Team 11:

Jose Martinez Carvajal

Muhammad Habib

We plan on using YouTube Statistical data from Kaggle and from the YouTube Data API. The data from Kaggle provided information for the most trending videos of 2017.

**Extract:**

From Kaggle we extracted a csv and json file. The US\_videos.csv file provided the name of the trending videos, the channel that made it, a category id, and information about user interaction like the number of comments.

In the category\_id.json file you could find the category name each video based on the category id that was included in the csv file.

From the YouTube Data API we wanted to collect data from the channels in our csv files like the amount of subscribers they had. We took that information, converted it into a data-frame and then made a csv file from it so that we could upload it to our database.

**Transform:**

We began our data cleaning with the csv file obtained from Kaggle.com. If we wanted to ask the question how long it take for a video for become trending, we would have found it difficult because the publish time and publish date was formatted as a string instead of datetime. Our first task was to convert these two columns into a datetime so we could do a time analyses if we so choose. In other columns we had values that were numbers but were formatted into strings instead of integers. We used the astype function to change the format of these values to integers. Furthermore, the csv file had a number of empty rows so we used the dropna() function to get rid of the null values. We also dropped columns that we found to hold irrelevant data for further analysis. The last piece of cleaning we did was renaming the column channel\_title to channel\_name so that we could merge it in pandas or joining it with the table made from the youtube data api.

To find the category names for each video we used the category\_id column in the csv and json file. Using the following function:

*with open('./youtube\_files/US\_category\_id.json', 'r') as json\_data:*

*data = json.load(json\_data)*

*for category in data['items']:*

*category\_name[category['id']] = category['snippet']['title']*

we could extract the category names for each category\_id in our csv file. Using the category\_id we could join the data from the json file to the csv. We used the insert function in pandas to insert a new category\_names column into our cleaned csv.

From the youtube data api we appended dictionaries with the data that we needed. When we tried to create the dataframe, we got the following error:

ValueError: arrays must all be same length.

We used this function:

def pad\_dict\_list(response\_dict, padel):

lmax = 0

for lname in dict\_list.keys():

lmax = max(lmax, len(response\_dict[lname]))

for lname in response\_dict.keys():

ll = len(response\_dict[lname])

if ll < lmax:

response\_dict[lname] += [padel] \* (lmax - ll)

which pads the rows with no values. The function in conjuction with the transpose() produced the dataframe. After the dataframe was created from the dictionaries we reformatted the columns so that it matched the csv file we cleaned from Kaggle. Both dataframes had the column channel\_name in common which we could use to merge in pandas or join in sql.

**Load:**

*\* The type of final production database to load the data into (relational or non-relational).*

Relational since we will be linking the channel\_name in 2 separate tables which will determine which we can use to determine the category of videos that become top trending, the amount of views and comment counts for a channel in our csv file, etc.

*\* The final tables or collections that will be used in the production database.*

We created the database and tables in Postgres SQL

**Uses for this data:**

Find the top trending video, not necessarily the most viewed video

What video is gaining traction and going viral?

What video is being talked about the most on the internet?