Final Project

Cao Yixiang SID: 4246043356

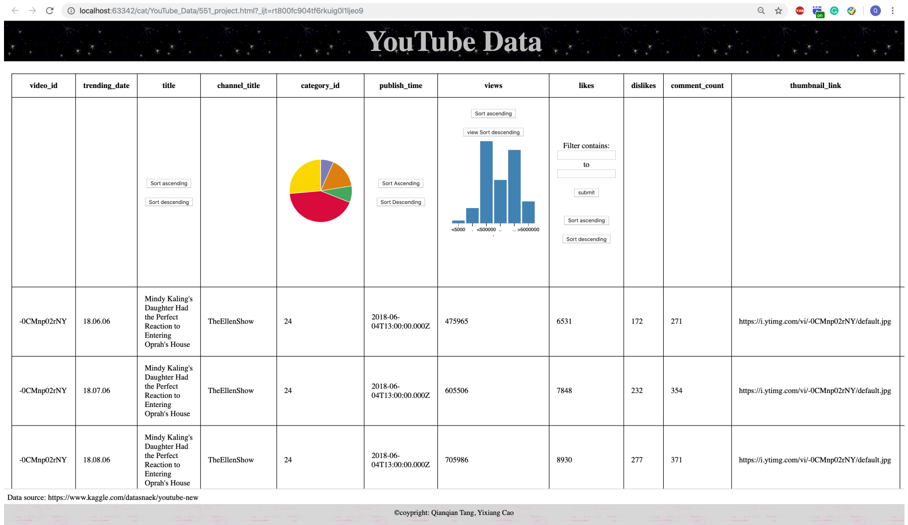
Tang Qianqian SID: 1289405231

I. Project Idea

First, we chose the dataset for Trending Youtube Video Statistics. It contains attributes like video_id, trending_date title, channel_title, category_id, publish_time, tags views, likes, dislikes, comment_count, thumbnail_link, comments_disabled, ratings_disabled, video_error_or_removed, description. We chose Firebase to store the data. And then, we implemented sorting functions on 3 numerical attributes: “likes”, “views” and “publish time”, and 1 non-numerical attribute: “title”. Also, we implemented filtering functions on one numerical attribute: “likes”, and one nonnumerical attribute: “tags”. To help the users more intuitive to observe data trends, we visualized 3 attributes: "category_id”, “views” and “tags”. Finally, we used the pie chart to present the proportion of “category_id”, used the statistical histogram to visualize “views” and used the word cloud to crawl keywords of “tags”.

II. Screenshot for each working component with a description.

Below are screenshots of the home page, which show all the data.

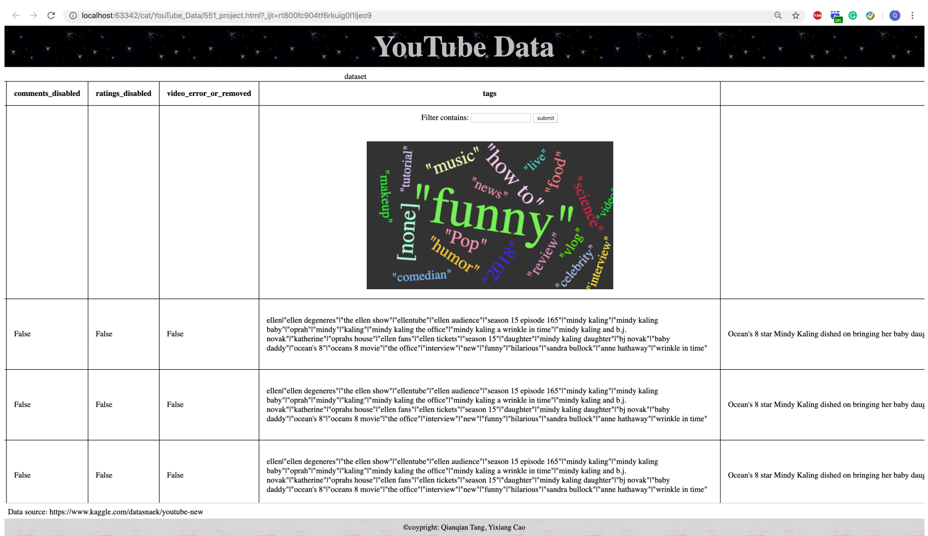


localhost:63342/cat/YouTube_Data/551_project.html?_ijt=rt800fc904tf6rkuig0l1ljeo9

YouTube Data

video_id	trending_date	title	channel_title	category_id	publish_time	views	likes	dislikes	comment_count	thumbnail_link
		Sort ascending Sort descending			Sort Ascending Sort Descending	Sort ascending view Sort descending <5000 . <500000 .. ... >5000000	Filter contains: to submit Sort ascending Sort descending			
-0CMnp02rNY	18.06.06	Mindy Kaling's Daughter Had the Perfect Reaction to Entering Oprah's House	TheEllenShow	24	2018-06-04T13:00:00.000Z	475965	6531	172	271	https://i.ytimg.com/vi/-0CMnp02rNY/default.jpg
-0CMnp02rNY	18.07.06	Mindy Kaling's Daughter Had the Perfect Reaction to Entering Oprah's House	TheEllenShow	24	2018-06-04T13:00:00.000Z	605506	7848	232	354	https://i.ytimg.com/vi/-0CMnp02rNY/default.jpg
-0CMnp02rNY	18.08.06	Mindy Kaling's Daughter Had the Perfect Reaction to Entering Oprah's House	TheEllenShow	24	2018-06-04T13:00:00.000Z	705986	8930	277	371	https://i.ytimg.com/vi/-0CMnp02rNY/default.jpg

Data source: https://www.kaggle.com/datasnaek/youtube-new



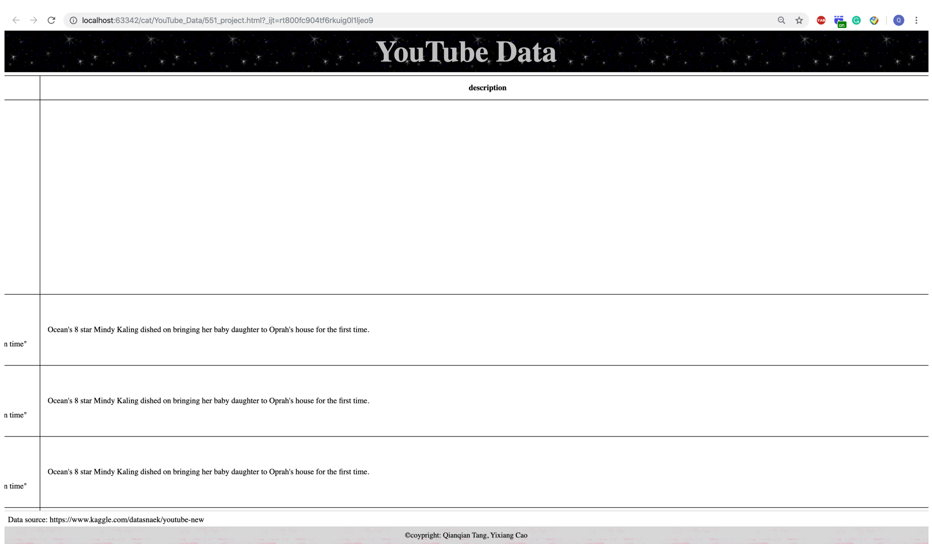
localhost:63342/cat/YouTube_Data/551_project.html?_ijt=rt800fc904tf6rkuig0l1ljeo9

YouTube Data

dataset

comments_disabled	ratings_disabled	video_error_or_removed	tags	
			Filter contains: submit "makeup" "tutorial" "music" "how to" "live" "food" "news" [none] "funny" "science" "video" "Pop" "humor" "2018" "review" "vlog" "celebrity" "interview" "comedian"	
False	False	False	ellen|"ellen degeneres"|"the ellen show"|"ellentube"|"ellen audience"|"season 15 episode 165"|"mindy kaling"|"mindy kaling baby"|"oprah"|"mindy"|"kaling"|"mindy kaling the office"|"mindy kaling a wrinkle in time"|"mindy kaling and b.j. novak"|"katherine"|"oprahs house"|"ellen fans"|"ellen tickets"|"season 15"|"daughter"|"mindy kaling daughter"|"bj novak"|"baby daddy"|"ocean's 8"|"oceans 8 movie"|"the office"|"interview"|"new"|"funny"|"hilarious"|"sandra bullock"|"anne hathaway"|"wrinkle in time"	Ocean's 8 star Mindy Kaling dished on bringing her baby daug
False	False	False	ellen|"ellen degeneres"|"the ellen show"|"ellentube"|"ellen audience"|"season 15 episode 165"|"mindy kaling"|"mindy kaling baby"|"oprah"|"mindy"|"kaling"|"mindy kaling the office"|"mindy kaling a wrinkle in time"|"mindy kaling and b.j. novak"|"katherine"|"oprahs house"|"ellen fans"|"ellen tickets"|"season 15"|"daughter"|"mindy kaling daughter"|"bj novak"|"baby daddy"|"ocean's 8"|"oceans 8 movie"|"the office"|"interview"|"new"|"funny"|"hilarious"|"sandra bullock"|"anne hathaway"|"wrinkle in time"	Ocean's 8 star Mindy Kaling dished on bringing her baby daug
False	False	False	ellen|"ellen degeneres"|"the ellen show"|"ellentube"|"ellen audience"|"season 15 episode 165"|"mindy kaling"|"mindy kaling baby"|"oprah"|"mindy"|"kaling"|"mindy kaling the office"|"mindy kaling a wrinkle in time"|"mindy kaling and b.j. novak"|"katherine"|"oprahs house"|"ellen fans"|"ellen tickets"|"season 15"|"daughter"|"mindy kaling daughter"|"bj novak"|"baby daddy"|"ocean's 8"|"oceans 8 movie"|"the office"|"interview"|"new"|"funny"|"hilarious"|"sandra bullock"|"anne hathaway"|"wrinkle in time"	Ocean's 8 star Mindy Kaling dished on bringing her baby daug

Data source: https://www.kaggle.com/datasnaek/youtube-new



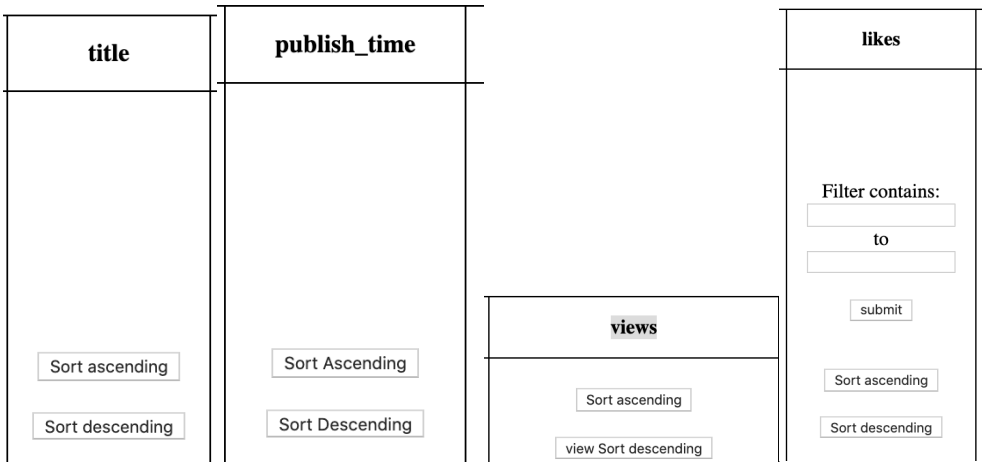
localhost:63342/cat/YouTube_Data/551_project.html?_ijt=rt800fc904tf6rkuig0l1ljeo9

YouTube Data

	description
n time"	Ocean's 8 star Mindy Kaling dished on bringing her baby daughter to Oprah's house for the first time.
n time"	Ocean's 8 star Mindy Kaling dished on bringing her baby daughter to Oprah's house for the first time.
n time"	Ocean's 8 star Mindy Kaling dished on bringing her baby daughter to Oprah's house for the first time.

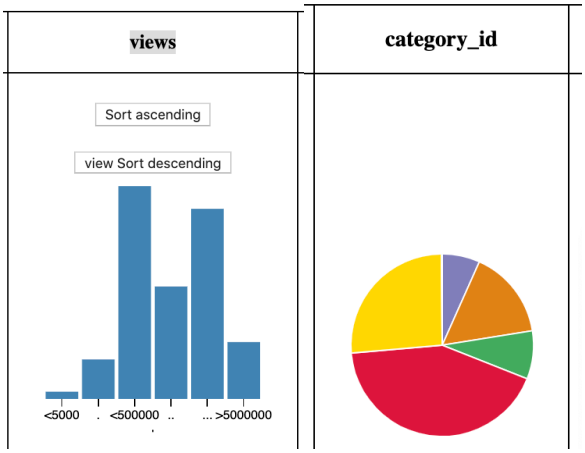
Data source: https://www.kaggle.com/datasnaek/youtube-new

The data is sorted according to the attribute “title”, “publish_time”, “views”, and “likes” alphabetically by clicking the buttons.

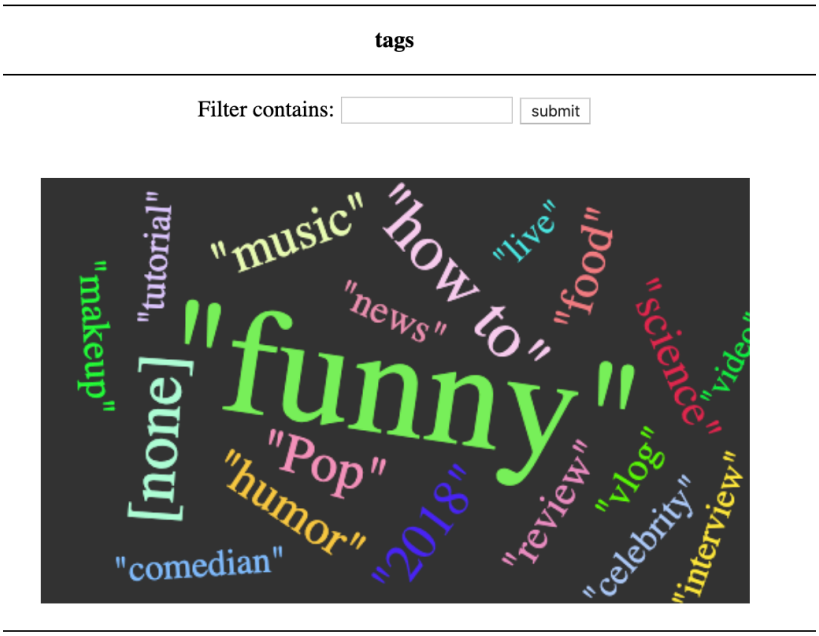


The data can be filtered by the attribute "likes" in a range. And the filter function also applied to attribute "tags" by inputting words and clicking the button.

The frequency of the attribute "views" of the data is presented by the histogram. And the proportion of the attribute “category _id” of the data can be demonstrated by the pie chart.



Also, the keywords of the tags with different sizes are presented based on the frequency.



II. Implementation

Pre-processing:

We firstly downloaded the data from Kaggle to the localhost. Because datatypes of all the attributes in the localhost CSV file are string, we wrote a python script to adjust the corresponding data type, then uploaded the data on firebase. Below is the screenshot of the python script:

```
import json
import csv
import requests
import codecs
csvFile = open('USvideos.csv', 'br')
jsonFile = open('file.json', 'w')
reader = csv.DictReader(codecs.iterdecode(csvFile, 'utf-8'))
out = json.dumps({"USvideos": [ row for row in reader ]}, sort_keys=True, indent=4, separators=(',', ':'))
jsonFile.write(out)
with open('file.json') as f:
  data = json.load(f)
for sub in data['USvideos']:
    for i in sub:
        if i=='category_id':
            sub[i]= int(sub[i])
        elif i=='views':
            sub[i]= int(sub[i])
        elif i=='likes':
            sub[i]= int(sub[i])
        elif i=='dislikes':
            sub[i]= int(sub[i])
        elif i=='comment_count':
            sub[i]= int(sub[i])
url = 'https://web-group-proj.firebaseio.com/'
response = requests.put(url+'USvideos.json', json.dumps(data['USvideos'], indent=4, separators=(',', ':')))
```

Web Frame:

We used HTML, CSS build a simple web frame, which contained a title and large table.

Show all data:

To show all of the data, we used 'firebase-analytics.js' library to connect our firebase, reading items in the database one by one and appended them into a table. Below is the JS function of showing data:

```
function showdatatab(dataref){
        dataref.forEach(
          function(snapshot){
            item=snapshot.val();
            var table='<tr><td>'+item.video_id+'</td><td>'+item.trending_date+'</td><td>'
            +item.title+'</td><td>'+item.channel_title+'</td><td>'+item.category_id+'</td><td>'+item.publish_time+
            '</td><td>'+item.views+'</td><td>'+item.likes+'</td><td>'+item.dislikes+'</td><td>'+item.comment_count+
            '</td><td>'+item.thumbnail_link+'</td><td>'+item.comments_disabled+'</td><td>'+item.ratings_disabled+
            '</td><td>'+item.video_error_or_removed+'</td><td>'+item.tags+'</td><td>'+item.description+'</td></tr>';
              $('#table tbody').append(table);
          }
          );
    }
```

Sorting functions:

For ascending sort functions, we wrote JavaScript functions to connect Firebase, making it sort items by the attributes we selected and returned the sorted data. Then we rewrite the whole table with sorted data. For example, the following JS function is used to sort views in ascending order:

```
function viewsortAscending(){
  $("#table tbody  tr:not(:first)").empty("");
  var sortref = firebase.database().ref("USvideos").orderByChild("views").limitToFirst(40);
     sortref.on('value',showdatatab);
}
```

For descending sort functions, there was a problem because firebase's 'orderByChild' function does not support descending sort. So we chose to write a new 'showdatatab' function for descending sort, which wrote items on the table from bottom to up. Below is the new 'showdatatab' function:

```
function showdatatab_d(dataref){
        dataref.on('value', showdatafun);
        var list=[0];
        var size=0;
        dataref.forEach(
          function(snapshot){
            item=snapshot.val();
            list.push(item)
            size=size+1
          } );
        for(i=0;i<size;i++){
          item=list.pop()
          var table='<tr><td>'+item.video_id+'</td><td>'+item.trending_date+'</td><td>'+item.title+'</td><td>'+item.channel_title+
          '</td><td>'+item.category_id+'</td><td>'+item.publish_time+'</td><td>'+item.views+'</td><td>'+item.likes+ '</td><td>'+
          item.dislikes+'</td><td>'+item.comment_count+'</td><td>'+item.thumbnail_link+'</td><td>'+item.comments_disabled+
          '</td><td>'+item.ratings_disabled+'</td><td>'+item.video_error_or_removed+ '</td><td>'+item.tags+'</td><td>'+item.description+'</td></tr>';
              $('#table tbody').append(table);
        }}
```

Finally, we created sort buttons in the header and set the corresponding sort functions to be triggered by manually clicking.

Filter functions:

For numerical filters, we wrote JavaScript function to connect Firebase, making it sorted by the attributes we selected and extracted the part of data in the input range. Then we rewrite the whole table with filtered data. For example, the following JS function is used to filter views:

```
function submittime(){
    timea=$("#time1").val();
    timeb=$("#time2").val();
    var filtref = firebase.database().ref("USvideos").orderByChild("video_id").startAt(timea);
    filtref=filtref.orderByChild("video_id").endAt(timeb)
    filtref.on('value',showdatatab);
  }
```

For the non-numerical filter, the filter of the string is a fuzzy search, so we used ‘indexOf’ function to get the index of input string when the firstly appear in tags, which will return -1 if the input string could not be found. Then all the items whose results of ‘indexOf’ function are not -1 are what we wanted.

```
function showdatatab_filter_non_num(dataref){
      var arr=[];
      var list=[];
      var size=0;
  dataref.forEach(
    function(snapshot){item=snapshot.val(); list.push(item);size=size+1}  );
    for(var i = 0;i<size; i++){
      item = list[i]
      keyword = item.tags
      kw2 = keyword.toLowerCase()
      inp2 = inputtag.toLowerCase()
      if (kw2.indexOf(inp2) != -1){arr.push(list[i]) }}
    for(i=0;i<arr.length;i++){
    item=arr[i]
    var table='<tr><td>'+item.video_id+'</td><td>'+item.trending_date+'</td><td>'+item.title+'</td><td>'+item.channel_title+'
    </td><td>'+item.category_id+'</td><td>'+item.publish_time+'</td><td>'+item.views+'</td><td>'+item.likes+'</td><td>'+
    item.dislikes+'</td><td>'+item.comment_count+'</td><td>'+item.thumbnail_link+'</td><td>'+item.comments_disabled+'</td><td>'
    +item.ratings_disabled+'</td><td>'+item.video_error_or_removed+'</td><td>'+item.tags+'</td><td>'+item.description+'</td></tr>';
        $('#table tbody').append(table); } }
```

Visualization:

We used the D3 library to draw a statistical histogram of views and a pie chart of category_id.

```
function histoGram(fD){
  var hG={},    hGDim = {t: 10, r: 10, b: 20, l: 10};
  hGDim.w = 200 - hGDim.l - hGDim.r,
    hGDim.h = 200 - hGDim.t - hGDim.b;
  var hGsvg = d3.select(id).append("svg")
    .attr("width", hGDim.w + hGDim.l + hGDim.r)
    .attr("height", hGDim.h + hGDim.t + hGDim.b).append("g")
    .attr("transform", "translate(" + hGDim.l + "," + hGDim.t + ")");
 var x = d3.scale.ordinal().rangeRoundBands([0, hGDim.w], 0.1)
    .domain(fD.map(function(d) { return d[0]; }));
  hGsvg.append("g").attr("class", "x axis")
    .attr("transform", "translate(0," + hGDim.h + ")")
    .call(d3.svg.axis().scale(x).orient("bottom"));
  var y = d3.scale.linear().range([hGDim.h, 0])
    .domain([0, d3.max(fD, function(d) { return d[1]; })]);
  var bars = hGsvg.selectAll(".bar").data(fD).enter()
    .append("g").attr("class", "bar");
  bars.append("rect")
    .attr("x", function(d) { return x(d[0]); })
    .attr("y", function(d) { return y(d[1]); })
    .attr("width", x.rangeBand())
    .attr("height", function(d) { return hGDim.h - y(d[1]); })
    .attr('fill',barColor);
  var legend = d3.select(id).append("table").attr('class','legend');
  return hG;
}
```

```
function pieChart(pD){
  var pC ={},    pieDim ={w:150, h: 150};
  pieDim.r = Math.min(pieDim.w, pieDim.h) / 2;
  var piesvg = d3.select(id).append("svg")
    .attr("width", pieDim.w).attr("height", pieDim.h).append("g")
    .attr("transform", "translate("+pieDim.w/2+","+pieDim.h/2+")");
  var arc = d3.svg.arc().outerRadius(pieDim.r - 10).innerRadius(0);
  var pie = d3.layout.pie().sort(null).value(function(d) { return d.freq; });
  var arcs = piesvg.selectAll("path").data(pie(pD)).enter().append("path").attr("d", arc)
    .each(function(d) { this._current = d; })
    .style("fill", function(d) { return segColor(d.data.type); });

  arcs.append("text")
    .attr("transform",function(d){
    return "translate(" + arc.centroid(d) + ")";
})
    .attr("text-anchor","middle")
    .text("hahah");

  function arcTween(a) {
    var i = d3.interpolate(this._current, a);
    this._current = i(0);
    return function(t) { return arc(i(t));    };
  }
  return pC; }
```

And we used ‘wordcloud.js’ library to extract the frequency of keywords in all video tags and draw word cloud graph.

```
function showwordcloud(dataref){
    var dic = new Array()
    data=dataref.val()
    var check=0
    for (var i in data){
        check=check+1
        var tag=data[i].tags
        var taglist=tag.split("|")
        for (var j=0;j<taglist.length;j++){
            var temp=taglist[j]
            if(dic[taglist[j]]!=undefined){dic[temp]=dic[temp]+1}
            else{dic[temp]=0}}}
    var dic_s=Object.keys(dic).sort(function(a,b){return dic[b]-dic[a]});
    var wordlist=[]
    for (var i=0;i<20;i++){
        tempkey=dic_s[i]
        tempvalue=dic[tempkey]
        wordlist.push([tempkey,tempvalue])}
var options = eval({
    "list": wordlist,
    "gridSize": 16,
    "weightFactor": 0.03,
    "fontWeight": 'normal',
    "fontFamily": 'Times, serif',
    "color": 'random-light',
    "backgroundColor": '#333',
    "rotateRatio": 1 });
var canvas = document.getElementById('canvas');
WordCloud(canvas, options);}
```

IV. Group Formation and responsibility.

Team member 1:

Name: Cao Yixiang

Responsibility: filter non-numerical attributes, visualization

Team member 2:

Name: Qianqian Tang

Responsibility: web frame, sorting function, filter numerical attribute

v. Reference:

[1] Dataset: https://www.kaggle.com/datasnaek/youtube-new

[2] D3 histogram: https://observablehq.com/@d3/histogram

[3] D3 pie chart: http://bl.ocks.org/NPashaP/96447623ef4d342ee09b

[4] Wordcloud library: https://cdn.bootcss.com/wordcloud2.js/1.1.0/wordcloud2.js