Visualisation of pedestrian activity in the city of Melbourne

Chafin Abdou*
Claude Bernard University - Lyon 1

Essohanam Kouyou†
Claude Bernard University - Lyon 1

Quentin Sorel‡
Claude Bernard University - Lyon 1

Abstract— This article is about a project of interactive graphs visualization realised thanks to the visualizations JavaScript library D3.js (Data Driven Documents). It was realised by three students of the Master 2 Data Science (Mathematical cursus) of the Universit Claude Bernard Lyon 1, during the school year 2017/2018. Particularly interested in the pedestrian activity and in how it can be helpful, we decided to seek for data containing observations able to describe pedestrian informations. We then choose the city of Melbourne in Australia to complete this project, with the goal of building a visualization that could help and be used in different fields: urbanism, tourism, even individuals. The project delivery consists in a website describing the pedestrian activity of Melbourne in November 2017. Given a day and an hour, the website displays the number of people at different locations in the city-centre. The site also allows you to view a various diagrams as well as an icon showing the weather of the selected day.

Index Terms: Visualization - Interactive graphs - Transport - Melbourne - BubbleMap

1 Introduction

In a world where sustainable development assumes an ever more pertinent position in the global imaginary, walking, travel by foot, has become revered as a viable alternative to the use of automobiles. More than a mere mode of transport, walking has revealed itself to be a veritable vector of life and major contributor to urban dynamism; for the past 10 years pedestrians have been implicated at the very heart of urban practices [4]. It is therefore logical to ascertain that numerous issues linked to urbanism and security in relation to the pedestrian body have appeared. In order to respond to this issue, it seems first crucial to study the behavior of pedestrians at the heart of the urban landscape [9].

To create our visualization, we needed data from a city. Our choice fell on the city of Melbourne in Australia because it provides free data.

With the aim to better understand pedestrian activity, the city of Melbourne has put into place an automatic system designed to count and record pedestrian numbers. In order to achieve this, pedestrian counting devices have been placed at certain populous locations within the city. The number of counting devices present in the city has risen from 18 in May 2009 to 44 at present. These pedestrian counting devices collect information which permits the temporal and spatial examination of pedestrian habits. The information collected is used by the city notably in the imposition of certain commercial strategies as well as in urban planning. More than that, such data can be useful in other various and numerous areas, for instance for transport companies, to point where and when people are situated. And that is why the study of pedestrian behavior is important.

We intended to utilize this data in order to create a visualization of pedestrian activity in Melbourne. In order to create such a visualization, we equally intend to use meteorological data in order to explicate the correlation between the number of citizens at each location and the external factors influencing their presence there.

The final visualization must therefore provide a clear and adequate representation of the data studied; we thus intend to investigate different means through which to present the data. Our project will investigate numerous visual tools including maps, which will be further explicated with advanced and personal ocular techniques in order to obtain an efficient and demonstrative visualization of the data studied.

*e-mail: chafin.abdou@gmail.com
†e-mail: essokouyou@gmail.com
‡e-mail: quentinsorel@outlook.com

This article is a summary of the work done, including steps of creation, as well as an inventory of related works.

2 Related Work

Visualization of spatial and temporal data is no longer to be introduced. Indeed several works have already been done in this area, we can think of Google Maps for example which is a free online mapping service. The service was created by Google and was launched in 2004 in the United States and Canada before arriving in Europe in 2005 in Great Britain. It is a service available on PC, tablet and smartphone that allows, from the scale of a country, to zoom up to the scale of a street. A virtual navigation service (Google Street View) showing the details of certain streets is also accessible via a gateway. Two types of views are available in Google Maps: a classic plan view, with street names, neighborhoods, cities, and a satellite view, which now covers the entire world.

We also thought to D3.js (or D3 for Data-Driven Documents), which is "a JavaScript library for manipulating documents based on data". To put it more simply, D3.js is a data visualization library. It was developed by Mike Bostock with the idea of bridging the gap between static display of data, and interactive and animated data visualizations. It allows the display of digital data in a graphic and dynamic form. This is an important tool for conforming to W3C standards that uses common SVG, JavaScript and CSS technologies for data visualization. It has been used for spatio-temporal visualization design.

These two technologies (Google Maps and D3.js) complement each other very well for creating quite nice visualizations, and we choose to use them for our work.

The following parts will now deal with the few already existing works and which we used for the establishing of our visualization

2.1 OOMCreative Melbourn Visualisation

We did not start from nothing, we based on an already existing visualization of the city of Melbourne designed by OOMCreative.

The city of Melbourne offers to visualize the data that we study in free access, a map visualization realized by **OOMCreative** (*figure 1*) is indeed available. It presents a global visualization of pedestrian activity by describing hour by hour the number of individuals detected at the locations where the sensors are placed. It also compares the current inflow with an average of the inflows collected on the same day at the same time. Thus the visualisation gives a detailed graphical representation of the information collected in relation to historical trends (*figure 2*).

It is this work that will serve as a basis for our project. Simple to manipulate and understand, this visualization only provides a simple approach to describing data. Our job will be to flesh out this work, to make it more comprehensive, highlighting other statistics. We will

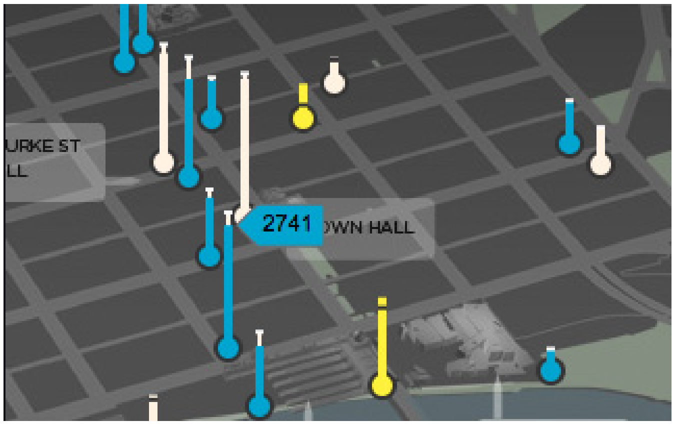

Fig. 1. Pedestrian activity visualisation in Melbourne by OOMCreation

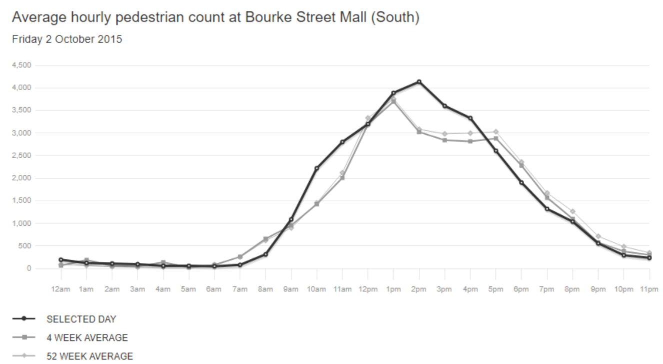

Fig. 2. Comparison of counts from a specific day to a 4 weeks average and a one year average

also consider external factors, such as weather or the holding of events in certain parts of the city and at a specific time, that will explain the values of some of our data. Our visualization should therefore be able to represent and reveal the possible correlations between this type of data and our initial dataset counting the number of people.

2.2 Weather Website : Wunderground

Data that could be used to embellish the initial model are meteorological data[8]. Pedestrians are totally exposed to the climate, and we tought that adding the weather in the visualization would allow users to visualize well the aftermaths of bad weather on the pedestrian activity. Meteorological data can be represented in many ways. We had to make a choice about how the weather will be displayed in our work. The first idea was not to overload the rendering : first for the map remains easily understandable, and in a second way to be able to add some different and important factor or tools to the map without having to think about how will we add that so that the map stay clear and clean. Meteorological data can be represented as little icons (*figure 3*), therefore we had the idea to represent them in the header of the visualisation map. Although simple, this idea was for us the most affordable and the most interesting.

The website *wunderground.com* is a site that provide the weather in a city, provided the date and the hour.

2.3 Snazzy Maps

As the owners of the website *https://snazzymaps.com* said :

"We want to help developers and designers match the Google Map on their website with their websites theme."

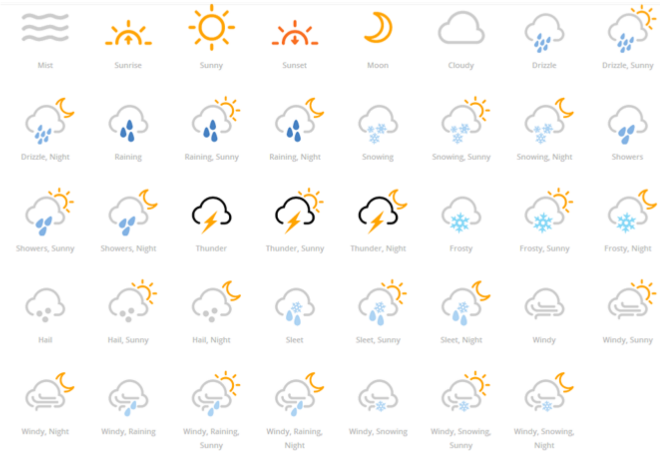

Fig. 3. Examples of weather icons

Snazzy Maps [6] is a website which provides a community built around creating and sharing great looking styles for Google Maps. It provides a repository of different styles for Google Maps aimed towards web designers and developers. We use for the background of our map the style : **ultra-light-with-labels** reachable here [7]. This style use light and gray colors to improve Google Maps design, it was chosen for its simplicity and elegance.

3 Project Description

Although we are based on a model aldready satisfacting, some points to improve visually remain. In particular the representation of the different counts for each sensor. Our idea is to represent them with circles, larger or smaller, depending on the number of pedestrians captured by the sensor. This would clearly visualize where pedestrians are most present.

Also, in our visualization sketch we decided to present the comparison data with the averages over several days in a block apart with objective not to overload our map. In the same way an overview of the general statistics of our data will be presented in a block under our map visualization. This block will also allow us to focus on more targeted statistics such as pedestrian accounts by date (time, day, month), or to view accurate data per sensor. To create the visualization background we use Snazzy Maps (detailled in the related works section).

Currently, there are three main components to Snazzy Maps. First, users can explore and use map styles created by others - all styles are licensed under creative commons and are completely free to use. Second, users can create their own style using our map styling tool. Third, users can use these styles in our dedicated map builder or easily apply them in WordPress with our Snazzy Maps WordPress Plugin.All styles are licensed under creative commons and are completely free to use.

3.1 Technical aspects

3.1.1 Datasets

We must have some real datasets to make our visualization and hope to show off behaviors with this visualisation. As we want to show Melbourne pedestrian activity we use the sensor datasets proposed on their website. the dataset is composed of two datasets :

- **Sensor locations** : this dataset provide all the sensor placed in the city with their places coordinates. Every sensor has an unique id

- **Pedestrian volume** : this dataset provide given the number of people counted every hour beside a sensor.

Sadly, Melbourne website doesn't provide Melbourn weather. So in order to add our weather icons to the map, we wrote python script to partially fill the weather dataset. That weather datset was filled in two steps which are:

- get the weather of Melbourne in 2017 per day by scrapping a weather website named *www.wunderground.com*.
- manually add links to the icon that corresponds to a weather.

3.1.2 Hosting of the site

The project directory is on github [2] as specified in the recommandations of projects. GitHub provide websites as GitHub Pages [?]. The site is thus, a GitHub page, avalaible on the *https://essohanamkouyou.github.io/dataviz_melbournCity/*. The project folder and the data can be found on the link *https://github.com/EssohanamKouyou/dataviz_melbournCity*.

3.2 Main functions and attributes

The project is built with web languages like HTML, CSS, Javascript, d3 [1][5] especially the fourth version and Google Maps API. The main difficulty of this project was the fact that Javascript doesn't guarantee the linear execution of the code. This section presents the principal D3 functions and the ones we wrote. For most of them we add the code definition. The principal components of our visualization are:

1. **D3 functions** : D3 is a very versatile library. It can help display simple lines as well as very complicated interactives visualisations. We use many D3 functions in order to create our graphs. Among them we can cite:

   - d3.select(): used to select html parts in the Dom. It is very useful for creating new elements. It is the most used function in d3
   - d3.timeParse : to define a format in which parse date
   - d3.scaleLinear() : scale continious values
   - Mouse events: used to add events on components. Very useful for interactive projects
   - d3.nest: most use to group data by a key value. We use it to have the average per sensor.

   ```
   var averages =
   d3.nest()
   .key(function(d) {
        return d.key;
       })
   .rollup(function(d){
           return d3.mean(d, function(g) {
                   return g.value;
           });
   })
   .entries(donnees);
   ```

2. **Map**: represents the map printed on the visualisation

   ```
   var map = new google
           .maps
           .Map(d3.select("#test").node(),{
             zoom: 14,
             center: {lat, lng},
             styles: style
   });
   ```

3. **Overlay[3]**: Overlays are objects on the map that are bound to latitude/longitude coordinates.

   Google Maps has several types of overlays:

   Marker - Single locations on a map. Markers can also display custom icon images Polyline - Series of straight lines on a map Polygon - Series of straight lines on a map, and the shape is "closed" Circle and Rectangle Info Windows - Displays content within a popup balloon on top of a map Custom overlays

4. **Marker**: represents all the conponents shown on the map The Marker constructor creates a marker. (Note that the position property must be set for the marker to display).

   Add the marker to the map by using the setMap() method:

   ```
   var marker =
   new google.maps.Marker(position);
   marker.setMap(map);
   ```

   Add animation to the marker :

   ```
   var marker = new google.maps.Marker({
     position:myCenter,
     animation:google.maps.Animation.BOUNCE
     });
   marker.setMap(map);
   ```

   Example of creation of the circles

   ```
   var cir = new google.maps.Circle({
     center:amsterdam,
     radius:20000,
     strokeColor:"#0000FF",
     strokeOpacity:0.8,
     strokeWeight:2,
     fillColor:"#0000FF",
     fillOpacity:0.4
   });
   ```

5. **LatLng**: it is the coordinates of the sensors. We get these coordinates from the sensor_locations.csv file:

   ```
   d3.csv("Pedestrian_sensor_locations.csv",
   function(data) {
     for(k in data){
     var latLng = new google.maps.LatLng
     (data[k].Latitude,data[k].Longitude)
     }
   }
   ```

6. **transfoDate(date)**: transform the data provided in the weather datset in the right format of Date. This function looks like:

   ```
    function transfoDate(date){
   p=date.split("/");
   a="";
   b="";
   if(p[0].length==2){a="0"}
   if(p[1].length==1){b="0"}
   return a+p[0].substring(1)+"/"+b+p[1]+"/"+p[2]
   }
   ```

7. **test2()**: print the circle conponent on the map once the website is launched

8. **Affichage2(hour, date)** : refresh the component on the map by deleting olders and creating new ones which are the ones of the hour and the date provided

9. **afficherbarchart10()** : print bar chart for the sum of volume counted per day in a month.

   ```
   var x = d3.scaleBand()
             .rangeRound([0, width])
             .padding(0.1)
   var y = d3.scaleLinear()
             .rangeRound([height, 0]);
   ```

```
var svg = d3.select("#barchart10")
             .append("svg")
            .attr("width", width
                  + margin.left
                  + margin.right)
            .attr("height", height
                  + margin.top
                  + margin.bottom)
    .append("g")
            .attr("transform",
            "translate(" + margin.left
            + "," + margin.top + ")");
```

10. **statjour2()** : print bar chart for the average volume counted on a date. It's definition looks like :

```
var x = d3.scaleBand()
          .rangeRound([0, width])
          .padding(0.1)
var y = d3.scaleLinear()
          .rangeRound([height, 0]);

 x.domain(averages.map(function(d) {
         return d.key; }));

y.domain([0, d3.max(averages, function(d) {
  return d.value; })]);

svg.append("g")
  .attr("class", "axis axis--x")
  .attr("transform", "translate(0," + height + ")")
  .call(d3.axisBottom(x))
  .selectAll("text")
  .attr("transform", function(d) {
        return "rotate(-65)" })
  .attr("dx", "-3.2em")
  .attr("dy", ".7em");

svg.append("g")
    .attr("class", "axis axis--y")
    .call(d3.axisLeft(y))
    .append("text")
    .attr("transform", "rotate(-90)")
    .attr("y", 6)
    .attr("dy", "0.71em")
    .attr("text-anchor", "end")
    .text("Nombre de personnes capts");


svg.selectAll(".bar")
    .data(averages)
    .enter().append("rect")
    .attr("class", "bar")
    .attr("x", function(d) {
              return x(d.key); })
    .attr("y", function(d) {
             return y(d.value); })
    .attr("width", x.bandwidth())
    .attr("height", function(d) {
            return height - y(d.value); });
}
```

11. **afficherbarchart30()** : given a sensor it prints in light color the volumes of the date selected and in darker color the average volume of the day beside that sensor.

12. **Boulbi()** : which is called to refresh the weather icon when the date of visualization is changed.

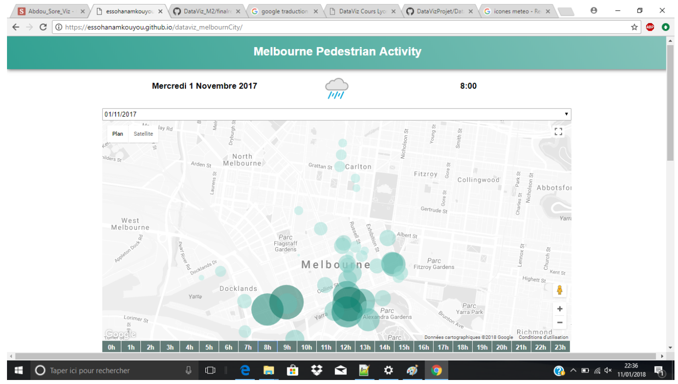

Fig. 4. Map part of the visualization

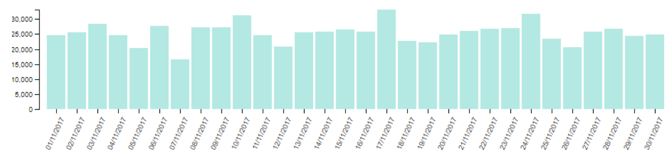

Fig. 5. Pedestrian Activity month statistics per day

```
function Boulbi(curd,curh){
   d3.select("#date")
   .text(jours[format(curd)
   .getDay()]+" "+format(curd)
   .getDate()+" "
   +mois[format(curd).getMonth()]+" 2017")

 d3.select("#heureact").text(curh+":00")

 d3.csv("meteo.csv", function(rob) {
cur=0;
while(curd !== transfoDate(rob[cur].date)){
  cur++;
}
d3.select("#meteo").select("svg").remove()
d3.select("#meteo").append("svg")
 .append("image")
                 .attr("xlink:href", rob[cur].links);
});
 }
```

3.3 Interactivity

Interactivity is one of the main points of this project, it allows the user to select the period and to define the localisation on which he wants to pay attention or wants more information. In this project the most users interaction are :

- zoom on the map for more details
- select a sensor for statistics
- select a day
- select an hour

3.4 Visualisation

This section presents in pictures what our website/visualization looks like and explain how users can interact with it. The images show the different options of our visualization.

The website is very simple to use. At the launch the website, the first you can see is the title : "Melbourne Pedestrian Activity". Just under the title section there is a white stripe which prints the date

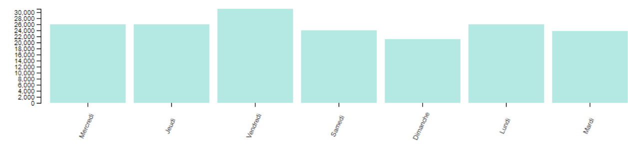

Fig. 6. Pedestrian Activity week statistics for the whole month

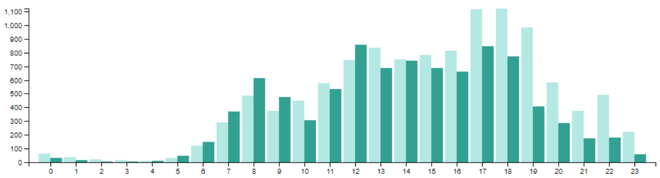

Fig. 7. Statistics of the activity beside a sensor : comparison between the actual active day and the mean on the month

selected, the hour selected and the wheater icon which represent the weather corresponding to the date printed. After the title comes the map with circles components representing the activity on the 1st, November 2017 at 10AM. The circles represent the number of person presents beside a sensor.

On the top of the map, a date picker is provided so that the user can interact with the visualization by navigating in days. At the end of the map , you can see a list of button , also provided to do an hour navigation on the map.

At the end of the map you can see two bart charts: the first represents the sum of volume counted per day in a month and the second reprents the average volume counted on the selected day .

The possibility to zoom on the circles is provided, the user just need to double click on the map. When a mouse is on a sensor circle, there is a little animation on it to invite the user to click it and see some statistcs relative to the clicked sensor. Once a circle is clicked, a third bar chart is created. That bar chart shoes in light color the volumes of the date selected and in darker color the average volume. (here of the whole month, ideally, if all data were represented, it would be the average affluency at this location at a specific hour from 15 days before the selected to 15 days after the selected day. So that you can have and make a great comparison in time). The *figure 7* has also been added here for more sensor details.

Also, the intensity of the bubbles color indicates if there is less (lighter) or more (darker) people than the habit. It is a great indicator to see if there are differences or if particular events are happening.

The *figure 4*, *figure 5* and *figure 6* shows the interface at launch of the website.

We see a Google Map centered on the city of Melbourne with possibility of zooming on the page. This same map shows details of the main places of Melbourne Hoddle Grid, which allows to see the most frequented places in that area and so on. we decided to make the zoom an unconditionnal of our visualization, especially to see precisely the location of the sensors. Certainly the representation in circles makes it possible to see the differences between the volumes measured at the level of the sensors but they do not give figures to know approximately what is the average proportion for example during a week. That's why we chose to add statistical graphs (here Bart Chart).

4 Discussion and Evaluation

Due to the fact that the library D3 provides maps visualisation, some people can ask why we choose to use Google Maps instead. Thus we made a little comparison of the two methods.

4.1 Advantages of Google Charts over D3.js:

- Google Charts cant create many variety of graphs when compared to d3.js, but what it can, it does it perfectly.
- It has a very easy-to-use API, compared to d3.js where you have to build every axes, line, and bar from scratch.
- Google Charts can be picked up by a non-tech person as well but the same can not be said for d3.js, where at least some basic knowledge of JavaScript is required to start making graphs.

4.2 Advantages of D3.js over Google Charts:

- One of the major points in favor of D3.js is that you can make your visualizations the way you want. Quite everything is reachable thanks with D3. This is not possible in Google Charts where you can only create some frequently used charts like bar, line, column, etc.
- While some of the Google Charts do not support large amount of data and have some limit, d3.js can be worked with data, ranging from bits to gigabits and it will still go strong.

Both of these technologies are interesting to use. Google Charts are best for those who want basic and simple graphs without wasting much time and D3.js graphs are for those who want to create some complex graphs or general graphs with customizations. The learning process for D3.js can be slow, but, once you get used to it, D3.js is fun and it becomes very pleasant to play with. For this project we choose to combine them because they perfectly complete each other.

4.3 Some tracks of investigations

We are well aware that our site as well as functional remains to be completed. There is still much more to add to make it a good visualization. One of the first thing that can be done is to release data to represent a much longer period. We decided for the need of the visualization to cut data and only based our work for a month (more than 150Mo to represent the full period).

Many other tracks can be interesting and are likely to bring a consequent visual contribution to our work. Whether because of technical problems or lack of time, many of the ideas mentioned could not be investigated or realized :

1. It would be essential in the future that this visualization brings more relevant statistical information. In particular to detect trends that are not obvious (some days rather than others, at a certain time, in some places, combinations of factors ...).
2. The weather must also have a full place in the statistics in order to concretely analyze its influence
3. An addition of transitions and aesthetic improvements can be considered to make the work more enjoyable for the eye.
4. The idea of the weather was interesting, but there may be other factors to add, such as the representation of events and to get some correlation between the presence at a certain place and the holding of an event.
5. A more difficult idea to put in place and which requires more documentation would be to be able to represent trajectories that pedestrians are likely to take and to analyze the links that can have two sensors (textit example: if I pass to a sensor , will I pass to such another sensor?)

5 CONCLUSION

In this article, we have detailed the description of the site representing "Melbourne Pedestrian Activity". We identified all recovery, cleaning and data processing steps used in the design of the site. Likewise, we cited all the technologies we used in this project.

As can be seen on the visualization, the weather can affect the volume of people present at a place as well as time (time). As expected there are fewer people late at night and many people at the hours when people go to work. This is perfectly normal. We also note on the map that there is a large concentration of the population at certain locations such as the train station in the center of Melbourne, at the stadium on match days. This visualization is to be completed with data of factors that can still really influence the pedestrian activity such as events in the city, and many others, in order to visualize more abstract causalities that can affect pedestrian comportments.

To conclude, it would be interesting to find some data to continue the visualisation that turns out to be a very interesting topic for the municipality, public transport companies, tourist offices and many others.

Also, this project is an example that shows how Google Maps and D3.js complete each other in order to create interesting visualizations.

REFERENCES

[1] D3 data driven documents. https://d3js.org/.
[2] Github repositories. https://github.com.
[3] Google api javascript documentation. https://developers.google.com/maps/documentation/javascript/.
[4] Melbourne pedestrian couting. http://www.pedestrian.melbourne.vic.gov.au/.
[5] Mike bostock. https://bost.ocks.org/mike/.
[6] Snazzy maps. https://snazzymaps.com/.
[7] Snazzy maps template. https://snazzymaps.com/style/151/ultra-light-with-labels.
[8] Wunderground. https://www.wunderground.com/history/.
[9] C. Ferrier. Le pieton, la voiture et la ville. *Metropoliques*, Apr. 2013.