Process Book

YouTube viz - Process book

1. Welcome in the world of YouTube Videos

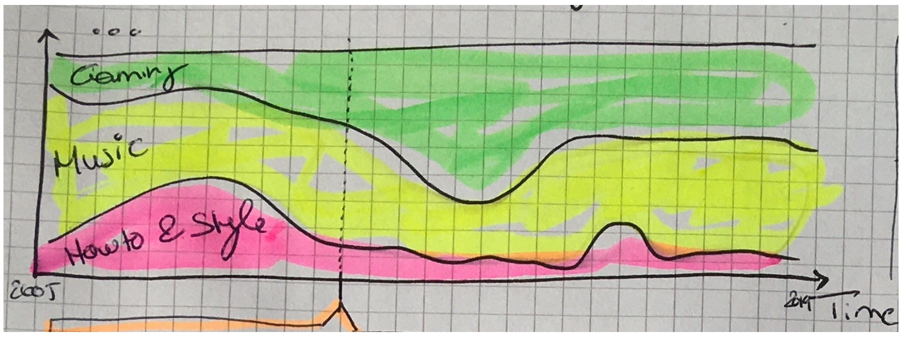
1.1 Origin of the idea

Our main aim with this project was to learn valuable skills from this class, while creating a visualization that was meaningful and usable in purposes outside of the class. To combine these two requirements, we asked the dlab where Timoté is doing his semester project, if any PhD students would need an interactive visualization for his research. Manoel Horta provided us with his recent scrap of 96 million YouTube videos metadata that he was using to study the evolution of trends on YouTube from 2008 to 2019. We were given *carte blanche* on how to visualize this data but kept in touch with him for feedbacks and insightful discussions.

1.2 Early visualization ideas

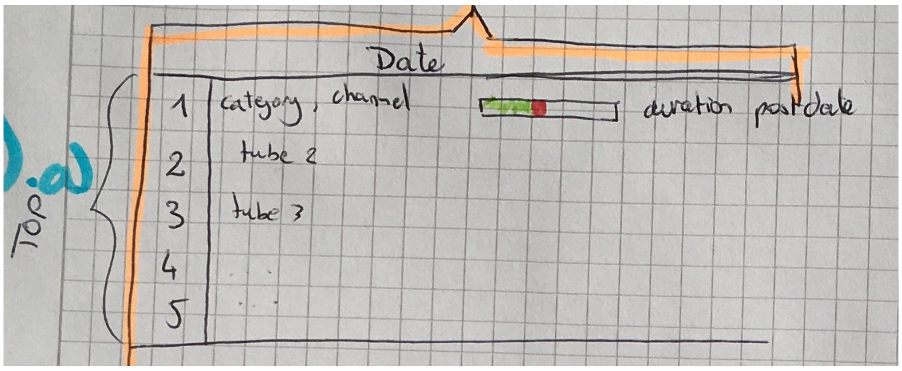
To stay within the aim of the research and make something that could be accompany the paper submission, we wanted to show the different kinds of content and formats that prospered on YouTube over the last decade. The solution we found the most appropriate to illustrate the above consisted of several graphical elements.

A. A stacked area plot scaled to 100%



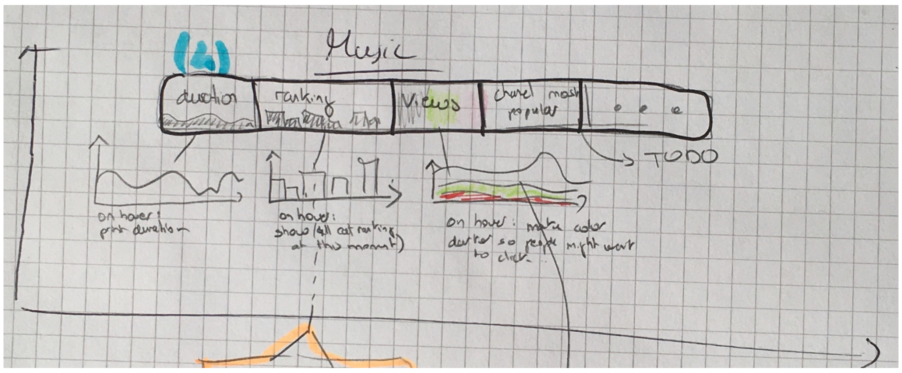
This chart would illustrate the popularity of the different categories of videos on YouTube over time. The initial idea was to be able to compare their relative importance over the years. The stacked area would highlight a region on hover to show it is clickable to filter the information presented below by category.

B. The most popular videos table



On hover, a table would have displayed the best videos on YouTube at a given time. We thought we might display the id of the video, the channel that posted it, the number of view and a link to watch the video on YouTube.

C. A more detailed view per category


(4)
Music
duration
ranking
Views
chanel most popular
TODO
on hover: print duration
on hover: show (all cat ranking at this moment)
on hover: make color darker so people might want to click

By clicking a category in the stacked area plot, we wanted to make appear a third interactive graph below the table of the best videos. This graph was intended to display different aspect of the category selected over the time. By clicking on a segmented control, the user could decide to visualize either the duration, the number of view, the ratio like/dislike or the average ranking of a category over time.

1.3 Intended Preprocessing challenges

A. Data Processing

One of the first issues we ran into is the sheer size of our dataset. Compressed it is over 10gb. We could never hope to fit that into any browser. The first step was to aggregate the data in order to reduce it to a much smaller file that could be easily sent over when loading the page.

Thus, the data was aggregated at a weekly and category level using *PySpark*. For each week and category, we compute a score. We also precompute histograms for the views, like and dislike counts and the video duration distribution. The bins are fixed and known so we don't need to write them in the files. We can later easily aggregate this data over multiple categories or weeks. Finally, we only keep the top 20 videos per category and week for the top videos table in order to reduce the file size to less than 8mb compressed.

B. Scoring scheme

We have been mentioning giving a score for each category, but how do we get it? For a given category C_i and a time period T, its score $score(C_i, T)$ can be defined as

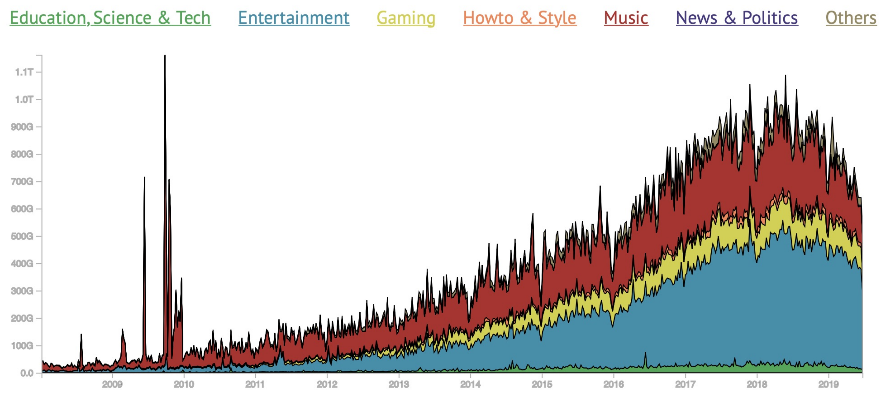
$$score(C_i, T) = \sum_{c_j = C_i \wedge t_j \in T} v_j \cdot w_j$$

Where for a given video j: c_j is its category, t_j its publication date, v_j its number of views and w_j the weight associated with the channel that published the video. This weight is inversely proportional to the probability of a channel being sampled during the scrap phase and was provided by Manoel. Rarer channels have higher weights because they are underrepresented in our sample. The score is then scaled down by 1 million to have a comprehensible Y axis.

2. A path full of pitfalls

2.1 A stacked area only? Not such a good idea!

When we first succeed at representing the categories of videos in a static graph, we faced a problem: the graph was not as smooth as expected. Instead of a constant and regular evolutions of the trends over time, the weekly granularity chosen made the paths look sharp, with a lot of peaks and depths. Moreover, we realized that the categories were unbalanced.



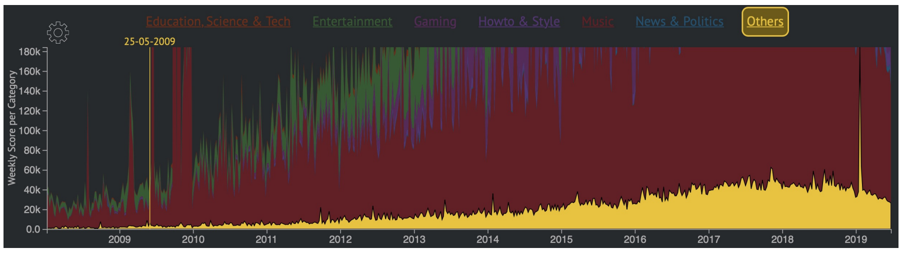
Induced problems:

- The spikes of the paths made it difficult to visualize any category that was not displayed at the bottom of the graph. Since the superposition of the chart in the graph increases this spiking effect, the category that were displayed on the top of the graph were quite unreadable.
- The fact that the categories were unbalanced made some of them look ridiculously small: this approach was not optimal.
- With a normal stacked graph, we could not tell easily which of the category was the most important at a certain date. In other terms, it was quite difficult to have an idea of the ranking of the categories at a given point of time.

Approached solutions:

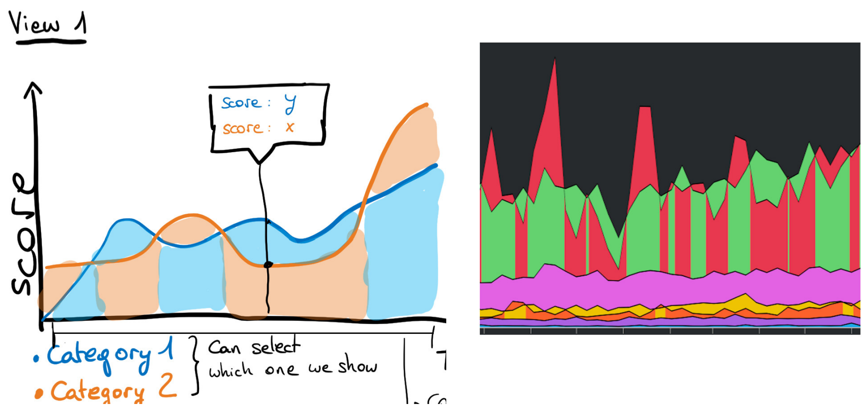
As we could not imagine a single plot that would have solved all the above problems, we decided to implement two different graphs:

The first one was a **clever** stacked area:



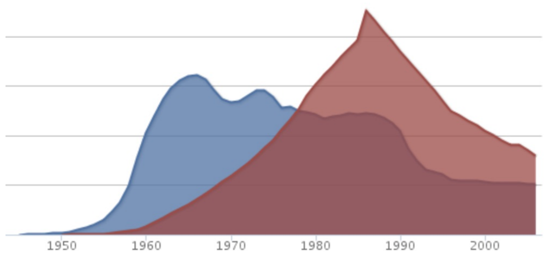
By selecting any category, it should have changed the chart order, to make the category appear bottommost. That way, the category would not have been distorted by a superposition with charts below it. Moreover, selecting this category dynamically adapts the y-axis, in order to maximize the space occupied by this category on screen.

We called our second solution the **Interleaving Chart**:



On the left: what we wanted to design, on the right: our results

The origin of the idea came to our mind when we were looking at some area chart example found over internet like this one

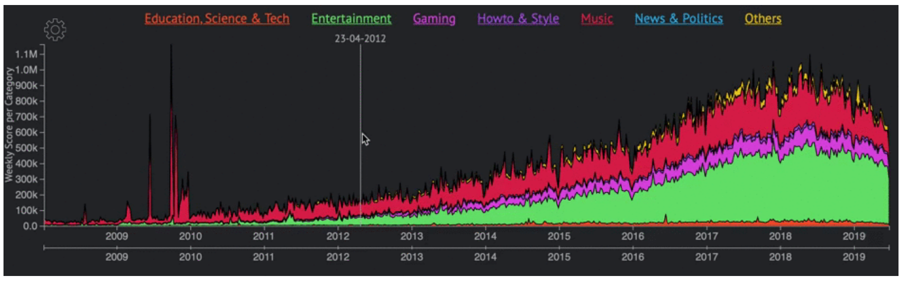


These plots are basically a superposition of charts which are filled with transparency. This allow the user to see clearly the exact shape of each chart. The limitation is that the more categories there are, the more confusing the plot becomes. As we had 7 categories to plot, it was too much for such a graph. Hence, we decided to implement something new: it works the same that the above area chart, but without transparency: we only show the lowermost color. At any timestamp, you can know the categories ranking by looking at vertical color interleaving.

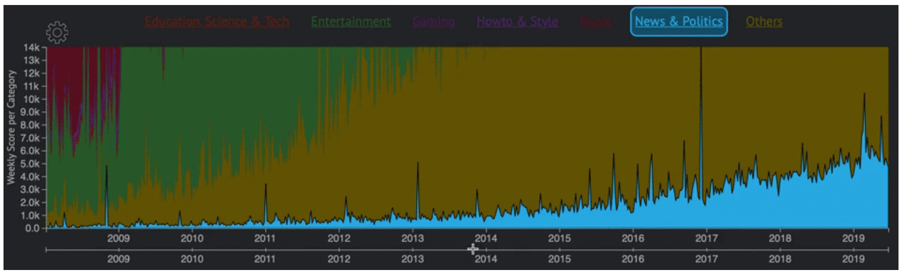
The implementation of this graph was quite challenging. To get coherent result, we had to look at our time series to find the critical indexes where any two series were crossing each other. Then we had to stroke the charts. With the help of the method getPointAtLength(), we could iterate along the paths to find the exact timestamp (with some delta approximation, expressed in pixel) where the charts were crossing. Once we had the exact intersections timestamps, we then fill the chart in the order they had to appear between any 2 critical timestamps with the help of a clip-path. As this computation was taking a lot of time, we had to optimize it by retaining only the indexes that were in the visible area of the graph and computing the intersection value with an error that was expressed in pixel. This was the most efficient way to provide to the user eyes a clean result independently of how much he was zooming in the graph. At that time, the computation time was small enough to provide the user a nice experience, but there were too many elements in the canvas. This was making the web page lag when the mouse was on hover the graph. To get rid of this, we had to skip the rendering of the charts which width were smaller than a certain pixel tolerance. In the final version, we decided to compute the chart interleaving 500ms after the user ended to scroll in the graph.

2.2 A fixed x-axis? Not enough!

As our time series were long enough to make it necessary, we decided to let the user zoom in the graph.



You can even click and drag in the graph,



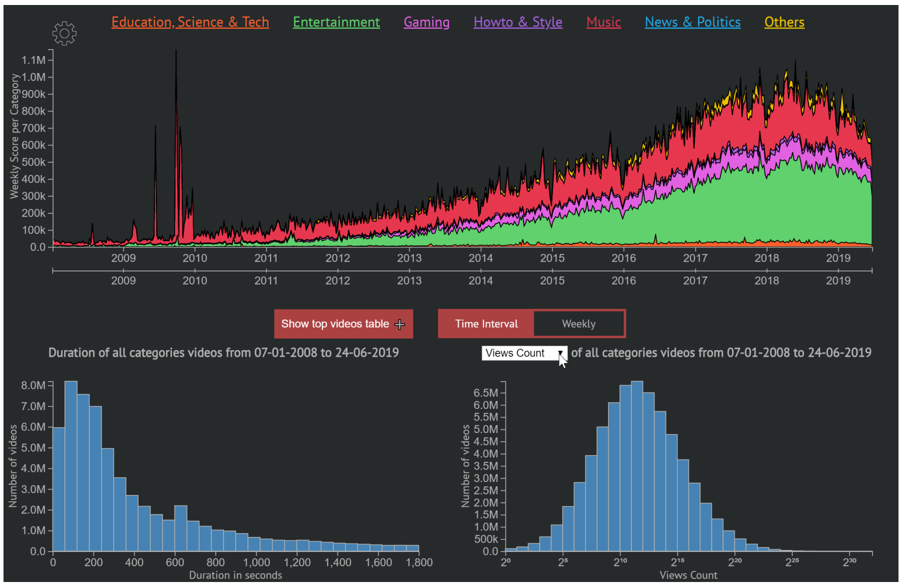
or use the d3 brush to zoom and explore the data.

2.3 Check out the best videos out there

Below the main visualization, we present the top videos in a table in which you can play it using our embed player while continuing to explore the dataset. The table is the only "plot" that is in `dc.js` , the other ones being in pure `d3.js` . It shows information such as the publication date, the number of views, likes, dislikes and its duration. The data is sorted by number of views. We use `crossfilter` in order to be able to filter the large data that we have over a given period of time or a given category. The filters are coordinated with your actions on the main visualization.

2.4 Let's learn more about our data

We mentioned in the preprocessing section that we also pre-computed some histograms. They are displayed below the table and also listen to the same `crossfilter` filters. They allow the user to learn more about the distribution of 4 important metrics: videos duration, views, likes and dislikes count. Compared to the early version, we polished the results and separated it into the 2 histograms you can see. In our opinion, it is important to be able to visualize the duration distribution separately from the counts that are inherently linked together: more views usually mean more likes/dislikes.

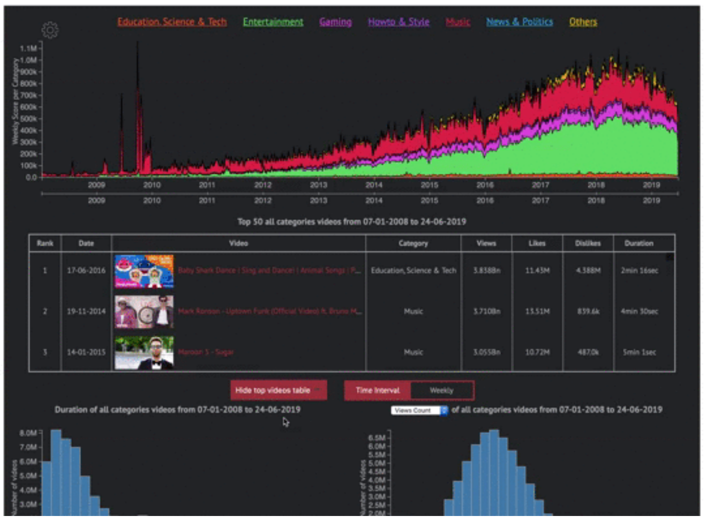


Education, Science & Tech
Entertainment
Gaming
Howto & Style
Music
News & Politics
Others
Weekly Score per Category
Show top videos table
Time Interval
Weekly
Duration of all categories videos from 07-01-2008 to 24-06-2019
Views Count
of all categories videos from 07-01-2008 to 24-06-2019
Number of videos
Duration in seconds
Views Count

We added transitions to accentuate the change when a filter is applied.

2.5 A Simple redirection to YouTube? Not cool enough!

While we might have put a link to YouTube to let the user watch the video he found interesting, he could have been sad not to be offered the possibility to watch his video directly in our website 😁



Education, Science & Tech
Entertainment
Gaming
Howto & Style
Music
News & Politics
Others
Weekly Score per Category
Hide top videos table
Time Interval
Weekly
Duration of all categories videos from 07-01-2008 to 24-06-2019
Views Count
of all categories videos from 07-01-2008 to 24-06-2019
Number of videos

Top 50 all categories videos from 07-01-2008 to 24-06-2019

Rank	Date	Video	Category	Views	Likes	Dislikes	Duration
1	17-06-2016	Baby Shark Dance | Sing and Dance! | Animal Songs | P...	Education, Science & Tech	3.838Bn	11.43M	4.388M	2min 16sec
2	19-11-2014	Mark Ronson - Uptown Funk (Official Video) ft. Bruno M...	Music	3.710Bn	13.51M	839.6k	4min 30sec
3	14-01-2015	Maroon 5 - Sugar	Music	3.055Bn	10.72M	487.0k	5min 1sec

We added transitions to accentuate the change when a filter is applied.

Hence, we decided to put an embedded YouTube video player directly in our web page: the user can browse the content of YouTube in our web page while listening to his favorite music. Finally, this player is completely draggable, resizable or minimizable.

2.6 Accessibility

After a discussion with our professor Laurent Vuillon, we decided to choose colors for our graph that are colorblind-friendly. Yet having to represent 7 different categories, it is very hard to find a palette that suits all the type of colorblind people. We tried our best and decided to include some highlighting with strong contrast on hover to help those people.

We hope you visit our visualization by clicking here

Acknowledgement

We first wish to thank Manoel Horta, PhD student at dlab EPFL, for providing us the dataset as well as many insightful feedbacks that helped us create the Visualization that you find on our homepage. We are also grateful to dlab EPFL, headed by Robert West, that let us use their infrastructure to handle the preprocessing. Finally, we thank Laurent Vuillon that mentored us during this project.

Peer assessment

As we were a group of two students, no one could take the opportunity to hide behind the rest of the group. Thus, we both contributed equally to this project.

Jonathan Kaeser

Jonathan spent most of his time on the visualizations for the first graph, as well as the embedded YouTube player. He oversaw the visual direction of this project.

Timoté Vaucher

Timoté took care of the rest of the visual interface. Besides that, he handled the data preprocessing, the communication with the dlab and the website integration on GitHub Pages.