




# Activity Breakdown

---

Lynda Negron, Jordan Dominguez, and Brandon Tineo



This project analyzes and breakdowns YouTube activity by major city. Within our visualization, we plot the biggest cities in the world on a map, showing the amount of videos uploaded within that city by major category. We also show how popular those videos are, and are thus able which cities provide the most popular videos of each YouTube category.

# **YouTube Activity Breakdown**

---



# PROCESS BOOK

---



# INITIAL STEPS

Project proposal, initial sketches, first design studio

---

Since its creation in 2005, Youtube has grown quite a bit in terms of popularity. Currently, Youtube claims that roughly 60 hours of new videos are uploaded to the site every single minute and boasts 800 million unique users a month. Given that the Youtube interface can be accessed in 61 different languages, it comes at no surprise that three quarters of the material uploaded to Youtube every single day comes from outside the United States.

However, culture within the United States and culture across different parts of the world is extremely different and thus we can expect that way in which users use Youtube is different from one place to another. For example, while one country might value humorous videos a lot, another country might value sports videos more. Value in this case can be measured in a variety of ways (i.e view counts, rating...). Taking all of this into consideration is what initially sparked out interest in this project. Further motivation comes from the fact that three of us are very active users of Youtube and are well aware of the plethora of information that can be accessed from the Youtube API.

Finally, we think this project is extremely interesting because of how widely applicable our visualizations will be. Up to date, it might seem surprising, but there are no effective visualizations of the way in which Youtube users use the website, or what they find appealing. Our visualization can be useful to the every day Youtube user, as well as to companies/organizations looking to effectively target information to certain groups of people.

# Background and Motivation

---

- We want to visualize how Youtube use differs around the world, including which parts of the world have the most viewers and what genre of videos are most popular.

# Initial Objectives

---

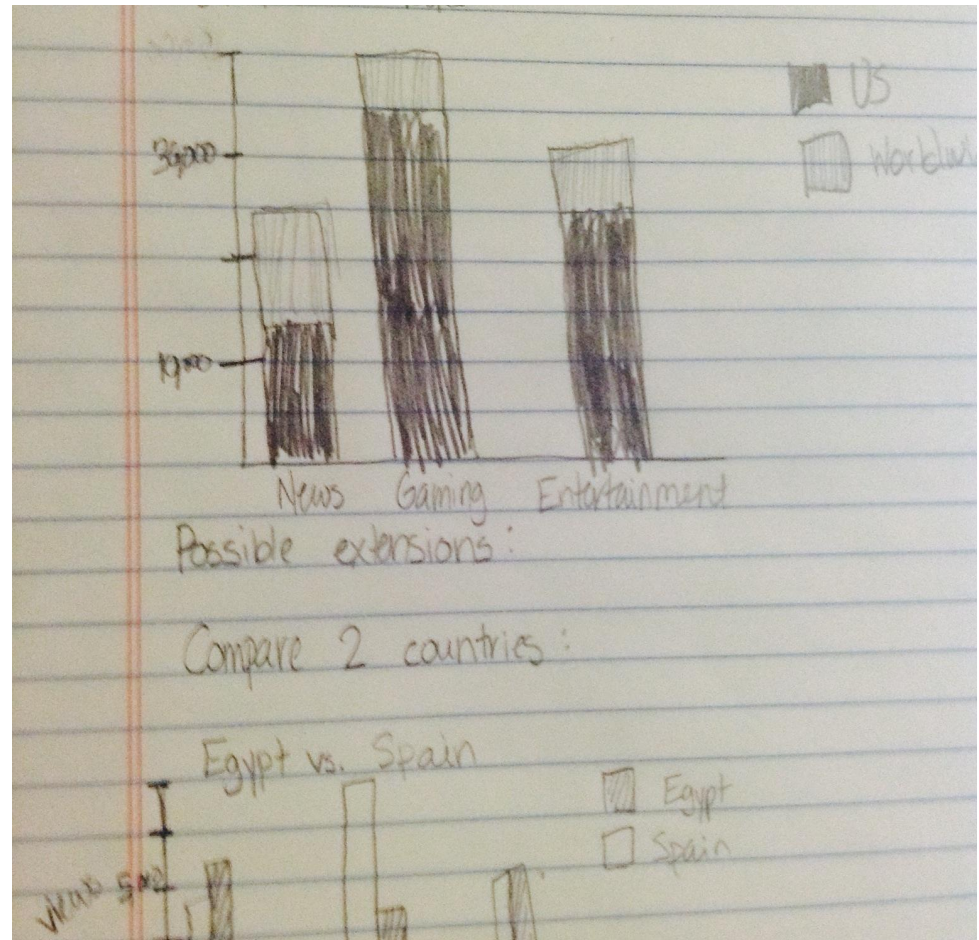


Chloropleth map to visualize amount of videos uploaded total videos of selected category in every country

# First Sketches

---

These sketches would provide alternative data regarding the information. One will be a bar chart that, on a mouseover of the country, would show the number of videos in each category. Another, on the side, would change when the map data changed, showing which countries had the most amount of videos or watch counts per videos within that category.



# First Sketches



- We soon found out that the YouTube API (at least, the updated version of it) would only provide which *city* these videos were uploaded in rather than the country.

# Problems

---



# **CHANGE OF DIRECTION**

---

- Instead of creating a choropleth map, we decided to plot the major cities of the world on a world map and choropleth color & base radius on amount of videos uploaded per city based on category.

# New Data?

---

- The feedback that we received during the design studio was extremely helpful. Some of the concerns raised by the group we discussed our project with were:
  - Cities chosen:
    - Cities chosen (top 100) might lead to overly concentrated areas and overly empty areas
    - Dots for each city need to be carefully sized as to not get too messy
    - Map more cities than just the top 100
  - Categories chosen:
    - Good idea to structure our visualization around categories
    - Category information most likely will appeal to advertisers and thus providing information on the demographics of these cities and the categories these cities enjoy most is key.
  - Visualization:
    - Probably best to not color our city points choroplethically because points are too small.
    - because the points weren't
- New ideas we got from the design studio included
  - Adding both a side visualization and a tooltip.
  - Showing information that supplements our understanding to the data we would show on the map.

# Design Studio Feedback

---



---

# New Sketches

---



# DATA COLLECTION

Data scraping, cleaning, aggregation, API calls methods etc.

---

- Our first question was which cities we wanted to look at YouTube information from.
- We decided to do some research on the top 100 cities in terms of population and scrape their longitude and latitude using JavaScript.

**The Biggest cities of the World (TOP 1000 cities)**

N	Name	Population	Altitude	Country	Latitude	Longitude
1	Tokyo :	31 480 498	20 m	JP	35°41'6.0" N	139°45'3.6" E
2	Beijing Shi :	14 933 274	59 m	CN	39°55'44.0" N	116°23'16.8" E
3	Shanghai :	14 608 512	6 m	CN	31°0'18.0" N	121°24'32.4" E
4	Buenos Aires :	13 076 300	14 m	AR	34°35'15.0" S	58°40'21.0" W
5	Bombay :	12 692 717	35 m	IN	18°58'30.0" N	72°49'32.9" E
6	Karachi :	11 627 378	14 m	PK	24°52'0.1" N	67°2'60.0" E
7	Constantinople :	11 581 707	157 m	TR	41°1'7.0" N	28°57'52.9" E
8	Mexico (México) :	11 285 654	2 224 m	MX	19°23'60.0" N	99°2'60.0" W
9	Delhi :	10 928 270	215 m	IN	28°40'0.1" N	77°13'0.1" E
10	Manila :	10 443 877	19 m	PH	14°36'15.1" N	120°58'55.2" E
11	Moscow :	10 381 288	151 m	RU	55°45'7.9" N	37°36'56.2" E
12	Seoul :	10 323 448	46 m	KR	37°33'59.0" N	127°0'0.0" E
13	Sao Paulo (S?o Paulo) :	10 021 437	631 m	BR	23°31'59.9" S	46°37'0.1" W
14	Istanbul :	9 797 536	28 m	TR	41°1'7.0" N	28°57'52.9" E
15	Lagos :	8 789 133	39 m	NG	6°27'11.0" N	3°23'45.0" E
16	Mexico :	8 720 916	2 224 m	MX	19°26'3.1" N	99°8'19.0" W

# City Data Collection

- Scraping the city data was only the first step in getting a usable city dataset to use in our project.
- In wrangling our city we had to make sure our latitudes and longitudes were formatted the correct way so that they matched the expected input of our map. This involved splitting the latitudes and longitudes returned by the scrape and returns integers that accounted for whether they were North, South, West or East.
- Our final step in wrangling our city data was to create a CSV file and a JSON file that could be easily accessed by other parts of our program.

# City Data Wrangling

---



- We started by exploring a variety of the information that the YouTube API could return to us.
- As previously mentioned we found that the version of the API that would be most suitable for us to use (Version 2) would not allow us to collect information about countries.
- At this point however we had collected all of the city data we needed and so we were ready to look into the kinds API calls that would return city information

## **YouTube API – Exploration Phase**

---

- We quickly realized that it would be best for us to select a few categories to work with.
- In selecting categories we visited the YouTube.com actual page and looked through which categories seemed to be most popular. It came to our attention that YouTube consistently placed the following categories at the top of their suggested categories: Comedy, Sports, Movies, Music, and Gaming.

# **YouTube API – Selecting Categories**

---

- One we had selected our categories we moved on to actually making our API calls and building an object. This step was particularly cumbersome given the asynchronous nature of the various YouTube API calls we were making.
- To solve this issue we proceeded to wrap the API calls within a function that would allow us to make sure every API call was completely done before moving on to the rest of the code.
- The YouTube API returns XML which required us to parse through the different tags, ids and attributes to gather the information we wanted.
- After exploring the <feed></feed> returned by each API call we proceeded to clean and merge our data into an object. Our object is currently sorted by categories since our visualization will allow user to change what they see based on what category they select.
- One final problem we encountered was handling the scope of our object.

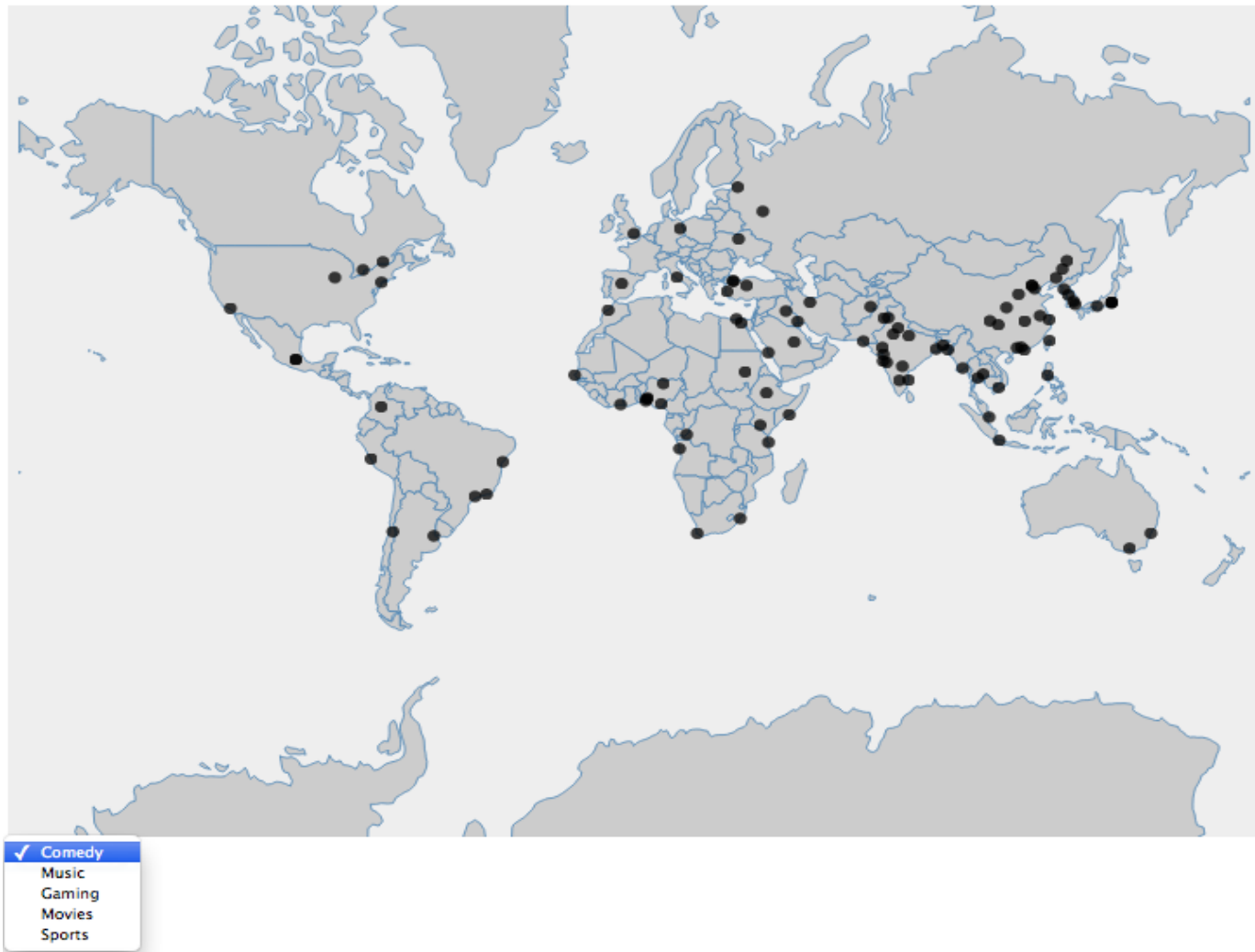
# YouTube API – API Calls + Data Wrangling

---

- We first started by plotting a map of the world by following the way we did this in the extra credit of the last homework. Given the amount of data we were plotting we wanted to add a little more functionality to our map than we did in the extra credit of HW4. The first of these added functionalities were geometric zoom (to allow users to zoom into particular parts of the map), and tooltips (to provide additional information about the countries/cities).
- We then implemented the drop down menu which would give users the ability to interact with the visualization.
- Finally, we plotted a map on the map to represent the 100 cities we were currently focusing on.

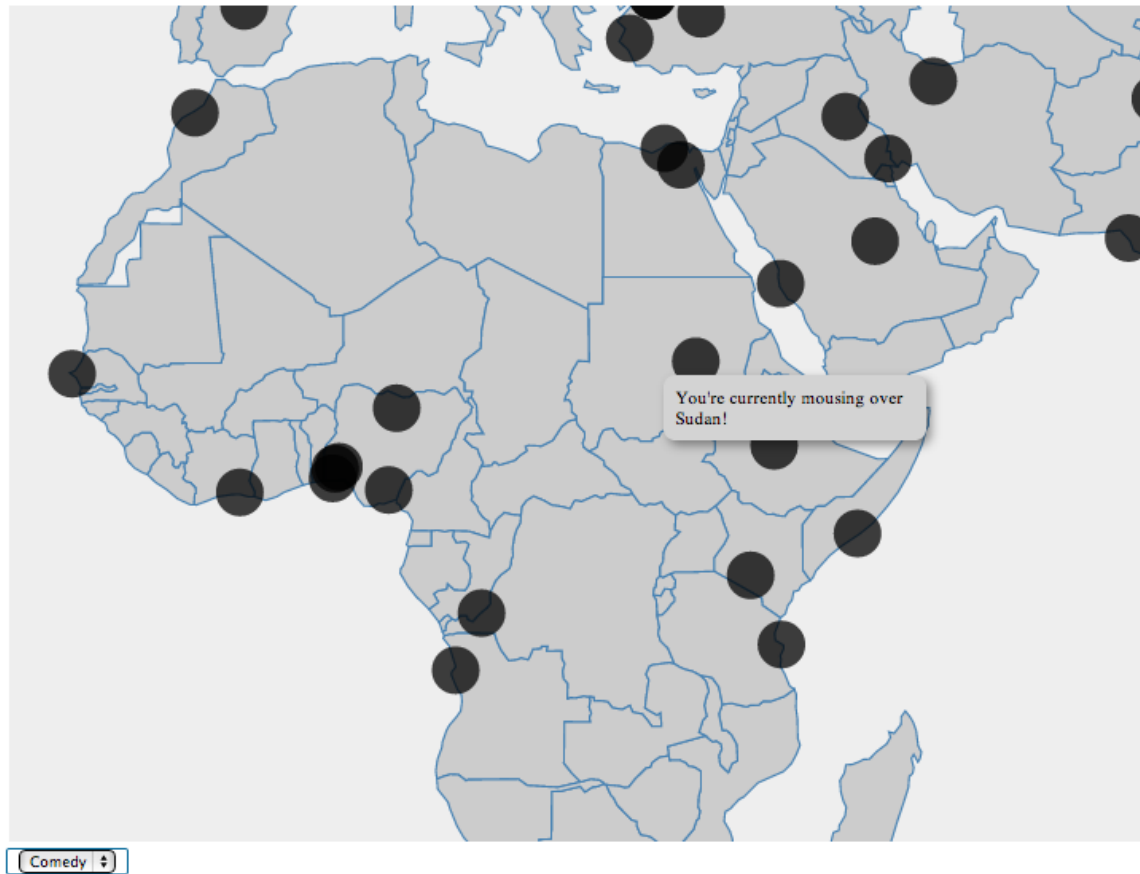
# Implementing Visualization – Initial Steps

---



# Visualization – First Look

---



# Visualization – First Look

---

- We are currently working on making the dots on the map sized and colored by the number of YouTube videos a particular city had in a particular category.
- However, we are running into some problems with asynchronicity when trying to use the data collected from our API calls
- We are currently working to solve our problems with asynchronicity using the queue methods in d3, but are finding that when we defer the necessary functions the visualization does not load

- 

## **Implementing Visualization – Current Status**

---

- First, we want to solve our problem with asynchronicity
- Next, we want to change the size of our circles so that they are sized by number of videos in each category selected in the dropdown menu
- Finally, we want to implement our side/detail visualizations - displaying how many videos in each category were posted, the relative popularity of various categories, and perhaps even the most popular videos overall.

# Implementing Visualization – Next Steps

---



- Change tooltip to display more important information
  - i.e. the most popular YouTube link for the city in the selected category
- Implement side visualization
- [Perhaps] color the map in a way that displays some relevant information

# Implementing Visualization – Next Steps

---



# FIRST MILESTONE

First implementation, feedback on progress

---



# **FIRST COMPLETE SCREENCAST**

Full functional prototype, second round of feedback from peers, reincorporation

---



# FINAL PRODUCT

Final screencast

---