Group number: 11
Members: Noah Britt, Matthew Re, Wei Hao
Repo: https://github.com/MRe6030/MRe6030.github.io
Page: https://mre6030.github.io/
Video: https://www.youtube.com/watch?v=3TaCkdItnU8

Overview and Motivation:

YouTube is the largest and most popular video-sharing platform and it continues expanding around the world. As a result, it is beneficial to investigate the trending videos and learn the characteristics of them. One way some videos can reach a broader range of viewers that would not normally engage with a video's particular content is through the YouTube trending tab. From Google's help center page, we can see that Trending will surface videos that "Are appealing to a wide range of viewers, are not misleading, clickbaity or sensational, and showcase a diversity of creators." Our dataset contains the top trending videos on YouTube in 2021 from ten different regions/countries. To better learn the characteristics of trending videos in this project, our goal is to examine this dataset and visualize the commonalities within the videos that could be found in YouTube's trending tab. Results from our eventual visualization can potentially be used to either examine if there is truth behind Google's help center page on what leads a video to becoming Trending (such as its claim of clickbaity or misleading videos do not appear on Trending) or show the patterns that the top trending videos share that Google is not explicitly stating, such as the possibility that maybe videos with more than 10 tags are frequently trending.

Related Work:

The stacked bar chart is inspired by the stacked bar chart taught in class. The stacked bar chart is useful since we can compare the magnitude of different groups by the stacked bar throughout another discrete attribute (e.g., year, video types). In our project, we use color as a channel to represent clickbait and non-clickbait. Green and red are used to distinguish videos with and without clickbait, this is also inspired by the visualization taught in class that uses color channels to compare the differences between groups. The interactions provided on the website are inspired by websites that have tabs with different contents.

Questions:

One of the main questions we wanted to look into was how clickbaited videos and channels that abuse clickbait tend to perform when compared to non-clickbait videos and channels. To do this, we wrote a quick algorithm that checks the videos title for common signs of clickbait. Namely, we looked for whether the title is in all caps (such as "MY SUMMER VACATION") or whether a title contained any of the substrings from a list of substrings we considered common in clickbait. Examples of these substrings are words like "believe" (ex. title: "you will not believe what happened"), "*" (ex. title: "my last birthday *emotional*), and "!!" (ex. title: "my first day of graduate school!!!!"). This isn't the best or most effective way to measure clickbait, as this algorithm resulted into a few false positives or negatives and if we had more time we could write an AI script that looks for red arrows in the thumbnail, but for the purposes of this project we found it to be sufficient for goals.

Data:

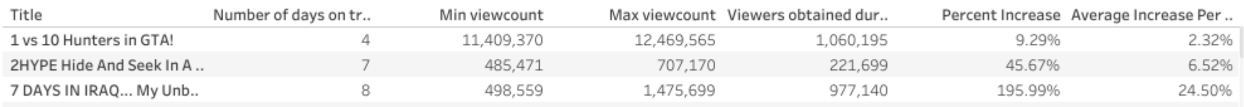
Our dataset is hosted on Kaggle.com and was created by user Jyot Makadiya. Within the data, the folder is ten CSV files, one for each region: BR, CA, DE, FR, GB, IN, JP, KR, MX, RU, and US. Each of these CSV files contains a total of 6000 rows and 16 columns. Our goal is to investigate YouTube's problems with clickbait. Clickbait has been a point of discussion in recent years due to YouTube viewers that clickbait results in them being misled or lied to about the content and YouTube creators believing that clickbait results in them being misled or lied to about the contents and YouTube creators believing that clickbait is required for a video to get discovered. We are not solely interested in proving or disproving negative aspects of content creation like clickbait, we are also interested in finding trends that could prove useful for content creator in improving the chances that their video gets put on trending.We expect that there will be a number of conclusions we can draw using the visualizations from this dataset.

By observation, very little data cleaning was needed. Technically, none was necessary as none of the data was corrupted or invalid/illogical. We did find that the dataset provides a "categoryID" field, but this field is just an integer value. While that makes for easy comparisons in Tableau or D3, it does not actually tell us what this integer value means. These YouTube categories, however, are known and are listed in different places, so we went in and replaced the integer values with their appropriate English categoryName. When we explored the dataset using Tableau, we found that some columns were not being recognized in their proper formats, but that was remedied using a Tableau calculation and did not require any modifications to the physical CSV.

Exploratory Data Analysis:

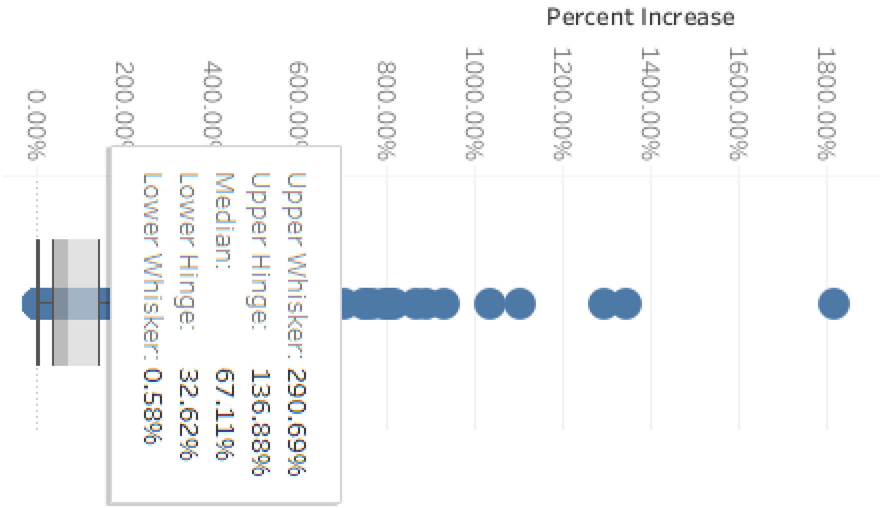
When we were attempting to explore the data for the first time, we put the dataset into Tableau and played around with the recommended visualizations for the data. We weren't given a whole lot of information about what the dataset contained, mostly just a few of the attributes, but we weren't told any specifics about the attributes, the range of dates for the videos, etc. The main visualizations that we used to initially explore the data was a standard table and box-and-whisker plot on some of the attributes.

Example of table:



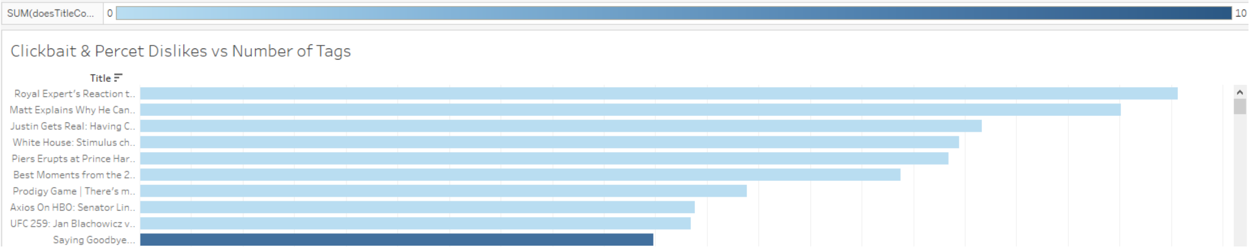
Title	Number of days on tr..	Min viewcount	Max viewcount	Viewers obtained dur..	Percent Increase	Average Increase Per ..
1 vs 10 Hunters in GTA!	4	11,409,370	12,469,565	1,060,195	9.29%	2.32%
2HYPE Hide And Seek In A ..	7	485,471	707,170	221,699	45.67%	6.52%
7 DAYS IN IRAQ... My Unb..	8	498,559	1,475,699	977,140	195.99%	24.50%

Example of Box-and-whisker plot:



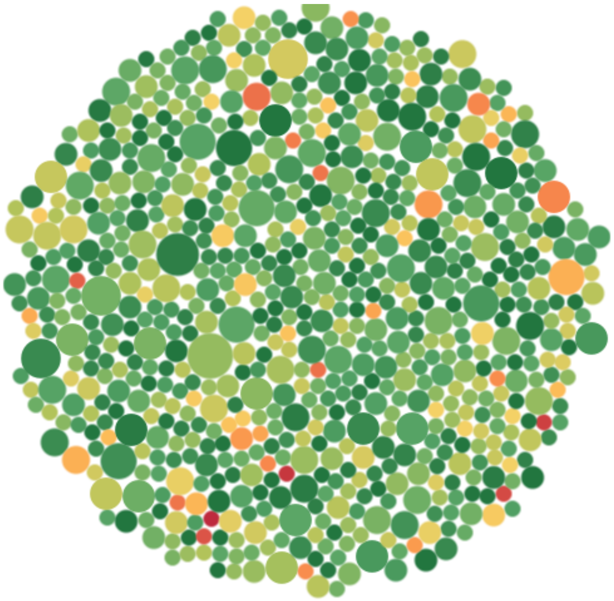
The goal of these visualizations wasn't to make anything interesting or see or see what we could and couldn't do with the data. Instead, these very simple visualizations allowed us to take a look at the data from a holistic standpoint and identify which attributes are related to each other.

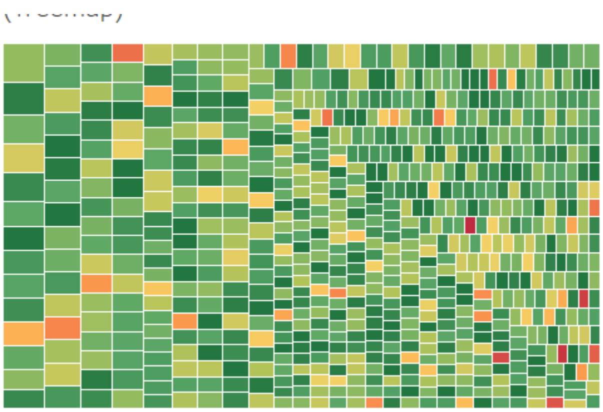
It was also through these initial visualizations that we made the discovery that the dataset does not contain 6000 different unique videos. While the dataset had 6000 rows (which is what made us believe it was 6000 videos) we noticed that any specific video was showing up with multiple different view counts and like/dislike numbers. As shown in the below image, the temporary measure that we were using for clickbait (which should have either been 1 or 0) was showing up in numbers as high as 10.


SUM(doesTitleCo...
0
10
Clickbait & Percet Dislikes vs Number of Tags
Title
Royal Expert's Reaction t..
Matt Explains Why He Can..
Justin Gets Real: Having C..
White House: Stimulus ch..
Piers Erupts at Prince Har..
Best Moments from the 2..
Prodigy Game | There's m..
Axios On HBO: Senator Lin..
UFC 259: Jan Blachowicz v..
Saying Goodbye...

Turns out, it was compounding all the scores for each day the video was trending. This is what led us to discover that instead of it being 6000 videos, it was about 1020 unique videos and Youtube had a total of 6000 total, but not unique, videos on trending.

Once we had a better understanding of our dataset overall, we began playing with what the data would look like in some of the different visualizations (bar vs line). An example of this is given below:





We decided to just put the data into the viable visualization styles and see what A) looked the best and B) gave us the most potential for interaction. Throughout the development of the final project, we often continued this process of "testing" a visualization idea in tableau first since it was easier to make the visualizations in tableau compared to d3.

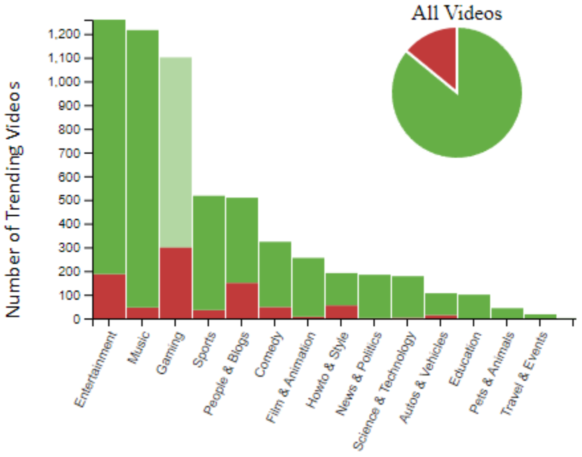
Design Evolution:

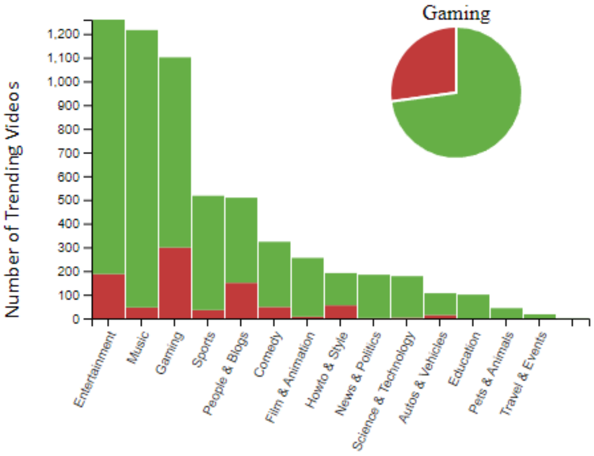
Our group took quite a while to get on the right track for our project, so it ended up taking several different iterations. Initially, we didn't design any of the visualizations to interact with one another. They were *intractable* but the idea was quite off the mark because the interaction was limited to the single visualization. For instance, we had used the treemap (in the above picture) in the initial design. It was "interactable" in that we could select what data we wanted to show in that treemap (either likes/dislikes or top X videos/channels) but that treemap didn't actually "talk" to any of the other visualizations. We made this initial design exclusively with Tableau, and while in hindsight that was a terrible idea, the intent was to see what we could work with in Tableau since Noah was the only group member with a comfortable amount of experience in d3 and javascript. The initial design also put way too much focus on the website to the point where the website was more interesting than the visualizations themselves were. This obviously was a problem, and we as a group met and decided we had to start again from scratch.

To get the final design up and running, we looked at the visualizations we had in tableau and took just their concept and moved it into d3.js. While we didn't like the lack of the visualizations talking to one another, and we didn't like the variables we had chosen to represent, we still did like the idea of using a treemap and bar chart as the two primary visualizations. Using what we had talked about in class, and using feedback from meetings with the professor, we decided that the interaction/talking between visualizations would come from how we make the data get smaller and smaller. The bar chart is our "starting point," and it represents the broadest representation of the data we had to work with, which was the category. Once we had that starting point, the only question really was which visualization type we would pick for the next two steps of the data (channels, then videos). Using the notes from class and more feedback, we landed on treemap for channels, and bubbles for videos. With that being said, the stack plot was also created since we wanted a way to look at the same attributes (the number of trending videos for a particular channel across multiple days). We realized that this wasn't really needed to fulfill the “start from broad categories to specific videos” idea we were going for, but since we created it, and it still served a purpose, we decided to keep it in the final design so long as we had room for it, which we ended up having.

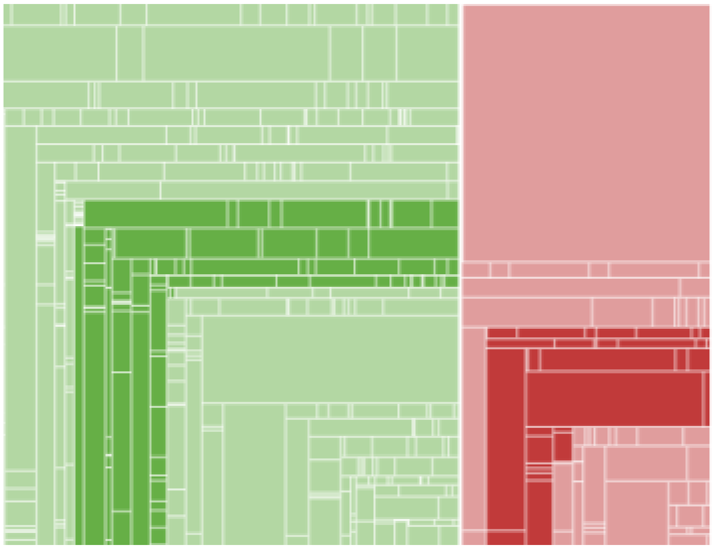
Implementation:

The first visualization, the bar chart, serves as our “starting point” for the project. While not completely required to start by interacting with the bar chart, the bar chart is what visualizes our broadest grouping of the data- categories.

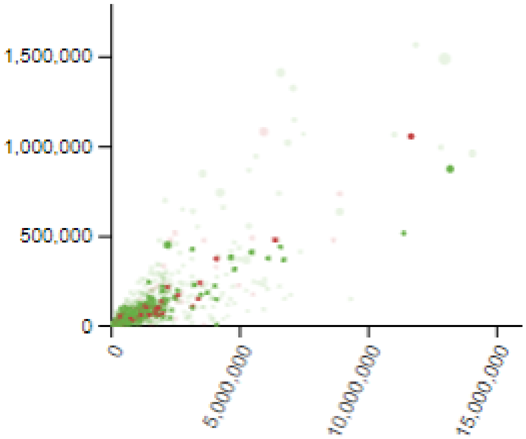




As shown above, by clicking on any of the bars, you can begin highlighting the data in the treemap and the bubble chart.

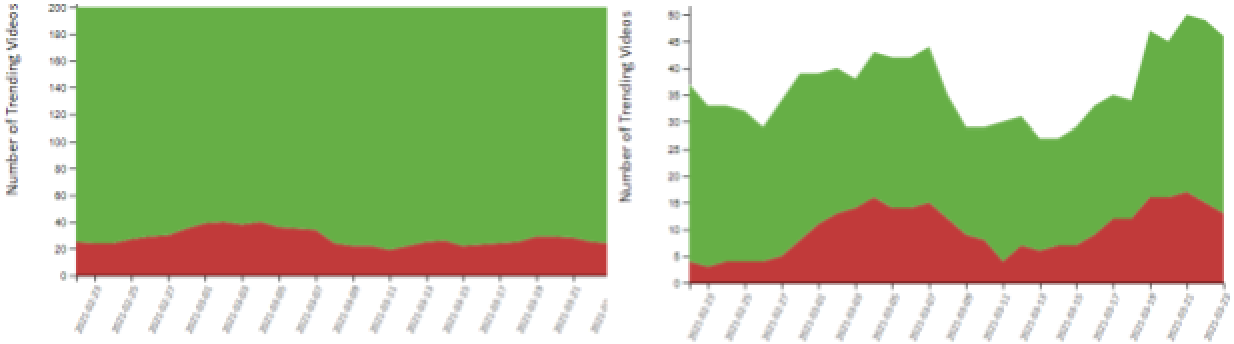


Treemap highlighting Channels with videos in the "Gaming" category.



Highlighted bubbles are videos in the gaming category

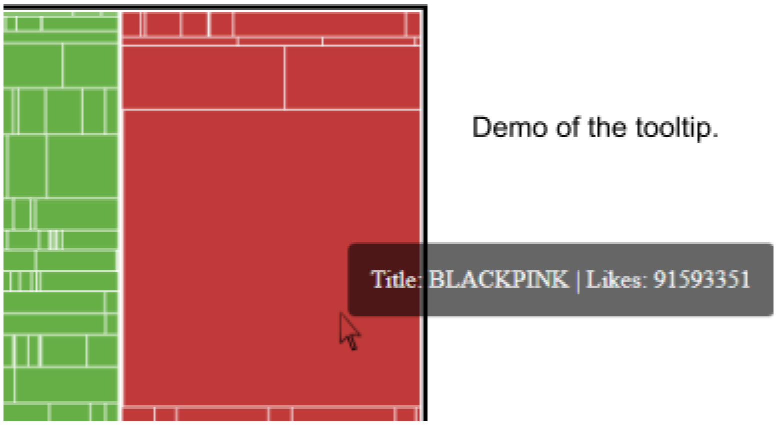
For the stackplot, we played around with the idea of highlighting that data too instead of filtering it, but we did not have any reasonable means to highlight the data based on the category since the highlighting would have to be a complete change in color. Since color is the only channel for the visualization, and it's being used for clickbait, we did not want to impede on distinction between clickbait and non-clickbait so we went with just filtering the data instead.



Left: default stackplot | Right: stackplot filtered for just gaming videos

Given that the bar chart changes the above three visualizations in these ways, it’s why we decided on it being our starting point. However, if you do not want to view the data based on any individual category, all the other interaction elements of the tables are still completely functional.

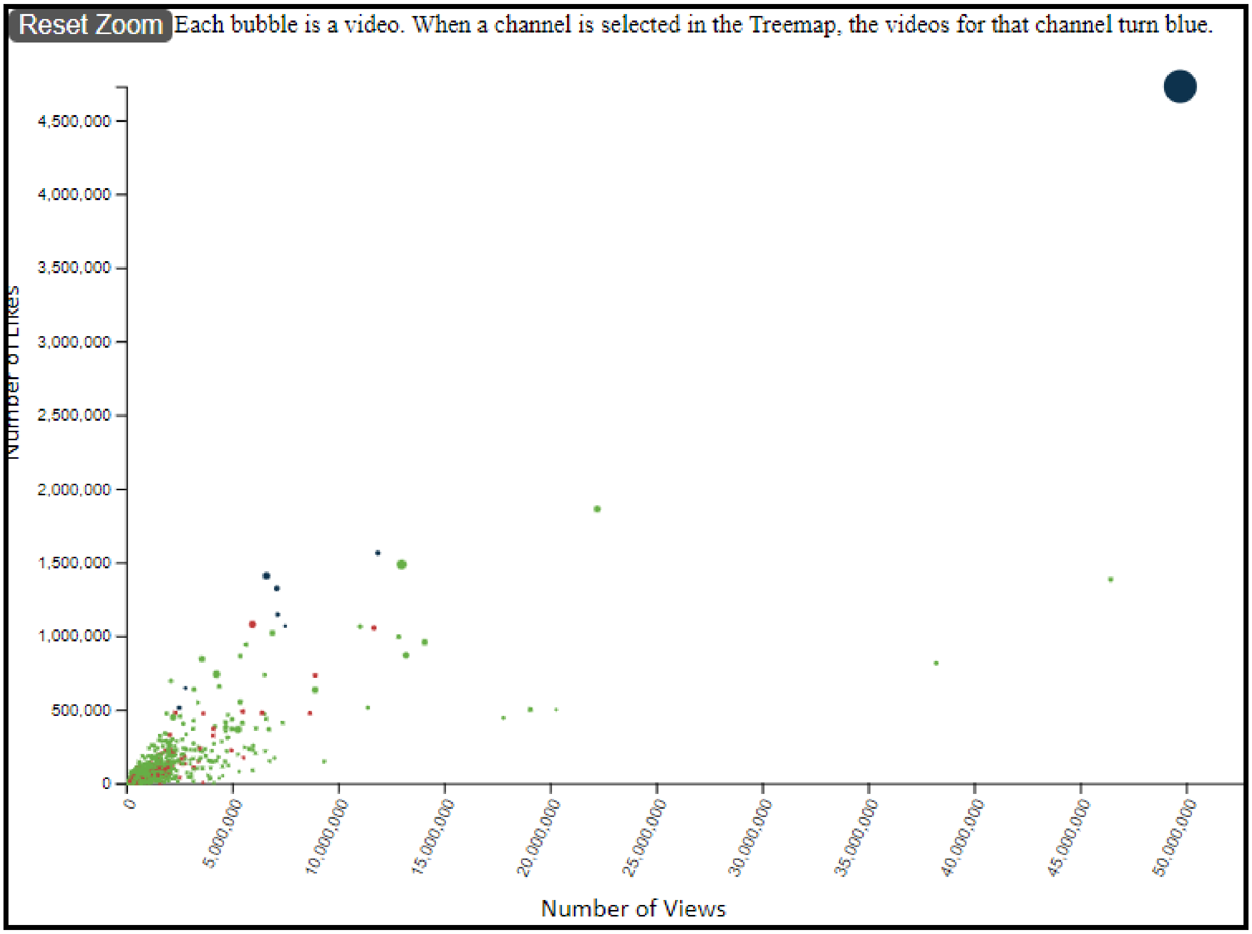
For the treemap, we allow users to pan and zoom given how small some of the squares are. Beyond this, we also made it so that the stackplot and bubble chart are also influenced by interacting with the treemap. Normally, when the user mouseovers any of the squares on the treemap, they’ll see the channel that square represents and how many likes that channel has across all of their videos.


Title: BLACKPINK | Likes: 91593351

Demo of the tooltip.

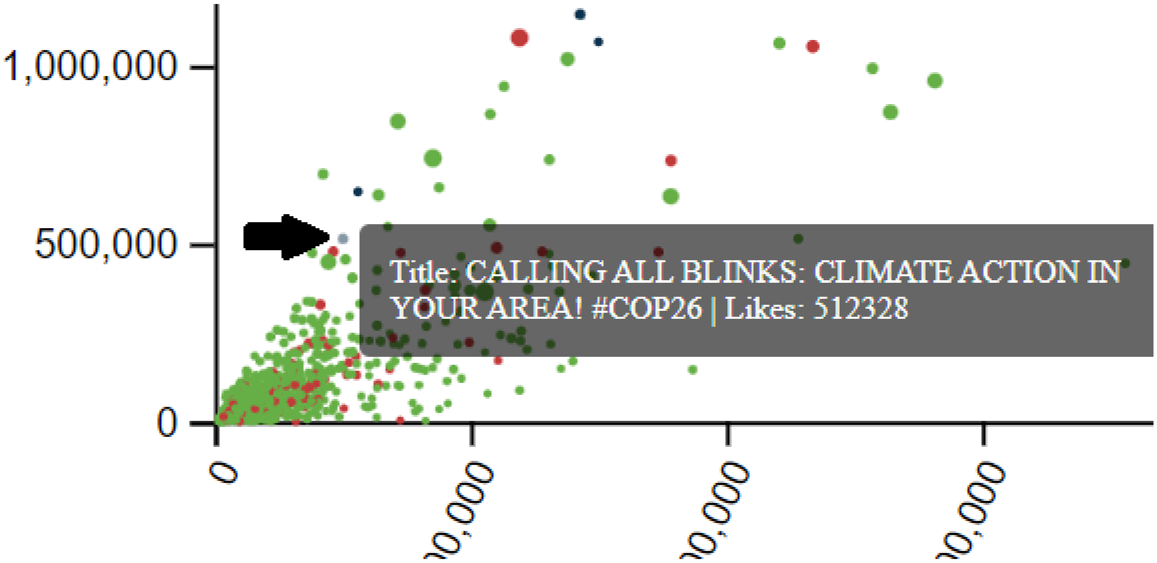
We wanted to be able to create a means by which we could visualize whether we can see any noticeable patterns of clickbait and how many likes a channel receives across their trending video, and we believe the treemap serves this function well. On the other hand, if we wanted to look at individual videos instead of channels, we have to look at the bubble chart.

The bubble chart serves to put a channel’s likes (from the treemap) into a bit of perspective. As shown in the image above, the channel BLACKPINK has the most amount of likes by a very large margin. However, the square is red, meaning that the channel, according to our algorithm, is guilty of potentially clickbaiting their videos. Does this mean that channels that clickbait are destined to get more likes? The bubble chart helps us answer, or at least put some doubt, on that question. When I click on the BLACKPINK square in my treemap, the bubble chart will turn all of BLACKPINK’s videos dark blue:



Each bubble is a video. When a channel is selected in the Treemap, the videos for that channel turn blue.

As we can see by the chart, over half of BLACKPINK's likes come from a singular video (the one in the top right corner). While not to discredit the channel's accomplishment in a successful video relative to their other ones, because that one video is an outlier in the general trends of our data, we get a rather skewed perception of what the data is trying to say. The reason the channel was flagged as potentially using clickbait was because of one of their videos:



Title: CALLING ALL BLINKS: CLIMATE ACTION IN YOUR AREA! #COP26 | Likes: 512328

As we can see, this video has a title in all caps (which caused our algorithm to believe it to be clickbait) and it is actually the worst performing video that channel has in terms of both likes and views. Now, just like how we can't jump to conclusions when we asked previously "Does this mean that channels that clickbait are destined to get more likes?" we also cannot just assume that the reason this video did relatively poorly is because it used components of clickbait. The reason we bring attention to this is because it demonstrates that to get a holistic view of the data, we have to combine the strengths and information of multiple visualizations, namely in this case, the treemap and the bubble chart.

This BLACKPINK example shows our intent for how we can use the interactive components of our visualizations to learn something from the data that the numbers alone can't easily tell us. Not only could we do the same process with other channels and videos, but we can do it with entire categories using the highlighting capabilities of the bar chart.

Evaluation:

The biggest thing that we learned about the data is that Youtube trending is a page created by an algorithm that nobody seems to understand. When looking at the visualizations, there does not seem to be any particular, or at least obvious, trends or patterns that indicate a definitive way to get a video on trending. over 90% of the videos on the trending page have a total view count (which means the view count including the days it was actively trending) below the 7.5m views mark. Another large percentage of the views are below 500,000 likes. However, since there are still a non-negligible number of videos outside of these ranges, we can't claim that YT trending exists to help give that push a video needs to take off. On the topic of clickbait, we didn't see any particular or obvious patterns in that data either. While we did learn that a large majority (over 70%) of the videos are not guilty of clickbait according to our algorithm, which is nice because it tells us that you don't *need* to clickbait a video to get it trending, we did not find any proof that states clickbait videos do definitively worse (or better) than non-clickbaity videos. In terms of just one-off points we learned about, we found that "Gaming" is the most guilty of clickbaiting their videos and that there were several categories (albeit with a lower sample size) that had very low or no traces of clickbait at all. The way that we came to conclusions (or the conclusion that we can't draw conclusions) was by following that pattern explained in the above implementation section.

We believe that our visualizations worked well, but we also understand that they are far from perfect and can always be improved. For instance, as mentioned in the Questions section of this book, we would have liked to truly refine our clickbait algorithm to remove the chances of false positives and negatives, as well as include the ability to scan a video's thumbnail to detect signs of clickbait there (such as red arrows or shocked faces). However, that requires an amount of skill in AI and machine learning that is way beyond the scope of this course, but in a perfect world it could and should be done. The other ways we think we could have improved our overall design was to include a few more of the attributes we had access to. We started running out of space (and visualization types to pick from without being repetitive), but we still had a few more attributes, such as comment count, whether likes and comments were enabled or disabled, and the entire video's description to work with. We had a few ideas on how we could have implemented more visualizations that utilized these unused attributes, but due to time constraints since we started from scratch after the feedback from the initial prototype and the aforementioned lack of screen space, we decided against including any visualizations for these attributes. We had sketched out some ideas such as using a gauge chart for these attributes or individual bar charts, but we couldn't settle on a design that seemed "worth it" and didn't just clutter the website. Again, in a perfect world, we'd have more screen space, more time, and frankly more experience in d3. These would allow us to add these visualizations, since we do believe the more data we can show the better so long as that extra data is shown in a way that is easy to comprehend and it fits with the overall design.