Process Book
Website: https://exploring-youtube.github.io/
Sophie Fuller
sfuller2@dons.usfca.edu
UID: 20352370

Jennifer Cruz-Hernandez
jjcruzhernandez@dons.usfca.edu
UID: 20328721

Exploring YouTube Data: Examining the Effects of Policy Change

Background and Motivation.

We are interested in the correlation between YouTube's policy change and the videos of YouTubers.. The recent shooting at YouTube was our main motivation for choosing this project. The shooter, Nadim Aghdam, claimed that YouTube began censoring her videos after their changed policies in February following a video posted by YouTuber, Logan Paul, depicting a dead body in Japan's "Suicide Forest". YouTube began demonetizing videos that they deemed inappropriate for all age groups, affecting the amount of money YouTubers were making from their new and previously posted videos. This change also influenced their subscriber and view count for their pages, further affecting the amount of money they could earn from posting videos on YouTube. The shooter claimed that these policies affected her viewership and as a result, attacked the YouTube headquarters. We are going to examine if there is any truth in her claims by analyzing the views of videos on the trending page before and after the policy change as well as run sentiment analysis on the comments to see if there is a noticeable difference in viewership.

Project Objectives:

1) Primary Questions:
    a) Is there a significant change in the types of videos that make the trending page following the policy changed YouTube implemented?
    b) Have videos from some YouTubers received less views or less interaction following these changes?
    c) Have the types of interactions (ie. likes and dislikes, comments) been affected by the policy changes?
2) We would like to use our analysis and visualizations to draw conclusions regarding the shooter's claim that YouTube's changing policy have affected the viewership of certain pages and videos. While her specific videos may not be in our dataset, we can look at other trending videos before and after the policy change to see if there is any widespread truth to her claims.
3) Benefits:
    a) Draw conclusions on the effect of the policy changed on viewership. Was there a substantial effect on some YouTubers' channels or were there no or negligible effects?

Data:

Data will be collected from a Kaggle Data Set as well as the YouTube API. The Kaggle dataset contains 57 MB worth of data. This dataset contains the data of YouTube videos that make the trending page daily. Each video has it's unique video id. The csv files contain comments and video statistics. This data was collected based on video uploaded a year ago. Since then YouTube has implemented new policies (February 2018) which YouTubers claim have negatively affected their views, so we will use the YouTube API to collect current data from trending videos. Based on the YouTube API guide the data will be collected into a JSON file. This dataset will contain similar, if not more, of the same categories that the Kaggle dataset contains.

Data Sources:

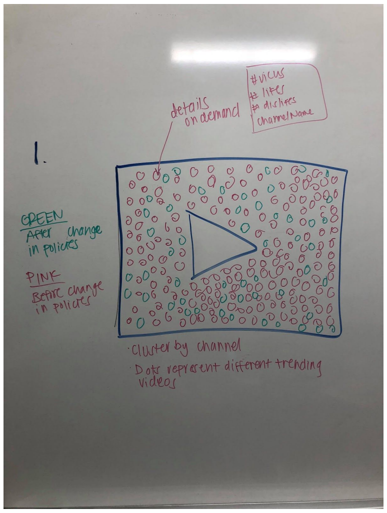
https://developers.google.com/youtube/v3/getting-started

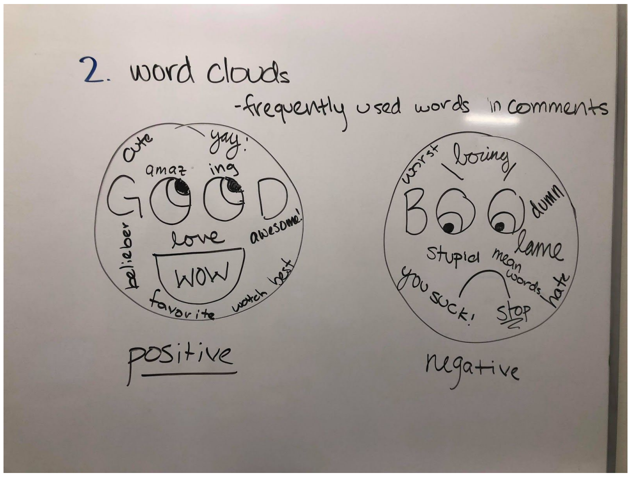
Data Processing:

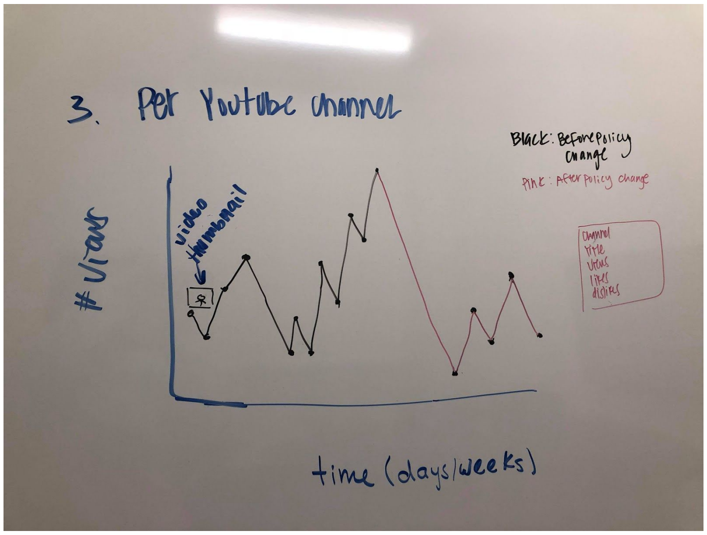
Yes. The Kaggle dataset is nicely organized already, the one thing that will need cleaning are the comments for videos. Since we are planning to run a sentiment analysis on the comments using Naive Bayes Classifier, we will be using R to clean up the comments data. Here we will be filtering out any hashtags, links, tagged names, numbers and unnecessary spaces. We will also focus on the likes and dislikes of videos. This will be used to determine if there is any relationship between the sentiment analysis of a video and the number of like/dislikes the video receives. The YouTube API dataset will be collected into a JSON which we will convert into a csv to keep it consistent with the Kaggle dataset. The same cleaning process will be done with the comments. We will also filter out any columns/categories that we will not be using in our visualisation. With the YouTube API data we'll focus on the number of views a certain video gets as well as the total number of dislike/likes a channel gets, which ultimately determines how much attention the channel's video is getting. So we will get data from the channels that are being used in the Kaggle dataset to get more accurate results on whether the new YouTube policies drastically affect how much attention a channel's video receives. We will be using Python to retrieve the data from the YouTube API, but use R to clean the data.

Visualization Design.

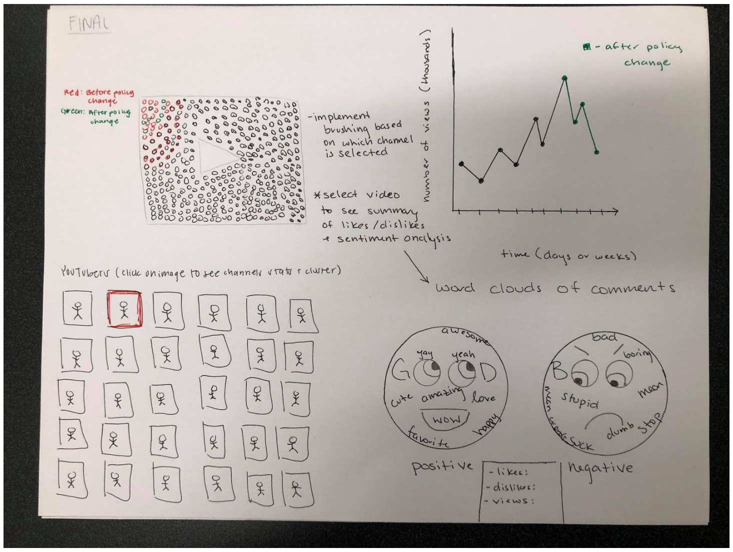
Prototypes:



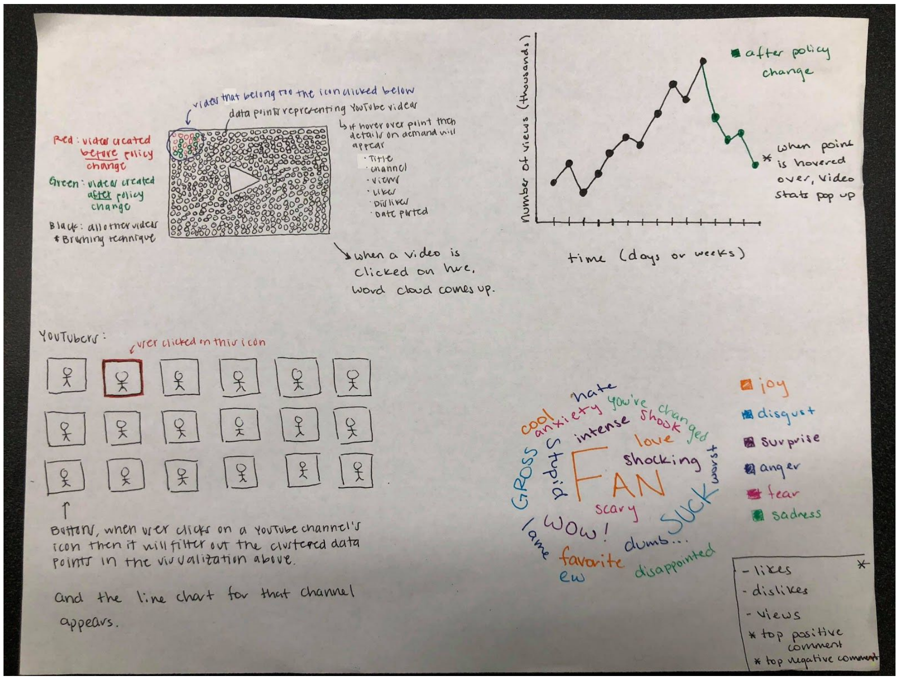




First Iteration:


FINAL
Red: Before policy change
Green: After policy change
-implement brushing based on which channel is selected
*select video to see summary of likes/dislikes + sentiment analysis
number of views (thousands)
■ - after policy change
time (days or weeks)
YouTubers (click on image to see channels stats + cluster)
word clouds of comments
awesome
yay
yeah
GOOD
cute
amazing
love
wow
favorite
happy
positive
bad
boring
BAD
mean
stupid
mean words
suck
dumb
stop
negative
- likes:
- dislikes:
- views:

Second Iteration:



videos that belong to the icon clicked below

data points representing YouTube videos

↳ if hover over point then details on demand will appear
· Title
· channel
· views
· likes
· Dislikes
· Date posted

Red: videos created before policy change

Green: videos created after policy change

Black: all other videos

* Brushing technique

↘ when a video is clicked on here, word cloud comes up.

user clicked on this icon

Buttons, when user clicks on a YouTube channel's icon then it will filter out the clustered data points in the visualization above.

and the line chart for that channel appears.

* when point is hovered over, video stats pop up

- likes
- dislikes
- views
* top positive comment
* top negative comment

Must-Have Features:

1. Storytelling with Data:
    a. we want to explore the implementation of YouTube's policies and the direct effects on the YouTuber community.
2. Details on Demand:
    a. a user should be able to dive deeper into a data point to get more details about the viewership of that video and the YouTuber who created it.

Optional Features:

1. Brushing:
    a. a user should be able to highlight a section of the visualization by date of upload to examine effects of policy changes at a certain date. Other points should still be viewable but muted.
2. It would be nice to have a little gif of the video play when a point is clicked but that may be outside the scope of our ability/this project. (May also be distracting from the actual visualization)

Project Schedule:

A. Week of 4/9 (Preparation for Alpha Release):
    a. Have code cleaned and processed.

b. Run the sentiment analysis and begin classifying comments under TBD classifiers.

c. Finalize plan for which data visualizations to use.

B. Week of 4/16 (Preparation for Beta Release):

a. Gather examples from bl.ocks and other resources on visualization similar to the one we want to implement.

b. Develop a prototype of the visualization (content is viewable and the direction is clear)

c. Implement some interactivity (doesn't have to be complete).

C. Week of 4/23 (Finalization of Visualization)

a. Finish the visualization and fully implement interactivity.

b. Begin working on presentation and how we want to display our findings to the class.

c. Work on project repository and make sure all code is commented and easy to read and navigate.

D. Week of 4/30 (Finish the Presentation)

a. Finish the presentation

b. Finalize repository and website.

c. Practice. Practice. Practice.

Related Work and Inspiration:

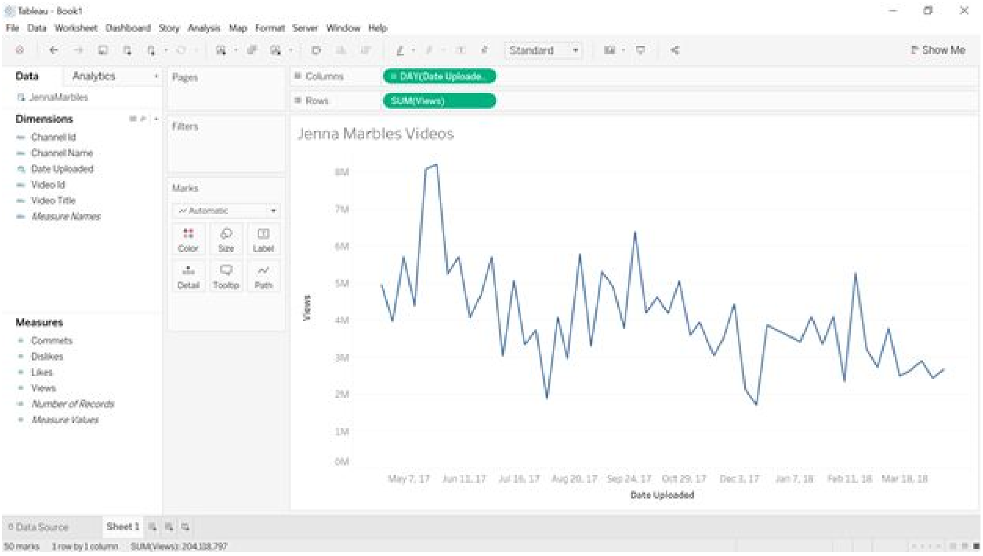
- Shirley Wu's Tweety Visualization: http://sxywu.com/tweety/

Process Book (Week 1)

Cleaning Data:

- We realized that we would not be able to use the Kaggle dataset anymore as it does not have the data we actually needed. The Kaggle dataset does not separate videos by channel which we need for clustering and tracks them over a period over a few weeks of adding new videos. So we turned our attention to the YouTube API and getting all of our data through that source.
- Because of this, we had to change our plan slightly. Instead of looking at videos on the trending page, we selected a few genres of videos including gaming, vlogging, DIY, and travel and chose specific popular channels from each of those categories. From there, the plan is the same - run sentiment analysis and look at views of videos over time focusing on a change (if one exists) after the policy change in February.
- We decided to gather video information from the following channels (Channel Name, ID)
    - Personal Vloggers:
        - Jake Paul (JakePaulProductions)
        - Logan Paul (UCG8rbF3g2AMX70yOd8vqIZg)
        - Jenna Marbles (JennaMarbles)

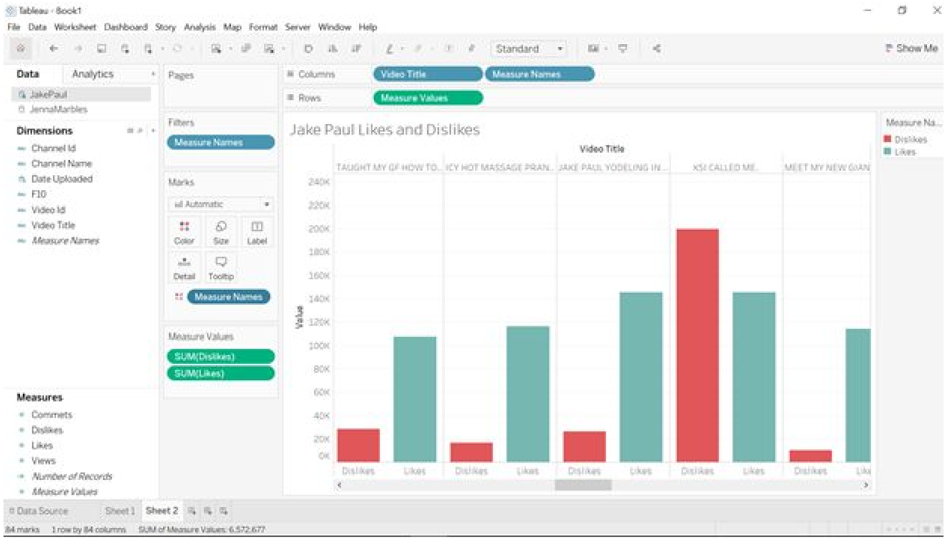
        - Shane Dawson (shane)
        - Liza Koshy (UCxSz6JVYmzVhtkraHWZC7HQ)
        - Superwoman (IISuperwomanII)
        - David Dobrik (UCmh5gdwCx6lN7gEC20leNVA)
        - Kian and JC (KianAndJc)
    - Family Vloggers:
        - Ace Family (UCWwWOFsW68TqXE-HZLC3WIA)
        - RomanAtwoodVlogs (RomanAtwoodVlogs)
        - Shaytards (SHAYTARDS)
        - Jason Nash Family(UCWcVikU5Sv-AqMtjV-BVsLQ)
    - Gamers:
        - KYR sp33dy (KYRSP33DY)
        - Fitz (UCtb8P4rf_1n8KS8eZk_lNNw)
        - Seananners (SeaNanners)
        - Vanoss (VanossGaming)
        - Hutch (shaun0728)
    - Makeup Gurus:
        - Jeffree Star (jeffreestar)
        - MannyMUA (MannyMua733)
        - James Charles (UCucot-Zp428OwkyRm2I7v2Q)
        - Jaclyn Hill (Jaclynhill1)
    - Other:
        - Sara Beauty Corner (SaraBeautyCorner)
        - Laura DIY (LaurDIY)
        - Bethany Mota (Macbarbie07)
        - Safiya Nygaard (UCbAwSkqJ1W_Eg7wr3cp5BUA)
        - BuzzFeed(BuzzFeedVideo)
- After Jen scraped the API and cleaned the data, we were able to make some preliminary visualizations in Tableau.



Columns DAY(Date Uploade..

Rows SUM(Views)

50 marks 1 row by 1 column SUM(Views): 204,108,797

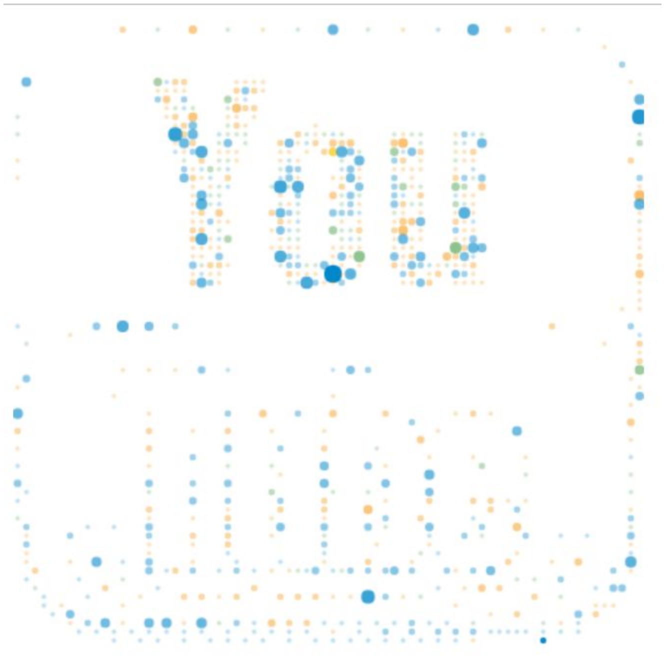


- The data is capable of making clean visualizations in Tableau so now we can move on to making customized visualizations using D3.

Week 2:

Starting the Visualizations

- Using Shirley Wu's tweety visualization as inspiration proved to be much harder than we thought (I sense a theme here). First of all, she doesn't use D3 for any part of it except for the fisheye. So that's out. We are still going to use a similar idea but use different resources.
- Luckily for us, Shirley circa 2015 comes to the rescue. She has image processing bl.ocks on how to take an image and link them to tweets. http://bl.ocks.org/sxywu/a3b6a4715c14f5d945b4
    - In this bl.ock, she takes her twitter profile picture, uses another bl.ock to downscale it then converts those pixels to colors and links them to a specific tweet. It's exactly what we need.
- We took a square YouTube logo from Google Images and ran it through the same bl.ock she did to generate a JSON array. http://blockbuilder.org/enjalot/a74e8599c359fac3f829
    - We had to try this several times. The bl.ock is very specific for what types of files it will do this properly for. The file must be a JPG and to work with Shirley's code, it must be an exact square image. Anything else will not work.
    - Using Shirley's code and our image, we generated the following:

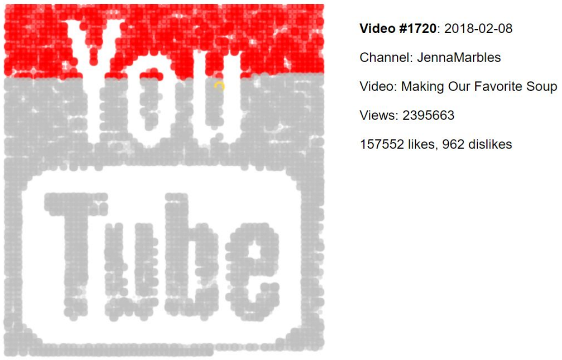


    - Each pixel, when hovered over, is a tweet and we can easily edit this to work for a Youtube video.
- From this, we are going to make a pop up of the stats for the video that is hovered over and this will be our demo for the beta release.
- Next steps: make the thumbnails and link them to line charts. Combine the thumbnails with the image to implement brushing. Link each point to a word cloud of the sentiment analysis (if we have time).

Week 3:

Continuing to Build:

- After our Beta Release, we have most of our visualizations running. Our YouTube logo has been updated and looks much better. Jen found a better image that allowed for more videos to be displayed which bulks up the visualization.



Video #1720: 2018-02-08

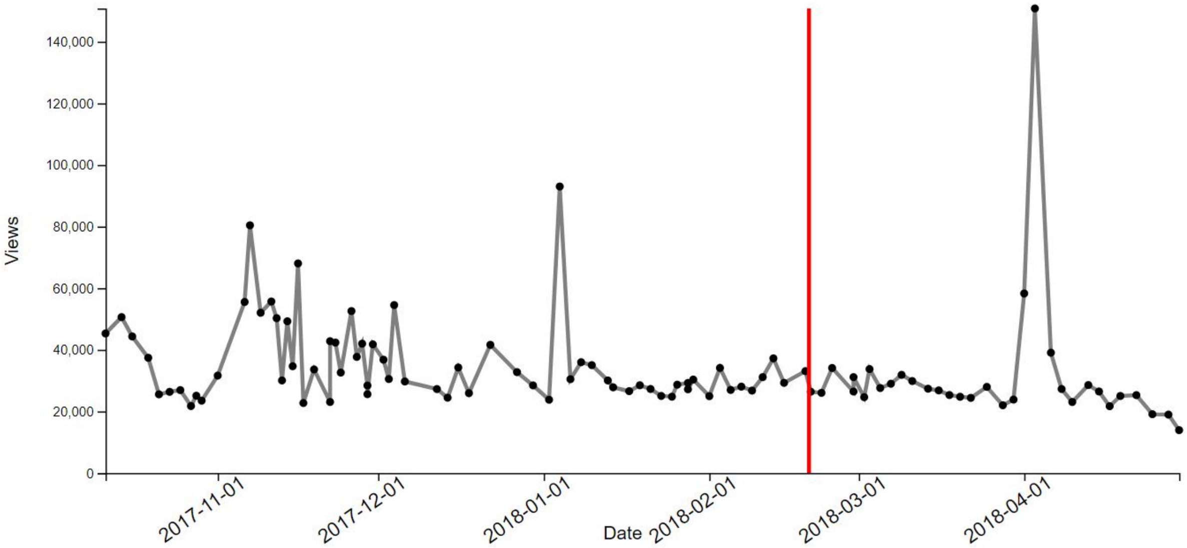
Channel: JennaMarbles

Video: Making Our Favorite Soup

Views: 2395663

157552 likes, 962 dislikes

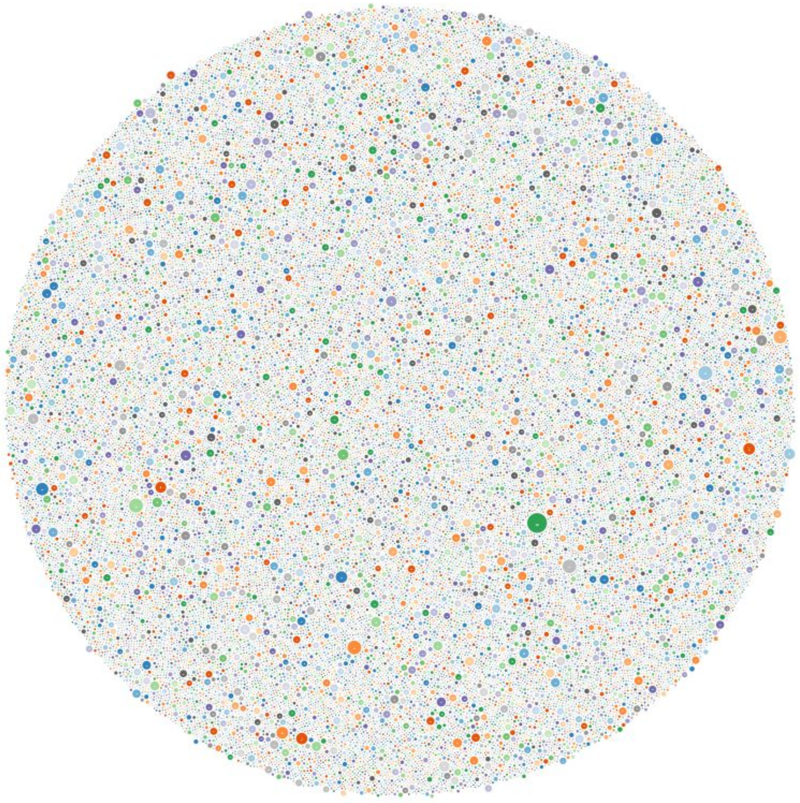
- From there, we are still working on linking our visualizations. We ran into an issue with the clickable buttons in that they were printing multiple times on top of each other making the text not able to show just once. Although when you logged it into the console, it did print the correct Channel ID. We had to change the way we approached this problem and now we have circular clickable buttons that we are working on linking to the line charts. We are almost there however have run into the problem where, once clicked, the line chart doesn't then disappear once another one is clicked. We can probably put some sort of "active" boolean variable on it (like when we did the line charts on assignment 5) and that might fix the problem.
- Additionally, we are working on word clouds. We are no longer going to do sentiment analysis as the code that Jen produced for another assignment produces a bar chart of how much of each emotion each tweet (in her case) displays. We would need one that classifies the YouTube comment as a certain emotion. For our purposes, the sentiment analysis would just be too much to implement for a visualization assignment. Instead, we are going to focus on the frequency of the words in the comments and choose an appropriate color scheme that assigns color based on frequency. Jason Davies (https://github.com/jasondavies/d3-cloud) has developed an algorithm to create a word cloud and, because it looks really complicated, we are going to edit this code to fit our purposes instead of attempting to write our own.
- Line charts proved to be a little challenging as well. Linking them to the buttons so that the line chart updates when a new YouTuber is clicked was a hard problem. Once we got that working, Alark had us add a bar on the graph that demarcated the date of the policy change.



Week 4:

Final Presentation Prep

- After many, many failed attempts at a word cloud (it's a lot harder than we expected it to be - we're still on theme), we finally gave up and headed into bubble chart territory. The idea was to encode the area of the bubbles based on the frequency of the word used in the comments. The larger the radius, the more that word was used in the comments for that video. Some problems we are going to run into: these datasets are ginormous. We processed the comment data in Python, creating a dictionary of unique video ideas where the value was a dictionary of unique words found in the comments of that video. The value of the word dictionary is just the frequency of that word. From there, we found a Mike Bostock bl.ock that implemented a simple bubble chart https://bl.ocks.org/mbostock/4063269 and used this as the base code for the chart. Another issue we quickly ran into was that our data is not hierarchical and literally every single example online is hierarchical. I wasn't quite sure how to solve this but found a (tricky) solution that doesn't make a whole lot of sense but fixed the problem. Somehow it was able to put our csv into classes that functions could process as hierarchical data. Bubble chart of 200 comments each from 100 videos from Jake Paul's channel below:



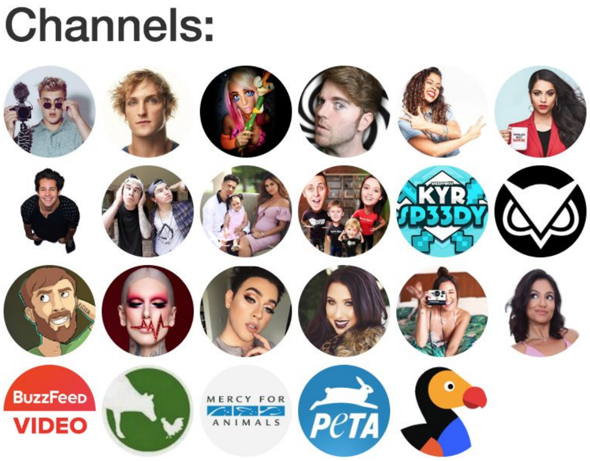
- It's not super clear, but there is text in those little bubbles (the word that it represents) and if hovered over, will give the word and the count. We haven't separated it by video yet, that is coming up.
- The next task of the week was formatting the dashboard to all work together and fit on one page. We had to resize our buttons to all fit on a single line. Preferably, we would have them in multiple small rows off to the side and next to the YouTube logo but there are two problems with that: "padding" in a category div element doesn't do what you think it would do and the Canvas that we put our YouTube logo on seems to be a immovable object.

- Luckily, Jen figured out how to make the YouTube logo move and we were able to put the buttons in a single row. The logo will go next to the buttons and the line chart and bubble chart will sit beneath them, updating as a new channel/video is clicked.
- Finally, the last thing we did this week was change which channels we were examining. As the YouTube shooter was an animal rights and vegan activist, we included some similar channels such as Mercy for Animals, PETA, The Dodo, and Farm Sanctuary onto our list of channels to extract information from. Additionally, we had to drop a few of our original channels for issues with their data.

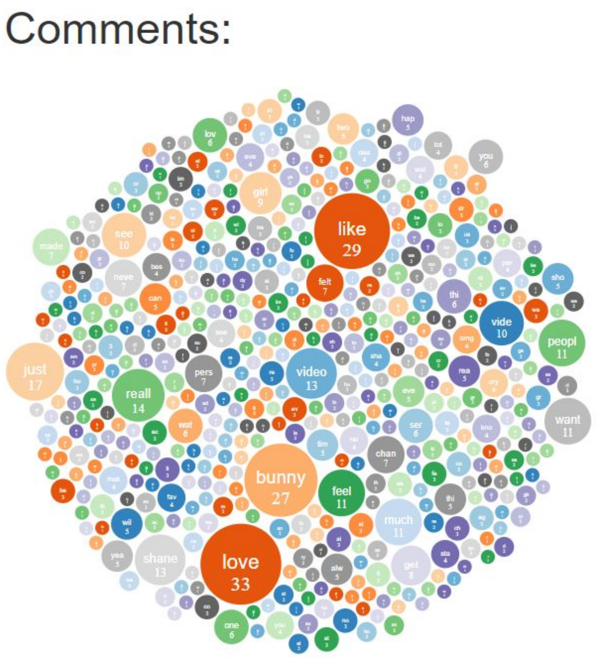
Week 5:

Final tweaks:

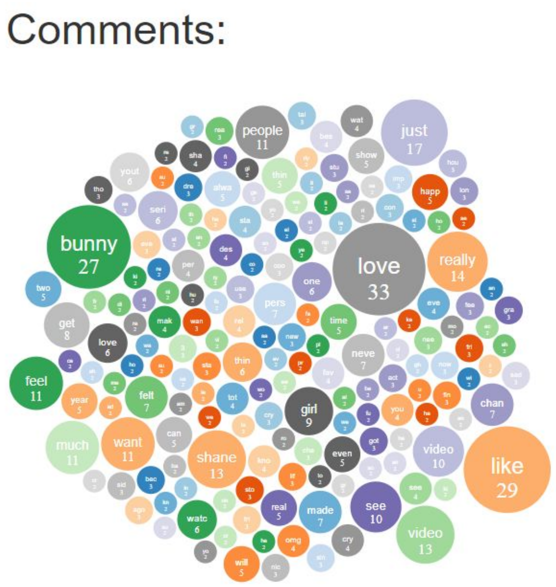
- We had to change, once more the list of YouTubers included in our data. The final list is: Jake Paul, Logan Paul, Jenna Marbles, Shane Dawson, Liza Kochy, Superwoman, David Dobrik, Kian and JC, the Ace Family, Hutch, Jeffree Starr, Vannoss Gaming, KYR SP33dy, MannyMUA, Jacklyn Hill, Laura DIY, Bethany Mota, Buzzfeed, Farm Sanctuary, Mercy for Animals, PETA, The Dodo, and Roman Atwood Vlogs.



- We were able to remove all the stopwords from the bubble chart so it looks much better. I'm still not very happy with the colors but that can still be fixed. Right now a typical bubble chart looks like this:

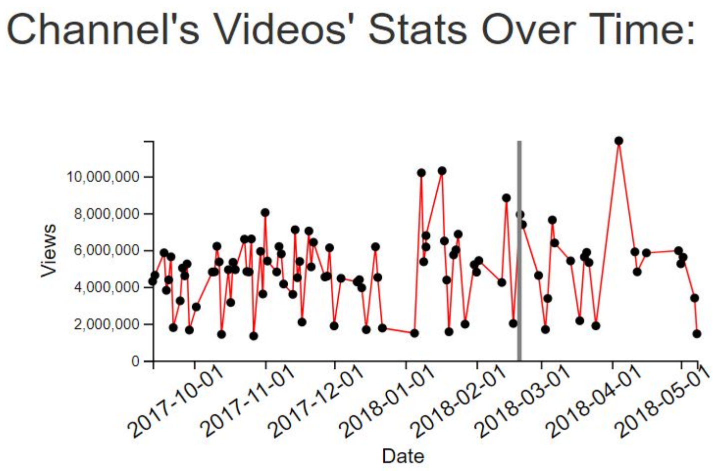

Comments:

- ○ Each word can be hovered over and the full word will appear as some bubbles are too small to read. The next step is to no longer include words with a frequency of one as you can barely see them and they're obviously less significant than other words and take up more space.

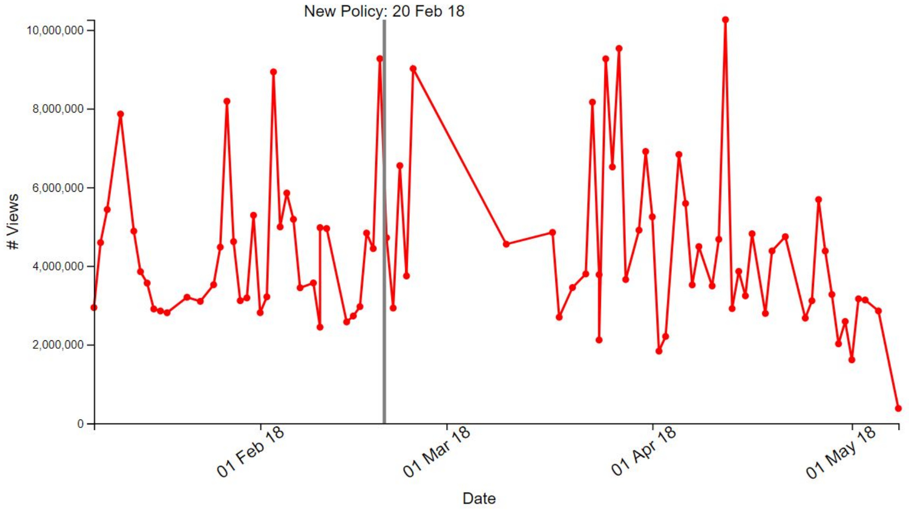

Comments:

- ○ Even though it's not perfect, it looks much better without the count 1 words taking up a lot of space.

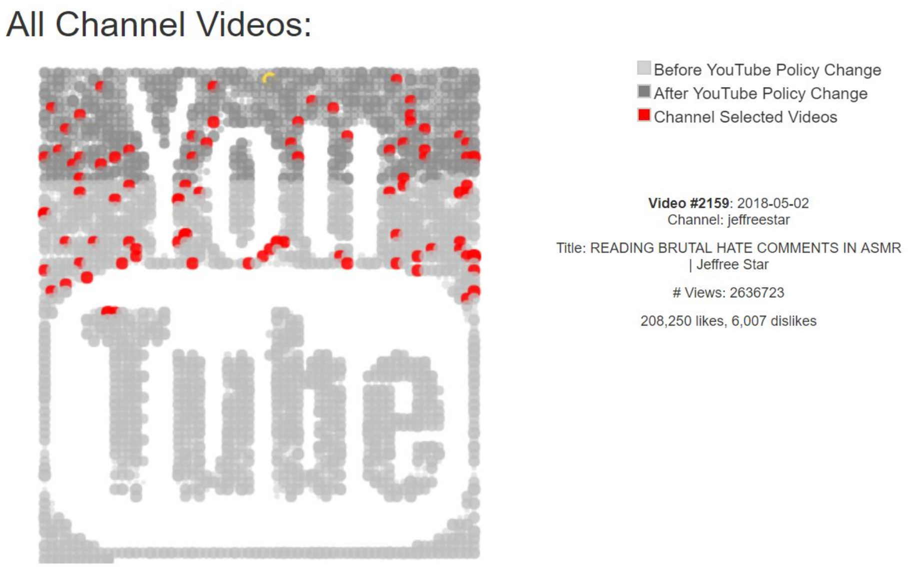
- The other issue we are working to solve is how to make our line chart look better. Right now, it is rather small and cramped and clunky looking and doesn't look great on the website:



- To fix it, we are making the line and all the points red, maintaining color consistency with the YouTube logo legend, and making it much bigger. We experimented with a curved line but as we have defined data points for almost every day (sometimes more than once a day) the sharp line looked better. We left in the dark grey vertical line denoting the policy change, another color choice to be consistent with the logo above. Now it looks like this which is much better:



- The final iteration of the YouTube logo with legend included:



All Channel Videos:

⬜ Before YouTube Policy Change
⬛ After YouTube Policy Change
🟥 Channel Selected Videos

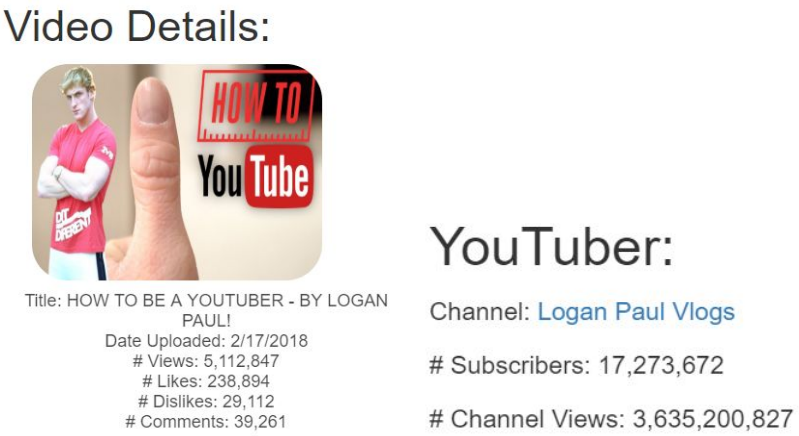
Video #2159: 2018-05-02
Channel: jeffreestar

Title: READING BRUTAL HATE COMMENTS IN ASMR | Jeffree Star

Views: 2636723

208,250 likes, 6,007 dislikes

- The final things we implemented were being able to visit a YouTuber's channel by selecting their Channel ID and view a video by clicking on the thumbnail next to the line chart:



Video Details:

Title: HOW TO BE A YOUTUBER - BY LOGAN PAUL!
Date Uploaded: 2/17/2018
Views: 5,112,847
Likes: 238,894
Dislikes: 29,112
Comments: 39,261

YouTuber:

Channel: Logan Paul Vlogs

Subscribers: 17,273,672

Channel Views: 3,635,200,827

Conclusions:

Overall, our data showed us that there was no significant change in any YouTuber's account following the policy change in February of this year. The only person who was affected a lot by it that we noticed was Logan Paul, the reason behind the change. Even then, his views are only about 1 million fewer on average, a very small number for him. There is no significant difference in likes, views, comments, or dislikes since the change for any of the YouTubers we looked at.

Outside of just the conclusions on our data, we learned a lot about JavaScript and D3. Jen learned Canvas to be able to implement the YouTube logo successfully and we both had to become adept at switching between v3 and v4 of D3 as a lot of the sample code we used was written in v3. We ran into lots of little errors along the way but we accomplished everything we wanted to implement and more.