Process book

Devrim Celik, Huan-Cheng Chang, Nina Mainusch

Data Visualization Project 2021 - Teamtwitch

1 Problematic

We want to focus on how the Corona Pandemic influences the interests of the Twitch.tv[4] users. In this setting, we aim at incorporating the following questions into our visualisation:

1. Firstly, we want to get a picture of the general trends of certain categories over the last few years. For example, we want to find out which categories were constantly popular and which ones were rather short-lived.

2. The Corona pandemic changed our lives in a multitude of ways, one of them being that many individuals work from home. As a result, we ask whether the amount of time that individuals spent on recreational activities on the web, such as watching gaming live-streams, increased. And if it did, did it increase within the progression of the pandemic?

3. We are interested in whether the life-altering changes that came with the current pandemic have an influence on our interests. Specifically, we are interested whether certain channels have become unexpectedly more popular under the pandemic. One channel that we are very intrigued in is chess. Twitch.tv has noticed a huge surge of interest in this channel over the year 2020. Many argue that this is due to release of the popular TV show Queen's Gambit[4]. We want to investigate the big picture: Did the pandemic have an influence in the surge of this category? What other factors played a role?

What are we trying to show?
We want paint a picture of what Twitch.tv is and what kind of trends were observable before the pandemic. Afterwards, we will contrast these findings with trends after the pandemic. Lastly, we want to introduce the chess phenomenon on Twitch.tv and find out how it came to be.

2 Data Collection

Our data comes from SullyGnome[3], a website that collects statistics and analytics of games streamed on Twitch. Data is aggregated by month, starting from January 2016. It is further broken down by

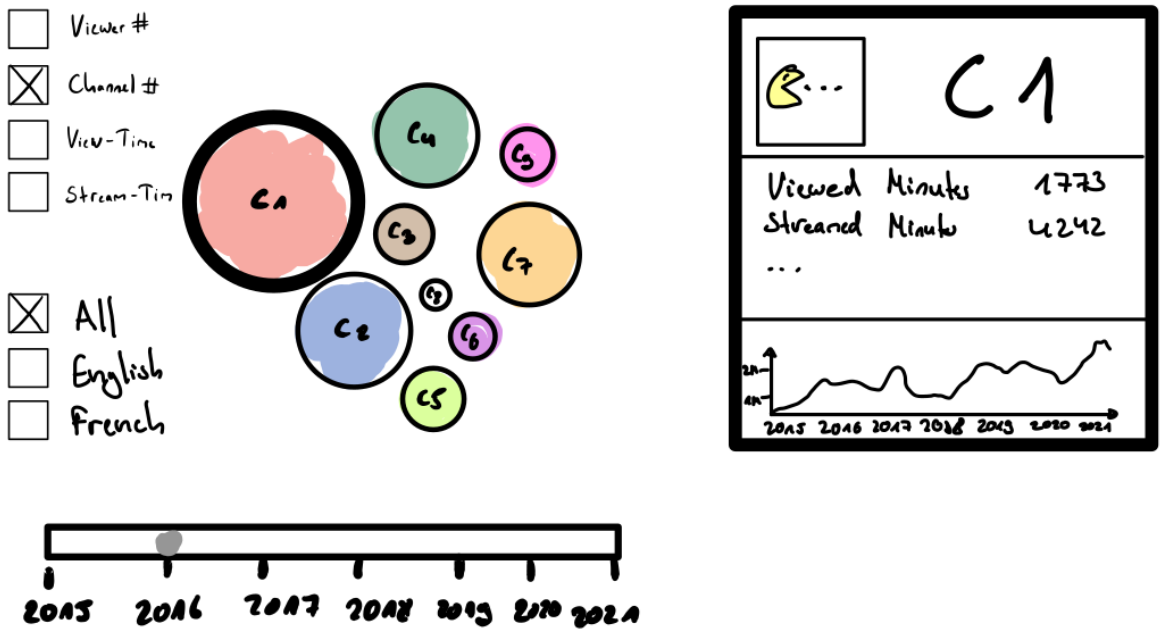


Figure 1: Final version of the Bubble Chart, now including a overview over the selected category.

the language of channels streaming the games. There are 37 different languages available. However, some languages have missing data, so eventually we choose 11 of them.

A python script is written to parse data from the website. In the script, HTTP requests are sent to the API endpoint, and then results are parsed and dumped to files in CSV format.

2.1 Challenges

To maintain portability, the script was run in Google Colab[2] and data was written to Google Drive directly. The main challenge was that HTTP requests must carry a specific header but the header expires after some time. The header is generated by SullyGnome's UI. We did not study their code thoroughly, so the header was copied from the UI and updated manually after a certain amount of time.

3 Data Visualisation

3.1 Bubble Chart

After having fixed all the minuscule issues we described in the last milestone, we were greeted with a basically finished plot. However, we were not completely satisfied with the result: Although the bubble chart on its own did convey information about popular categories at different time points, there were two pieces of information that were not shown at all:

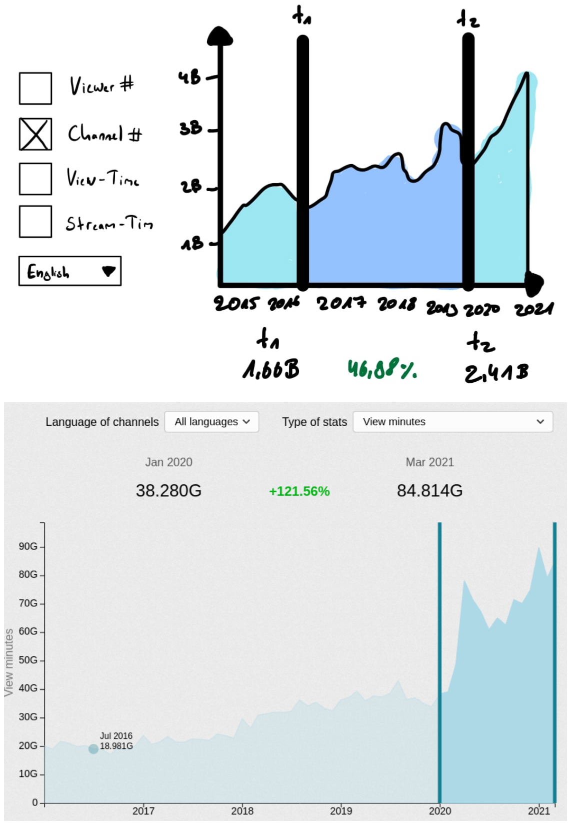
1. The development of each individual category over time?

2. The bubble sizes are relative to the sizes of the other bubbles for a given month. That means, there was no way of accessing the absolute size.

To address these points, we decided to include a line chart in the info panel for a selected bubble, that would depict the development of the selected category over the whole time interval between 2015 and 2021, see Fig. 1. From a practical perspective, adding the line chart was no major hurdle, however, the data we were given was only available per month. This means, that we had to aggregate the data of the months for the selected categories over the complete time-interval. We selected to collect information only about those categories, that are, at one month or the other, represented by a bubble.

3.2 Area Chart

This chart displays how the popularity of Twitch changed over time. The area represents the type of statistics that is displayed, e.g., view minutes, average number of viewers, etc.. Users can choose the language of channels to compare the trend of Twitch in different regions of the world. When the cursor hovers over the chart, an indicator shows up and displays the data of that time in the chart. The most interesting thing in this chart is that users can interact with the vertical bars, as displayed in Figure 2. The range wrapped by the two bars is in a darker colour for clarity, and the data points represented by both bars are displayed on the top of the chart. Users can therefore see how exactly the numbers changed in the selected period of time. For instance, one might be interested in how the total view minutes changed before and after COVID-19 hit the world. Moving the bars to January 2020 and March 2021 respectively demonstrates this change.



Language of channels: All languages
Type of stats: View minutes

Figure 2: Area chart. Top: the design; bottom: actual implementation.

The data displayed in this chart is the aggregation of all games of each language. Since the distribution of games is highly skewed, the aggregation of all games does not reflect the overall trends or the order of magnitude of the statistics well. We attempted to display data with top-10 and top-50 popular games together in one chart with two areas in different colours in order to compare changes of popular games and super popular games. The idea is that possibly only the super popular games became even more popular while others remain almost the same. Nevertheless, displaying the two aggregations together put too much information in one place and brought extra cognitive load to users, and it is not straightforward to find a good aggregation for all stats available in the dataset, so eventually we display only 3 types of stats with the full dataset: view minutes, streamed minutes, and number of unique channels. We think these stats are enough to demonstrate how the popularity of Twitch changed.

In the initial design described in previous milestones, this plot was a bar chart with two bars only that showed the data of two different time. The downside of the bar chart is that users cannot see the trend over time. They can learn about the difference of data in two different time period, but they have no clue in how it changed in the middle. Also, it is hard to fix the scale of the y-axis since there were only two values in the chart and they changed a lot when the selected time period changed. Using the current area chart brings the following improvements:

- Displaying the whole time span gives users the whole picture of the trends.
- Letting users select the range they are more curious about brings more flexibility and fun of mining insights from the data.
- Using a light colour for the unselected area helps users identify the action they are taking now and focus on the region they select.

3.3 Chess Plot

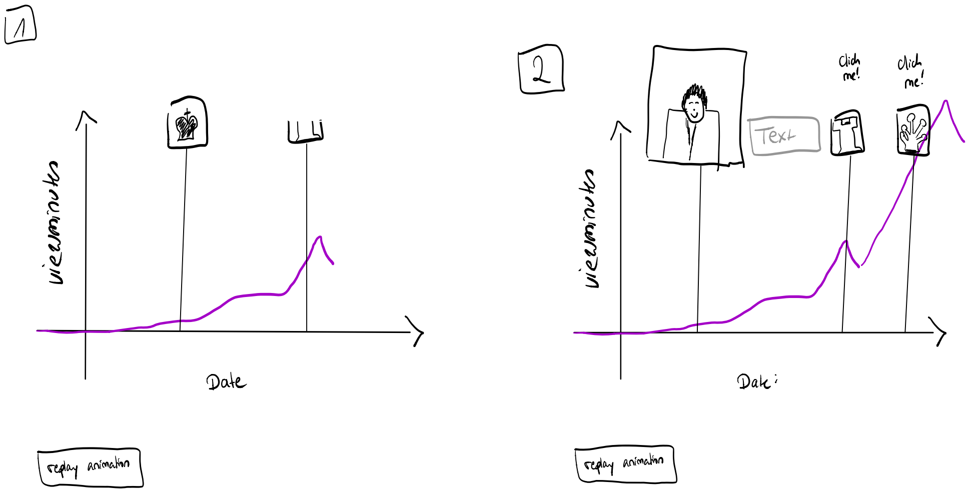


Figure 3: Final implementation of our chess plot. In 1), we can observe how the plot evolves over time, in 2) we see the complete version.

Regarding the chess plot, it still explores the chess category on Twitch.tv and implementing the first draft was rather straight forward. First, we set up the data, the axes and a static line connecting the data points. On hovering over the line, the date and the corresponding viewminutes get displayed.

Originally, we intended to represent the events as lines with a square box with a number, where below the plot a textbox would explain the event. We decided to modify this slightly, by making the plot interactive. We still have boxes representing the events, but now the user can click on the boxes, they will expand into a bigger box displaying some photo and a text will appear right next to the plot explaining the event. If the user clicks on one of the other boxes, the current box transforms back and the new text and image gets displayed.

Another change we made was to let the plot evolve over time. Doing this, the user can experience the development of the chess-channel in this third time dimension. When the website is loaded, the line starts at position $(0, 0)$ and develops fully within 10 seconds. On passing one of the three dates, the corresponding line starts to build itself up, resulting in the formation of the box. Here the hardest challenge was the development of the line over time. We finally found the `d3 .ease()` function, which made this effect possible. For each of the lines and boxes we manually had to set the time for their respective development. Our final plot is sketched in Fig. 3, where you can see in the upper plot the chart after approx five seconds, and in the lower plot the final version. On the bottom of the plot we added a "replay" button, such that the user can repeat the effect of the development of the line.

Initially, we planned on doing an extension to this plot, which would have basically been a visual embellishment. We have decided to not do this, but instead to integrate the interactive feature of clicking the boxes and the evolving line, which will ideally also lead to better memorability and furthermore contribute to what we want to transmit with the visualisation: the user should notice that there is an increasing trend and which underlying events might be responsible for this trend.

3.4 Scatter Plot

Our scatter plot has the purpose to answer the question what the secret behind a successful channel is. Does more stream time imply more viewminutes? Or can you only be successful, if your channel is in a popular category? Inspect a sketch of the final plot in Fig. 4.

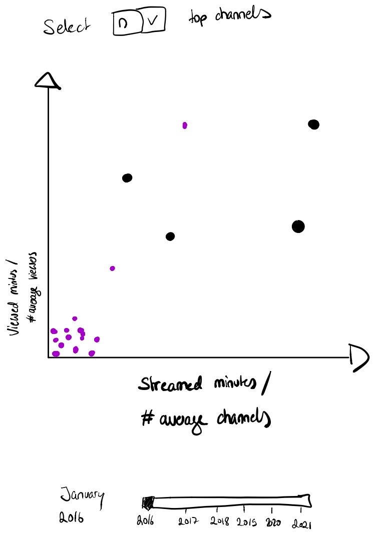


Figure 4: Final implementation of our scatter plot.

After thinking about what to best plot on the axis, we settled for streamed minutes normalised by the number of unique channels on the x-axis and viewed minutes normalised by the average number of viewers on the y-axis. On top of the time slider we planned before, we introduced a drop-down menu for letting the user select n, the number of highest channels that he wants to mark. This will allow him to see where a successful channel is placed in the graph. On fixing

n the highest channels will change their color and increase their size for better identifiability. We reduced the number of displayed channels to $N = 40$, since the plot looks to crowded for higher numbers of N. Out of this 40 channels, 20 are selected because they have the highest amount of viewminutes in that month, the other 20 are selected at random.

One of the challenges for this plot was to adjust the data when the time-slider is used, which we overcame with the `.transition()` and a custom implemented `changeDots` function.

4 Website

4.1 Set-up

The web app is built on React.js for better modularity and fewer nuisances when merging changes made by different people. For the page layout design, we use fullPage.js[1] to display individual pages easily. The web app is deployed and hosted on GitHub using `gh-pages`.

4.2 Layout

Regarding the layout, we decided to keep it rather simple and use the theme color purple of Twitch.tv as our theme color. The web app consists of pages. Each page displays a section or a plot. Users can scroll up and down to move between the pages. The background color is a neutral paper grey, and the first two pages of our website serve to introduce the reader to our topic and not to overwhelm him with too much content.

The first page is the welcome page where the title of this project is displayed. In the second page, a brief introduction to Twitch.tv is presented for readers who have little knowledge about this platform. For each plot section, a text block for the description is placed on the left and the plot is placed on the right to comply with the general left-to-right reading experience. However, the plots themselves are designed to be self-explanatory.

5 Peer Assessment

Task	Responsible Member
Data Collection	Huan
Exploratory Data Analysis	Nina
Organization & Coordination	Devrim
M1 Report - EDA Writing	Nina
M1 Report - Rest of Writing	Devrim
Setting up Website Skeleton	Huan
Visualization Propositions for M2	All
Writing Bubble Chart	Devrim
Writing Area Chart	Huan
Writing Chess Chart	Nina, Devrim
Writing Scatter Plot	Nina
Website set-up	Huan
Website-layout	Huan, Nina
Video Script	Devrim
Process book	All

References

[1] fullPage.js. https://github.com/alvarotrigo/react-fullpage. Accessed: 2021-05-30.

[2] Google Colab. https://colab.research.google.com/. Accessed: 2021-04-25.

[3] SullyGnome - Twitch stats and analysis. https://www.sullygnome.com/. Accessed: 2021-04-25.

[4] Twitch. https://www.twitch.tv/. Accessed: 2021-05-21.