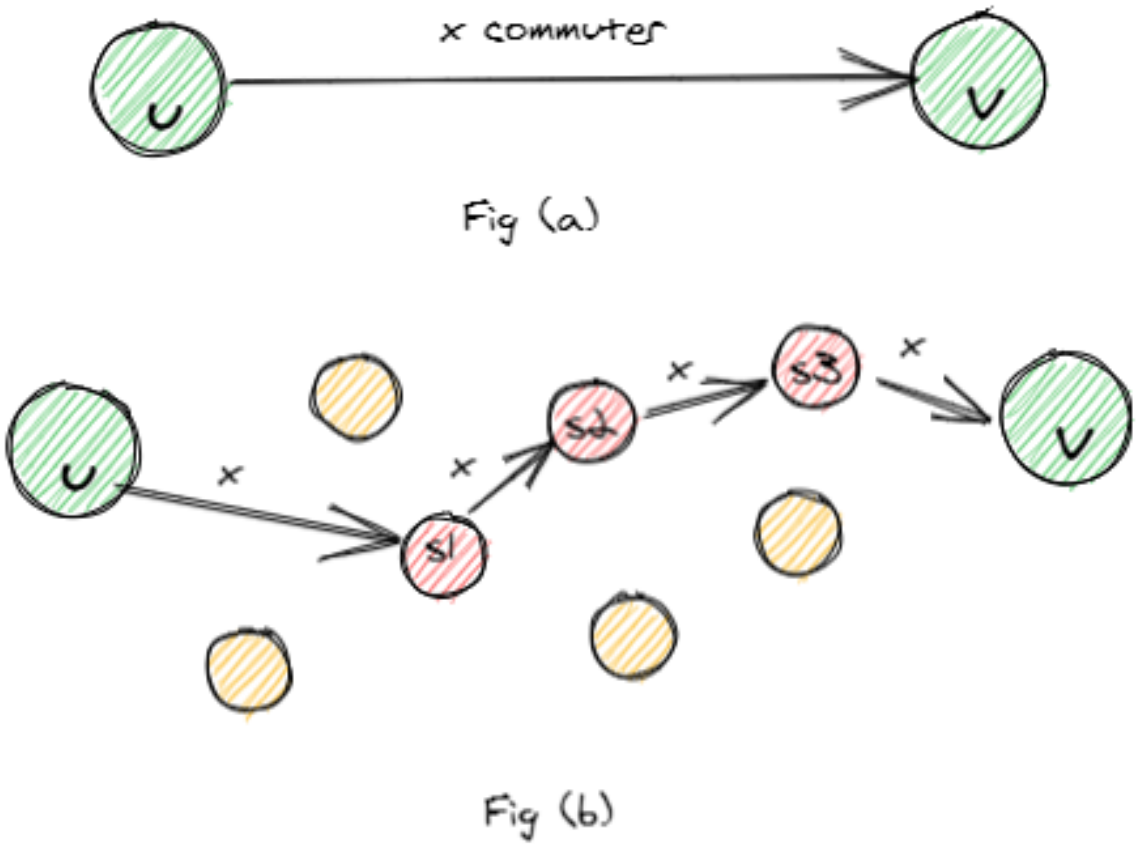

FIGURE 3.2 – An example of path generalization

Mixing journey data and physical network, we could create a transportation network with many attributes. As our goal is to find the safe path, we had to add another attribute to our network's arcs that could help us compute the risk of Covid-19 in a rout. To this ends, we used Covid-19 death data set to assign a Covid-19 weight to each station in the network. Covid-19 death has two features : a geopandas polygon (area) and its associated death per thousand. For each station in our network, we computed a coordinate based on the geaopandas standard. Next, for each of them, we found the closest polygon and assigned death to them as a covid-19 weight accordingly. Although death cannot be an

exact measure of covid risk in a station, in a study in [3], this was highly correlated with the number of covid-19 cases in an area—a 0.96 correlation. The higher death rate in an area, the higher Covid issues in that area, thereby getting infected. This weight is only assigned to the stations, but what if we want to find a covid-weight for the arcs to compute the risk of a route ? Please take the following statements [4] [1] into account.

1-Sitting in the same row, especially adjacent, carried the highest risk in this particular setting. (Stress)

2-Longer journeys, perhaps unsurprisingly, increased the risk, even for those sitting a couple of rows away (Journey Time Length)

These two statements above suggest that the chance of getting infected during commuting using public transportation can be a function of Stress and Journey Time Length. It is indeed a multiplication of tress and Journey Time Length. The fact is that the more we stay in a closed and crowded place, the higher chance we get infected. But this chance is also dependent upon the number of infected people in the secure place. To estimate, we can use the covid-weight of each station because if a station is in a red zone area, there might be more infected commuters going to the station who do not show any symptoms and can potentially transmit the virus. Therefore, given that $S(u, v, t)$ is the stress (Number of passengers) of arc (u, v) at time t, $L(u, v, t)$ is the journey length at time t, and $Covid(v)$ is the death rate at node v, we can approximately formulate the covid-19 weight of each arc at a time t as follow :

$$Contamination - Risk(U, V, t) = S(u, v, t) * Covid(v) \tag{3.1}$$

$$Weight(u, v, t) = L(u, v, t) * Contamination - Risk(U, V, t) \tag{3.2}$$

By optimizing a path concerning $Weight(u, v, t)$, we can detect a safe and relatively short route since the weight function is also a journey length function. To optimize the path or route, we can use any single-source shortest-path algorithm like Dijkstra. Figure 3.3 depicts a diagram of the procedure of creating the network.

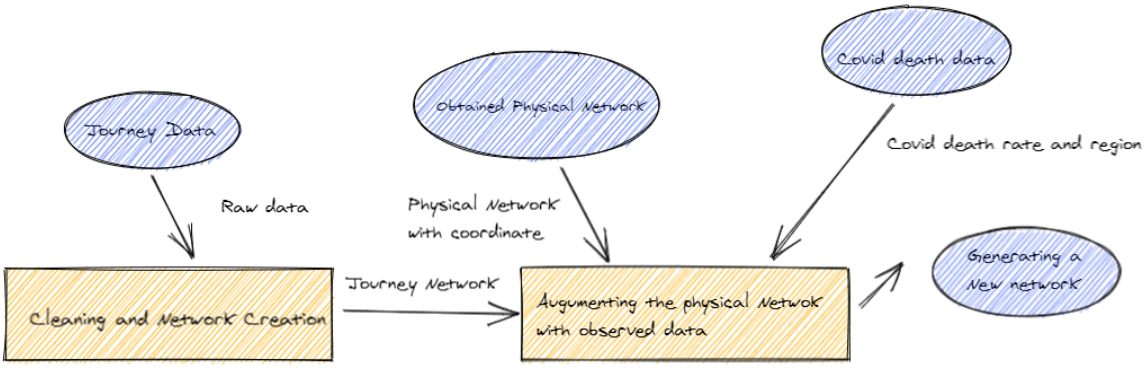

FIGURE 3.3 – A diagram of network creation procedure

Here are two example of the same path at two different hours :

1. They were gathered from a BBC Scientific article.

FIGURE 3.4 – Shortest and safest paths between green-park and st-johns-wood at 8 :00

FIGURE 3.5 – Shortest and safest paths between green-park and st-johns-wood at 21 :00

As you can see, the two paths are not the same. At 8 AM, the shortest and safest ways are the same because no safest path satisfies time and safety. The risk of contamination is 214, which is high but remember this is a directed network so the alternative routes could be too long and end up by crossing many stations and rise the total risk.

However, at 21, the shortest path is different and has a risk of 204 a tiny bit lower than the previous, but here, the shortest path has a chance of 439 which is more than the double. So at this point, it is worth to take the safest course despite the increase in length. The variation of duration is 2 minutes.

At 8 AM, the safe path we found for 21, might be higher. The traffic is a function of time, as well as our network.

But our problem is that this is valid only if one person queries it in a large amount of time. Indeed, the safety concept relies on the assumption that this path won't be too much occupied. If everyone takes this path, the assumption breaks down.

To fix that, we add a weight based on the number of people that make the query. After each path recommendation, the network updates the weights of the crossed arcs. So it can create a new proposal based on the affluence of the queries.

This is the distribution of people in the eight paths our algorithm recommended for 600 people between the same two stations in the afternoon.

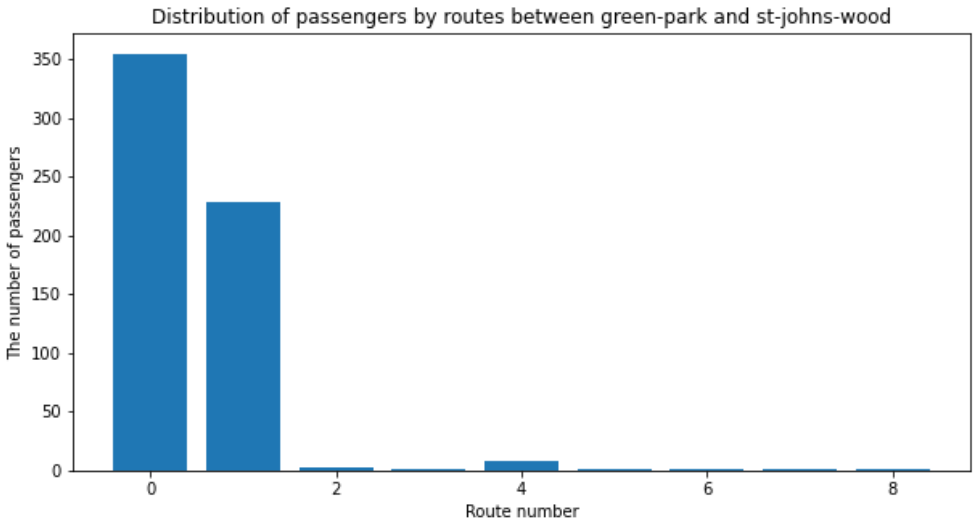

FIGURE 3.6 – Path distributions

As we can see, only two paths are highly recommended. This confirms our previous assumption that these two stations do not have many efficient ways to communicate. And this also explains the high average risk because everyone thus uses the same path.

Now let's visualise it :

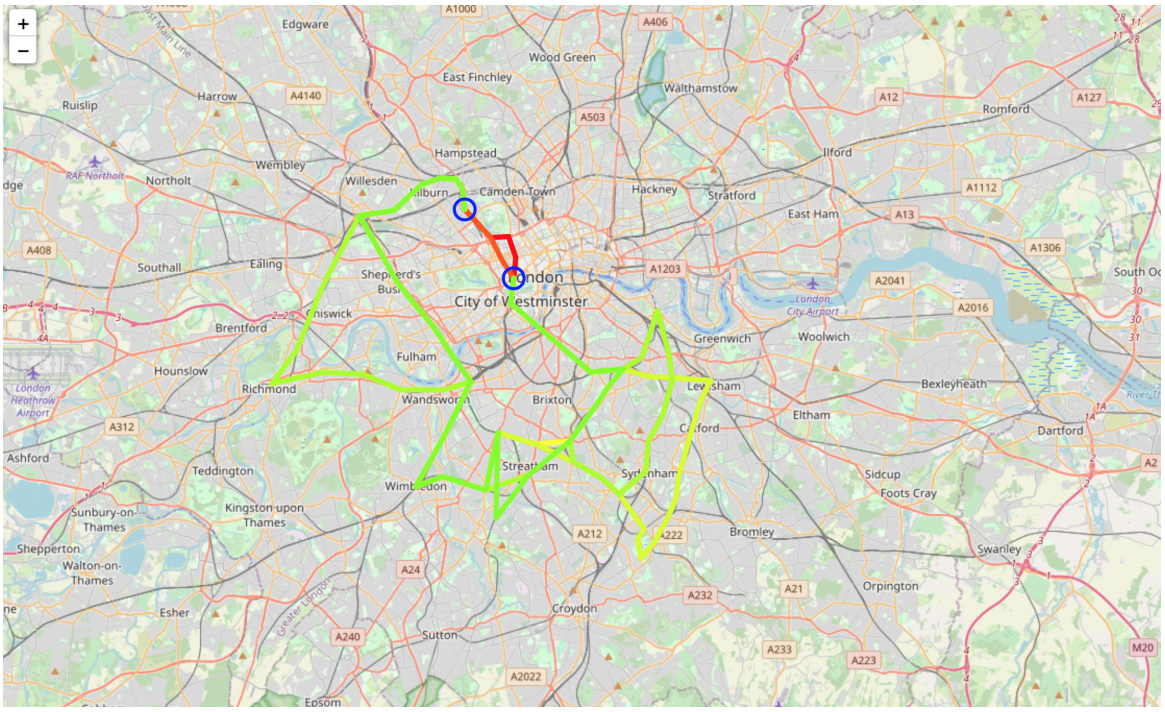

FIGURE 3.7 – Path distributions

As you can see the other paths are indeed much longer than the previous ones and that's the reason why they are so less recommended.

4 Final considerations

Conclusions

In this project, we scrutinized London transportation network. We first began our study by analyzing a journey data-set and then extended it by proposing a shortest-path framework to reducing the risk of Covid-19 infection. As we are experiencing an unprecedented moment due to the current pandemic, coming up with a solution for mitigating the risk of contamination in the public transportation is worth studying. The framework that we proposed here epitomizes the power of data science. It is worth noting that all we accomplished here was based on some separated data-sets that were gathered in different time periods. Indeed, it was an attempt to gather valuable information to cope with the current challenges in the transportation system in London. Due to the fact that the data sets were goatherd in different time periods, we should remark that our results may be flawed. However, if we have more reliable data or even real-time, we could clearly solve the safety issue more efficiently.

5 References

1) TFL Stations Information.
Sources : *https* : *//www.doogal.co.uk/london*$_s$*tations.php*
2) TFL Journey's Information.
Sources : *https* : *//www.kaggle.com/astronasko/transport* $-$ *for* $-$ *london* $-$ *journey* $-$ *information*
3) Relationships of total COVID-19 cases and deaths with ten demographic.
Sources : Dimitar Valev, Relationships of total COVID-19 cases and deaths with ten demographic, economic and social indicators, Sep. 2020.
4) $Covid - 19 : Howtotravelsafelyonthebus, trainandsubway.$
Sources : https ://www.bbc.com/future/article/20200904-covid-19-how-to-travel-safely-on-the-bus-train-and-subway