

3_Analysis

December 16, 2021

```
[50]: ### Importing packages
import pandas as pd
import pickle
import os
import numpy as np
import math
import matplotlib.pyplot as plt

from sklearn.decomposition import NMF
from sklearn.feature_extraction.text import TfidfTransformer
from scipy.cluster.hierarchy import linkage, leaves_list, cut_tree
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cluster import KMeans
from youtube_transcript_api import YouTubeTranscriptApi

from time import process_time

path = os.chdir("C:\\Users\\guill\\OneDrive - University of_
↳Pittsburgh\\Texto\\Project\\Final")
```

1 Radical YouTubers?

1.1 Introduction

Social media consumption has been increasing in the last two decades. Data from Pew Research Center show that 81% of adults in the United States use YouTube. YouTube was launched in 2005 and is still growing in the audience—its audience was 73% of adults in 2019 (Auxier & Anderson, 2021). The effects of YouTube are multiple. On the one hand, YouTube makes a profit from ads, and it pays to the content creators through the YouTube Partner Program. According to YouTube, this program paid 30 billion dollars to creators in the last three years (Mohan, 2021). These payments stimulate creators to produce new, diverse, and exciting content, which could help to increase the number of choices each viewer has available. Platforms that aggregate and distribute content contribute with variety and lower search costs to consumers' welfare (Athey, Mobius, & Pal, 2021)..

On the other hand, the presence of black-box algorithms that create filter bubbles (Pariser, 2011)

prevents us from understanding all their effects. There are rising concerns about the potential negative impacts of social media. Although there is a small but growing literature about the effects of the internet and social media on polarization (Zhuravskaya, Petrova, & Enikolopov, 2020), we know less about whether content creators face incentives to provide radicalized content (as highlighted by the media, e.g., Roose (2019)). This project contributes to filling this gap by measuring polarization on Youtube by looking at the text from video's transcripts. This is a descriptive paper where I plan to look at the recent evolution of the political ideologies of Youtube's videos. In particular, I will address the question of whether there has been an increasing polarization in the period 2016-2022.

This project studies how social media that pays their creators and individualizes content could exacerbate polarization. I analyze the production of content that is not affiliated with mainstream media. Some of these content-creators started streaming from their homes, and thanks to the income paid by YouTube, they were able to build channels with more than 1 million subscribers and produced videos watched by 500,000 viewers.

I estimate the extent to which content-creators ideological choice and differentiation evolve from the first quarter of 2016 and the last quarter of 2021. I use computational text analysis methods to analyze YouTubers' political ideology and the formed ideological clusters. Up to my knowledge, this project is the first research to look at differentiation in political content on YouTube's videos using their transcripts.

I find that polarization in the videos has not been increasing as it is expected. However, the five methods to calculate polarization fail to match other classification algorithms which raises questions about the approach.

1.2 Data

In order to measure polarization, I use YouTube video transcripts. I chose 67 content-creators that uploaded videos discussing political issues. To get the list of these creators, I looked at a website that already has labeled content-creators according to their political ideology (transparency.tube). The existence of these labels helps me to discuss the results later.

The transcripts come in the form of a dictionary with text and the time of the video they appear.

```
[51]: # As an example: Elon Musk OWNED On Twitter After Attacking Bernie
# https://www.youtube.com/watch?v=Luu7a6zrLHY
YouTubeTranscriptApi.get_transcript('Luu7a6zrLHY')
```

```
[51]: [{'text': '-We are honored to have\nour first guest with us tonight.',
      'start': 1.534,
      'duration': 3.434},
      {'text': 'He served for 36 years\nas a senator from Delaware.',
      'start': 4.968,
      'duration': 3.799},
      {'text': 'He was Vice President\nunder the Obama Administration,',
      'start': 8.767,
      'duration': 3.067},
      {'text': 'and now he serves as the 46th\nPresident of the United States.',
      'start': 11.834,
```

```

    'duration': 4.5},
{'text': 'This is his first time\non a late night program',
 'start': 16.334,
 'duration': 2.866},
{'text': 'as the sitting president.', 'start': 19.2, 'duration': 1.334},
{'text': 'Please welcome\nPresident Joe Biden!',
 'start': 20.534,
 'duration': 2.566},
{'text': '[ Cheers and applause ]', 'start': 23.1, 'duration': 1.501},
{'text': '[ The Roots play\n"Hail to the Chief" ]',
 'start': 24.601,
 'duration': 1.699},
{'text': '[ Laughs ] Yes!', 'start': 26.3, 'duration': 2.034},
{'text': '-Hey, Jimmy.\nHow are you, man?',
 'start': 28.334,
 'duration': 1.833},
{'text': '-[ Laughs ]', 'start': 30.167, 'duration': 2.701},
{'text': 'President Biden,\nI appreciate you being here.',
 'start': 32.868,
 'duration': 2.299},
{'text': "It's your first time on our show\nas President Biden,",
 'start': 35.167,
 'duration': 2.801},
{'text': "and I'm honored.", 'start': 37.968, 'duration': 1.733},
{'text': "-I'm honored to be on.\nThanks for inviting me.",
 'start': 39.701,
 'duration': 2.766},
{'text': '-We have lots to talk about.', 'start': 42.467, 'duration': 1.9},
{'text': "Let's start with\nsome good news here.",
 'start': 44.367,
 'duration': 1.766},
{'text': 'You passed\nyour infrastructure bill.',
 'start': 46.133,
 'duration': 3.034},
{'text': 'This is a big deal.', 'start': 49.167, 'duration': 0.966},
{'text': 'Passed!', 'start': 50.133, 'duration': 3.368},
{'text': 'What are some changes', 'start': 53.501, 'duration': 1.266},
{'text': 'that Americans can expect\nfrom passing the bill?',
 'start': 54.767,
 'duration': 2.9},
{'text': '-Well, you know, we used to be\nthe number one,',
 'start': 57.667,
 'duration': 2.8},
{'text': 'have the best infrastructure\nin the world --',
 'start': 60.467,
 'duration': 1.666},
{'text': 'highways, bridges,\nthe whole works.',

```

```

    'start': 62.133,
    'duration': 3.0},
{'text': 'And now we rank number 13\nin the world.',
 'start': 65.133,
 'duration': 3.468},
{'text': 'And that has a lot\nto do with quality of life.',
 'start': 68.601,
 'duration': 3.033},
{'text': 'It has a lot to do\nwith whether or not',
 'start': 71.634,
 'duration': 2.1},
{'text': "We're going to be competitive.",
 'start': 73.734,
 'duration': 1.266},
{'text': 'For example,\none thing it does --',
 'start': 75.0,
 'duration': 1.634},
{'text': "We're going to replace\nall the lead pipes in America",
 'start': 76.634,
 'duration': 2.967},
{'text': 'that are causing people\nto get sick, get cancer,',
 'start': 79.601,
 'duration': 2.966},
{'text': 'and kids to be able\nto drink clean water.',
 'start': 82.567,
 'duration': 2.8},
{'text': "We're going to make sure\nthat we have",
 'start': 85.367,
 'duration': 1.9},
{'text': 'everywhere in the country,\nrural and urban and suburban,',
 'start': 87.267,
 'duration': 3.866},
{'text': 'we have Internet connections.', 'start': 91.133, 'duration': 1.501},
{'text': 'Because the world is changing,',
 'start': 92.634,
 'duration': 1.733},
{'text': "and we're so far\nbehind the curve.",
 'start': 94.367,
 'duration': 1.7},
{'text': "But the big thing\nit's going to do,",
 'start': 96.067,
 'duration': 1.3},
{'text': "it's going to create\nmillions of jobs,",
 'start': 97.367,
 'duration': 2.0},
{'text': "and it's going to change\nthe quality of life",
 'start': 99.367,

```

```

    'duration': 1.867},
{'text': 'for an awful lot of people\naround the country.',
 'start': 101.234,
 'duration': 1.799},
{'text': "-Yeah, that's what\nI'm talking about.",
 'start': 103.033,
 'duration': 1.1},
{'text': "[ Cheers and applause ]\nThat's great.",
 'start': 104.133,
 'duration': 3.768},
{'text': "And I know you've been\nworking hard",
 'start': 107.901,
 'duration': 1.266},
{'text': 'on the Build Back Better Plan.\n-Yes.',
 'start': 109.167,
 'duration': 2.933},
{'text': '-Some people may not\nfully understand',
 'start': 112.1,
 'duration': 1.434},
{'text': "what's actually in this plan.", 'start': 113.534, 'duration': 2.0},
{'text': 'Is there a simple way\nto break it down?',
 'start': 115.534,
 'duration': 2.067},
{'text': "-Yeah.\nWhat's in it is the ability",
 'start': 117.601,
 'duration': 2.2},
{'text': 'for working class\nand middle class people',
 'start': 119.801,
 'duration': 2.499},
{'text': 'to have the cost of living\ncut significantly.',
 'start': 122.3,
 'duration': 2.7},
{'text': "For example, there's money\nin there to make sure",
 'start': 125.0,
 'duration': 2.968},
{'text': 'that there is the ability\nto care for childcare.',
 'start': 127.968,
 'duration': 2.966},
{'text': "Right now,\nif you're in New York",
 'start': 130.934,
 'duration': 1.867},
{'text': "and you're a man or a woman", 'start': 132.801, 'duration': 1.966},
{'text': 'having childcare\nfor your children,',
 'start': 134.767,
 'duration': 1.834},
{'text': "you're paying $21,000 to $25,000\na year to be able to do that.",
 'start': 136.601,

```

```

    'duration': 4.566},
{'text': "Well, most people can't do that,",
 'start': 141.167,
 'duration': 1.6},
{'text': 'so people leave the work force.',
 'start': 142.767,
 'duration': 1.934},
{'text': "This will mean you don't have to\npay more than 7% of your income,",
 'start': 144.701,
 'duration': 3.633},
{'text': 'saving tens of thousands\nof dollars over time.',
 'start': 148.334,
 'duration': 3.267},
{'text': 'We also provided an ability,\nBuild Back Better Initiative,',
 'start': 151.601,
 'duration': 4.466},
{'text': "we're going to make sure\nyou're able to send",
 'start': 156.067,
 'duration': 1.867},
{'text': 'your 3- and 4- and 5-year-olds\nto school,',
 'start': 157.934,
 'duration': 2.8},
{'text': 'which will increase\nexponentially',
 'start': 160.734,
 'duration': 1.533},
{'text': 'their chance of succeeding.', 'start': 162.267, 'duration': 1.634},
{'text': 'All of the studies show, Jimmy,',
 'start': 163.901,
 'duration': 1.633},
{'text': 'that 47% better chance\nof graduating from high school,',
 'start': 165.534,
 'duration': 3.267},
{'text': 'no matter what the background\nyou come from,',
 'start': 168.801,
 'duration': 2.266},
{'text': 'no matter what home\nyou come from,',
 'start': 171.067,
 'duration': 2.233},
{'text': 'and to be able to go on\nto get to school.',
 'start': 173.3,
 'duration': 1.867},
{'text': 'What better can happen\nfor America',
 'start': 175.167,
 'duration': 2.4},
{'text': 'than have a better\neducated population?',
 'start': 177.567,
 'duration': 1.7},

```

```

{'text': '-Absolutely.', 'start': 179.267, 'duration': 1.367},
{'text': "-There's a whole range\nof things that are in there",
  'start': 180.634,
  'duration': 2.0},
{'text': 'that are going to really reduce,\nessentially,',
  'start': 182.634,
  'duration': 2.466},
{'text': 'the cost of living for people\nin a reasonable, rational way.',
  'start': 185.1,
  'duration': 4.434},
{'text': 'And by the way, every single bit\nof that bill is paid for.',
  'start': 189.534,
  'duration': 4.533},
{'text': 'No one making\nless than $400,000 a year',
  'start': 194.067,
  'duration': 2.534},
{'text': 'will pay\nan additional cent in taxes,',
  'start': 196.601,
  'duration': 4.066},
{'text': 'as well as it will not increase\nthe deficit one penny.',
  'start': 200.667,
  'duration': 4.867},
{'text': '-Do you think this will pass\nthe Senate before the year?',
  'start': 205.534,
  'duration': 4.233},
{'text': '-Well, I hope it will.', 'start': 209.767, 'duration': 1.433},
{'text': "I don't have a single Republican\nvote right now to pass it.",
  'start': 211.2,
  'duration': 3.467},
{'text': 'And, uh --\nWow!', 'start': 214.667, 'duration': 1.5},
{'text': "-So, it's going to be tough,\nbut I think we can get it done.",
  'start': 216.167,
  'duration': 3.3},
{'text': "If we don't get it, I'm going to\nkeep at it until we get it done.",
  'start': 219.467,
  'duration': 2.467},
{'text': '-I was at the Kennedy Center,',
  'start': 221.934,
  'duration': 2.034},
{'text': 'and I saw Republicans\nand Democrats in the crowd.',
  'start': 223.968,
  'duration': 4.0},
{'text': 'Everyone was laughing at jokes\nand listening to great music.',
  'start': 227.968,
  'duration': 3.266},
{'text': 'And everyone seemed\ncordial and nice.',
  'start': 231.234,

```

'duration': 2.667},
 {'text': 'And I go, "This seems like\nthis should be much easier',
 'start': 233.901,
 'duration': 3.033},
 {'text': 'to get along with everybody."',
 'start': 236.934,
 'duration': 1.767},
 {'text': '-[Laughs]', 'start': 238.701, 'duration': 1.167},
 {'text': '[Laughter]\n-Well, Jimmy, look,',
 'start': 239.868,
 'duration': 3.633},
 {'text': 'the truth is\nthat before COVID hit,',
 'start': 243.501,
 'duration': 4.866},
 {'text': 'the Trump arm\nof the Republican Party,',
 'start': 248.367,
 'duration': 4.534},
 {'text': 'which seems to be\nthe Republican Party now,',
 'start': 252.901,
 'duration': 3.933},
 {'text': 'did not even show up\nat the Kennedy Center.',
 'start': 256.834,
 'duration': 2.433},
 {'text': "They wouldn't go to the awards.",
 'start': 259.267,
 'duration': 1.8},
 {'text': 'So all the people you have\nthat we saw at that event,',
 'start': 261.067,
 'duration': 3.5},
 {'text': 'which was a very fulsome event,',
 'start': 264.567,
 'duration': 2.533},
 {'text': 'were people who in fact\nare Republicans,',
 'start': 267.1,
 'duration': 2.234},
 {'text': "but they're the kind --\nlike for example,",
 'start': 269.334,
 'duration': 2.9},
 {'text': 'a lot of Republicans\nand Democrats',
 'start': 272.234,
 'duration': 1.667},
 {'text': 'used to get\nreally on well together.',
 'start': 273.901,
 'duration': 1.766},
 {'text': 'For example, one of the reasons',
 'start': 275.667,
 'duration': 1.1},


```

{'text': 'I was a little bit late\ncoming on',
 'start': 276.767,
 'duration': 1.967},
{'text': 'is because -- not coming on,\nbut to do the taping here,',
 'start': 278.734,
 'duration': 3.399},
{'text': "is because\nI did Bob Dole's eulogy today.",
 'start': 282.133,
 'duration': 3.401},
{'text': 'He asked me on his deathbed\nwhether I would do his eulogy.',
 'start': 285.534,
 'duration': 2.933},
{'text': 'We were friends. We disagreed,\nbut we were friends.',
 'start': 288.467,
 'duration': 3.0},
{'text': 'We used to have an awful lot\nof that relationship,',
 'start': 291.467,
 'duration': 2.401},
{'text': 'and it still exists,\nexcept that the Qanon',
 'start': 293.868,
 'duration': 3.532},
{'text': 'and the extreme elements\nof the Republican Party',
 'start': 297.4,
 'duration': 2.301},
{'text': 'and what Donald Trump\nkeeps sort of, seems to me,',
 'start': 299.701,
 'duration': 4.267},
{'text': 'feeding, you know,\nwith the big lie,',
 'start': 303.968,
 'duration': 4.499},
{'text': 'it makes it awful hard.', 'start': 308.467, 'duration': 1.367},
{'text': 'And I think most Republicans --',
 'start': 309.834,
 'duration': 3.733},
{'text': "there's an awful lot\nof Republicans in Congress",
 'start': 313.567,
 'duration': 1.833},
{'text': 'I think would agree with that.', 'start': 315.4, 'duration': 1.401},
{'text': '-Yeah, I thought it was great\nto see you there.',
 'start': 316.801,
 'duration': 4.266},
{'text': 'We gave you a standing ovation\nbecause I go, "Here he is.',
 'start': 321.067,
 'duration': 2.434},
{'text': 'He\'s bringing class back.\nHe\'s a classy guy."',
 'start': 323.501,
 'duration': 2.933},

```

```

{'text': "And you're bringing class\nback to the office.",
 'start': 326.434,
 'duration': 1.966},
{'text': "-You're very kind to say that.\n-And I thought it was amazing",
 'start': 328.4,
 'duration': 1.633},
{'text': 'that you did that.\n-Thank you.',
 'start': 330.033,
 'duration': 2.768},
{'text': '-The COVID problem.', 'start': 332.801, 'duration': 2.033},
{'text': "I don't know what else to do,", 'start': 334.834, 'duration': 1.5},
{'text': 'and how to tell people\nto get the vaccine.',
 'start': 336.334,
 'duration': 3.933},
{'text': 'At this point, I think people\nare going to do',
 'start': 340.267,
 'duration': 1.833},
{'text': 'what they will, I guess.', 'start': 342.1, 'duration': 1.234},
{'text': "I don't know what to say.", 'start': 343.334, 'duration': 1.3},
{'text': 'Is it the youth\nthat we have to hope',
 'start': 344.634,
 'duration': 3.466},
{'text': 'the kids get us out of this?', 'start': 348.1, 'duration': 1.234},
{'text': 'Because, I mean, I got vaxxed,\nI got boosted,',
 'start': 349.334,
 'duration': 2.1},
{'text': 'I got whatever you got.', 'start': 351.434, 'duration': 1.9},
{'text': 'If you want me to wear\nred pants,',
 'start': 353.334,
 'duration': 1.6},
{'text': "I'll wear red pants.", 'start': 354.934, 'duration': 1.034},
{'text': 'I want this thing to go away.',
 'start': 355.968,
 'duration': 2.332},
{'text': '-Well, I think what we ought\nto do is have more people',
 'start': 358.3,
 'duration': 1.668},
{'text': 'listen to your song.', 'start': 359.968, 'duration': 1.599},
{'text': "-Yeah that's what\nI'm talking about, yeah.",
 'start': 361.567,
 'duration': 1.833},
{'text': '[ Applause ]', 'start': 363.4, 'duration': 1.468},
{'text': '-I think that will\nget them moving.',
 'start': 364.868,
 'duration': 1.633},
{'text': '-I gotta tell you --\nSo I had this song come out,',
 'start': 366.501,

```

```

    'duration': 4.433},
{'text': 'it\'s called "It Was\ na Masked Christmas" this week,',
 'start': 370.934,
 'duration': 2.567},
{'text': 'with Ariana Grande\ nand Megan Thee Stallion, and --',
 'start': 373.501,
 'duration': 3.4},
{'text': '[ Cheers and applause ]\ nThank you.',
 'start': 376.901,
 'duration': 1.766},
{'text': 'You were nice enough\ nto tweet about it --',
 'start': 378.667,
 'duration': 3.934},
{'text': '"\'Tis the season\ n(to get boosted)."',
 'start': 382.601,
 'duration': 1.666},
{'text': 'This is amazing\ nthat the president did this.',
 'start': 384.267,
 'duration': 3.367},
{'text': 'I am so excited\ nthat you did this.',
 'start': 387.634,
 'duration': 3.1},
{'text': 'It changed everything.', 'start': 390.734, 'duration': 1.033},
{'text': 'All these radio stations\ nwant the song.',
 'start': 391.767,
 'duration': 2.134},
{'text': "Now they're playing the song.",
 'start': 393.901,
 'duration': 2.533},
{'text': 'But you said\ nin another comment,',
 'start': 396.434,
 'duration': 1.9},
{'text': 'you said the message\ nis very important --',
 'start': 398.334,
 'duration': 2.267},
{'text': 'get your booster.', 'start': 400.601, 'duration': 1.9},
{'text': "-Well, it is.\ nIt's really important, Jimmy.",
 'start': 402.501,
 'duration': 1.866},
{'text': "Look, we're so far ahead\ nof the rest of the world",
 'start': 404.367,
 'duration': 3.633},
{'text': 'in terms\ nof the science and technology',
 'start': 408.0,
 'duration': 1.901},
{'text': 'to produce the vaccines.', 'start': 409.901, 'duration': 1.766},
{'text': 'And now we found out that', 'start': 411.667, 'duration': 2.466},

```

```

{'text': 'if you have your two shots\nfor Moderna or from Pfizer',
 'start': 414.133,
 'duration': 4.201},
{'text': 'or the one shot from J&J,\nwhat happens is,',
 'start': 418.334,
 'duration': 4.066},
{'text': 'this booster shot', 'start': 422.4, 'duration': 1.534},
{'text': 'not only brings\nthe levels back up of immunity,',
 'start': 423.934,
 'duration': 3.166},
{'text': 'I mean, to deal with fighting\noff the virus,',
 'start': 427.1,
 'duration': 2.967},
{'text': 'any of the viruses,', 'start': 430.067, 'duration': 1.467},
{'text': 'but it brings it up\nexponentially.',
 'start': 431.534,
 'duration': 2.499},
{'text': "So if you want to be sure\nyou're going to be okay --",
 'start': 434.033,
 'duration': 2.534},
{'text': 'And the other good news, Jimmy,',
 'start': 436.567,
 'duration': 1.6},
{'text': 'is that we now have\n5-, 6-, and 7-year-olds',
 'start': 438.167,
 'duration': 3.434},
{'text': 'are able to be vaccinated,\nas well as teenagers.',
 'start': 441.601,
 'duration': 3.566},
{'text': "And so we're moving.\nWe're moving.",
 'start': 445.167,
 'duration': 2.534},
{'text': "Now, I don't,\nthe politicalization of this,",
 'start': 447.701,
 'duration': 3.566},
{'text': 'making it become\na political statement',
 'start': 451.267,
 'duration': 2.701},
{'text': "if you get the shot and somehow,\nI don't know what the h--",
 'start': 453.968,
 'duration': 2.866},
{'text': 'Anyway.\n[ Laughter ]', 'start': 456.834, 'duration': 2.333},
{'text': "Here's the deal, Jimmy.", 'start': 459.167, 'duration': 4.866},
{'text': 'The bottom line is that the way\nto avoid this virus',
 'start': 464.033,
 'duration': 6.367},
{'text': 'is to get two shots\nand then get the booster shot.',

```

'start': 470.4,
 'duration': 4.201},
 {'text': "It's available.\nWe have the medicines available.",
 'start': 474.601,
 'duration': 3.133},
 {'text': 'It will make\na gigantic difference.',
 'start': 477.734,
 'duration': 2.2},
 {'text': "And even if you don't care\nso much about yourself",
 'start': 479.934,
 'duration': 2.466},
 {'text': 'think about your kids,\nyour brother',
 'start': 482.4,
 'duration': 1.267},
 {'text': 'your sister, your co-worker.', 'start': 483.667, 'duration': 2.301},
 {'text': 'And this stuff about Biden', 'start': 485.968, 'duration': 2.065},
 {'text': 'is mandating\nthese things happening',
 'start': 488.033,
 'duration': 1.801},
 {'text': "and it's un-American,", 'start': 489.834, 'duration': 2.134},
 {'text': 'look at it this way --', 'start': 491.968, 'duration': 1.132},
 {'text': "it's patriotic to get this done.",
 'start': 493.1,
 'duration': 2.434},
 {'text': 'Not a joke.', 'start': 495.534, 'duration': 1.133},
 {'text': "It's patriotic to get it done.\n-Yeah.",
 'start': 496.667,
 'duration': 2.466},
 {'text': 'Do you, um, how much do you pay\nattention to approval ratings?',
 'start': 499.133,
 'duration': 3.967},
 {'text': '-Well, not anymore.', 'start': 503.1, 'duration': 2.234},
 {'text': "I'm joking.", 'start': 505.334, 'duration': 1.267},
 {'text': '[Laughter]', 'start': 506.601, 'duration': 2.799},
 {'text': 'I was paying attention\nwhen it was in the mid 60s.',
 'start': 509.4,
 'duration': 2.834},
 {'text': "Now I'm in the mid 40s,\nI don't pay attention.",
 'start': 512.234,
 'duration': 2.433},
 {'text': '-I appreciate the honesty,\nyeah.',
 'start': 514.667,
 'duration': 2.433},
 {'text': 'I mean, because you came in --\nyou came in hot.',
 'start': 517.1,
 'duration': 3.1},
 {'text': "And then you got to medium,\nbut now you're at a low.",

'start': 520.2,
 'duration': 2.933},
 {'text': "But you've just got to keep\nyour head down, right,",
 'start': 523.133,
 'duration': 1.334},
 {'text': 'and just keep\ndoing the right thing?',
 'start': 524.467,
 'duration': 2.0},
 {'text': "-Well, look, here's the deal.",
 'start': 526.467,
 'duration': 1.501},
 {'text': "I think that it's --\nWe've been in less than a year.",
 'start': 527.968,
 'duration': 4.533},
 {'text': 'A lot has happened.', 'start': 532.501, 'duration': 1.433},
 {'text': 'And look, people are afraid.', 'start': 533.934, 'duration': 1.466},
 {'text': 'People are worried.', 'start': 535.4, 'duration': 1.501},
 {'text': 'And people are getting so much\ninaccurate information to them.',
 'start': 536.901,
 'duration': 5.299},
 {'text': "I don't mean about me,\nbut about the situation.",
 'start': 542.2,
 'duration': 3.467},
 {'text': "And so, you know, they're being\ntold that, you know,",
 'start': 545.667,
 'duration': 5.1},
 {'text': 'Armageddon is on the way.', 'start': 550.767, 'duration': 1.834},
 {'text': 'The truth is, the economy\nis growing more than it has',
 'start': 552.601,
 'duration': 2.833},
 {'text': 'any time in close to 60 years.',
 'start': 555.434,
 'duration': 2.267},
 {'text': 'The unemployment rate\nis down to 4.2%.',
 'start': 557.701,
 'duration': 2.432},
 {'text': "It's going to go lower,\nin my view.",
 'start': 560.133,
 'duration': 2.034},
 {'text': 'We do have inflation on things',
 'start': 562.167,
 'duration': 1.701},
 {'text': "that in fact matter\nto people's lives.",
 'start': 563.868,
 'duration': 2.833},
 {'text': 'You know, when you show up\nat a gas station',
 'start': 566.701,

'duration': 1.633},
 {'text': "and it's \$3.50 at your pump,", 'start': 568.334, 'duration': 2.6},
 {'text': 'although now because\nI took 50 million barrels,',
 'start': 570.934,
 'duration': 5.466},
 {'text': 'I called the oil petroleum\nreserve,',
 'start': 576.4,
 'duration': 3.468},
 {'text': 'the gasoline is down below \$3\nin many places.',
 'start': 579.868,
 'duration': 2.466},
 {'text': "It's going to come down.", 'start': 582.334, 'duration': 1.367},
 {'text': "It's going to move.", 'start': 583.701, 'duration': 1.2},
 {'text': 'But in the meantime,\npeople are worried.',
 'start': 584.901,
 'duration': 1.9},
 {'text': "And they're worried about\nwhether or not",
 'start': 586.801,
 'duration': 2.766},
 {'text': 'they have a brother,\nsister, uncle, aunt',
 'start': 589.567,
 'duration': 2.401},
 {'text': 'who has gotten COVID,', 'start': 591.968, 'duration': 1.466},
 {'text': "and they don't know\nwhat to do about it.",
 'start': 593.434,
 'duration': 1.534},
 {'text': 'I mean, so there is\na lot of anxiety,',
 'start': 594.968,
 'duration': 2.466},
 {'text': 'and my job is to be\nstraightforward,',
 'start': 597.434,
 'duration': 2.633},
 {'text': 'shoot from the shoulder,', 'start': 600.067, 'duration': 1.467},
 {'text': 'let people know exactly\nwhat the truth is,',
 'start': 601.534,
 'duration': 2.566},
 {'text': "and lay out how I'm going to try\nto make life better for them.",
 'start': 604.1,
 'duration': 3.9},
 {'text': '-Yeah. Does anyone really\nunderstand how hard your job is?',
 'start': 608.0,
 'duration': 4.701},
 {'text': 'Like, who, I mean --\nbesides my uncle, apparently',
 'start': 612.701,
 'duration': 3.733},
 {'text': 'who knows everything.', 'start': 616.434, 'duration': 1.933},
 {'text': 'He knows everything.\nBut I mean --',

'start': 618.367,
 'duration': 1.666},
 {'text': '-Well, send him over.', 'start': 620.033, 'duration': 1.267},
 {'text': '[Laughter]', 'start': 621.3, 'duration': 1.501},
 {'text': "-Yeah, no, you don't\nwant to talk to this guy.",
 'start': 622.801,
 'duration': 2.366},
 {'text': '[Laughter]', 'start': 625.167, 'duration': 1.4},
 {'text': 'But I mean, even just being\nin D.C. the other day',
 'start': 626.567,
 'duration': 2.934},
 {'text': 'and driving around,', 'start': 629.501, 'duration': 1.233},
 {'text': 'you see all these giant\nbuildings and you just --',
 'start': 630.734,
 'duration': 1.967},
 {'text': 'you feel kind of nervous\nand you feel like --',
 'start': 632.701,
 'duration': 2.167},
 {'text': "I feel important when I'm there,",
 'start': 634.868,
 'duration': 1.432},
 {'text': "but you're the leader\nof our country.",
 'start': 636.3,
 'duration': 2.668},
 {'text': 'I mean, the pressure.', 'start': 638.968, 'duration': 1.666},
 {'text': 'And then you have to do\n20 events',
 'start': 640.634,
 'duration': 2.9},
 {'text': "during the day\nthat we don't even see.",
 'start': 643.534,
 'duration': 1.833},
 {'text': '-Well, look, I was\nVice President for eight years.',
 'start': 645.367,
 'duration': 2.666},
 {'text': "I knew what the job was,\nand I don't --",
 'start': 648.033,
 'duration': 2.568},
 {'text': 'no one should feel sorry for me.',
 'start': 650.601,
 'duration': 1.499},
 {'text': "It's the greatest honor anybody\ncan have in their whole life,",
 'start': 652.1,
 'duration': 2.868},
 {'text': 'in my view, is to be\nPresident of the United States.',
 'start': 654.968,
 'duration': 1.799},
 {'text': 'But what the difference is --',

'start': 656.767,
 'duration': 3.101},
 {'text': "the difference is that\nwhen you're in Congress --",
 'start': 659.868,
 'duration': 4.532},
 {'text': 'and I had important jobs\nin the Congress as a senator.',
 'start': 664.4,
 'duration': 2.401},
 {'text': 'I was chairman\nof important committees.',
 'start': 666.801,
 'duration': 2.366},
 {'text': 'But being President is like\nbeing chairman',
 'start': 669.167,
 'duration': 2.1},
 {'text': 'of every single committee.', 'start': 671.267, 'duration': 1.8},
 {'text': "Nothing gets to my desk unless\nit hasn't been able to be solved",
 'start': 673.067,
 'duration': 4.033},
 {'text': 'before it got to my desk.', 'start': 677.1, 'duration': 1.701},
 {'text': "So by definition,\nit's more difficult.",
 'start': 678.801,
 'duration': 2.566},
 {'text': 'Harry Truman had a great line.', 'start': 681.367, 'duration': 2.1},
 {'text': 'He said about being president,\nhe said, "The buck stops here."',
 'start': 683.467,
 'duration': 3.967},
 {'text': 'I used to be -- I kid at Barack,',
 'start': 687.434,
 'duration': 2.333},
 {'text': 'President Obama,\nwho was a good friend',
 'start': 689.767,
 'duration': 2.134},
 {'text': 'and we served eight years\nas friends together.',
 'start': 691.901,
 'duration': 2.633},
 {'text': 'And I used to say to him --\ninstead of him thanking me',
 'start': 694.534,
 'duration': 3.0},
 {'text': 'I got to be the last guy\nin the room to give him advice.',
 'start': 697.534,
 'duration': 2.367},
 {'text': 'That was the deal we had.', 'start': 699.901, 'duration': 1.566},
 {'text': 'And he\nd thank me, and I get\nto walk out and say, "Look',
 'start': 701.467,
 'duration': 3.134},
 {'text': 'I give you the advice,\nyou got to make the decision."',
 'start': 704.601,

```

    'duration': 2.399},
{'text': '-Yeah.\n-"And, you know, what happens',
 'start': 707.0,
 'duration': 2.601},
{'text': 'is going to be on you."', 'start': 709.601, 'duration': 1.599},
{'text': "But that's part of the deal.", 'start': 711.2, 'duration': 1.501},
{'text': 'But look, I have\nenormous confidence --',
 'start': 712.701,
 'duration': 2.933},
{'text': "You're going to realize\nwhat a naive guy I am.",
 'start': 715.634,
 'duration': 5.2},
{'text': 'But I really have faith\nat the end of the day',
 'start': 720.834,
 'duration': 3.4},
{'text': 'the American people\nare going to get it.',
 'start': 724.234,
 'duration': 2.033},
{'text': 'They understand.\nThey understand.',
 'start': 726.267,
 'duration': 2.4},
{'text': 'Given half a shot,\nthey have never, ever, ever,',
 'start': 728.667,
 'duration': 3.034},
{'text': 'ever let the country down.', 'start': 731.701, 'duration': 1.599},
{'text': "We've just to give them a shot.\n-That's right.",
 'start': 733.3,
 'duration': 2.401},
{'text': 'More with President Joe Biden\nwhen we come back, everybody.',
 'start': 735.701,
 'duration': 3.267}]

```

Most of the transcripts are automatically generated, and they do not include any punctuation. I first chose the political commentators and then downloaded transcripts from every video uploaded on YouTube. Only transcripts in English are downloaded. I aim to get videos from around 8000 content creators. Thus I use the already downloaded transcripts. When I downloaded the transcripts, given the size of the text files, I vectorized them and saved them as JSON files. Therefore, my project starts at the stage of processing these JSON files. However, I also attached a Jupyter Notebook that shows how I got the dataset.

Once I defined the 67 content creators, I imported all their video transcripts after January 2016. Following the literature that measures polarization, I aggregated all the videos of a content-creator in a quarter. In the end, I have 795 vectors. Each vector represents a quarter's production of a content creator. As seen from that figure, not all content creators produce at every period.

In the beginning, 195 thousand tokens were included. However, my computer could not handle some of the steps to process that dataset. Thus, I erased any term that included any character that is not a letter or something different from the plain letter (for example, accents and numbers).

I also deleted those words with a frequency of fewer than two occurrences in the whole dataset. These were primarily words that were included by mistake by the automatic generator. In the end, I have 63363 words.

1.3 Methodology

My data lack partisanship information that I can use as labels. Thus, I will use unsupervised learning algorithms to look at word frequencies. The first question is how to measure polarization. When labels exist, it is easy to identify clusters and from there, polarization could be estimated using that information, since we would look at how two well-defined groups behave.

Polarization means people think very differently (difference in means), and overlappings are not usual (cluster variance is small). Clusters that are homogeneous and separated represent a polarized situation. However, I do not have pre-existing labels in my dataset. Thus, my strategy is to construct two clusters and then measure polarization using these two clusters. I measure polarization following Mehlhaff (2021). I chose to use the cluster-polarization coefficient (CPC) because of its straightforward implementation to my data. This measure is designed to scale to high-dimensional analysis that incorporates distance and concentration of a distribution's component groups.

$$CPC = \frac{\sum_{k=1}^{n_k} \sum_{j=1}^{n_j} (\mu_{k[j]} - \mu_j)^2}{\sum_{k=1}^{n_k} \sum_{j=1}^{n_j} (\mu_{k[j]} - \mu_j)^2 + \sum_{k=1}^{n_k} \sum_{i \in C_k} \sum_{j=1}^{n_j} (x_{i[k][j]} - \mu_{k[j]})^2}$$

where each individual i in cluster k holds a position on dimension j .

To construct the two clusters, I follow five different strategies. First, I transform the document-term matrix to a tf-idf matrix. Given that some words are very frequent in speech, I prefer to give them less relevance than those that could be the ones that determine the conflict.

1.3.1 NMF

I use Non-Negative Matrix Factorization (NMF) at every quarter with two topics. The entry j in the row i of the U matrix for a specific quarter correspond to the nonnegative components of the content-creator i in that quarter for the topic j . In my case, topics are what I consider as sources of conflict (or soft clusters) in the political dimension. If a document belongs to cluster j on a strong way, then it will have a very positive coordinate in the entry j of the row i . The coordinate will be zero or close to it if the document is not well represented by that cluster.

To project these coordinates to a latent ideological position, I calculate the angle of the row vector of the U matrix (that represents document location across the two topics) (Lahoti, 2018). The measure goes between 0 and 1. It is equal to 0 when it has 0 entry on the 1st column and 1 when it has a 0 entry on the second column. Once I have these projections, I cluster at every quarter, either using hierarchical clustering or k-means. Once I cluster and have the distances to the centroid of the clusters on this $[0,1]$ interval, I can compute the CPC.

As a recap, these are the steps:

0. Transform the document-term matrix into the tf-idf at each quarter
1. factorize the matrix at each quarter to reduce the dimensionality to two clusters.
2. With the U matrix I project each creator to the $[0,1]$ interval.
3. Clusterize the estimations in step 2 in two clusters, using k-means and hierarchical clustering (two different outcomes)

4. With the estimations in step 2 and clusters from step 3 I calculate a CPC for each quarter.

1.3.2 Hierarchical Clustering

Second, I cluster at every quarter using the term-document matrix for that period. I do that using hierarchical clustering. Once I have two clusters, I calculate the CCP using the term-document matrix and the terms as the dimensions for the estimation.

As a recap, these are the steps: 0. Transform the document-term matrix into the tf-idf at each quarter 1. Clusterize the tf-idf in step 2 in two clusters 2. With the tf-idf, taking each term as a dimension, and clusters from step 1 I calculate a CPC for each quarter.

1.3.3 Wordfish

Finally, to compare the results of the previous methods, I ran the R code to apply the Wordfish method (Slapin and Proksch, 2006). In this case, I use the matrix with the word counts instead of the tf-idf. After getting ideal positions for each creator-quarter, I cluster them into two groups (using either hierarchical clustering or k-means) and compute CPC at each quarter.

As a recap, these are the steps: 1. Calculate latent positions for content-creators at each quarter 2. Clusterize the estimations in step 2 in two clusters, using k-means and hierarchichal clustering (two difwerent outcomes) 3. With the estimations in step 1 and clusters from step 2 I calculate a CPC for each quarter.

1.3.4 Validation

In order to validate, I will look at the cluster purity measure. In the five mentioned methods, I always end up having two groups. I would expect those clusters represent the left-right dimension of conflict. Thus, I compare the cluster that emerges from my methods with the labels from transparency.tube.

1.3.5 Processing Data

```
[2]: gf = pd.read_pickle('foo')
```

```
[3]: D=gf.to_numpy()
      E=D.transpose()
      E
```

```
[3]: array([[0., 0., 0., ..., 0., 0., 0.],
            [0., 0., 0., ..., 0., 0., 0.],
            [0., 0., 0., ..., 0., 0., 0.],
            ...,
            [0., 0., 0., ..., 0., 0., 0.],
            [0., 0., 0., ..., 0., 0., 0.],
            [0., 0., 0., ..., 0., 0., 0.]])
```

```
[4]: ### Index list is id-month-year. I need to extract month and year from it.
      ### The output is a new column with the quarter: Q0 is first quarter of 2016.
      g1=gf.transpose()
```

```

gl = gl[gl.columns[gl.sum()>2]]
li=gl.index.values.tolist()
di=[k.split('-') for k in li]
for k in range(0,len(di)):
    if len(di[k])==4:
        di[k][0 : 2]='-'.join(di[k][0 : 2])
    elif len(di[k])==5:
        di[k][0 : 3]='-'.join(di[k][0 : 3])
    elif len(di[k])==6:
        di[k][0 : 4]='-'.join(di[k][0 : 4])

t=0
for j in range(2016,2022):
    for i in range(0,4):
        for k in range(0,len(di)):
            if di[k][1]==str(i) and di[k][2]==str(j):
                f='Q'+str(t)
                di[k].append(f)
            t+=1

q=list(range(0,len(di)))
for k in range(0,len(di)):
    q[k]=di[k][3]
gl['Quarter'] = q

```

1.3.6 Projections into the dimension of conflict and clustering

```

[6]: ## Quarter data + Tf-idf
tfidf_transformer = TfidfTransformer()
Dict = {}
t=0
for j in range(2016,2022):
    for i in range(0,4):
        f='Q'+str(t)
        A=gl[gl['Quarter'] == f]
        A = A.drop(columns=['Quarter'])
        A_tfidf = tfidf_transformer.fit_transform(A)
        gl[gl['Quarter'] == f].index.values.tolist()
        Dict[f] = [A, A_tfidf, gl[gl['Quarter'] == f].index.values.tolist()]
        t+=1

```

```

[7]: ## NMF
# From this I get a dictionary with quarters, the U matrix, the projections_
→into the L/R dimension and the clustering over it.
nmf = NMF(n_components=2, solver="mu", max_iter=2000, tol=1e-10, beta_loss=1)
tfidf_transformer = TfidfTransformer()

```

```

DictNMF = {}
t=0
for j in range(2016,2022):
    for i in range(0,4):
        f= 'Q'+str(t)
        E= Dict[f][1].toarray()
        np.random.seed(42)
        Unnmf = nmf.fit_transform(E)
        VTnnmf= nmf.components_
        m=[None] * len(Unnmf)
        nmfhard=[None] * len(Unnmf)
        for j in range(0,len(Unnmf)):
            m[j]=np.arctan(Unnmf[j][0]*Unnmf[j][1])/math.pi
            nmfhard[j]=Unnmf[j][0]>Unnmf[j][1]
        linked = linkage(np.reshape(m, (len(m), 1)), 'ward')
        kmeans = KMeans(n_clusters=2, random_state=42).fit(np.reshape(m,
↪(len(m), 1)))
        DictNMF[f] = [E, Unnmf, m, nmfhard, linked, kmeans]
        t+=1

```

```

[8]: ## Hierarchical clustering
# From this I get a dictionary with quarters and the clustering over tf-idf
↪matrix for each quarter.
Dict_hi = {}
t=0
for j in range(2016,2022):
    for i in range(0,4):
        f='Q'+str(t)
        A=gl[gl['Quarter'] == f]
        A = A.drop(columns=['Quarter'])
        A_tfidf = tfidf_transformer.fit_transform(A)
        A_m=A_tfidf.toarray()
        linked = linkage(A_m, 'ward')
        Dict_hi[f] = [A_m, linked ]
        t+=1

```

```

[9]: ## Wordfish
# From this I get a dictionary with quarters and the clustering over the
↪coefficients calculated in wordfish
df = pd.read_csv(r'values_wf.csv', header=None)
df=df.iloc[:, 1]
wf = {}
wf= [gl.index.values.tolist(),df, gl['Quarter']]
wf=pd.DataFrame.from_dict(wf).T
w={}
for k in range(0,24):
    t=str(k)

```

```

f='Q'+t
A=wf[wf[2] == f][1]
linked = linkage(np.reshape(list(A), (len(A), 1)), 'ward')
kmeans = KMeans(n_clusters=2, random_state=42).fit(np.reshape(list(A),
→(len(A), 1)))
w[f]=[A, linked, kmeans]

```

1.3.7 CPC

```

[10]: #NMF -> Projection -> Hierarchical Clustering
CPC_NMF_ph={}
for k in range(0,24):
    t=str(k)
    f='Q'+t
    z=0
    y=0
    yr=0
    ze=0

    for i in range(0,len(DictNMF[f][1])):
        if cut_tree(DictNMF[f][4], n_clusters=[2])[i]==1:
            y += 1
            yr += DictNMF[f][2][i]
        elif cut_tree(DictNMF[f][4], n_clusters=[2])[i]==0:
            ze += DictNMF[f][2][i]
            z += 1
    ce_0 = ze/z
    ce_1 = yr/y
    centroid = (ze + yr)/(z+y )
    di_ce=0
    for i in range(0,len(DictNMF[f][1])):
        if cut_tree(DictNMF[f][4], n_clusters=[2])[i]==1:
            di_ce += (DictNMF[f][2][i]-ce_1)**2
        elif cut_tree(DictNMF[f][4], n_clusters=[2])[i]==0:
            di_ce += (DictNMF[f][2][i]-ce_0)**2
    di_cl = (ce_1-centroid)**2 + (ce_0-centroid)**2
    CPC_NMF_ph[f]=di_ce/(di_ce + di_cl)

```

```

[11]: #NMF -> Projection -> K-means Clustering

```

```

CPC_NMF_pk={}
for k in range(0,24):
    t=str(k)
    f='Q'+t
    z=0
    y=0

```

```

ce_0 = DictNMF[f][5].cluster_centers_[0]
ce_1 = DictNMF[f][5].cluster_centers_[1]
di_ce=0
for i in range(0,len(DictNMF[f][1])):
    if DictNMF[f][5].labels_[i]==1:
        y += 1
        di_ce += (DictNMF[f][2][i].item()-ce_1)**2
    elif DictNMF[f][5].labels_[i]==0:
        z += 1
        di_ce += (DictNMF[f][2][i]-ce_0)**2
centroid = (ce_0*z+ce_1*y)/(z+y)
di_cl = (ce_1-centroid)**2 + (ce_0-centroid)**2
CPC_NMF_pk[f]=di_ce/(di_ce + di_cl)

```

[14]: *#NMF -> Hierarchical Clustering*

```

t1_start = process_time()

CPC_h={}
for k in range(0,24):
    t=str(k)
    f='Q'+t

    A=pd.DataFrame(Dict_hi[f][0]).loc[:, pd.DataFrame(Dict_hi[f][0]).
↪sum(axis=0)>0]

    B = {}
    lat_list = [item for sublist in cut_tree(Dict_hi[f][1], n_clusters=[2]) for
↪item in sublist]
    di_cl=[0.0 * len(A.loc[0,:])]
    di_ce0=[0.0 * len(A.loc[0,:])]
    di_ce1=[0.0 * len(A.loc[0,:])]
    for j in range(0,len(A.loc[0,:])): #loop over words
        ce_0 = A.iloc[:,j].groupby(lat_list).mean()[0]
        ce_1 = A.iloc[:,j].groupby(lat_list).mean()[1]
        centroid = A.iloc[:,j].mean()
        di_cl[j] = (ce_1-centroid)**2 + (ce_0-centroid)**2
        di_ce1[j] = (A.iloc[:,j]-ce_1)**2
        di_ce0[j] = (A.iloc[:,j]-ce_0)**2
    x = pd.DataFrame(di_ce1).T.groupby(lat_list).sum().iloc[1,:].sum() + pd.
↪DataFrame(di_ce0).T.groupby(lat_list).sum().iloc[0,:].sum()
    q = np.sum(di_cl)
    CPC_h[f]=x/(x + q)

```

[12]: *#Wordfish -> Hierarchical Clustering*

```

CPC_wf_h={}

```



```

for k in range(0,24):
    t=str(k)
    f='Q'+t
    z=0
    y=0
    yr=0
    ze=0

    for i in range(0,len(wf[wf[2] == f][1])):
        if cut_tree(w[f][1], n_clusters=[2])[i]==1:
            y += 1
            yr += wf[wf[2] == f][1][i:i+1].item()
        elif cut_tree(w[f][1], n_clusters=[2])[i]==0:
            ze += wf[wf[2] == f][1][i:i+1].item()
            z += 1
    ce_0 = ze/z
    ce_1 = yr/y
    centroid = (ze + yr)/(z+y )
    di_ce=0
    for i in range(0,len(wf[wf[2] == f][1])):
        if cut_tree(w[f][1], n_clusters=[2])[i]==1:
            di_ce += (wf[wf[2] == f][1][i:i+1].item()-ce_1)**2
        elif cut_tree(w[f][1], n_clusters=[2])[i]==0:
            di_ce += (wf[wf[2] == f][1][i:i+1].item()-ce_0)**2
    di_cl = (ce_1-centroid)**2 + (ce_0-centroid)**2
    CPC_wf_h[f]=di_ce/(di_ce + di_cl)

```

```

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:15:
FutureWarning: `item` has been deprecated and will be removed in a future
version
    from ipykernel import kernelapp as app
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:13:
FutureWarning: `item` has been deprecated and will be removed in a future
version
    del sys.path[0]
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:25:
FutureWarning: `item` has been deprecated and will be removed in a future
version
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:23:
FutureWarning: `item` has been deprecated and will be removed in a future
version

```

[13]: *#Wordfish -> K-Means Clustering*

```

CPC_wf_k={}
for k in range(0,24):
    t=str(k)

```

```

f='Q'+t
z=0
y=0
ce_0 = w[f][2].cluster_centers_[0]
ce_1 = w[f][2].cluster_centers_[1]
di_ce=0
for i in range(0,len(w[f][0])):
    if w[f][2].labels_[i]==1:
        y += 1
        di_ce += (w[f][0][i:i+1].item()-ce_1)**2
    elif w[f][2].labels_[i]==0:
        z += 1
        di_ce += (w[f][0][i:i+1].item()-ce_0)**2
centroid = (ce_0*z+ce_1*y)/(z+y)
di_cl = (ce_1-centroid)**2 + (ce_0-centroid)**2
CPC_wf_k[f]=di_ce/(di_ce + di_cl)

```

```

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:16:
FutureWarning: `item` has been deprecated and will be removed in a future
version
    app.launch_new_instance()
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:13:
FutureWarning: `item` has been deprecated and will be removed in a future
version
    del sys.path[0]

```

1.4 Results

I could only get from the results that polarization has not dramatically changed over the last five years. In all measures, we see some fluctuations, but the trend seems to be constant. Using Wordfish, polarization seems to be cyclical, and we are in an upward trend now. However, this is not something we can see with the other three measures.

Thus, it seems complicated to support a statement that we see rising polarization and bias on social media content and this is robust to the different strategies. Although this is a small sample of content creators, there is still no evidence that shows that polarization is increasing on the content. This result is surprising given that, as mentioned in the introduction, YouTube is considered a source of radicalization.

The result means that the polarization in vocabulary use between the two most prominent groups (in terms of these differences) has not increased. There are two factors that affect this measure: distance between clusters and homogeneity of each of both clusters. Future work could look at both dimensions and see if both do not show any change, or maybe they went in different directions.

```

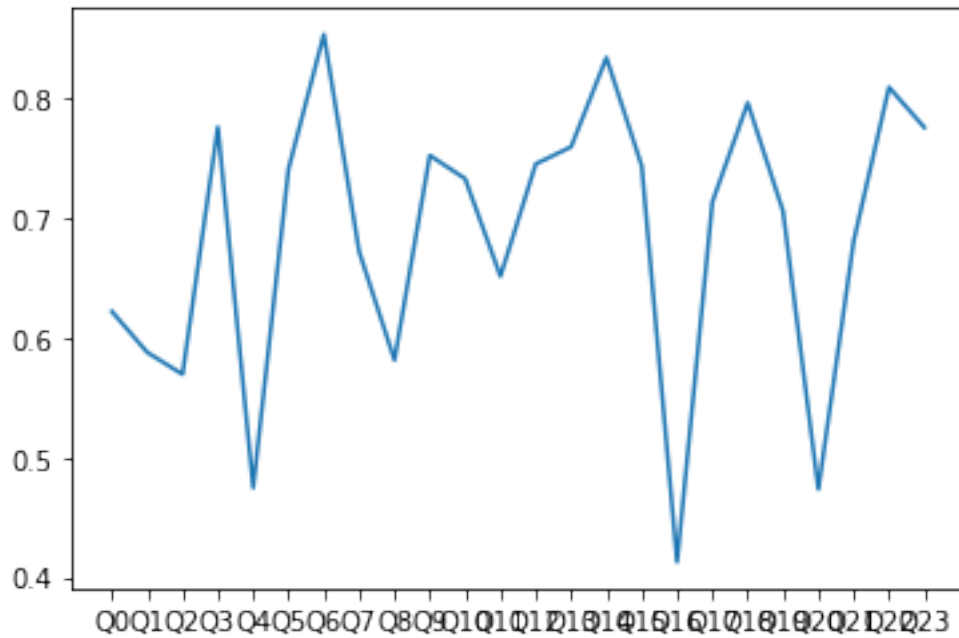
[17]: plt.plot(list(CPC_NMF_ph.keys()),list(CPC_NMF_ph.values()))
      plt.title("CPC for NMF -> Projection -> Hierarchichal Clustering")

```

```

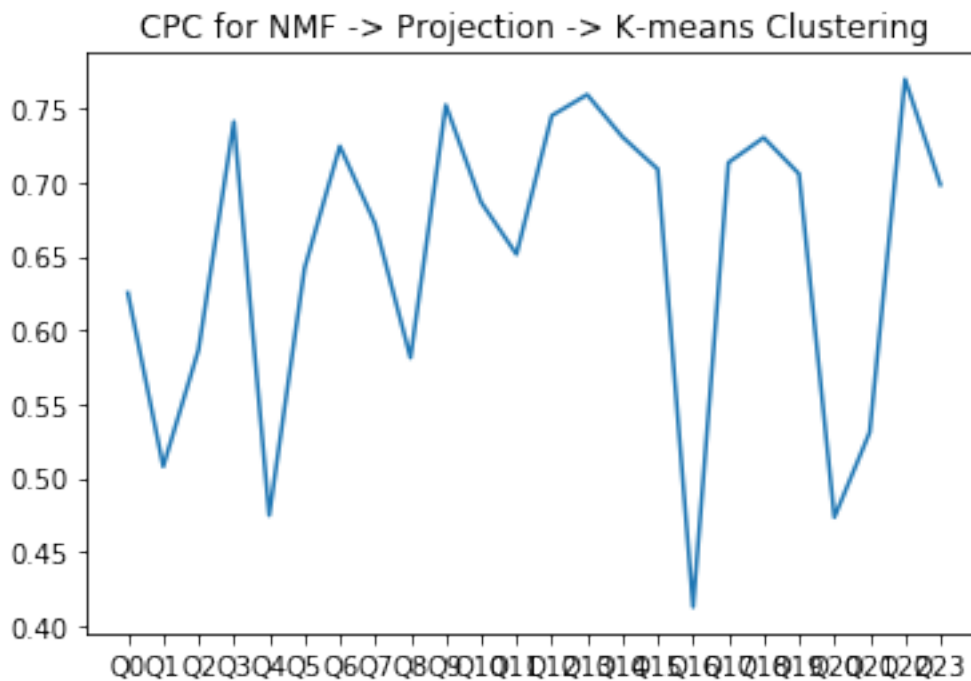
[17]: [<matplotlib.lines.Line2D at 0x2508dd05f48>]

```



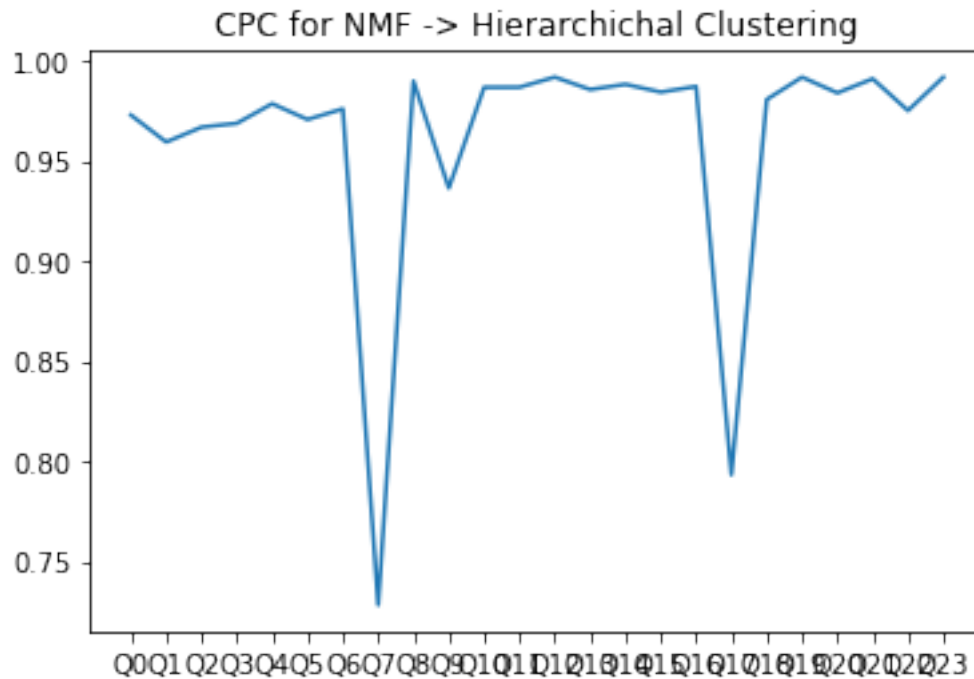
```
[46]: plt.plot(list(CPC_NMF_pk.keys()),list(CPC_NMF_pk.values()))
plt.title("CPC for NMF -> Projection -> K-means Clustering")
```

```
[46]: Text(0.5, 1.0, 'CPC for NMF -> Projection -> K-means Clustering')
```



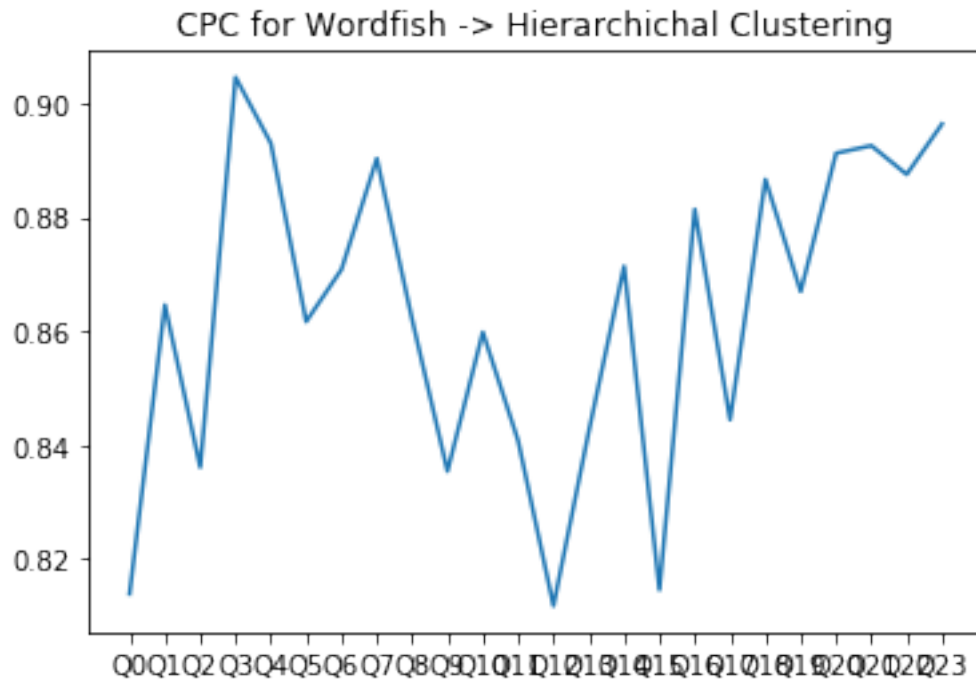
```
[47]: plt.plot(list(CPC_h.keys()),list(CPC_h.values()))
plt.title("CPC for NMF -> Hierarchichal Clustering")
```

```
[47]: Text(0.5, 1.0, 'CPC for NMF -> Hierarchichal Clustering')
```



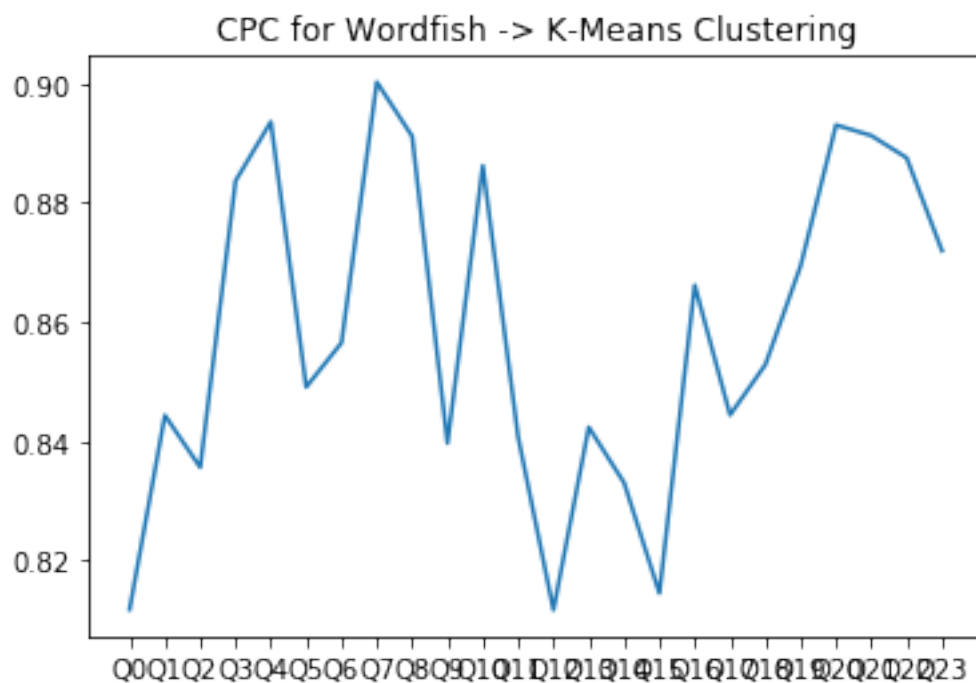
```
[48]: plt.plot(list(CPC_wf_h.keys()),list(CPC_wf_h.values()))
plt.title("CPC for Wordfish -> Hierarchichal Clustering")
```

```
[48]: Text(0.5, 1.0, 'CPC for Wordfish -> Hierarchichal Clustering')
```



```
[49]: plt.plot(list(CPC_wf_k.keys()),list(CPC_wf_k.values()))
plt.title("CPC for Wordfish -> K-Means Clustering")
```

```
[49]: Text(0.5, 1.0, 'CPC for Wordfish -> K-Means Clustering')
```



1.5 Validation

```
[20]: df = pd.read_csv(r'channels_lab.csv')
```

```
[21]: [item for sublist in cut_tree(Dict_hi[f][1], n_clusters=[2]) for item in
      ↪sublist];
```

```
[22]: li=gl.index.values.tolist()
      li
```

```
[22]: ['UUORJJ_Wm7jyOU9eY10LgcwA-0-2016',
      'UUORJJ_Wm7jyOU9eY10LgcwA-1-2016',
      'UUORJJ_Wm7jyOU9eY10LgcwA-2-2016',
      'UUORJJ_Wm7jyOU9eY10LgcwA-3-2016',
      'UUORJJ_Wm7jyOU9eY10LgcwA-0-2017',
      'UUORJJ_Wm7jyOU9eY10LgcwA-1-2017',
      'UUORJJ_Wm7jyOU9eY10LgcwA-2-2017',
      'UUORJJ_Wm7jyOU9eY10LgcwA-3-2017',
      'UUORJJ_Wm7jyOU9eY10LgcwA-0-2018',
      'UUORJJ_Wm7jyOU9eY10LgcwA-1-2018',
      'UUORJJ_Wm7jyOU9eY10LgcwA-2-2018',
      'UUORJJ_Wm7jyOU9eY10LgcwA-3-2018',
      'UUORJJ_Wm7jyOU9eY10LgcwA-0-2019',
      'UUORJJ_Wm7jyOU9eY10LgcwA-1-2019',
      'UUORJJ_Wm7jyOU9eY10LgcwA-2-2019',
      'UUORJJ_Wm7jyOU9eY10LgcwA-3-2019',
      'UUORJJ_Wm7jyOU9eY10LgcwA-0-2020',
      'UUORJJ_Wm7jyOU9eY10LgcwA-1-2020',
      'UUORJJ_Wm7jyOU9eY10LgcwA-2-2020',
      'UUORJJ_Wm7jyOU9eY10LgcwA-3-2020',
      'UUORJJ_Wm7jyOU9eY10LgcwA-0-2021',
      'UUORJJ_Wm7jyOU9eY10LgcwA-1-2021',
      'UUORJJ_Wm7jyOU9eY10LgcwA-2-2021',
      'UUOrZoXAD5lxBHMsjrGwWWQ-0-2016',
      'UUOrZoXAD5lxBHMsjrGwWWQ-1-2016',
      'UUOrZoXAD5lxBHMsjrGwWWQ-2-2016',
      'UUOrZoXAD5lxBHMsjrGwWWQ-3-2016',
      'UUOrZoXAD5lxBHMsjrGwWWQ-0-2017',
      'UUOrZoXAD5lxBHMsjrGwWWQ-1-2017',
      'UUOrZoXAD5lxBHMsjrGwWWQ-2-2017',
      'UUOrZoXAD5lxBHMsjrGwWWQ-3-2017',
      'UUOrZoXAD5lxBHMsjrGwWWQ-0-2018',
      'UUOrZoXAD5lxBHMsjrGwWWQ-1-2018',
      'UUOrZoXAD5lxBHMsjrGwWWQ-2-2018',
```

'UUOrZoXAD5lXgBHMsjrGwWWQ-3-2018',
'UUOrZoXAD5lXgBHMsjrGwWWQ-0-2019',
'UUOrZoXAD5lXgBHMsjrGwWWQ-1-2019',
'UUOrZoXAD5lXgBHMsjrGwWWQ-2-2019',
'UUOrZoXAD5lXgBHMsjrGwWWQ-3-2019',
'UUOrZoXAD5lXgBHMsjrGwWWQ-0-2020',
'UUOrZoXAD5lXgBHMsjrGwWWQ-1-2020',
'UUOrZoXAD5lXgBHMsjrGwWWQ-2-2020',
'UUOrZoXAD5lXgBHMsjrGwWWQ-3-2020',
'UUOrZoXAD5lXgBHMsjrGwWWQ-0-2021',
'UUOrZoXAD5lXgBHMsjrGwWWQ-1-2021',
'UUOrZoXAD5lXgBHMsjrGwWWQ-2-2021',
'UUOrZoXAD5lXgBHMsjrGwWWQ-3-2021',
'UUOuVZd8N7FfIZnPu0y7o95A-0-2016',
'UUOuVZd8N7FfIZnPu0y7o95A-1-2016',
'UUOuVZd8N7FfIZnPu0y7o95A-2-2016',
'UUOuVZd8N7FfIZnPu0y7o95A-3-2016',
'UUOuVZd8N7FfIZnPu0y7o95A-0-2017',
'UUOuVZd8N7FfIZnPu0y7o95A-1-2017',
'UUOuVZd8N7FfIZnPu0y7o95A-2-2017',
'UUOuVZd8N7FfIZnPu0y7o95A-3-2017',
'UUOuVZd8N7FfIZnPu0y7o95A-0-2018',
'UUOuVZd8N7FfIZnPu0y7o95A-1-2018',
'UUOuVZd8N7FfIZnPu0y7o95A-2-2018',
'UUOuVZd8N7FfIZnPu0y7o95A-3-2018',
'UUOuVZd8N7FfIZnPu0y7o95A-0-2019',
'UUOuVZd8N7FfIZnPu0y7o95A-1-2019',
'UUOuVZd8N7FfIZnPu0y7o95A-2-2019',
'UUOuVZd8N7FfIZnPu0y7o95A-3-2019',
'UUOuVZd8N7FfIZnPu0y7o95A-0-2020',
'UUOuVZd8N7FfIZnPu0y7o95A-1-2020',
'UUOuVZd8N7FfIZnPu0y7o95A-2-2020',
'UUOuVZd8N7FfIZnPu0y7o95A-3-2020',
'UUOuVZd8N7FfIZnPu0y7o95A-0-2021',
'UUOuVZd8N7FfIZnPu0y7o95A-1-2021',
'UUOuVZd8N7FfIZnPu0y7o95A-2-2021',
'UUOuVZd8N7FfIZnPu0y7o95A-3-2021',
'UU17knbFWwBQOSaLCiT0_5jw-2-2016',
'UU17knbFWwBQOSaLCiT0_5jw-3-2016',
'UU17knbFWwBQOSaLCiT0_5jw-2-2020',
'UU1Y7SRZ1f6BNmBPakawhLXA-0-2017',
'UU1Y7SRZ1f6BNmBPakawhLXA-1-2017',
'UU1Y7SRZ1f6BNmBPakawhLXA-2-2017',
'UU1Y7SRZ1f6BNmBPakawhLXA-3-2017',
'UU1Y7SRZ1f6BNmBPakawhLXA-0-2018',
'UU1Y7SRZ1f6BNmBPakawhLXA-1-2018',
'UU1Y7SRZ1f6BNmBPakawhLXA-0-2020',

'UU1Y7SRZ1f6BNmBPakawhLXA-1-2020',
'UU1Y7SRZ1f6BNmBPakawhLXA-3-2020',
'UU2PA-AKmVpU6NKCGtZq_rKQ-0-2016',
'UU2PA-AKmVpU6NKCGtZq_rKQ-1-2016',
'UU2PA-AKmVpU6NKCGtZq_rKQ-2-2016',
'UU2PA-AKmVpU6NKCGtZq_rKQ-3-2016',
'UU2PA-AKmVpU6NKCGtZq_rKQ-0-2017',
'UU2PA-AKmVpU6NKCGtZq_rKQ-1-2017',
'UU2PA-AKmVpU6NKCGtZq_rKQ-2-2017',
'UU2PA-AKmVpU6NKCGtZq_rKQ-3-2017',
'UU2PA-AKmVpU6NKCGtZq_rKQ-0-2018',
'UU2PA-AKmVpU6NKCGtZq_rKQ-1-2018',
'UU2PA-AKmVpU6NKCGtZq_rKQ-2-2018',
'UU2PA-AKmVpU6NKCGtZq_rKQ-3-2018',
'UU2PA-AKmVpU6NKCGtZq_rKQ-0-2019',
'UU2PA-AKmVpU6NKCGtZq_rKQ-1-2019',
'UU2PA-AKmVpU6NKCGtZq_rKQ-2-2019',
'UU2PA-AKmVpU6NKCGtZq_rKQ-3-2019',
'UU2PA-AKmVpU6NKCGtZq_rKQ-0-2020',
'UU2PA-AKmVpU6NKCGtZq_rKQ-1-2020',
'UU2PA-AKmVpU6NKCGtZq_rKQ-2-2020',
'UU2PA-AKmVpU6NKCGtZq_rKQ-3-2020',
'UU2PA-AKmVpU6NKCGtZq_rKQ-0-2021',
'UU2PA-AKmVpU6NKCGtZq_rKQ-1-2021',
'UU2PA-AKmVpU6NKCGtZq_rKQ-2-2021',
'UU2PA-AKmVpU6NKCGtZq_rKQ-3-2021',
'UU2w1CQqPli4w94-PCg2UvJQ-1-2021',
'UU2w1CQqPli4w94-PCg2UvJQ-3-2021',
'UU3dZNn_1-kP76SmIqy4rHSw-0-2016',
'UU3dZNn_1-kP76SmIqy4rHSw-1-2016',
'UU3dZNn_1-kP76SmIqy4rHSw-2-2016',
'UU3dZNn_1-kP76SmIqy4rHSw-3-2016',
'UU3dZNn_1-kP76SmIqy4rHSw-1-2017',
'UU3dZNn_1-kP76SmIqy4rHSw-2-2017',
'UU3dZNn_1-kP76SmIqy4rHSw-3-2017',
'UU3dZNn_1-kP76SmIqy4rHSw-2-2021',
'UU3FD64RRsrCLpiZNkq7ZkSg-0-2016',
'UU3FD64RRsrCLpiZNkq7ZkSg-1-2016',
'UU3FD64RRsrCLpiZNkq7ZkSg-2-2016',
'UU3FD64RRsrCLpiZNkq7ZkSg-3-2016',
'UU3FD64RRsrCLpiZNkq7ZkSg-0-2017',
'UU3FD64RRsrCLpiZNkq7ZkSg-1-2017',
'UU3FD64RRsrCLpiZNkq7ZkSg-2-2017',
'UU3FD64RRsrCLpiZNkq7ZkSg-3-2017',
'UU3FD64RRsrCLpiZNkq7ZkSg-0-2018',
'UU3FD64RRsrCLpiZNkq7ZkSg-2-2018',
'UU3FD64RRsrCLpiZNkq7ZkSg-0-2019',

'UU3FD64RRsrCLpiZNkq7ZkSg-1-2019',
'UU3FD64RRsrCLpiZNkq7ZkSg-2-2019',
'UU3FD64RRsrCLpiZNkq7ZkSg-3-2019',
'UU3FD64RRsrCLpiZNkq7ZkSg-0-2020',
'UU3FD64RRsrCLpiZNkq7ZkSg-1-2020',
'UU3FD64RRsrCLpiZNkq7ZkSg-2-2020',
'UU3FD64RRsrCLpiZNkq7ZkSg-3-2020',
'UU3FD64RRsrCLpiZNkq7ZkSg-0-2021',
'UU3FD64RRsrCLpiZNkq7ZkSg-1-2021',
'UU3FD64RRsrCLpiZNkq7ZkSg-2-2021',
'UU3FD64RRsrCLpiZNkq7ZkSg-3-2021',
'UU3M718ved_rYQ45AVzSORGA-0-2016',
'UU3M718ved_rYQ45AVzSORGA-1-2016',
'UU3M718ved_rYQ45AVzSORGA-2-2016',
'UU3M718ved_rYQ45AVzSORGA-3-2016',
'UU3M718ved_rYQ45AVzSORGA-0-2017',
'UU3M718ved_rYQ45AVzSORGA-1-2017',
'UU3M718ved_rYQ45AVzSORGA-2-2017',
'UU3M718ved_rYQ45AVzSORGA-3-2017',
'UU3M718ved_rYQ45AVzSORGA-0-2018',
'UU3M718ved_rYQ45AVzSORGA-1-2018',
'UU3M718ved_rYQ45AVzSORGA-2-2018',
'UU3M718ved_rYQ45AVzSORGA-3-2018',
'UU3M718ved_rYQ45AVzSORGA-0-2019',
'UU3M718ved_rYQ45AVzSORGA-1-2019',
'UU3M718ved_rYQ45AVzSORGA-2-2019',
'UU3M718ved_rYQ45AVzSORGA-3-2019',
'UU3M718ved_rYQ45AVzSORGA-0-2020',
'UU3M718ved_rYQ45AVzSORGA-1-2020',
'UU3M718ved_rYQ45AVzSORGA-2-2020',
'UU3M718ved_rYQ45AVzSORGA-3-2020',
'UU3M718ved_rYQ45AVzSORGA-0-2021',
'UU3M718ved_rYQ45AVzSORGA-1-2021',
'UU3M718ved_rYQ45AVzSORGA-2-2021',
'UU3M718ved_rYQ45AVzSORGA-3-2021',
'UU3X81QvudZJ6fMsL7WwqEDw-0-2016',
'UU3X81QvudZJ6fMsL7WwqEDw-1-2016',
'UU3X81QvudZJ6fMsL7WwqEDw-2-2016',
'UU3X81QvudZJ6fMsL7WwqEDw-3-2016',
'UU3X81QvudZJ6fMsL7WwqEDw-0-2017',
'UU3X81QvudZJ6fMsL7WwqEDw-1-2017',
'UU3X81QvudZJ6fMsL7WwqEDw-2-2017',
'UU3X81QvudZJ6fMsL7WwqEDw-3-2017',
'UU3X81QvudZJ6fMsL7WwqEDw-0-2018',
'UU3X81QvudZJ6fMsL7WwqEDw-1-2018',
'UU3X81QvudZJ6fMsL7WwqEDw-2-2018',
'UU3X81QvudZJ6fMsL7WwqEDw-3-2018',

'UU3X81QvudZJ6fMsL7WwqEDw-0-2019',
'UU3X81QvudZJ6fMsL7WwqEDw-2-2019',
'UU3X81QvudZJ6fMsL7WwqEDw-3-2019',
'UU3X81QvudZJ6fMsL7WwqEDw-0-2020',
'UU3X81QvudZJ6fMsL7WwqEDw-1-2020',
'UU3X81QvudZJ6fMsL7WwqEDw-2-2020',
'UU3X81QvudZJ6fMsL7WwqEDw-3-2020',
'UU3X81QvudZJ6fMsL7WwqEDw-3-2021',
'UU4-AGCUwX7C-D4BsBook9fQ-2-2016',
'UU4-AGCUwX7C-D4BsBook9fQ-3-2018',
'UU4-AGCUwX7C-D4BsBook9fQ-1-2019',
'UU4-AGCUwX7C-D4BsBook9fQ-2-2019',
'UU4-AGCUwX7C-D4BsBook9fQ-3-2019',
'UU4-AGCUwX7C-D4BsBook9fQ-0-2020',
'UU4-AGCUwX7C-D4BsBook9fQ-2-2020',
'UU4-AGCUwX7C-D4BsBook9fQ-3-2020',
'UU4-AGCUwX7C-D4BsBook9fQ-0-2021',
'UU4-AGCUwX7C-D4BsBook9fQ-1-2021',
'UU4-AGCUwX7C-D4BsBook9fQ-2-2021',
'UU4-AGCUwX7C-D4BsBook9fQ-3-2021',
'UU4GzVIbK8bria6A0y43hY4Q-0-2020',
'UU4GzVIbK8bria6A0y43hY4Q-1-2020',
'UU4GzVIbK8bria6A0y43hY4Q-3-2020',
'UU4GzVIbK8bria6A0y43hY4Q-0-2021',
'UU4GzVIbK8bria6A0y43hY4Q-1-2021',
'UU4GzVIbK8bria6A0y43hY4Q-2-2021',
'UU4GzVIbK8bria6A0y43hY4Q-3-2021',
'UU4YMONX86SgxJRf8y0lQ3gA-2-2018',
'UU4YMONX86SgxJRf8y0lQ3gA-3-2018',
'UU51_awyLXl2nxCduuzzmw4w-0-2016',
'UU51_awyLXl2nxCduuzzmw4w-1-2016',
'UU51_awyLXl2nxCduuzzmw4w-2-2016',
'UU51_awyLXl2nxCduuzzmw4w-3-2016',
'UU51_awyLXl2nxCduuzzmw4w-0-2017',
'UU51_awyLXl2nxCduuzzmw4w-1-2017',
'UU51_awyLXl2nxCduuzzmw4w-2-2017',
'UU51_awyLXl2nxCduuzzmw4w-3-2017',
'UU51_awyLXl2nxCduuzzmw4w-0-2018',
'UU51_awyLXl2nxCduuzzmw4w-1-2018',
'UU51_awyLXl2nxCduuzzmw4w-2-2018',
'UU51_awyLXl2nxCduuzzmw4w-3-2018',
'UU51_awyLXl2nxCduuzzmw4w-0-2019',
'UU51_awyLXl2nxCduuzzmw4w-1-2019',
'UU51_awyLXl2nxCduuzzmw4w-2-2019',
'UU51_awyLXl2nxCduuzzmw4w-3-2019',
'UU51_awyLXl2nxCduuzzmw4w-0-2020',
'UU51_awyLXl2nxCduuzzmw4w-1-2020',

'UU51_awyLXl2nxCduuzzmw4w-2-2020',
'UU51_awyLXl2nxCduuzzmw4w-3-2020',
'UU51_awyLXl2nxCduuzzmw4w-0-2021',
'UU51_awyLXl2nxCduuzzmw4w-1-2021',
'UU51_awyLXl2nxCduuzzmw4w-2-2021',
'UU51_awyLXl2nxCduuzzmw4w-3-2021',
'UU554eY5jNUfDq3yD0JYir0Q-0-2016',
'UU554eY5jNUfDq3yD0JYir0Q-1-2016',
'UU554eY5jNUfDq3yD0JYir0Q-2-2016',
'UU554eY5jNUfDq3yD0JYir0Q-3-2016',
'UU554eY5jNUfDq3yD0JYir0Q-0-2017',
'UU554eY5jNUfDq3yD0JYir0Q-1-2017',
'UU554eY5jNUfDq3yD0JYir0Q-2-2017',
'UU554eY5jNUfDq3yD0JYir0Q-3-2017',
'UU554eY5jNUfDq3yD0JYir0Q-0-2018',
'UU554eY5jNUfDq3yD0JYir0Q-1-2018',
'UU554eY5jNUfDq3yD0JYir0Q-2-2018',
'UU554eY5jNUfDq3yD0JYir0Q-3-2018',
'UU554eY5jNUfDq3yD0JYir0Q-0-2019',
'UU554eY5jNUfDq3yD0JYir0Q-1-2019',
'UU554eY5jNUfDq3yD0JYir0Q-2-2019',
'UU554eY5jNUfDq3yD0JYir0Q-3-2019',
'UU554eY5jNUfDq3yD0JYir0Q-0-2020',
'UU554eY5jNUfDq3yD0JYir0Q-1-2020',
'UU554eY5jNUfDq3yD0JYir0Q-2-2020',
'UU554eY5jNUfDq3yD0JYir0Q-3-2020',
'UU554eY5jNUfDq3yD0JYir0Q-0-2021',
'UU554eY5jNUfDq3yD0JYir0Q-1-2021',
'UU554eY5jNUfDq3yD0JYir0Q-2-2021',
'UU554eY5jNUfDq3yD0JYir0Q-3-2021',
'UU595wqznMGuY2mi6DKx-qnQ-0-2016',
'UU595wqznMGuY2mi6DKx-qnQ-1-2016',
'UU595wqznMGuY2mi6DKx-qnQ-2-2016',
'UU595wqznMGuY2mi6DKx-qnQ-3-2016',
'UU595wqznMGuY2mi6DKx-qnQ-0-2017',
'UU595wqznMGuY2mi6DKx-qnQ-1-2017',
'UU595wqznMGuY2mi6DKx-qnQ-2-2017',
'UU595wqznMGuY2mi6DKx-qnQ-3-2017',
'UU595wqznMGuY2mi6DKx-qnQ-0-2018',
'UU595wqznMGuY2mi6DKx-qnQ-1-2018',
'UU595wqznMGuY2mi6DKx-qnQ-2-2018',
'UU595wqznMGuY2mi6DKx-qnQ-3-2018',
'UU595wqznMGuY2mi6DKx-qnQ-0-2019',
'UU595wqznMGuY2mi6DKx-qnQ-1-2019',
'UU595wqznMGuY2mi6DKx-qnQ-2-2019',
'UU595wqznMGuY2mi6DKx-qnQ-3-2019',
'UU595wqznMGuY2mi6DKx-qnQ-0-2020',

'UU595wqznMGuY2mi6DKx-qnQ-1-2020',
'UU595wqznMGuY2mi6DKx-qnQ-2-2020',
'UU595wqznMGuY2mi6DKx-qnQ-3-2020',
'UU595wqznMGuY2mi6DKx-qnQ-0-2021',
'UU595wqznMGuY2mi6DKx-qnQ-1-2021',
'UU595wqznMGuY2mi6DKx-qnQ-2-2021',
'UU595wqznMGuY2mi6DKx-qnQ-3-2021',
'UU5bEfSFTYQVfLCwkhBt8NtQ-0-2016',
'UU5bEfSFTYQVfLCwkhBt8NtQ-1-2016',
'UU5bEfSFTYQVfLCwkhBt8NtQ-2-2016',
'UU5bEfSFTYQVfLCwkhBt8NtQ-3-2016',
'UU5bEfSFTYQVfLCwkhBt8NtQ-0-2017',
'UU5bEfSFTYQVfLCwkhBt8NtQ-1-2017',
'UU5bEfSFTYQVfLCwkhBt8NtQ-2-2017',
'UU5bEfSFTYQVfLCwkhBt8NtQ-3-2017',
'UU5bEfSFTYQVfLCwkhBt8NtQ-0-2018',
'UU5bEfSFTYQVfLCwkhBt8NtQ-1-2018',
'UU5bEfSFTYQVfLCwkhBt8NtQ-2-2018',
'UU5bEfSFTYQVfLCwkhBt8NtQ-3-2018',
'UU5bEfSFTYQVfLCwkhBt8NtQ-0-2019',
'UU5bEfSFTYQVfLCwkhBt8NtQ-1-2019',
'UU5bEfSFTYQVfLCwkhBt8NtQ-2-2019',
'UU5bEfSFTYQVfLCwkhBt8NtQ-3-2019',
'UU5bEfSFTYQVfLCwkhBt8NtQ-0-2020',
'UU5bEfSFTYQVfLCwkhBt8NtQ-1-2020',
'UU5bEfSFTYQVfLCwkhBt8NtQ-2-2020',
'UU5bEfSFTYQVfLCwkhBt8NtQ-3-2020',
'UU5bEfSFTYQVfLCwkhBt8NtQ-0-2021',
'UU5bEfSFTYQVfLCwkhBt8NtQ-1-2021',
'UU5bEfSFTYQVfLCwkhBt8NtQ-2-2021',
'UU5bEfSFTYQVfLCwkhBt8NtQ-3-2021',
'UU5fdssPqmmGhkhsJi4VcckA-0-2016',
'UU5fdssPqmmGhkhsJi4VcckA-1-2016',
'UU5fdssPqmmGhkhsJi4VcckA-2-2016',
'UU5fdssPqmmGhkhsJi4VcckA-0-2017',
'UU5fdssPqmmGhkhsJi4VcckA-2-2017',
'UU5fdssPqmmGhkhsJi4VcckA-3-2017',
'UU5fdssPqmmGhkhsJi4VcckA-0-2018',
'UU5fdssPqmmGhkhsJi4VcckA-1-2018',
'UU5fdssPqmmGhkhsJi4VcckA-2-2018',
'UU5fdssPqmmGhkhsJi4VcckA-3-2018',
'UU5fdssPqmmGhkhsJi4VcckA-0-2019',
'UU5fdssPqmmGhkhsJi4VcckA-2-2019',
'UU5fdssPqmmGhkhsJi4VcckA-3-2019',
'UU5fdssPqmmGhkhsJi4VcckA-1-2020',
'UU5fdssPqmmGhkhsJi4VcckA-2-2020',
'UU5fdssPqmmGhkhsJi4VcckA-0-2021',

'UU5fdssPqmmGhkhsJi4VcckA-1-2021',
'UU5fdssPqmmGhkhsJi4VcckA-2-2021',
'UU5ft8escJDZqT-QIWkHsImQ-2-2020',
'UU5tEELgWBfKbA9fVPRzBzPQ-3-2021',
'UU61qQBrHiNKprDXW5ZnLPyQ-3-2016',
'UU61qQBrHiNKprDXW5ZnLPyQ-0-2017',
'UU61qQBrHiNKprDXW5ZnLPyQ-1-2017',
'UU61qQBrHiNKprDXW5ZnLPyQ-0-2018',
'UU61qQBrHiNKprDXW5ZnLPyQ-1-2018',
'UU61qQBrHiNKprDXW5ZnLPyQ-0-2019',
'UU61qQBrHiNKprDXW5ZnLPyQ-2-2019',
'UU61qQBrHiNKprDXW5ZnLPyQ-0-2020',
'UU61qQBrHiNKprDXW5ZnLPyQ-1-2020',
'UU61qQBrHiNKprDXW5ZnLPyQ-2-2020',
'UU61qQBrHiNKprDXW5ZnLPyQ-3-2020',
'UU61qQBrHiNKprDXW5ZnLPyQ-0-2021',
'UU61qQBrHiNKprDXW5ZnLPyQ-3-2021',
'UU66IBH4uFkBTzEh2o1FFJSg-3-2016',
'UU66IBH4uFkBTzEh2o1FFJSg-0-2017',
'UU66IBH4uFkBTzEh2o1FFJSg-1-2018',
'UU66IBH4uFkBTzEh2o1FFJSg-1-2019',
'UU66_4Nrei9ZR4kk3uoWSSBw-0-2016',
'UU66_4Nrei9ZR4kk3uoWSSBw-1-2016',
'UU66_4Nrei9ZR4kk3uoWSSBw-2-2016',
'UU66_4Nrei9ZR4kk3uoWSSBw-3-2016',
'UU66_4Nrei9ZR4kk3uoWSSBw-0-2017',
'UU66_4Nrei9ZR4kk3uoWSSBw-1-2017',
'UU66_4Nrei9ZR4kk3uoWSSBw-2-2017',
'UU66_4Nrei9ZR4kk3uoWSSBw-3-2017',
'UU66_4Nrei9ZR4kk3uoWSSBw-0-2018',
'UU66_4Nrei9ZR4kk3uoWSSBw-1-2018',
'UU66_4Nrei9ZR4kk3uoWSSBw-2-2018',
'UU66_4Nrei9ZR4kk3uoWSSBw-3-2018',
'UU66_4Nrei9ZR4kk3uoWSSBw-0-2019',
'UU66_4Nrei9ZR4kk3uoWSSBw-1-2019',
'UU66_4Nrei9ZR4kk3uoWSSBw-2-2019',
'UU66_4Nrei9ZR4kk3uoWSSBw-3-2019',
'UU66_4Nrei9ZR4kk3uoWSSBw-0-2020',
'UU66_4Nrei9ZR4kk3uoWSSBw-1-2020',
'UU66_4Nrei9ZR4kk3uoWSSBw-2-2020',
'UU66_4Nrei9ZR4kk3uoWSSBw-3-2020',
'UU66_4Nrei9ZR4kk3uoWSSBw-0-2021',
'UU66_4Nrei9ZR4kk3uoWSSBw-1-2021',
'UU66_4Nrei9ZR4kk3uoWSSBw-2-2021',
'UU66_4Nrei9ZR4kk3uoWSSBw-3-2021',
'UU6cMYsKMx6XicFcFm7mTsmA-0-2016',
'UU6cMYsKMx6XicFcFm7mTsmA-1-2016',

'UU6cMYsKMx6XicFcFm7mTsmA-2-2016',
'UU6cMYsKMx6XicFcFm7mTsmA-3-2016',
'UU6cMYsKMx6XicFcFm7mTsmA-0-2017',
'UU6cMYsKMx6XicFcFm7mTsmA-1-2017',
'UU6cMYsKMx6XicFcFm7mTsmA-2-2017',
'UU6cMYsKMx6XicFcFm7mTsmA-3-2017',
'UU6cMYsKMx6XicFcFm7mTsmA-0-2018',
'UU6cMYsKMx6XicFcFm7mTsmA-1-2018',
'UU6cMYsKMx6XicFcFm7mTsmA-2-2018',
'UU6cMYsKMx6XicFcFm7mTsmA-3-2018',
'UU6cMYsKMx6XicFcFm7mTsmA-0-2019',
'UU6cMYsKMx6XicFcFm7mTsmA-1-2019',
'UU6cMYsKMx6XicFcFm7mTsmA-2-2019',
'UU6EyGxVgmTGt27_U9c_oz9g-3-2016',
'UU6EyGxVgmTGt27_U9c_oz9g-0-2017',
'UU6EyGxVgmTGt27_U9c_oz9g-2-2017',
'UU6EyGxVgmTGt27_U9c_oz9g-3-2017',
'UU6EyGxVgmTGt27_U9c_oz9g-0-2018',
'UU6EyGxVgmTGt27_U9c_oz9g-1-2018',
'UU6EyGxVgmTGt27_U9c_oz9g-2-2018',
'UU6EyGxVgmTGt27_U9c_oz9g-3-2018',
'UU6EyGxVgmTGt27_U9c_oz9g-0-2019',
'UU6Hwqnuqo1MMuJUslNlgqmw-0-2019',
'UU6Hwqnuqo1MMuJUslNlgqmw-1-2019',
'UU6Hwqnuqo1MMuJUslNlgqmw-2-2019',
'UU6Hwqnuqo1MMuJUslNlgqmw-3-2019',
'UU6Hwqnuqo1MMuJUslNlgqmw-0-2020',
'UU6Hwqnuqo1MMuJUslNlgqmw-1-2020',
'UU6Hwqnuqo1MMuJUslNlgqmw-2-2020',
'UU6Hwqnuqo1MMuJUslNlgqmw-3-2020',
'UU6Hwqnuqo1MMuJUslNlgqmw-3-2021',
'UU6P8LT59hNgA8bes61IA90w-0-2016',
'UU6P8LT59hNgA8bes61IA90w-1-2016',
'UU6P8LT59hNgA8bes61IA90w-2-2016',
'UU6P8LT59hNgA8bes61IA90w-3-2016',
'UU6P8LT59hNgA8bes61IA90w-0-2017',
'UU6P8LT59hNgA8bes61IA90w-1-2017',
'UU6P8LT59hNgA8bes61IA90w-2-2017',
'UU6P8LT59hNgA8bes61IA90w-3-2017',
'UU6P8LT59hNgA8bes61IA90w-0-2018',
'UU6P8LT59hNgA8bes61IA90w-1-2018',
'UU6P8LT59hNgA8bes61IA90w-2-2018',
'UU6P8LT59hNgA8bes61IA90w-3-2018',
'UU6P8LT59hNgA8bes61IA90w-0-2019',
'UU6P8LT59hNgA8bes61IA90w-1-2019',
'UU6P8LT59hNgA8bes61IA90w-2-2019',
'UU6P8LT59hNgA8bes61IA90w-3-2019',

'UU6P8LT59hNgA8bes61IA90w-0-2020',
'UU6P8LT59hNgA8bes61IA90w-1-2020',
'UU6P8LT59hNgA8bes61IA90w-2-2020',
'UU6P8LT59hNgA8bes61IA90w-3-2020',
'UU6P8LT59hNgA8bes61IA90w-0-2021',
'UU6P8LT59hNgA8bes61IA90w-1-2021',
'UU6P8LT59hNgA8bes61IA90w-2-2021',
'UU6P8LT59hNgA8bes61IA90w-3-2021',
'UU6x4zJADfr5Z4w-bf8lbrcg-2-2018',
'UU6x4zJADfr5Z4w-bf8lbrcg-0-2019',
'UU6x4zJADfr5Z4w-bf8lbrcg-1-2019',
'UU6x4zJADfr5Z4w-bf8lbrcg-1-2020',
'UU6x4zJADfr5Z4w-bf8lbrcg-0-2021',
'UU6x4zJADfr5Z4w-bf8lbrcg-1-2021',
'UU6x4zJADfr5Z4w-bf8lbrcg-2-2021',
'UU6x4zJADfr5Z4w-bf8lbrcg-3-2021',
'UU72vWiZdUN_Y20C2RgJ-dvA-0-2016',
'UU72vWiZdUN_Y20C2RgJ-dvA-1-2016',
'UU72vWiZdUN_Y20C2RgJ-dvA-2-2016',
'UU72vWiZdUN_Y20C2RgJ-dvA-3-2016',
'UU72vWiZdUN_Y20C2RgJ-dvA-0-2017',
'UU72vWiZdUN_Y20C2RgJ-dvA-1-2017',
'UU72vWiZdUN_Y20C2RgJ-dvA-2-2017',
'UU72vWiZdUN_Y20C2RgJ-dvA-3-2017',
'UU72vWiZdUN_Y20C2RgJ-dvA-0-2018',
'UU72vWiZdUN_Y20C2RgJ-dvA-1-2018',
'UU72vWiZdUN_Y20C2RgJ-dvA-2-2018',
'UU72vWiZdUN_Y20C2RgJ-dvA-3-2018',
'UU72vWiZdUN_Y20C2RgJ-dvA-0-2019',
'UU72vWiZdUN_Y20C2RgJ-dvA-1-2019',
'UU72vWiZdUN_Y20C2RgJ-dvA-2-2019',
'UU72vWiZdUN_Y20C2RgJ-dvA-3-2019',
'UU72vWiZdUN_Y20C2RgJ-dvA-0-2020',
'UU72vWiZdUN_Y20C2RgJ-dvA-1-2020',
'UU72vWiZdUN_Y20C2RgJ-dvA-2-2020',
'UU72vWiZdUN_Y20C2RgJ-dvA-3-2020',
'UU72vWiZdUN_Y20C2RgJ-dvA-0-2021',
'UU72vWiZdUN_Y20C2RgJ-dvA-1-2021',
'UU72vWiZdUN_Y20C2RgJ-dvA-2-2021',
'UU72vWiZdUN_Y20C2RgJ-dvA-3-2021',
'UU7e57qWIEAZTQAaFwHpKdlw-0-2016',
'UU7e57qWIEAZTQAaFwHpKdlw-1-2016',
'UU7e57qWIEAZTQAaFwHpKdlw-2-2016',
'UU7e57qWIEAZTQAaFwHpKdlw-3-2016',
'UU7e57qWIEAZTQAaFwHpKdlw-0-2017',
'UU7e57qWIEAZTQAaFwHpKdlw-1-2017',
'UU7e57qWIEAZTQAaFwHpKdlw-2-2017',

'UU7e57qWIEAZTQAaFwHpKdlw-3-2017',
'UU7e57qWIEAZTQAaFwHpKdlw-0-2018',
'UU7e57qWIEAZTQAaFwHpKdlw-1-2018',
'UU7e57qWIEAZTQAaFwHpKdlw-2-2018',
'UU7e57qWIEAZTQAaFwHpKdlw-3-2018',
'UU7e57qWIEAZTQAaFwHpKdlw-0-2019',
'UU7e57qWIEAZTQAaFwHpKdlw-1-2019',
'UU7e57qWIEAZTQAaFwHpKdlw-2-2019',
'UU7e57qWIEAZTQAaFwHpKdlw-3-2019',
'UU7e57qWIEAZTQAaFwHpKdlw-0-2020',
'UU7e57qWIEAZTQAaFwHpKdlw-1-2020',
'UU7e57qWIEAZTQAaFwHpKdlw-2-2020',
'UU7e57qWIEAZTQAaFwHpKdlw-3-2020',
'UU7e57qWIEAZTQAaFwHpKdlw-0-2021',
'UU7e57qWIEAZTQAaFwHpKdlw-1-2021',
'UU7e57qWIEAZTQAaFwHpKdlw-2-2021',
'UU7e57qWIEAZTQAaFwHpKdlw-3-2021',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-0-2016',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-1-2016',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-2-2016',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-3-2016',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-0-2017',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-1-2017',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-2-2017',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-3-2017',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-0-2018',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-1-2018',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-2-2018',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-3-2018',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-0-2019',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-1-2019',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-2-2019',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-3-2019',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-0-2020',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-1-2020',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-2-2020',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-3-2020',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-0-2021',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-1-2021',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-2-2021',
'UU7Edgk9RxP7Fm7vjQ1d-cDA-3-2021',
'UU7eWRnOTyqua812sDXGqiqw-3-2016',
'UU7eWRnOTyqua812sDXGqiqw-0-2017',
'UU7eWRnOTyqua812sDXGqiqw-3-2017',
'UU7eWRnOTyqua812sDXGqiqw-0-2018',
'UU7eWRnOTyqua812sDXGqiqw-1-2018',
'UU7eWRnOTyqua812sDXGqiqw-2-2018',

'UU7eWRnOTyqua812sDXGqiqw-3-2018',
'UU7eWRnOTyqua812sDXGqiqw-0-2019',
'UU7eWRnOTyqua812sDXGqiqw-1-2019',
'UU7eWRnOTyqua812sDXGqiqw-2-2019',
'UU7eWRnOTyqua812sDXGqiqw-3-2019',
'UU7eWRnOTyqua812sDXGqiqw-0-2020',
'UU7eWRnOTyqua812sDXGqiqw-1-2020',
'UU7eWRnOTyqua812sDXGqiqw-2-2020',
'UU7eWRnOTyqua812sDXGqiqw-3-2020',
'UU7kaw8NXcqb7jdRsRIHaswQ-0-2016',
'UU7kaw8NXcqb7jdRsRIHaswQ-1-2016',
'UU7kaw8NXcqb7jdRsRIHaswQ-2-2016',
'UU7kaw8NXcqb7jdRsRIHaswQ-3-2016',
'UU7kaw8NXcqb7jdRsRIHaswQ-0-2017',
'UU7kaw8NXcqb7jdRsRIHaswQ-1-2017',
'UU7kaw8NXcqb7jdRsRIHaswQ-2-2017',
'UU7kaw8NXcqb7jdRsRIHaswQ-3-2017',
'UU7kaw8NXcqb7jdRsRIHaswQ-0-2018',
'UU7kaw8NXcqb7jdRsRIHaswQ-1-2018',
'UU7kaw8NXcqb7jdRsRIHaswQ-2-2018',
'UU7kaw8NXcqb7jdRsRIHaswQ-1-2020',
'UU7kaw8NXcqb7jdRsRIHaswQ-3-2020',
'UU7kaw8NXcqb7jdRsRIHaswQ-0-2021',
'UU7kaw8NXcqb7jdRsRIHaswQ-1-2021',
'UU7kIy8fZavEni8Gzl8NLj0Q-1-2016',
'UU7kIy8fZavEni8Gzl8NLj0Q-2-2016',
'UU7kIy8fZavEni8Gzl8NLj0Q-3-2016',
'UU7kIy8fZavEni8Gzl8NLj0Q-0-2017',
'UU7kIy8fZavEni8Gzl8NLj0Q-1-2017',
'UU7kIy8fZavEni8Gzl8NLj0Q-2-2017',
'UU7kIy8fZavEni8Gzl8NLj0Q-3-2017',
'UU7kIy8fZavEni8Gzl8NLj0Q-0-2018',
'UU7kIy8fZavEni8Gzl8NLj0Q-1-2018',
'UU7kIy8fZavEni8Gzl8NLj0Q-2-2018',
'UU7kIy8fZavEni8Gzl8NLj0Q-3-2018',
'UU7kIy8fZavEni8Gzl8NLj0Q-0-2019',
'UU7kIy8fZavEni8Gzl8NLj0Q-1-2019',
'UU7kIy8fZavEni8Gzl8NLj0Q-2-2019',
'UU7kIy8fZavEni8Gzl8NLj0Q-3-2019',
'UU7kIy8fZavEni8Gzl8NLj0Q-0-2020',
'UU7kIy8fZavEni8Gzl8NLj0Q-1-2020',
'UU7kIy8fZavEni8Gzl8NLj0Q-2-2020',
'UU7kIy8fZavEni8Gzl8NLj0Q-3-2020',
'UU7kIy8fZavEni8Gzl8NLj0Q-0-2021',
'UU7kIy8fZavEni8Gzl8NLj0Q-1-2021',
'UU7kIy8fZavEni8Gzl8NLj0Q-2-2021',
'UU7kIy8fZavEni8Gzl8NLj0Q-3-2021',

'UU7KZFVVnKjiCzkxWD-Qrakg-0-2021',
'UU7KZFVVnKjiCzkxWD-Qrakg-1-2021',
'UU7KZFVVnKjiCzkxWD-Qrakg-2-2021',
'UU7KZFVVnKjiCzkxWD-Qrakg-3-2021',
'UU7Q4rvzJDbHeBHYk5rnvZeA-0-2016',
'UU7Q4rvzJDbHeBHYk5rnvZeA-1-2016',
'UU7Q4rvzJDbHeBHYk5rnvZeA-2-2016',
'UU7Q4rvzJDbHeBHYk5rnvZeA-3-2016',
'UU7Q4rvzJDbHeBHYk5rnvZeA-0-2017',
'UU7Q4rvzJDbHeBHYk5rnvZeA-1-2017',
'UU7Q4rvzJDbHeBHYk5rnvZeA-2-2017',
'UU7Q4rvzJDbHeBHYk5rnvZeA-3-2017',
'UU7Q4rvzJDbHeBHYk5rnvZeA-0-2018',
'UU7Q4rvzJDbHeBHYk5rnvZeA-1-2018',
'UU7Q4rvzJDbHeBHYk5rnvZeA-2-2018',
'UU7Q4rvzJDbHeBHYk5rnvZeA-3-2018',
'UU7Q4rvzJDbHeBHYk5rnvZeA-0-2019',
'UU7Q4rvzJDbHeBHYk5rnvZeA-1-2019',
'UU7Q4rvzJDbHeBHYk5rnvZeA-2-2019',
'UU7Q4rvzJDbHeBHYk5rnvZeA-3-2019',
'UU7Q4rvzJDbHeBHYk5rnvZeA-0-2020',
'UU7Q4rvzJDbHeBHYk5rnvZeA-1-2020',
'UU7Q4rvzJDbHeBHYk5rnvZeA-2-2020',
'UU7Q4rvzJDbHeBHYk5rnvZeA-3-2020',
'UU7Q4rvzJDbHeBHYk5rnvZeA-0-2021',
'UU7Q4rvzJDbHeBHYk5rnvZeA-1-2021',
'UU7Q4rvzJDbHeBHYk5rnvZeA-2-2021',
'UU7Q4rvzJDbHeBHYk5rnvZeA-3-2021',
'UU7qCAvNXXOPi05tufRDN0XQ-0-2016',
'UU7qCAvNXXOPi05tufRDN0XQ-1-2016',
'UU7qCAvNXXOPi05tufRDN0XQ-2-2016',
'UU7qCAvNXXOPi05tufRDN0XQ-3-2016',
'UU7qCAvNXXOPi05tufRDN0XQ-0-2017',
'UU7qCAvNXXOPi05tufRDN0XQ-1-2017',
'UU7qCAvNXXOPi05tufRDN0XQ-1-2018',
'UU7qCAvNXXOPi05tufRDN0XQ-2-2018',
'UU7qCAvNXXOPi05tufRDN0XQ-3-2018',
'UU7qCAvNXXOPi05tufRDN0XQ-0-2019',
'UU7qCAvNXXOPi05tufRDN0XQ-1-2019',
'UU7qCAvNXXOPi05tufRDN0XQ-2-2019',
'UU7qCAvNXXOPi05tufRDN0XQ-3-2019',
'UU7qCAvNXXOPi05tufRDN0XQ-0-2020',
'UU7qCAvNXXOPi05tufRDN0XQ-1-2020',
'UU7qCAvNXXOPi05tufRDN0XQ-2-2020',
'UU7qCAvNXXOPi05tufRDN0XQ-3-2020',
'UU7qCAvNXXOPi05tufRDN0XQ-0-2021',
'UU7qCAvNXXOPi05tufRDN0XQ-1-2021',

'UU7qCAvNXXOPi05tufRDN0XQ-2-2021',
'UU7qCAvNXXOPi05tufRDN0XQ-3-2021',
'UU7Q11bhBYk8LVMpZfkRZJGQ-2-2017',
'UU7Q11bhBYk8LVMpZfkRZJGQ-3-2017',
'UU7Q11bhBYk8LVMpZfkRZJGQ-0-2018',
'UU7Q11bhBYk8LVMpZfkRZJGQ-1-2018',
'UU7Q11bhBYk8LVMpZfkRZJGQ-2-2018',
'UU7Q11bhBYk8LVMpZfkRZJGQ-3-2018',
'UU7Q11bhBYk8LVMpZfkRZJGQ-0-2019',
'UU7Q11bhBYk8LVMpZfkRZJGQ-1-2019',
'UU7Q11bhBYk8LVMpZfkRZJGQ-2-2019',
'UU7Q11bhBYk8LVMpZfkRZJGQ-3-2019',
'UU7Q11bhBYk8LVMpZfkRZJGQ-0-2020',
'UU7Q11bhBYk8LVMpZfkRZJGQ-1-2020',
'UU7Q11bhBYk8LVMpZfkRZJGQ-2-2020',
'UU7Q11bhBYk8LVMpZfkRZJGQ-3-2020',
'UU7Q11bhBYk8LVMpZfkRZJGQ-0-2021',
'UU7Q11bhBYk8LVMpZfkRZJGQ-1-2021',
'UU7Q11bhBYk8LVMpZfkRZJGQ-2-2021',
'UU7Q11bhBYk8LVMpZfkRZJGQ-3-2021',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-2-2016',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-3-2016',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-0-2017',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-1-2017',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-2-2017',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-3-2017',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-0-2018',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-1-2018',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-2-2018',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-3-2018',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-0-2019',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-1-2019',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-2-2019',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-3-2019',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-0-2020',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-1-2020',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-2-2020',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-3-2020',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-0-2021',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-1-2021',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-2-2021',
'UU7SeFWZYFmsm1tqWxfuOTPPQ-3-2021',
'UU7UiChjgT_LDKcr_8NEEbMA-3-2019',
'UU7UiChjgT_LDKcr_8NEEbMA-0-2020',
'UU7UiChjgT_LDKcr_8NEEbMA-3-2020',
'UU7UiChjgT_LDKcr_8NEEbMA-2-2021',
'UU7UiChjgT_LDKcr_8NEEbMA-3-2021',

'UU87YBeLMwXhgaw5tcCxsXgQ-1-2016',
'UU87YBeLMwXhgaw5tcCxsXgQ-2-2016',
'UU87YBeLMwXhgaw5tcCxsXgQ-1-2017',
'UU87YBeLMwXhgaw5tcCxsXgQ-2-2017',
'UU87YBeLMwXhgaw5tcCxsXgQ-3-2017',
'UU87YBeLMwXhgaw5tcCxsXgQ-0-2018',
'UU87YBeLMwXhgaw5tcCxsXgQ-1-2018',
'UU87YBeLMwXhgaw5tcCxsXgQ-2-2018',
'UU87YBeLMwXhgaw5tcCxsXgQ-3-2018',
'UU87YBeLMwXhgaw5tcCxsXgQ-0-2019',
'UU87YBeLMwXhgaw5tcCxsXgQ-1-2019',
'UU87YBeLMwXhgaw5tcCxsXgQ-2-2019',
'UU87YBeLMwXhgaw5tcCxsXgQ-3-2019',
'UU87YBeLMwXhgaw5tcCxsXgQ-0-2020',
'UU87YBeLMwXhgaw5tcCxsXgQ-1-2020',
'UU87YBeLMwXhgaw5tcCxsXgQ-2-2020',
'UU87YBeLMwXhgaw5tcCxsXgQ-3-2020',
'UU87YBeLMwXhgaw5tcCxsXgQ-0-2021',
'UU87YBeLMwXhgaw5tcCxsXgQ-2-2021',
'UU87YBeLMwXhgaw5tcCxsXgQ-3-2021',
'UU8eE25pvmhz-nAMZATep-7g-1-2016',
'UU8eE25pvmhz-nAMZATep-7g-3-2016',
'UU8eE25pvmhz-nAMZATep-7g-2-2017',
'UU8eE25pvmhz-nAMZATep-7g-3-2017',
'UU8GA0CAJxBL4bExaUCvwL4Q-0-2016',
'UU8GA0CAJxBL4bExaUCvwL4Q-1-2016',
'UU8GA0CAJxBL4bExaUCvwL4Q-2-2016',
'UU8GA0CAJxBL4bExaUCvwL4Q-3-2016',
'UU8GA0CAJxBL4bExaUCvwL4Q-0-2017',
'UU8GA0CAJxBL4bExaUCvwL4Q-1-2017',
'UU8GA0CAJxBL4bExaUCvwL4Q-2-2017',
'UU8GA0CAJxBL4bExaUCvwL4Q-3-2017',
'UU8GA0CAJxBL4bExaUCvwL4Q-0-2018',
'UU8GA0CAJxBL4bExaUCvwL4Q-1-2018',
'UU8GA0CAJxBL4bExaUCvwL4Q-2-2018',
'UU8GA0CAJxBL4bExaUCvwL4Q-3-2018',
'UU8GA0CAJxBL4bExaUCvwL4Q-0-2019',
'UU8GA0CAJxBL4bExaUCvwL4Q-1-2019',
'UU8GA0CAJxBL4bExaUCvwL4Q-2-2019',
'UU8GA0CAJxBL4bExaUCvwL4Q-3-2019',
'UU8GA0CAJxBL4bExaUCvwL4Q-0-2020',
'UU8GA0CAJxBL4bExaUCvwL4Q-1-2020',
'UU8GA0CAJxBL4bExaUCvwL4Q-2-2020',
'UU8GA0CAJxBL4bExaUCvwL4Q-3-2020',
'UU8GA0CAJxBL4bExaUCvwL4Q-0-2021',
'UU8GA0CAJxBL4bExaUCvwL4Q-1-2021',
'UU8GA0CAJxBL4bExaUCvwL4Q-2-2021',

'UU8GAOCAJxBL4bExaUCvwL4Q-3-2021',
'UU8KbWhBEWVlEdkHmETx8MlA-2-2018',
'UU8KbWhBEWVlEdkHmETx8MlA-3-2018',
'UU8KbWhBEWVlEdkHmETx8MlA-0-2019',
'UU8KbWhBEWVlEdkHmETx8MlA-2-2019',
'UU8KbWhBEWVlEdkHmETx8MlA-0-2020',
'UU8RJExf7iNpV6tp6hvNtwww-0-2016',
'UU8RJExf7iNpV6tp6hvNtwww-2-2017',
'UU8RJExf7iNpV6tp6hvNtwww-3-2017',
'UU8RJExf7iNpV6tp6hvNtwww-0-2018',
'UU8RJExf7iNpV6tp6hvNtwww-1-2018',
'UU8RJExf7iNpV6tp6hvNtwww-2-2018',
'UU8RJExf7iNpV6tp6hvNtwww-3-2018',
'UU8RJExf7iNpV6tp6hvNtwww-0-2019',
'UU8RJExf7iNpV6tp6hvNtwww-1-2019',
'UU8RJExf7iNpV6tp6hvNtwww-2-2019',
'UU8RJExf7iNpV6tp6hvNtwww-3-2019',
'UU8RJExf7iNpV6tp6hvNtwww-0-2020',
'UU8RJExf7iNpV6tp6hvNtwww-1-2020',
'UU8RJExf7iNpV6tp6hvNtwww-2-2020',
'UU8RJExf7iNpV6tp6hvNtwww-3-2020',
'UU8RJExf7iNpV6tp6hvNtwww-0-2021',
'UU8RJExf7iNpV6tp6hvNtwww-1-2021',
'UU8RJExf7iNpV6tp6hvNtwww-2-2021',
'UU8RJExf7iNpV6tp6hvNtwww-3-2021',
'UU92qENDv1CKpNCkUwJnFdkA-2-2016',
'UU92qENDv1CKpNCkUwJnFdkA-3-2016',
'UU92qENDv1CKpNCkUwJnFdkA-0-2017',
'UU9f78Z5hgtDt0n8JWyfBk8Q-0-2016',
'UU9f78Z5hgtDt0n8JWyfBk8Q-1-2016',
'UU9f78Z5hgtDt0n8JWyfBk8Q-2-2016',
'UU9f78Z5hgtDt0n8JWyfBk8Q-3-2016',
'UU9f78Z5hgtDt0n8JWyfBk8Q-0-2017',
'UU9f78Z5hgtDt0n8JWyfBk8Q-1-2017',
'UU9f78Z5hgtDt0n8JWyfBk8Q-2-2017',
'UU9f78Z5hgtDt0n8JWyfBk8Q-3-2017',
'UU9f78Z5hgtDt0n8JWyfBk8Q-0-2018',
'UU9f78Z5hgtDt0n8JWyfBk8Q-1-2018',
'UU9f78Z5hgtDt0n8JWyfBk8Q-2-2018',
'UU9f78Z5hgtDt0n8JWyfBk8Q-3-2018',
'UU9f78Z5hgtDt0n8JWyfBk8Q-0-2019',
'UU9f78Z5hgtDt0n8JWyfBk8Q-1-2019',
'UU9f78Z5hgtDt0n8JWyfBk8Q-2-2019',
'UU9f78Z5hgtDt0n8JWyfBk8Q-3-2019',
'UU9f78Z5hgtDt0n8JWyfBk8Q-0-2020',
'UU9f78Z5hgtDt0n8JWyfBk8Q-1-2020',
'UU9f78Z5hgtDt0n8JWyfBk8Q-0-2021',

'UU9f78Z5hgtDt0n8JWyfBk8Q-1-2021',
'UU9f78Z5hgtDt0n8JWyfBk8Q-2-2021',
'UU9f78Z5hgtDt0n8JWyfBk8Q-3-2021',
'UU9infsKo33_2LUoiqXGgQWg-0-2016',
'UU9infsKo33_2LUoiqXGgQWg-1-2016',
'UU9infsKo33_2LUoiqXGgQWg-2-2016',
'UU9infsKo33_2LUoiqXGgQWg-3-2016',
'UU9infsKo33_2LUoiqXGgQWg-0-2017',
'UU9infsKo33_2LUoiqXGgQWg-1-2017',
'UU9infsKo33_2LUoiqXGgQWg-2-2017',
'UU9infsKo33_2LUoiqXGgQWg-3-2017',
'UU9infsKo33_2LUoiqXGgQWg-0-2018',
'UU9infsKo33_2LUoiqXGgQWg-1-2018',
'UU9infsKo33_2LUoiqXGgQWg-2-2018',
'UU9infsKo33_2LUoiqXGgQWg-3-2018',
'UU9infsKo33_2LUoiqXGgQWg-0-2019',
'UU9infsKo33_2LUoiqXGgQWg-1-2019',
'UU9infsKo33_2LUoiqXGgQWg-2-2019',
'UU9infsKo33_2LUoiqXGgQWg-3-2019',
'UU9infsKo33_2LUoiqXGgQWg-0-2020',
'UU9infsKo33_2LUoiqXGgQWg-1-2020',
'UU9infsKo33_2LUoiqXGgQWg-2-2020',
'UU9infsKo33_2LUoiqXGgQWg-3-2020',
'UU9infsKo33_2LUoiqXGgQWg-0-2021',
'UU9infsKo33_2LUoiqXGgQWg-1-2021',
'UU9infsKo33_2LUoiqXGgQWg-2-2021',
'UU9infsKo33_2LUoiqXGgQWg-3-2021',
'UU9ISPZsMaBi5mutsgX6LC1g-0-2016',
'UU9ISPZsMaBi5mutsgX6LC1g-1-2016',
'UU9ISPZsMaBi5mutsgX6LC1g-2-2016',
'UU9ISPZsMaBi5mutsgX6LC1g-3-2016',
'UU9ISPZsMaBi5mutsgX6LC1g-0-2017',
'UU9ISPZsMaBi5mutsgX6LC1g-1-2017',
'UU9ISPZsMaBi5mutsgX6LC1g-2-2017',
'UU9ISPZsMaBi5mutsgX6LC1g-3-2017',
'UU9ISPZsMaBi5mutsgX6LC1g-0-2018',
'UU9ISPZsMaBi5mutsgX6LC1g-1-2018',
'UU9ISPZsMaBi5mutsgX6LC1g-2-2018',
'UU9ISPZsMaBi5mutsgX6LC1g-3-2018',
'UU9ISPZsMaBi5mutsgX6LC1g-0-2019',
'UU9ISPZsMaBi5mutsgX6LC1g-1-2019',
'UU9ISPZsMaBi5mutsgX6LC1g-2-2019',
'UU9ISPZsMaBi5mutsgX6LC1g-3-2019',
'UU9ISPZsMaBi5mutsgX6LC1g-0-2020',
'UU9ISPZsMaBi5mutsgX6LC1g-1-2020',
'UU9ISPZsMaBi5mutsgX6LC1g-2-2020',
'UU9ISPZsMaBi5mutsgX6LC1g-3-2020',

```
'UU9ISPZsMaBi5mutsgX6LC1g-0-2021',
'UU9ISPZsMaBi5mutsgX6LC1g-1-2021',
'UU9ISPZsMaBi5mutsgX6LC1g-2-2021',
'UU9ISPZsMaBi5mutsgX6LC1g-3-2021',
'UUA071P11f2wk-B8Rkwt47bQ-3-2020',
'UUA071P11f2wk-B8Rkwt47bQ-0-2021',
'UUA071P11f2wk-B8Rkwt47bQ-1-2021',
'UUA071P11f2wk-B8Rkwt47bQ-2-2021']
```

```
[23]: di=[k.split('-') for k in li]
      for k in range(0,len(di)):
          if len(di[k])==4:
              di[k][0 : 2]='-'.join(di[k][0 : 2])
          elif len(di[k])==5:
              di[k][0 : 3]='-'.join(di[k][0 : 3])
          elif len(di[k])==6:
              di[k][0 : 4]='-'.join(di[k][0 : 4])
```

```
[25]: di = pd.DataFrame(di, columns =['id', 'm', 'y'])
      ## merge both dfs, merge with di
      df['id']=df['id'].str.strip()
      gi = pd.merge(di, df, on='id')
```

```
[28]: ei = [0.0] *len(gi)

      t=0
      for j in range(2016,2022):
          for i in range(0,4):
              for k in range(0,len(gi)):
                  if gi.loc[k,'m']==str(i) and gi.loc[k,'y']==str(j):
                      f='Q'+str(t)
                      ei[k]=f
                  t+=1

      gi['Quarter']=ei
```

```
[29]: dictpur={}
      for k in range(0,24):
          t=str(k)
          f='Q'+t
          lab_NMF_phi=[item for sublist in cut_tree(DictNMF[f][4], n_clusters=[2])
          ↪for item in sublist]
          lab_NMF_pk= DictNMF[f][5].labels_
          lab_wf_h= [item for sublist in cut_tree(w[f][1], n_clusters=[2]) for item
          ↪in sublist]
          lab_wf_k=w[f][2].labels_
```

```

lab_hi= [item for sublist in cut_tree(Dict_hi[f][1], n_clusters=[2]) for
↪item in sublist]
dictpur[f]= [gi[gi['Quarter']==f] ,lab_NMF_phi, lab_NMF_pk, lab_wf_h,
↪lab_wf_k,lab_hi]

```

```

[30]: dictpur2={}
for k in range(0,24):
    t=str(k)
    f='Q'+t
    lr = gi[gi['Quarter']==f]['classif2'].tolist()
    d = {"labels": lr, "lab_NMF_phi": dictpur[f][1], "lab_NMF_pk":
↪dictpur[f][2], "lab_wf_h": dictpur[f][3], "lab_wf_k": dictpur[f][4] ,
↪"lab_hi": dictpur[f][5]}
    df2 = pd.DataFrame(d)
    dictpur2[f] = df2

```

```

[31]: names=["lab_NMF_phi", "lab_NMF_pk", "lab_wf_h", "lab_wf_k", "lab_hi"]
pur3={}
for k in names:
    purli=[None] * len(range(0,24))
    purli2=[None] * len(range(0,24))
    for i in range(0,24):
        t=str(i)
        f='Q'+t
        purli[i] = (max(pd.crosstab(dictpur2[f][k],
↪dictpur2[f]['labels'])[1][0],pd.crosstab(dictpur2[f][k],
↪dictpur2[f]['labels'])[0][0])+max(pd.crosstab(dictpur2[f][k],
↪dictpur2[f]['labels'])[0][1],pd.crosstab(dictpur2[f][k],
↪dictpur2[f]['labels'])[1][1]))/len(dictpur2[f])
        purli2[i] = 1- dictpur2[f]['labels'].sum()/len(dictpur2[f])
    pur3[k]=[purli,purli2]

```

Validation Results I compute the cluster purity measure for each method following (KJDBKJB) (in Blue) in the following graphs. I also computed the share of right-wing political commentators since this would be the measure's outcome if clusters were random (in Orange).

The results from the cluster purity measure show that the clusterization presented here was not effective on matching clusterization made by transparency.tube; the measure is different from that that mimics a random assignment.

Does this mean the polarization measure is wrong? That is a possible answer. However, the lack of a trend's robustness and misclassification means that maybe conflict in YouTube political commentators is not about their ideologies. At least in terms of their vocabulary, we cannot see a split between Left-wing and Right-wing political commentators.

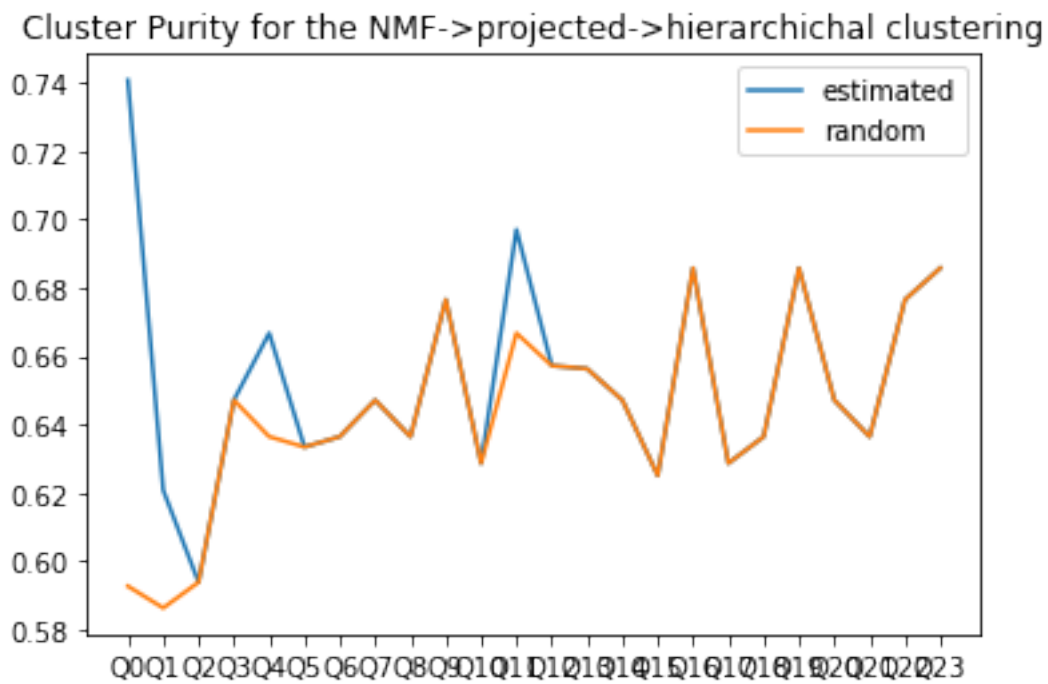
```

[41]: plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_NMF_phi'][0], label='estimated')
plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_NMF_phi'][1], label='random')

```

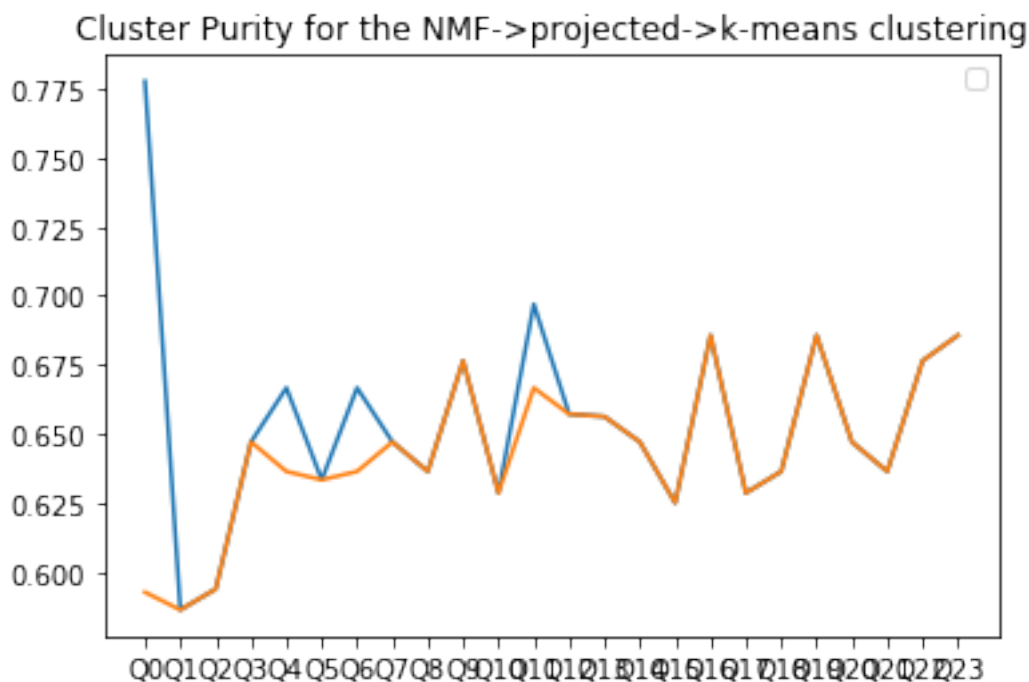


```
plt.title("Cluster Purity for the NMF->projected->hierarchichal clustering")
plt.legend();
```

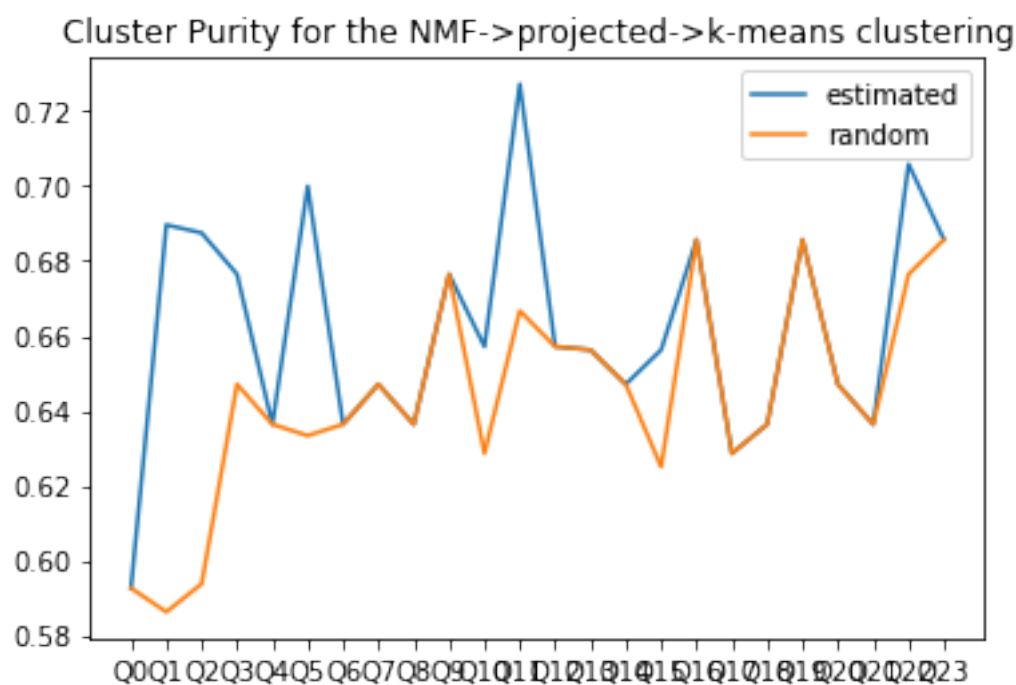


```
[42]: plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_NMF_pk'][0], label='estimated')
plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_NMF_pk'][1], label='random')
plt.title("Cluster Purity for the NMF->projected->k-means clustering")
plt.legend();
```

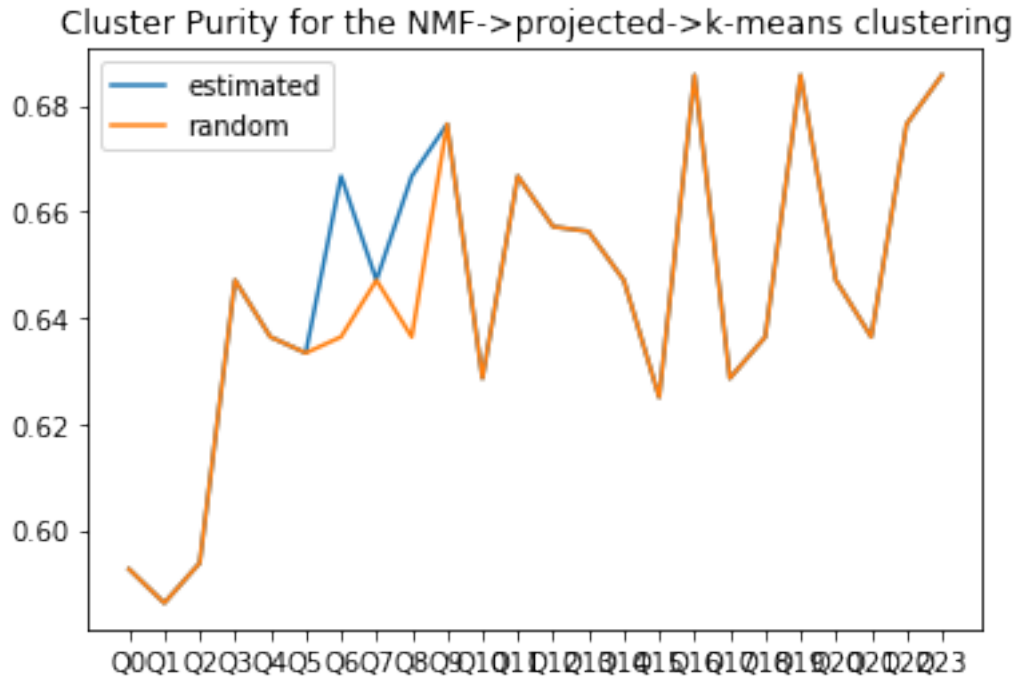
No handles with labels found to put in legend.



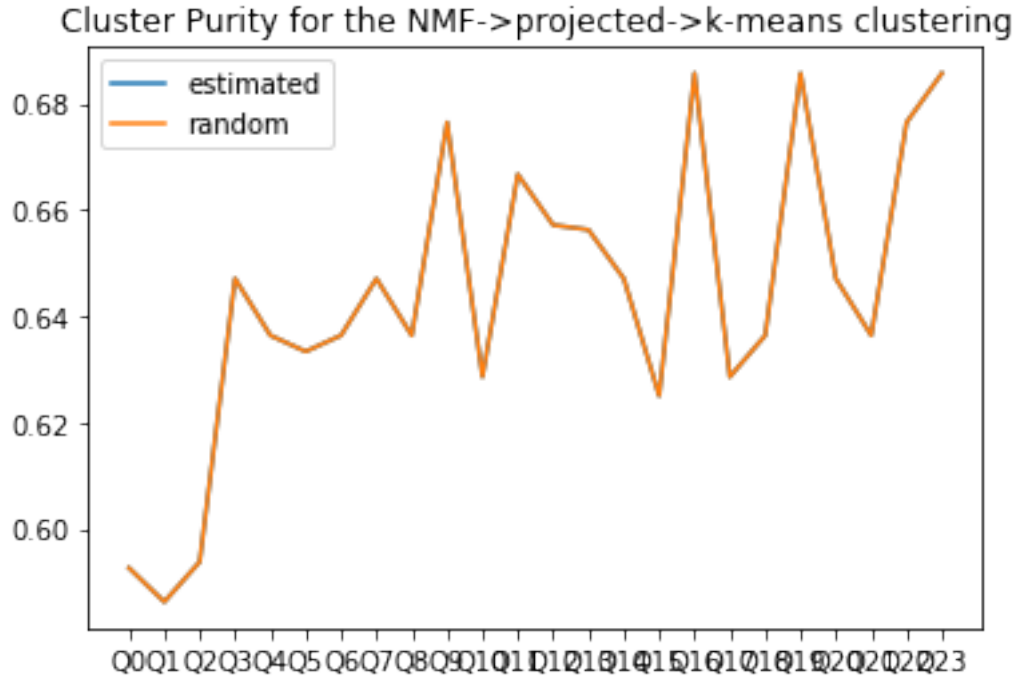
```
[43]: plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_hi'][0], label='estimated')
plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_hi'][1], label='random')
plt.title("Cluster Purity for the NMF->projected->k-means clustering")
plt.legend();
```



```
[44]: plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_wf_h'][0], label='estimated')
plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_wf_h'][1], label='random')
plt.title("Cluster Purity for the NMF->projected->k-means clustering")
plt.legend();
```



```
[45]: plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_wf_k'][0], label='estimated')
plt.plot(list(CPC_NMF_ph.keys()),pur3['lab_wf_k'][1], label='random')
plt.title("Cluster Purity for the NMF->projected->k-means clustering")
plt.legend();
```



1.6 Future Steps and Discussion

This project is the first step towards a project where I pretend to learn how each political commentator chooses to position given their past positions, reactions to it (views, subscribers, likes, and comments), and other commentators' positions. However, the lack of conclusive results means that improvements (or more validations) regarding the estimation of political choices using YouTube transcripts should be made.

In any case, I plan to improve the measure of polarization using the same dataset.

In terms of future directions, first, given the existing database of labels for political commentators, it could be good to measure polarization using supervised methods to check if results differ enough. To do this, I could use the labels made manually by other people or use speeches from people we know of their partisanship.

Second, a possible direction could be to focus on a narrower topic. By focusing on variation across quarters, I try to use variations about similar topics (everybody spoke about the elections in the last quarter of 2020). Goet (2019), for example, restricts to speeches related to economic issues. It is more challenging to do that on videos that could be about anything and are not labeled. However, tags exist, and titles and comments could also be analyzed.

Finally, I aim to incorporate video metadata into the analysis. YouTube videos contain information on how viewers react, which could also help understand the dynamics of polarization and political ideology choice. I plan to create a novel dataset with estimated political positions from texts and daily reactions to the most prominent YouTube political commentators' videos. To do that, I need to download data from YouTube every day, and I plan to start doing it soon.

1.6.1 References

- Athey, S., Mobius, M., & Pal, J. (2021). The impact of aggregators on internet news consumption (Tech. Rep.). National Bureau of Economic Research.
- Auxier, B., & Anderson, M. (2021). Social media use in 2021. Pew Research Center.
- Goet, N. D. (2019). Measuring polarization with text analysis: evidence from the UK House of Commons, 1811–2015. *Political Analysis*, 27(4), 518–539.
- Lahoti, P., Garimella, K., & Gionis, A. (2018, February). Joint non-negative matrix factorization for learning ideological leaning on twitter. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (pp. 351–359).
- Mehlhaff, I. D. (2021). A Group-Based Approach to Measuring Polarization. The University of North Carolina at Chapel Hill.
- Mohan, N. (2021). Responsibility is good for business and for the creator economy. <https://blog.youtube/inside-youtube/responsibility-good-business-and-creator-economy/>. (Accessed: 2021-12-03)
- Pariser, E. (2011). The filter bubble: What the internet is hiding from you. Penguin UK.
- Roose, K. (2019). The making of a YouTube radical. *The New York Times*, 8. (Accessed: 2021-12-03)
- Slapin, J. B., & Proksch, S.-O. (2006). A poisson scaling model for estimating time-series party positions from texts. *American Journal of Political Science*.
- Zhuravskaya, E., Petrova, M., & Enikolopov, R. (2020). Political effects of the internet and social media. *Annual Review of Economics*, 12, 415–438

[]: