**YouTube Data Documentation**

Quick Note: I forgot to add df.to\_csv("TATE\_youtube.csv", **index=False**)

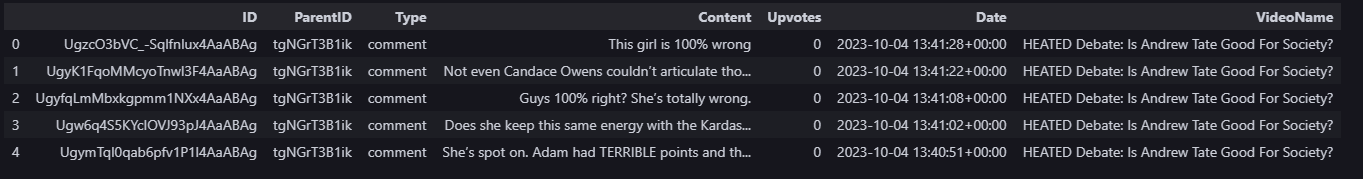
which is why the indexing is weird.

df = pd.read\_csv('Tate\_youtube.csv')

df = df.drop(columns=['Unnamed: 0'])

You can just drop it for future reference.

**Tate\_youtube.csv**



**ID –** Unique comment / reply ID

**ParentID –** Parent Comment ID & Video ID

**Type** **–**  comment or reply

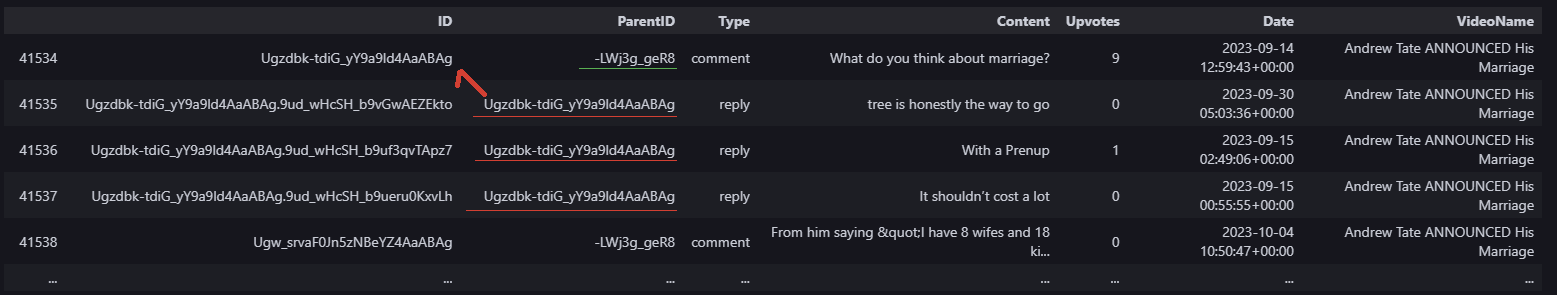
**Content -** Comment

**Date -** %Y-%m-%d %H:%M:%S

**VideoName –** Name of Video

**How the Comment Thread is Listed:**

1. **Parent Comment – First Comment in a Thread**
   1. **ParentID =** VideoID
   2. **Type** = comment
2. **Reply**
   1. **ParentID** = The ID of the *parent comment* being replied to.
   2. **Type** = reply



**Code**

df = pd.read\_csv('Tate\_youtube.csv')

df = df.drop(columns=['Unnamed: 0']) # forgot index=False when to\_csv()

videogroup = df.groupby('ParentID') # Where the Parent Comments are

video\_reply = {}

# Filtering out the videos to find the Parent comment (only need video\_no\_reply)

video\_no\_reply = {}

# Creating 2 DataFrames depending on if the df.Type is ‘comment’ or ‘reply’

for v\_id, g\_df in videogroup:

no\_reply = g\_df[g\_df.Type == "comment"]

# Technically if the lengths are the same, there's no reply

if len(g\_df) == len(no\_reply):

video\_no\_reply[v\_id] = no\_reply

else:

video\_reply[v\_id] = g\_df

# Combining all the DataFrames from the dictionary

df\_no\_reply = pd.concat(video\_no\_reply.values(), ignore\_index=True)

df\_no\_reply.head()

# I want the TopComment ParentID

video\_id = "-LWj3g\_geR8"

top\_level\_comments = df[(df.ParentID == video\_id) & (df.Type == "comment")]

top\_level\_comment\_ids = top\_level\_comments['ID'].tolist()

all\_comments\_replies = df[df['ID'].isin(top\_level\_comment\_ids) | df['ParentID'].isin(top\_level\_comment\_ids)]

all\_comments\_replies