Data Visualisation - Les Zinzins

Pierre Hellich, Gaston Wolfart and Kelyan Hangard

June 3, 2023

1 Idea progression

Initially, our goal was to create a dashboard where visitors to the website could view comprehensive information about Airbnb listings and filter them based on regions or other preferences. However, we realized that Airbnb itself is the best platform for browsing Airbnb offers. Therefore, we needed to find another approach that would truly add value. Inspired by "How to drink wine" video, we decided to incorporate a storytelling dimension.

We achieved this by guiding visitors through each section, focusing on different aspects of our dataset. After a brief introduction, visitors will notice that our entire site is designed around the Airbnb theme, with each "floor" of the "house" representing a different section. This concept brings a playful and cartoonish design to the visualizations, which visitors can interact with. We decided not to proceed with an idea from our milestone 2 report, which involved placing the start button where the door is and making the house's facade disappear to reveal a cross-section. This approach didn't provide an intuitive way to begin the experience. Instead, we added a door sound effect to the initial start button, creating the sense of visiting a house. This sound effect adds depth and enhances the overall experience.

One additional value our site offers compared to Airbnb is the inclusion of an extra "floor" with additional data, such as weather information. Visitors often look for more than just a house; they also consider factors like sunshine, warmth, or even snow. This additional data helps users make informed choices. With more time available, we would have liked to incorporate other data, such as cost of living, nightlife activity, and notable landmarks of each city.

To make visitors feel more at ease while browsing the site, we replaced the original section titles with more amusing and intriguing ones. The goal was to create a desire in visitors to explore the graphs, which are designed to be simple and quickly understood. For example, "Price overview" became "Where does your wallet want you to go?", "Distance overview" is now "Stay Close to the Heart", "Capacity overview" transformed into "One Size Doesn't Fit All", and "Weather overview" became "Sunshine Tracker". We also developed a consistent visual identity for the site, ensuring text homogeneity. The color palette used is attached in the appendix.

Given more time, we could have implemented several additional ideas. For instance, if a visitor didn't feel like clicking to another "floor," we would have added the title of the next section on the button, encouraging users to explore further. Another idea was to find a dataset that encompassed not only major European cities but a broader range of locations. Alternatively, we could have explored additional aspects of our current dataset that we didn't prioritize, such as cleanliness ratings or the proportion of superhosts.

The progression of ideas for each visualization will be discussed in subsequent sections of the report.

2 Challenges and design decisions

One key part of our website is the cut view of the interior of the house that serves as a navigation bar. This was challenging as it is not a classic way of doing navigation and we had to draw custom

polygons, using SVG to separate each area of the room. The idea is that all the rooms are turn off initialy (low opacity) and whenever the user hovers the mouse over a room, it lights up only a little bit, as if we opened a door towards a dark room and only a little bit of light is going in. Then if the user clicks on the house, the room lights up fully, showing that we are currently in this room. We can see this behavior in the Figure 1. The use is located in the section "Location" and hovers his mouse over next room, revealing the title of the next section: "Capacity".

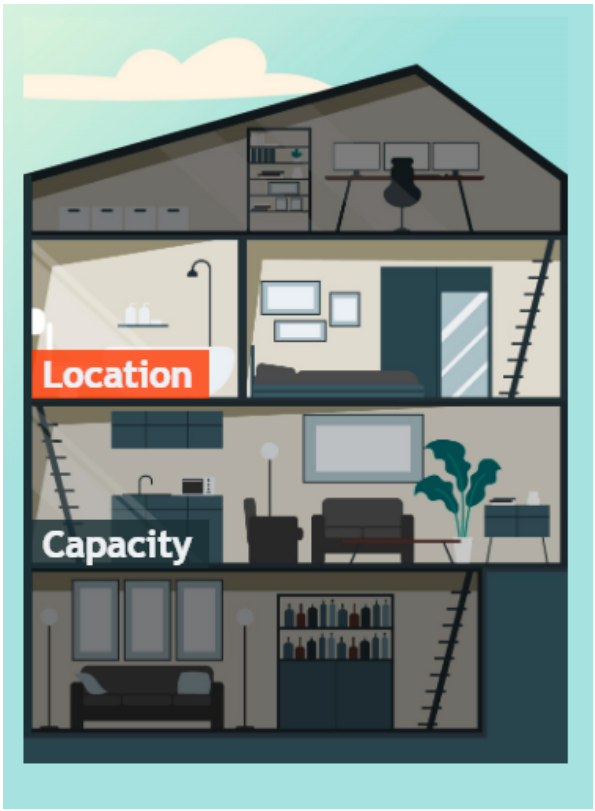

Figure 1: House navigation bar. Location is selected and we hover over Capacity

One challenge was to make this navigation bar follow what was showed on the right side of the screen (where the visualization is showed) if the user scrolled instead of using the house navbar.

To let the user know about the full potential of the visualizations, we've included for each section a tooltip, signified by a question mark icon next to the titles, that informs the user on the features of each visualization.

About the spider chart, there were no particular difficulties in implementing it. The only design change was to superimpose the two distances, on the same graph and at the same scale. Site visitors can, for example, observe that for Paris, the bird's-eye distance to the center is rather high on average. However, the very dense metro network means that the distance to the metro is very small.

Regarding the violin plot, the implementation was quite complicated. Initially, the goal was to implement the Kernel Density Operator (KDE) function as well as the box plot ourselves, in order to create the visualization entirely. However, we finally used D3 once again. The values of outliers from some cities had an undue impact on the shape of the violin, stretching it to sometimes approach zero values. This was resolved by removing outliers above 4000 euros. The visualization, while still crushed because it was very dispersed along the y-axis, had the violins truncated at 800. Visitors can still navigate through the visualization, as well as zoom in or out, in order to see higher prices.

The computation of the treemap was quite easy for two reasons: because our data had been efficiently preprocessed in Python, using the 'EDA.ipynb' file, and because Plotly, the library we chose for this purpose, is quite user-friendly. Despite the smooth start, we faced a challenge when it came to applying the treemap functions to all cities. Our initial approach was to create eleven different

functions, one for each city, such as 'generate_treemap_london' and 'generate_treemap_berlin'. However, this approach quickly proved to be inefficient. Recognizing the need for a cleaner, more concise solution, we decided to generalize our approach. We designed a single function, 'generateTreemaps', which took a list of cities as an argument. This function iterated through all cities in the CSV file, plotting only the treemaps associated with the city to be displayed. This made our code cleaner and more manageable. The main problem was the arrangement of the treemaps on the webpage. It proved quite difficult to display five treemaps simultaneously in a way that allowed the user to easily read each of them. Our initial efforts were focused on modifying various parameters in the CSS file, but despite our best attempts, this approach didn't provide the clarity we were looking for. We finally basically created a separate container for each of the treemaps, removed the margins of the Plotly graphics and used the grid property of css. This simple modification strongly improved the visibility of the treemaps.

For what concerns the geographical map with the meteorological data, the first step was to gather the data. This was done using the API from https://open-meteo.com/ to get both the number of rain hours per day and the mean temperature per day for the whole year of 2022. Then the data was processed using python and Jupyter notebooks [1] so that we won't have to fetch it again every time (the data would not change anyway).

Once this has been done, the map is displayed using TopoJSON and a vector file of Europe with a not very high resolution and with a Mercator projection so it has a nice and familiar look. We can here choose the Mercator projection as well as low resolution since we are not really interested in the geographical properties.

The number of sunny days (number of days where there was no rain), is shown as the area of the circles on the visualization. To translate the data into the area and not the diameter of the yellow circle, we use a square-root scale for accurate representation. A big challenge was to get the tooltip showing the mean temperature right. For the background of the tooltip, we used https://css.glass/ which permits to create these blurry glass containers.

3 Design evolution

The development of our website's design has been a step-by-step process, starting from basic sketches and ideas to the final visually appealing elements. We initially used a Jupyter notebook during our first milestone, which helped us select the main topics for the website and what we wanted to visualize. During this process, we found that Airbnb prices, distances to city centers, and Airbnb capacities would provide interesting data for comparison among European cities.

In the pursuit of the optimal way to visualize prices relative to cities, we experimented with box plots, violin plots, and probability density functions (PDFs). After some deliberation, we concluded that violin plots offered the most effective visualization, particularly with the addition of a distinction between weekend and weekday prices.

Originally, our concept for visualizing distances from Airbnbs to city centers and metros was to create unique plots where each city was represented as a collection of circles, as seen in Figure 2. The radius of these circles were intended to symbolize the distance between Airbnbs and either downtown or the metro, with three circles representing the first quartile, median, and third quartile respectively. However, we soon realized that these plots were challenging to interpret and consumed more space than desired. In response, we chose to use radar charts instead, shown in Figure 5, which summarized all average distances in a single plot for each city, proving easier to understand and more aesthetically visualisation. Then, we decided to display only one single radar chart at a time on our website, showing both downtown and metro distances at the same time.

In considering the visualization of Airbnb capacity distributions, our initial thought was to create pie charts for each city. However, after learning that humans often struggle to accurately estimate quantities based on angles, we opted to replace these with treemaps. These allowed for more precise comparisons of surfaces and included the percentage of each Airbnb category in each treemap area.

Finally, for weather visualization, it became apparent that an interactive map would be the most effective way to show variations in sunny days and temperatures across space and time. Initially, we planned on using a line plot for visualizing temperature changes throughout the year, as seen in

[1] Code can be found in the github under `rain_data_preprocessing/` folder

Figure 5. However, upon implementation, we found scatter plots to be more effective, despite being slightly more complex to read, as they better represented the temperature variation each month. As shown in Figure 5, we originally planned to display the number of sunny days per year for each city by basically printing it, but decided it would be more insightful to adjust the radius of each city point proportionally to the number of sunny days per year in each city.

We first designed each of the plots with its own colors. However, we realized further down the project that a common color palette for all the plots would more consistent. We then decided to use a common color palette to harmonize the look-and-feel of the whole website. To achieve this, colors where chosen based on the image of the house acting as a navigation bar (see Figure 3). The final color palette can be seen in Figure 4, we added a redish color to add some contrast and give more impact to our visualizations.

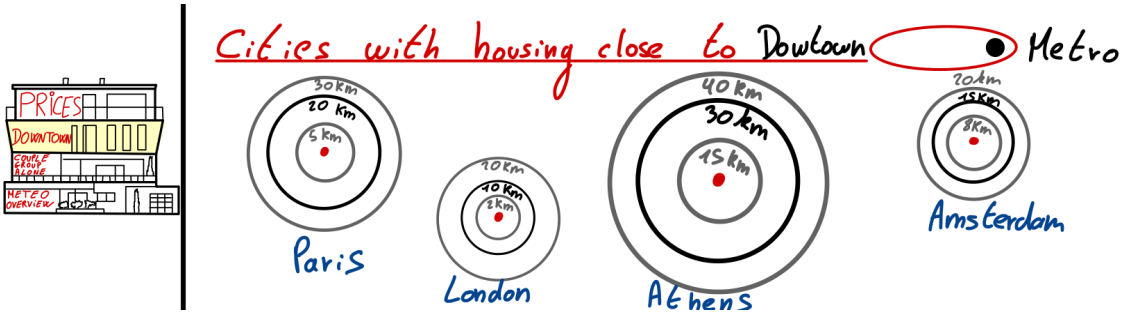

Figure 2: First idea of the plot comparing distances from airbnb to downtown and metro

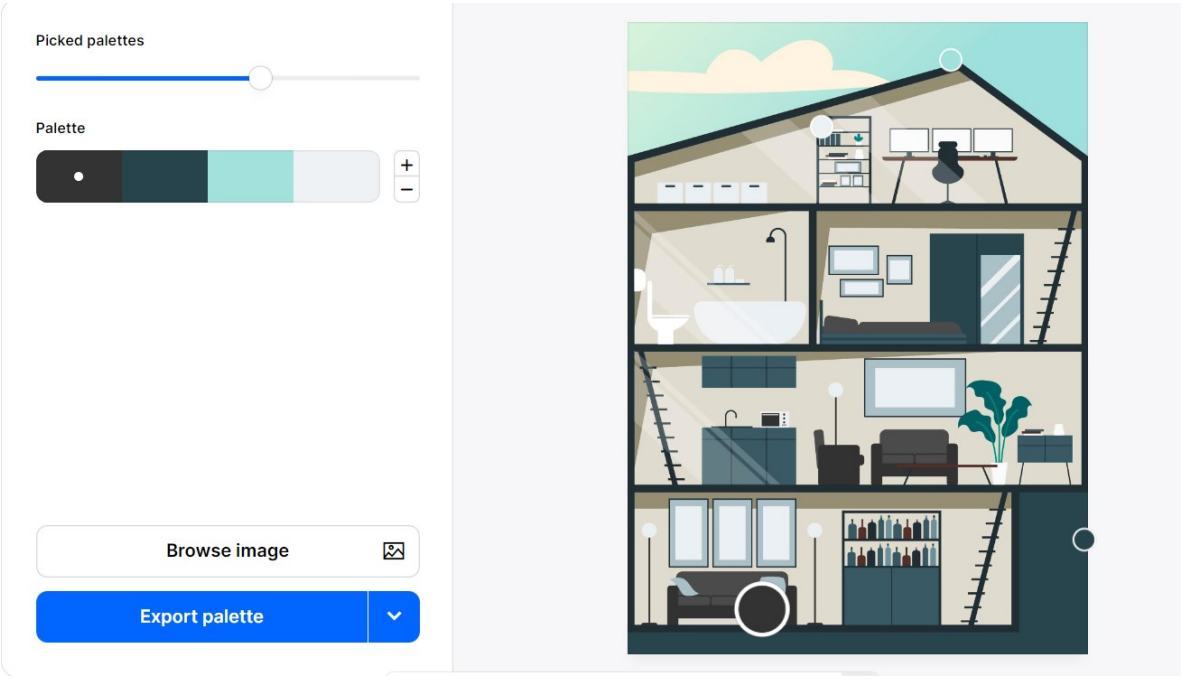

Figure 3: Extraction of the color palette

4 Peer assessment

The distribution of the workload was fair throughout the project. The first exchange of ideas took place in a meeting before the 1st milestone. For the second milestone, Gaston created the backbone of the site, while Kelyan was in charge of schematizing our final result and Pierre handled the report. For the final delivery, we divided the graphs. Pierre took care of the violin chat and the radar chart.

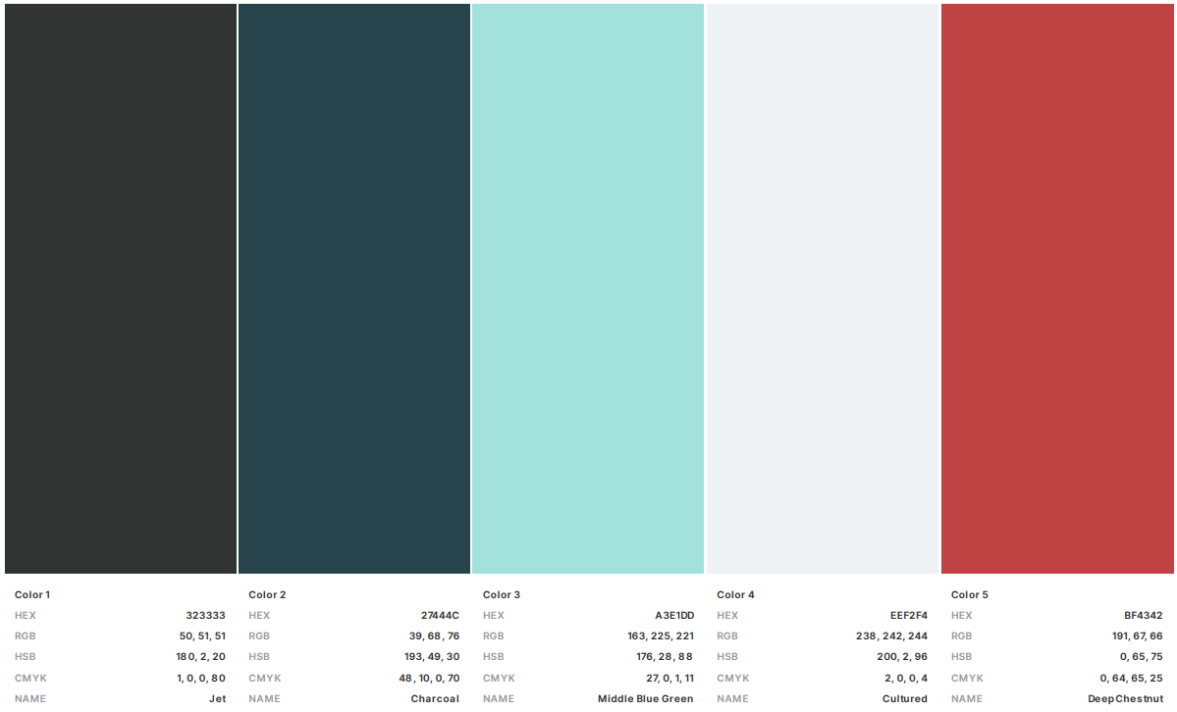

Figure 4: Final color palette of the website

Kelyan handled the treemaps. Gaston took care of the map with the meteorological data, and a few global improvements like the info point. Each member also contributed to the writing of the report and the creation of the video equally.

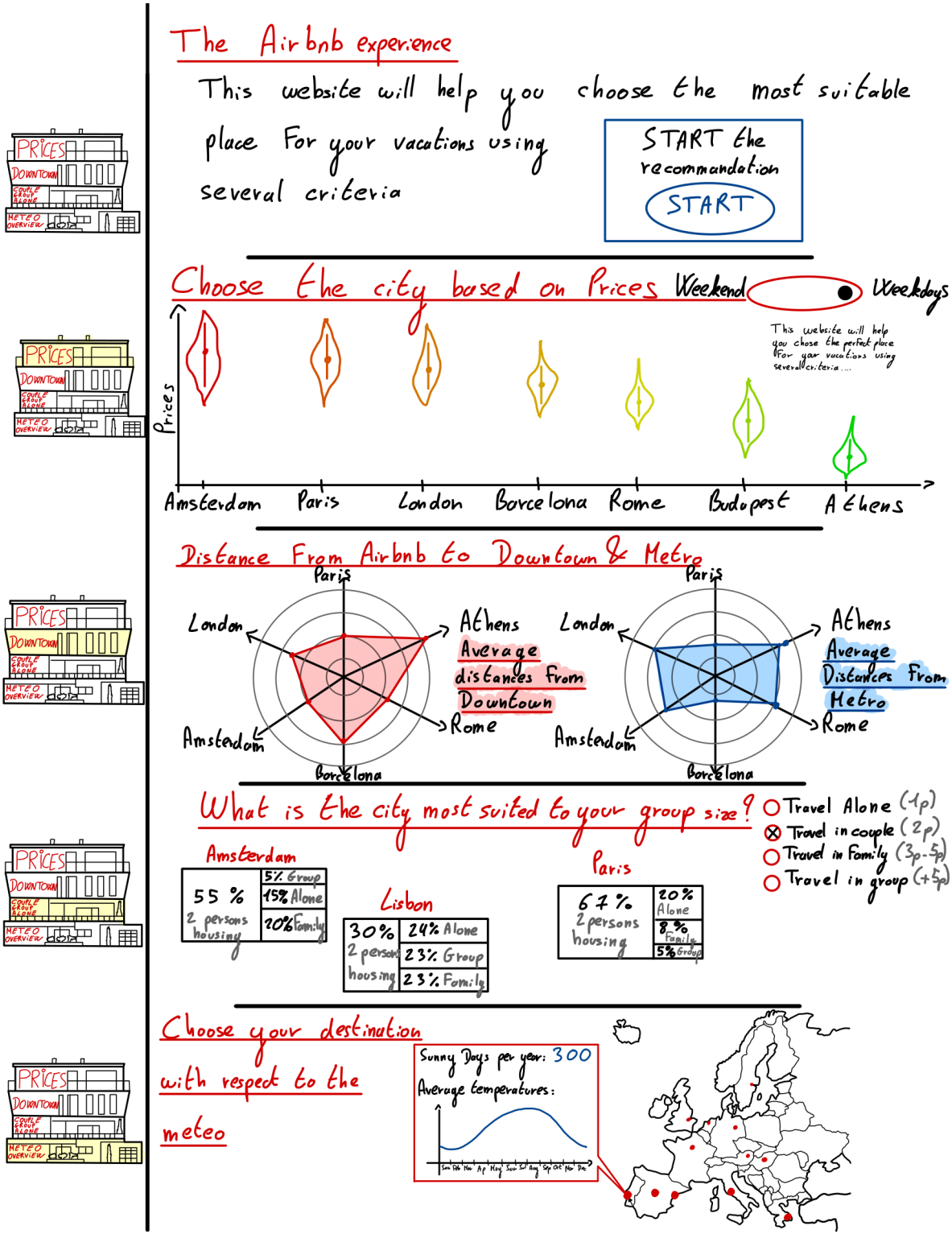

Figure 5: Website plan of the second Milestone