Data Visualization

Mobility in Paris

Julien Jampsin - Paul Woringer

Dec. 2022

Table of contents

Introduction

Daily commute can take quite a long time when you live in Paris, and one can wonder, what is the fastest transportation method in the city. In this project we offer a visualization of travel times by Uber ride, via public transportation (the subway), and by riding a bicycle. We have taken this data from a multitude of sources, mixing geographical data and travel durations. Our goal is to make a tool as interactive and easy to use as possible, to incite users to explore the data.

The entirety of the code and data is available on github at:
`https://github.com/paulwrg/Data-Vis.git`

Chapter 1

Description of our data set

Our project uses an agglomeration of smaller data sets:

- Geographical data: city plan of Paris
  - Roadways
  - Waterways
  - Green spaces
  - Contours...Iris definition of zones
- Uber travel times,
- RATP travel times,
- Bike travel times.

1.1 Geographical data

The geographical data is used to create the background map of our visualization. The data on roadways, waterways, and green spaces comes from Paris Data [3]. This is why this data covers only the city of Paris, and not the reste of the Ile-de-France. Our goal with them is to provide the user with basic geographical landmarks required for them to find their way around the map. However, displaying the green areas on top of the travel time data caused a lot of visual pollution, as it overlapped with the colored zones. This is why it is only displayed at the launch of the visualization tool, and not after the user begins to interact with it.

The data on Contours...Iris is provided by IGN Geoservices [1]. We selected this partition of space into zones as it what Uber uses in its data set Uber Movement [4].

All of this geographical data is given in .geojson or .json files which contain a list of polygons, given by the geographical coordinates of their vertices. The different zones of Contour...Iris also have an index "MOVEMENT_ID" and name, but the given name is just an address arbitrarily chosen in the zone. This name is therefore not very relevant, as simply displaying the location of the zone on the map gives much more information to the user than a random address.

1.2 Uber travel times

This data is provided on the Uber Movement [4] website in the company's initiative to make most of their data accessible to all, in a search for better road safety and urban planning. The

data available on their website is massive, with over one gigabyte of data per trimester. Our implementation was not able to deal with that much data, so we chose to only consider a small portion of the data: the monthly aggregates of one trimester of data. However, this is still plenty to create meaningful visualizations.

The data is split into two data sets, one that represents week-ends only, and the other week-days. This allows us to switch between the two and visualize them independently, as traffic can vary greatly depending on the day of the week. It also allows for more precise analysis of their differences.

Each entry is identified by a source and a destination given by the Movement_ID that defines the zones in Contours...Iris (see 1.1). This allows us to locate the start and end points of the trip. There is one entry per source/destination pair per month in the trimester. To make the computations faster, we chose to filter and use only one month of data instead of the three, but that does not change the result much.

Each entry contains the mean travel time and the standard deviation of the travel time over the chosen time period (one month in our case). This mean travel time is the main information that we display in our visualization tool.

1.3 RATP travel times

This data is obtained via the Navitia.io API [2]. We submit two types of request to this API:

- A request for a single travel, given by a pair of source and destination coordinates,
- A request for a general map that shows what distance a traveller can cover from a given source point in a given period of time.

The first returns a list of possible itineraries. Each is given with its different steps (e.g. take the metro from here to there, then walk this long...), the number of transfers, and many other pieces of information. The only information we chose to use is the duration of the itinerary and the CO2 emissions it causes.

The second returns a list of isochrone curves that outline the regions a traveller can reach in a given period of time (one curve per period of time given in the request).

Contrarily to the Uber Movement data, it is real-time, a request is sent to the website and the response depends on current travel conditions. Also, the two types of requests do not always return results that are coherent with one another: the shortest itinerary can be much longer/shorter than the time displayed on the isochrone map, without any clear explanation.

1.4 Bike travel times

We did not find any data that was accessible for free online in this regard. We therefore decided to produce our own approximation, estimating that the traveller rides their bike at a constant average speed of 15km/h, which corresponds to the average speed for commuters in urban areas. We also estimate that the length of the journey is the distance as the crow flies between the origin and the destination (since the speed is slightly underestimated, the result is rather coherent with what Google Maps gave in our tests).

Initially we wanted to do this with walking travel times, but this gave very large values which made the other data comparatively very small, and harder to compare between each other. This is why we switched to bicycle ride times.

1.5 CO2 emissions

We wanted to display this information as well, but it was only provided for public transports by the Navitia.io API. For the rest, we chose to create our own estimates, using an average CO2 emission per km for Uber and claiming that bicycle-riding is carbon neutral.

1.6 Overall analysis of our data set

Since we wanted to display a lot of different information in our visuals, we had to get data from a variety of sources, since we could not find a free alternative that offered a centralized data set. This is not really an issue for geographical data, since they are only landmarks used by the user to find their way around the map. However, comparing travel times and CO2 emissions that come from different sources, and might not be calculated in the same way, is not very accurate. Also, the fact that the results given by the Navitia.io API depend on the time of the request makes our visuals highly dependent on the current situation and not easily generalizable. We judged our results during the day to be accurate enough for this project, but it would be very interesting to adapt this tool to use a more coherent database for more accurate results.

Chapter 2

Visualization of the data

As mentioned above, our goal was to visualize both travel times and CO2 emissions. We created two different visuals: a map, and a side bar with charts. In the following sections we will detail both how we implemented our solution and the design choices we made.

2.1 Background map

The main goal of the background map is to give the user an idea of the overall trends for each visualization mode. For the visualization of travel times for each method of transportation, it gives an idea of the overall distance that can be covered in a given period of time. Visualizing differences in Uber travel times over the weekend on the background map dives a general idea of places where traffic is more dense during week-days than on the week-end, and vice-versa.

The overall background map is generated using the geographical data described in 1.1. We used the Lambert Conical projection, which is the convention for maps of France. We estimated that the roadways and waterways were essential for the user to be able to find their way around the map, so we keep them all the time. However, as mentioned before, the green spaces create quite a lot of overlap and visual pollution once we start displaying data for travel times, therefore we discard it once the user begins to interact with our tool. We still kept it at the beginning as it can be pretty useful to use parcs as landmarks to locate certain points in the map.

For both Uber travel times and bike travel times, we display the data zone-by-zone, which allows for an overall consistency in the code and the visuals. However, for travels done by public transportation, the API returns isochrone curves instead of a matrix of durations. These isochrones are not directly compatible with the partition of the map into zones, so we chose to display them as they are, to minimize the distortion of the given data.

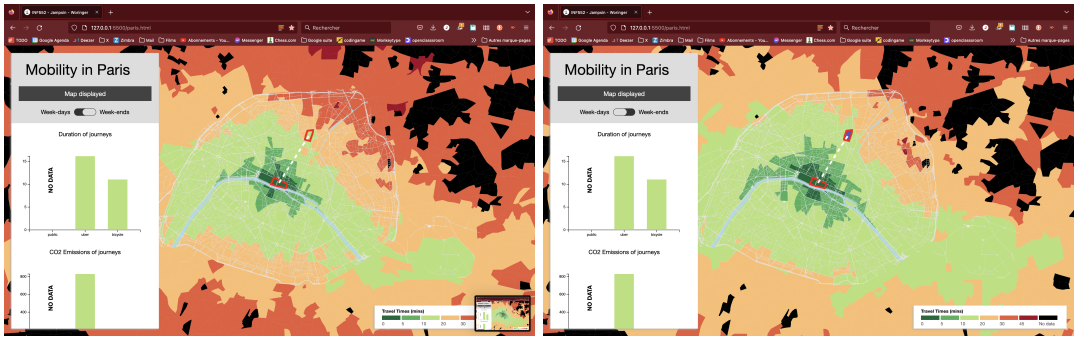

Figure 2.1: Difference between week days and week ends

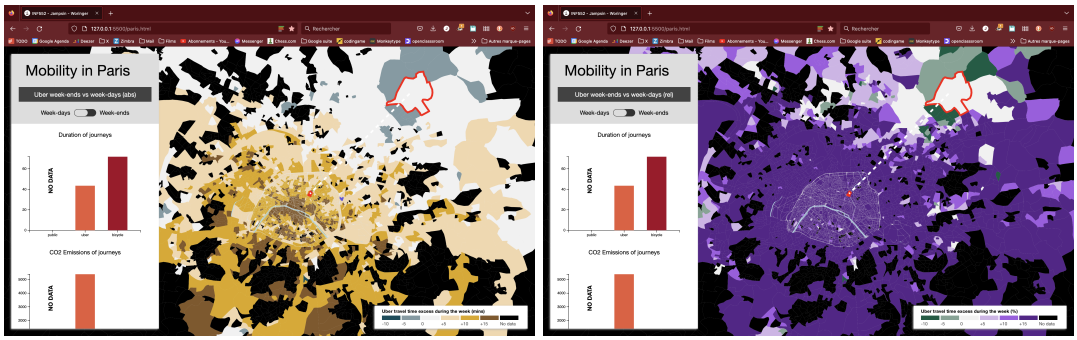

Figure 2.2: Absolute and relative difference between week day and week end

2.1.1 Interacting with the map

To make the tool as interactive as possible, we decided that clicking once on a zone made this zone the target of the trip, and double-clicking selected it to be the source. A simple overlay outlining the source and selected target, with a dashed line joining them makes the selection clear to the user.

The user can use the left panel to select they want to see. They can choose the means of transportation used (either travel times by Uber, public transportation or bicycle). They can also select whether to see data from week-ends or week-days which only has an influence for Uber travel times). Finally, they can choose to display the difference in Uber travel times between the week-ends and the week-days, either as an absolute value (difference in minutes) or as a relative value (percentage of change).

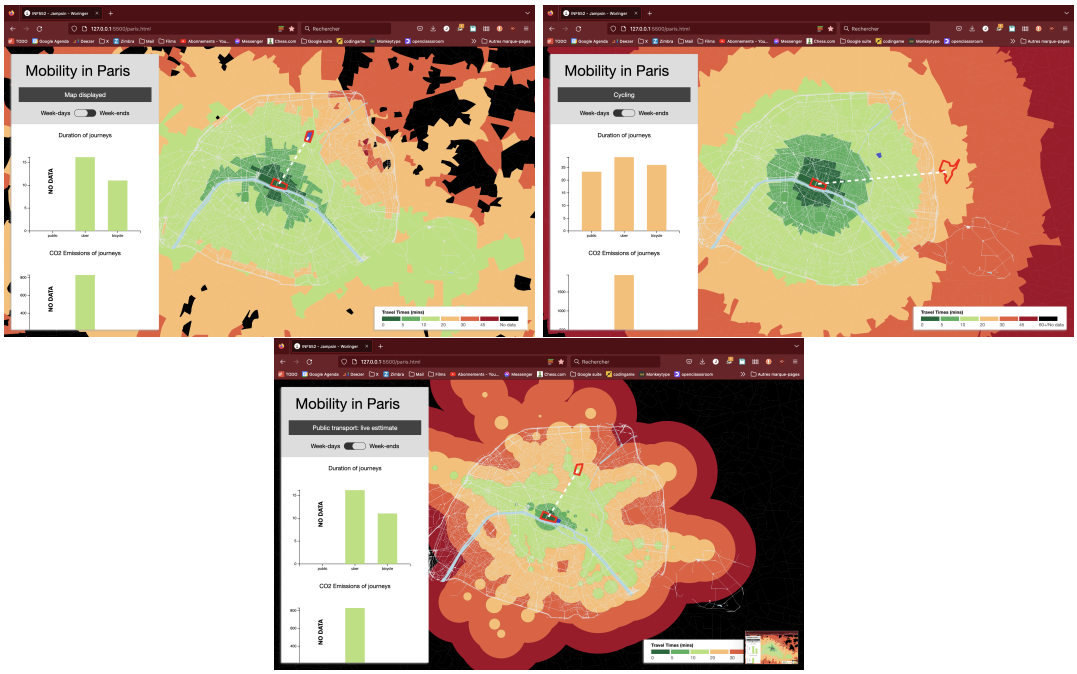

Figure 2.3: Different choices of means of transportation

2.1.2 Color choice

Base scale for travel times We use a simple green-yellow-red color scale to indicate the travel time from the chosen source to each zone. We chose the most natural color scheme, where green indicates short travel times and red are longer durations. We decided to leave zones for

which we have no data in black, signalling the absence of information. We also chose to use the same color scale for each map, to make for easier comparison of the data when we switch transportation method. This is all explained in a legend at the bottom of the visual.

Divergent scale for absolute difference in travel times When we display the difference in travel times between the week-ends and the week-days, we are displaying data that is not directly comparable to the other travel times mentioned above. To prevent the user from making subconscious comparisons, we therefore chose to change the color scheme entirely. We use a divergent scale, which is most adapted to visualizations of differences like this, that can be either positive or negative. Since it is mostly longer to travel during the week, the scale goes shows more categories in the positives than in the negatives. We kept the number of different colors rather small in order to make the map as easy to read as possible, despite this reducing the precision of the data displayed. Again, we used black to indicate the absence of data in order to remain coherent with our previous choices.

Divergent scale for relative difference in travel times Once more this is a completely different type of data displayed, which is why we chose to have yet another color scheme to avoid any possible confusion. Again, a divergent color scheme was naturally most appropriate here. Both color schemes can be seen in 2.2.

2.2 Bar charts in the side-panel

In the side-panel, we chose to display two different charts to give a more precise comparison between each method of transportation for a selected source and destination. This is relevant for users that regularly make the same trip (to work for example) and wish to evaluate their different options. In each case, if we have no available data, we simply display "NO DATA" in place of the bar as can be seen in 2.3.

The first bar chart generated gives directly the travel times using each method of transportations. A linear scale is used, with the origin set to zero, which allows for better comparisons in terms of proportion. We chose to re-use the same color scheme for travel times as what appears on the map, for maximal coherence. Also, repeating the same information via bar height and bar color makes for maximal impact.

Right below, another bar chart displays the CO2 emissions of each transportation method. The height of the bar indicates the quantity of emitted CO2. We chose to include biking, even though it was always zero, to insist on the fact that it was a carbon neutral transportation method. We also thought that placing the two charts right above one another allowed the user to take both informations into account. For example, when deciding on how to commute to work, if the duration of the ride on public transport is slightly longer than by driving (Uber), but the CO2 emissions are considerably lower, they might choose to make an effort and choose the slightly longer but much more eco-friendly option. To help make this connection between the two charts, we use the same color scale (the one used in the base map for durations).

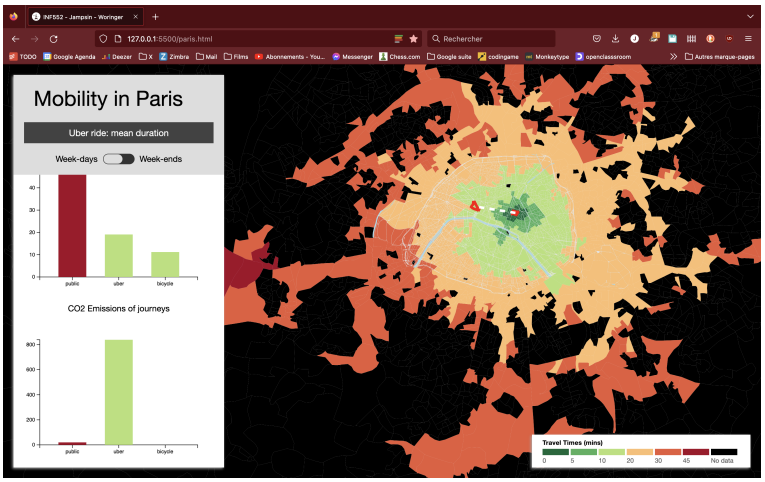

Figure 2.4: Different bar chart with color corresponding to distance

2.3 Implementation

We never really used CSS, JS or HTML and thought this project would be a good opportunity to learn more about it. That's why we only used d3, as it allows more flexibility and is lower level.

As we don't have any web-design background either, so we took inspiration from the Uber website [4]. Then we used elements based on previous labs of the Data Visualisation course to add other sources of data, compare them and display the information as clearly as possible.

Chapter 3

Further improvements

While coding the project, a lot of ideas came to mind, but we did not have time to implement all of them. Here is a list of what can be done to improve and extend on our project.

3.1 Aesthetic

First, we had ideas in terms of pure aesthetic of the webpage that did not change lot to the visualization of the data, but provided a better user experience.

3.1.1 Selected journey

As you can see in the actual state of the project, we link the source and the destination of the selected journeys with a simple line. This is not to much of an issue as the coloration of different zones let us know easily where is the start, but it could add even more clarity if we replaced it with an arrow. We tried to implement it, but the arrow was pointing outside of the zone on a dezoomed view, due to the head being added at the end of the line and not being a part of it.

Furthermore, we have access to the name of the zones, so we could display them in our left panel. For this, we thought of a representation a bit like in Google Maps for instance. Ideally, this could also be used as a seach bar to select the source and destination by entering an address.

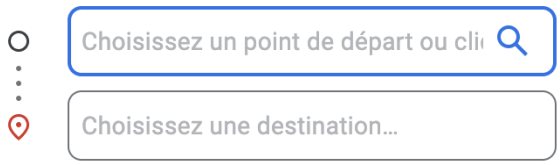

Figure 3.1: Example in Google Maps

3.1.2 User Interface

We would like to improve the way we navigate between our different views. For this, we have severals ideas : first of all, we would like the absolute and relative difference in travel time to be on the same page, and to switch between them as we switch between WeekDays and WeekEnds for instance. We would also like to have a button to come back to the initial state (with green spaces displayed) to allow a better localisation in the map.

3.1.3 Minor tweaks

Finally, we have some remaining ideas that we would like to have, like a loading screen while we load the data, or little purely aesthetic animations, like displaying the color of the zones in increasing order of distance.

3.2 Data

We also have ideas of changes to that would have more impact on the access to the data or in clarity of visualization.

3.2.1 How we display it

In our first iteration of the tiles coloring, we had a continuous linear scale, but we found it hard to read so we switched to the discrete one. However, we liked how it looked and wanted to have the possibility to switch between these two views.

The CO2 equivalent data have the advantage to be clear on the fact that Uber is consuming much more than other methods, but thus, we lack in clarity over the actual values. We could also display the same data with a logarithmic scale for users who want to have a more precise understanding of the data.

3.2.2 What we display

Speaking of precision, we would also like to have the possibility to display the exact value of each of our bars in the plot when you hover the mouse over them.

We also have access to data that we did not consider priority, but that we could have added to the visualization if we had more time. For instance, Uber [4] gives us the standard deviation of the travel time, and we could display it in the bar chart. Navitia.io [2] also gives us the decomposition of our journey (metro, walking, bus, etc). We could also display it in the bar chart by stacking the values.

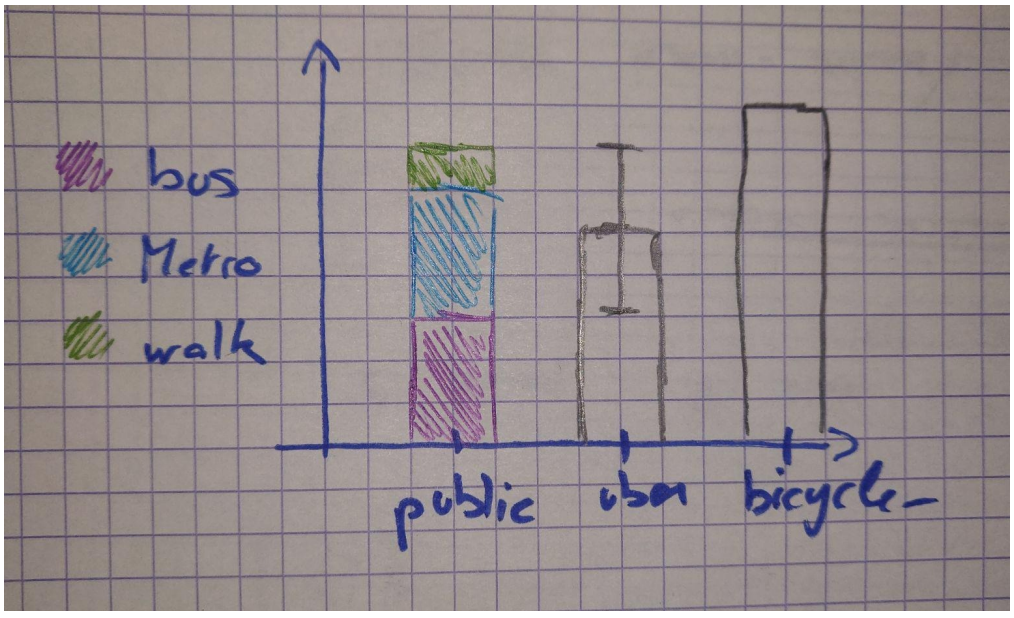

Figure 3.2: Ideas to add values to the bar chart

Conclusion

In this project, we created a very visually-pleasing tool to visualize travel times in Paris with different means of transportation. We also have implemented a view to compare the travel times by Uber on week-days and on the week-ends. We chose different color schemes to indicate the different types of data displayed. Overall, this project required an in-depth understanding of many of the concepts and implementation techniques seen in class, ranging from interactions with APIs and real-time data, to geovisualization.

Our biggest regret is that we were not able to find a unique source that could provide us with all the data we needed, and that therefore our data set somewhat lacks in consistency, but this did not affect the visualization too much.

Bibliography

[1] *Contours Iris - Geoservices IGN*. URL: https://geoservices.ign.fr/contoursiris.

[2] *Navitia.io*. URL: https://navitia.io/.

[3] *Paris Data*. URL: https://opendata.paris.fr/.

[4] *Uber Movement*. URL: https://movement.uber.com/?lang=fr-FR.