Data visualization - Process Book

Seasonal trends in hotel reviews

Hilda Abigél Horváth, Adrien Schurger-Foy, Julian Schnitzler

June 3, 2022

1 Development process

From the previous milestones we started the final milestone with a ready plan and a basic website. Our plan was to create two interactive maps. As we described in the previous milestones, the first of them is about the reviews given to hotels across Europe, and the other one is a connections map visualizing the reviewers' nationalities. We also attached the sketches below. In the following subsections, we describe the development process in approximate chronological order, thereby indicating the challenges and design decisions in the exact order we encountered them.

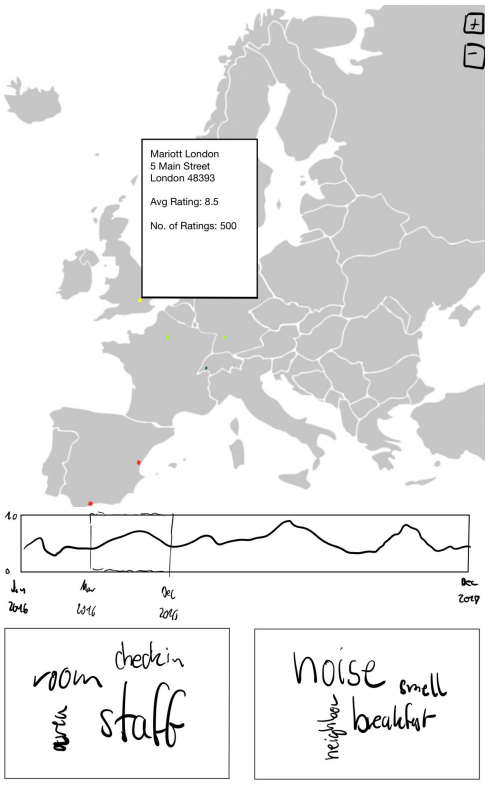


Figure 1: Sketch of the first visualization

Figure 1: Sketch of the second visualization

1.1 Prepare data

To not load the same data twice, we decided to put most of the needed data in a single .geojson file: it includes the names and locations of the hotels, as well as information about the visitor reviews. In preparation for the word clouds, we lemmatized the reviews and stored them with the hotel names, coordinates and ratings. To improve the performance of our webpage, we aimed to process the data beforehand as much as possible, since we understood very early that the computational power of a browser is quite limited, leading us to put as much of the required computational work as possible in our preprocessing script.

1.2 Maps

The first step for both of our visualizations was getting a base map. The basis of the two visualizations are two maps for which we chose to use different approaches. For the Europe map we decided to use a library called Leaflet, which allows us to easily make interactive maps. In the beginning, we only wanted to use topojson, but after some time we decided that hotels are too complicated to distinguish since there are no streets or any landmarks to orient on. It is also important for a tourist to see exactly how far are the attractions and transportation when choosing the right hotel, which is not possible without these tiles. Hence, we used OpenStreetMap tile layer as a basis of the map from the Mapbox provider. We choose to use a light theme to make the visuals simpler. For the other map, we used topojson since we need not display as much information, making it easier to distinguish the arrows from the background. In addition, it allows us to color the countries according to the number of tourists originating from there. To allow a user to zoom into and move the map, we added a d3.zoom to it. When the zoom event is triggered, we also need to set the transform attribute of all elements in the map to event.transform, since otherwise they do not join in the movement.

However, in connection with maps we faced the problematic of how to use these free but licensed tools in the right way, so we had to search for the way how to use it properly.

1.3 Europe map markers

The next step was to create markers for the hotels. Based on their geological location (longitude and latitude provided in the dataset), the Leaflet library offers a way to add markers on the map. The library also provides a way to show information about the hotels on click action. We decided to add the hotel name and the average rating to the tooltip. The average rating is rounded to one decimal, to make it readable and at the same time allow for distinguishing between hotel ratings.

1.4 Brush selector

To allow users to pick a specific timeframe, we put a brush with which the timeframe can be chosen. Since we already had the axes for the lineplot that should come in the background in our skeleton, we simply had to add a brushX on top.

The brush selector rounds up to the nearest month, since we want the user to be able to understand the timeframe chosen, directly from our axis that we provide below. The brush writes it's current selection

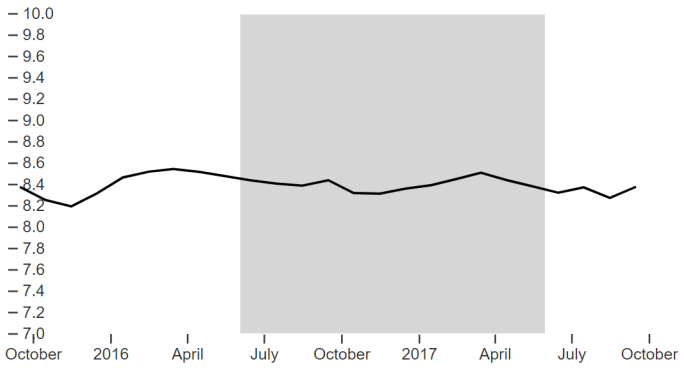


Figure 2: The final version of the time selector with average scores

into two global variables, which can then be read by the map and wordclouds to change the selection. On the brushend events, we need to update the wordclouds and the colors of the markers in the map.
In our version at this step, we just put the average over time for all hotels in the dataset. We use a granularity of one month, so each datapoint is the average of one month of ratings for all hotels in the dataset. This simplifies the data we work with, and allows us to include more processing in our preprocessing script rather than the webpage.

1.5 Word clouds

We first attempted to use the wordcloud2 library, but this library did not allow dynamically changing the words, something critical to our visualization. We instead used AnyChart along with a custom wrapper object to manage the word clouds. We decided to show word clouds of the reviews of only the hotel clicked, as using the reviews from every visible hotel would require too much processing time. Based on the time frame and hotel selected, a subset of the lemmatized reviews are taken to compute on the fly the frequency of each word. This is unfortunately something we cannot pre-compute, as there are too many possible selections of hotel and time frame. This results in a rather long processing time for every change in hotel or brush selection. We can speed up this update time by drawing just a few common words, although we initially used 20. Drawing 20 words took too much time, and many of the drawn words had quite low frequency anyway. We estimate 8 words should be a good balance between information and time on most modern computers and browsers. By drawing just 8 words, we sacrifice some words of lower importance, but the decrease in processing time leads to a more streamline visualization.
This example shows that seeing these words can be pretty useful for tourists, since it makes it obvious that most of the complaints are about the room, the bed and the breakfast, while one of the best things about this hotel are the helpful and friendly stuff.
By hovering over the words, one can further investigate the frequency of the words among the reviews and see the percentage of people who used that word in their review.

1.6 Markers on the world map with connection lines

For the world map, after drawing all countries in the map, we drew our own circles. With the fill and stroke style properties, we managed to create circles that act as markers on the map. When clicking one

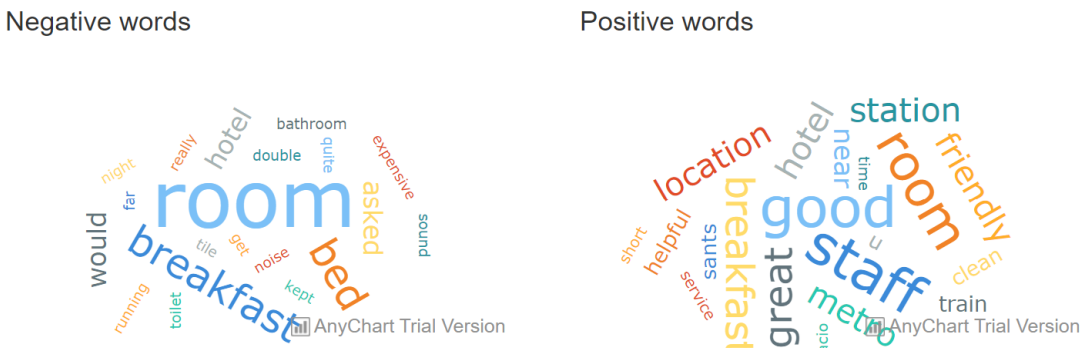


Figure 3: Example of the word clouds

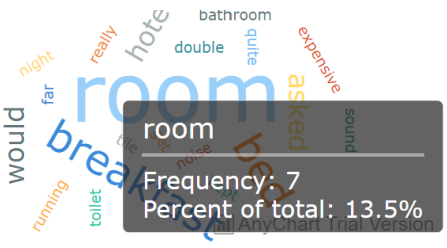


Figure 4: Example of the tooltip in the word clouds

of those, the lines and colors for the reviewer of this hotel are set. The lines are drawn by adding lines to the svg that is the map. We use a JavaScript object that maps hotels to their coordinates, and then input these into our geoprojection to get the position the markers should have on the map. The thickness of each line is proportional to the log of the number of reviewers from that specific country. When moving the mouse over the line, i.e. when the mouseover event is triggered, a tooltip, (a standard html div with some special properties), appears and indicates the country and the number of reviewers. Here, one of the main challenges we faced with the markers was changing the radius properly when zooming in and out. We found out that event.transform.k saves the factor of increase in size, and so we're able to divide by this factor to get the new radius of the circles when zooming in. Also, for the tooltip, we had to get the coordinates of it's position not with respect to the svg, but with respect to the webpage in the background. However, it turns out that in event.pageX and event.pageY the mouseover event saves the exact position where the mouse was hovering over the line, with respect to the coordinates of the page. Hence, we simply had to add some pixels to it to get a position near the cursor, to show a Another challenge here was to draw the correct lines still when zoomed into the map and chosing a different hotel, since the new lines need to get the transformation that is already there. We had to save the current transformation in a variable and add it as an attribute to the lines to make sure they work properly.

1.7 Colorful markers

For easier interpretation our plan was to give colors to the markers based on their review score. To this end we used the awesome-markers plugin. This plugin provided colorful markers with different icons on Leaflet visualizations. We decided to use 6 specific colors, instead of a continuous scale to choose from,

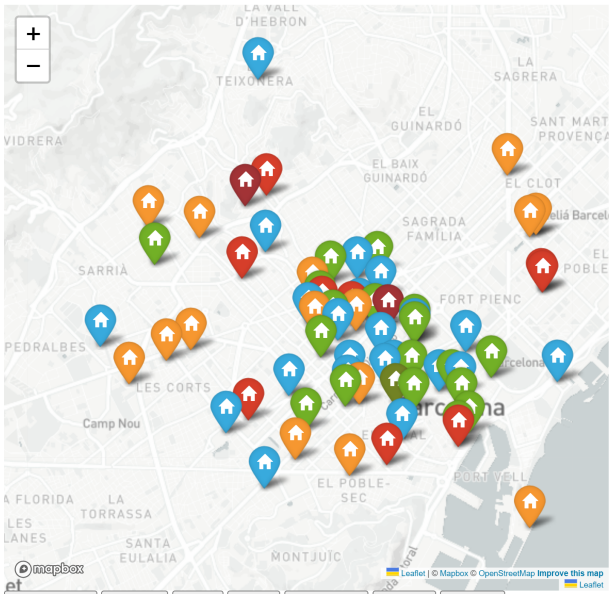


Figure 5: Colorful markers after using awesome-markers

because it makes easier to interpret which hotels are outstandingly good and which are really bad.

1.8 Zoom buttons for cities

As the dataset we use contains only hotels from 6 different cities, we decided to add buttons for each city allowing the user to easily zoom and navigate to the chosen city. Also a 'Europe' button was added to easily get the original zoom. We also had to choose the colors for these categories. We used the dark red - red - orange - blue - green - dark green scale, since the interpretation of red is bad green is good and blue is neutral is widely used. These buttons do not remove the ability to zoom and move on the map

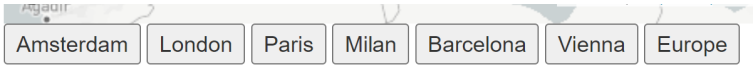


Figure 6: Buttons to navigate the users to cities on the map

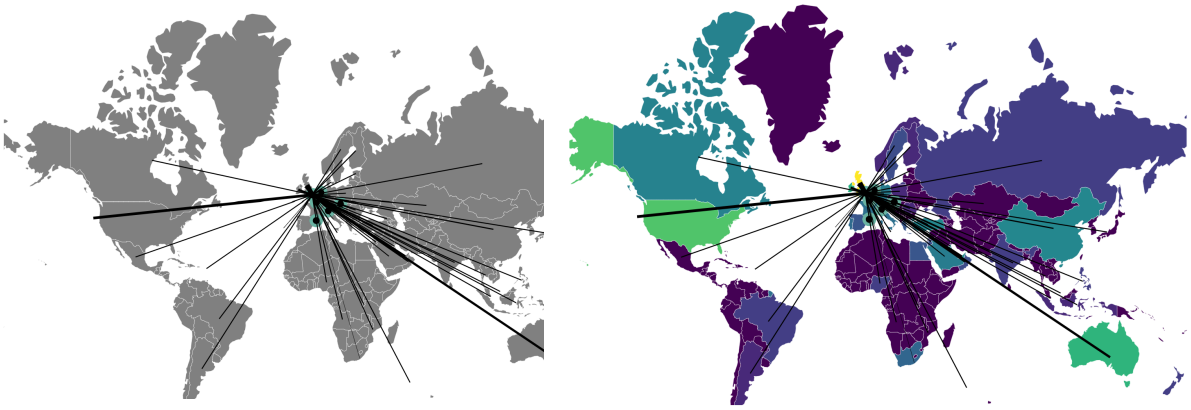
freely, just help to easily change views between cities and the Europe map.

1.9 Zoom buttons change lineplot

In the background of the brush selector, we display a plot of the average ratings of all hotels over time. When clicking on one of the buttons, this changes to only hotels of the selected city. We wanted to make it dependent on all hotels currently in the map, but couldn't follow this approach further since it would have crashed our page because it was too resource heavy. Hence, we decided for a slightly less resource-demanding version: as mentioned above, when clicking on one of the city buttons (or Europe), this changes to the respective city/whole Europe. When it is on one specific city and you zoom out, above a certain level, we switch it back to the average of all hotels. This arrangement moves client processing time to our preprocessing script.

1.10 Colors on the second map

Our plan was to not only visualize the number of reviewers via the connection lines' thickness, but also with colors of the countries as well. We realized that using the linear scale to convert to colors are not a good idea here, because there are many countries few tourists come from, but there are some that count hundreds or thousands. By using a logarithmic scale, the viewer can better understand the data.



(a) Grey Map

(b) Colorful map

Figure 7: Difference between the maps

As it can be seen in the figure, the colors help in understanding where the reviewers come from.

1.11 Connection between the two maps

As we wanted the two maps to interact, we added that on click of a hotel on the first map the connections appear in the second. This doesn't remove the possibility to change hotels on the second map, but makes it easier to gather all the information about a specific hotel. To do that, we simulate the same behavior that a user would do in that situation: In the id of each circle in the second map, we saved the name of the hotel. We can then get this circle by id and trigger a click event, for which then the lines are drawn, the colors are changed, and the tooltip with the hotel name is generated. With this feature, the user can easily see where the reviewers for this specific hotel came from.

1.12 Brush changes colors and scores

As a next step, we connected the marker color on the first map to the brush selecting a time frame. The colors change based on the average scores which are recalculated only on reviews from the selected time frame. For this, we needed to call the update functions that both the colors and scores provide, also on the brushend event.

1.13 Hotel names on second map

In order to help to identify the chosen hotel, we added that on click the name of the hotel will appear in the top left corner of the map. We again decided to use a html div, and simply add some CSS style like a solid border, background color and text-align center to make it look like a tooltip. However, compared to the first tooltip of that style, the one for the lines in the second map, it was this time a larger challenge,

since it is not possible to simply take the svg coordinates or, as we did for the line tooltip, use the event coordinates of the mousover event to determine the position to put the tooltip. We found out that the getScreenCTM function of the clicked circle gives the screen position of this circle, which can then be used together with the window offset to get the exact poisition of the circle on the webpage. Now, combining this with the knowledge of the position in the svg coordinate system, we were able to find the top left corner of the svg, and add the tooltip there, since it was the position that we considered the least distracting and least blocking any relevant information below.

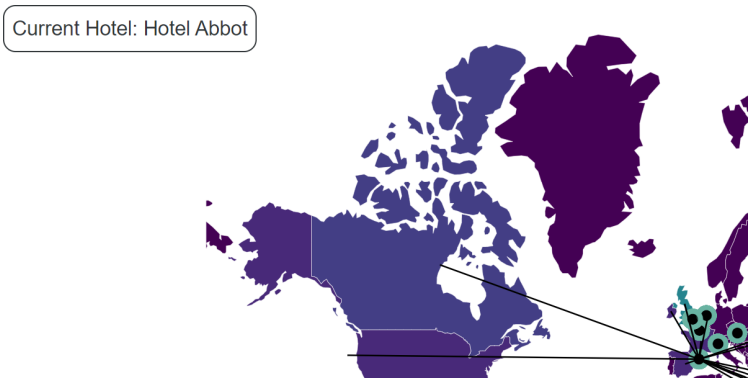


Figure 8: Example of the current name on the second map

2 Result

As a result of our work we could implement all of our goal set in the previous milestones. Although, we added a few more functionalities to help the users. These changes are the buttons, connection between the two maps and the colors on the second map. Furthermore, we slighly changed the layout of our webpage. We decided to add the brush and the word clouds on the right of the Europe map, because screens have more horizontal space, and it makes it easier to use these three components interactively.

3 Peer assessment

In the following, we list what each of our team members contributed to our project.

Julian:

- Map for the origin of reviewers
- Brush
- Lineplot
- Host large files with AWS S3
- Final report
- Final refactoring & code organization
- Screencast

Hilda:

- First map with markers
- buttons for the map
- Colors of the second map
- Layout of the webpage
- Final report
- Final refactoring & code organization
- Screencast

Adrien:

- preprocessing
- data management/selection helper functions
- wordclouds, dynamic connection to brush and hotel clicked
- Final report
- Map legend