Question 1

In [1]:
```
pip install tweepy
```

```
Requirement already satisfied: tweepy in /opt/anaconda3/lib/python3.7
/site-packages (3.8.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/anaco
nda3/lib/python3.7/site-packages (from tweepy) (1.3.0)
Requirement already satisfied: PySocks>=1.5.7 in /opt/anaconda3/lib/p
ython3.7/site-packages (from tweepy) (1.7.1)
Requirement already satisfied: six>=1.10.0 in /opt/anaconda3/lib/pyth
on3.7/site-packages (from tweepy) (1.12.0)
Requirement already satisfied: requests>=2.11.1 in /opt/anaconda3/lib
/python3.7/site-packages (from tweepy) (2.22.0)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/anaconda3/lib/
python3.7/site-packages (from requests-oauthlib>=0.7.0->tweepy) (3.1.
0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /opt/anaconda
3/lib/python3.7/site-packages (from requests>=2.11.1->tweepy) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /opt/anaconda3/l
ib/python3.7/site-packages (from requests>=2.11.1->tweepy) (2019.9.11
)
Requirement already satisfied: idna<2.9,>=2.5 in /opt/anaconda3/lib/p
ython3.7/site-packages (from requests>=2.11.1->tweepy) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.
1 in /opt/anaconda3/lib/python3.7/site-packages (from requests>=2.11.
1->tweepy) (1.24.2)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
pip install pytube3
```

```
Requirement already satisfied: pytube3 in /opt/anaconda3/lib/python3.
7/site-packages (9.6.4)
Requirement already satisfied: typing-extensions in /opt/anaconda3/li
b/python3.7/site-packages (from pytube3) (3.7.4.2)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install textblob
```

```
Requirement already satisfied: textblob in /opt/anaconda3/lib/python3
.7/site-packages (0.15.3)
Requirement already satisfied: nltk>=3.1 in /opt/anaconda3/lib/python
3.7/site-packages (from textblob) (3.4.5)
Requirement already satisfied: six in /opt/anaconda3/lib/python3.7/si
te-packages (from nltk>=3.1->textblob) (1.12.0)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
pip install 'clean-text[gpl]'
```

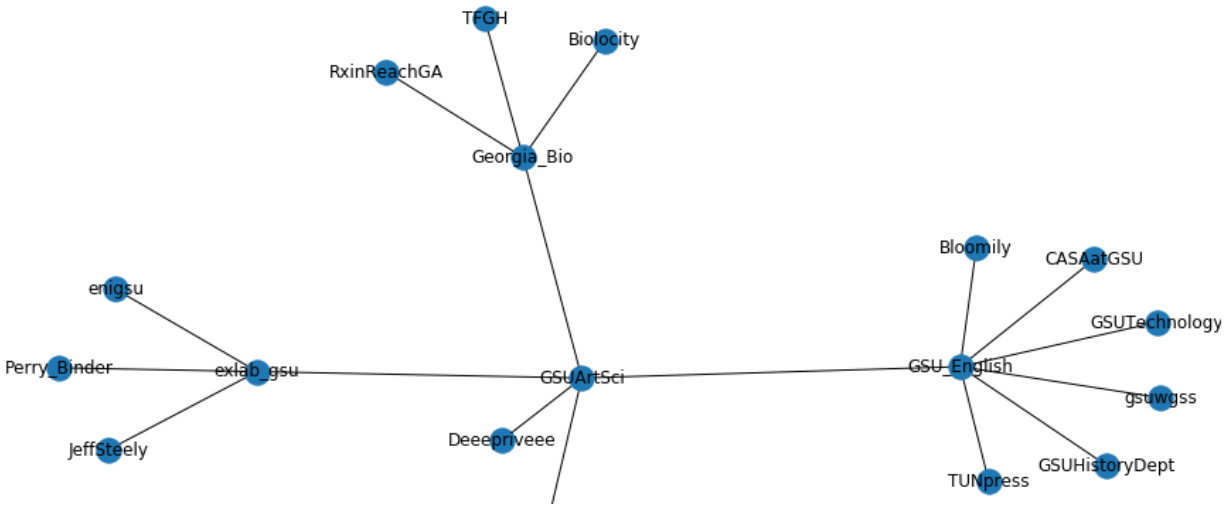
```
Requirement already satisfied: clean-text[gpl] in /opt/anaconda3/lib/
python3.7/site-packages (0.1.1)
Requirement already satisfied: ftfy in /opt/anaconda3/lib/python3.7/s
ite-packages (from clean-text[gpl]) (5.7)
Requirement already satisfied: unidecode; extra == "gpl" in /opt/anac
onda3/lib/python3.7/site-packages (from clean-text[gpl]) (1.1.1)
Requirement already satisfied: wcwidth in /opt/anaconda3/lib/python3.
7/site-packages (from ftfy->clean-text[gpl]) (0.1.7)
Note: you may need to restart the kernel to use updated packages.
```

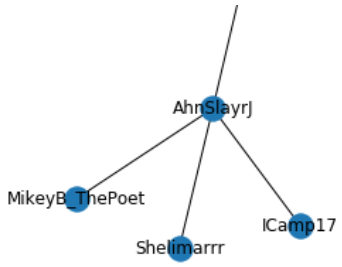
In [ ]:
```
```

In [10]:
```
import tweepy

auth = tweepy.OAuthHandler("WeaDO9F7S2RcngTSgfP6Srb2x",
                           "wIDHusSJ7thLhw7w9b1xJdSRxK6BPXkkkMtLkb5XhC

auth.set_access_token("2242904858-Bu49q8K0V1MlIGXnvA804cdyw49zfcVUQSSz
                      "T8ecdbA72rp0nPd47Y0wZPCyzl2JuLYcL7ORBV9WN91cr")

api = tweepy.API(auth, wait_on_rate_limit=True)

import pandas as pd

edge_list = pd.DataFrame(columns = ["source", "target"])
max_num_friends = 3
max_num_followers = 3

handle = "GSUArtSci"

user = api.get_user(handle)
friends = user.friends()

handle = "GSUArtSci"

for friend in friends[0:min(len(friends), max_num_friends)]:
    edge_list = edge_list.append({'source' : user.screen_name,
                                  'target' : friend.screen_name} ,
                                 ignore_index=True)

    friends_of_friends = friend.friends()

    for friend_of_friend in friends_of_friends[0:min(len(friends_of_fr
        edge_list = edge_list.append({'source' : friend.screen_name,
                                      'target' : friend_of_friend.scre
                                     ignore_index=True)
```

```
if user.protected == False:
    followers = user.followers()

    for follower in followers[0:min(len(followers), max_num_followers)
        edge_list = edge_list.append({'source' : follower.screen_name,
                                      'target' : user.screen_name} ,
                                     ignore_index=True)

        if(follower.protected == False):
            followers_of_followers = follower.followers()

            for follower_of_follower in followers_of_followers[0:min(1
                edge_list = edge_list.append({'source' : follower_of_f
                                              'target' : follower.scre
                                             ignore_index=True)

import networkx as nx
import matplotlib.pyplot as plt

G = nx.from_pandas_edgelist(df = edge_list,
                            source = "source",
                            target = "target",
                            create_using = nx.Graph)

fig_scale = 2
size = plt.gcf().get_size_inches()
plt.gcf().set_size_inches(size[0]*fig_scale, size[1]*fig_scale)

nx.draw(G, with_labels = True)
```

```
/opt/anaconda3/lib/python3.7/site-packages/networkx/drawing/nx_pylab.
py:579: MatplotlibDeprecationWarning:
The iterable function was deprecated in Matplotlib 3.1 and will be re
moved in 3.3. Use np.iterable instead.
  if not cb.iterable(width):
```





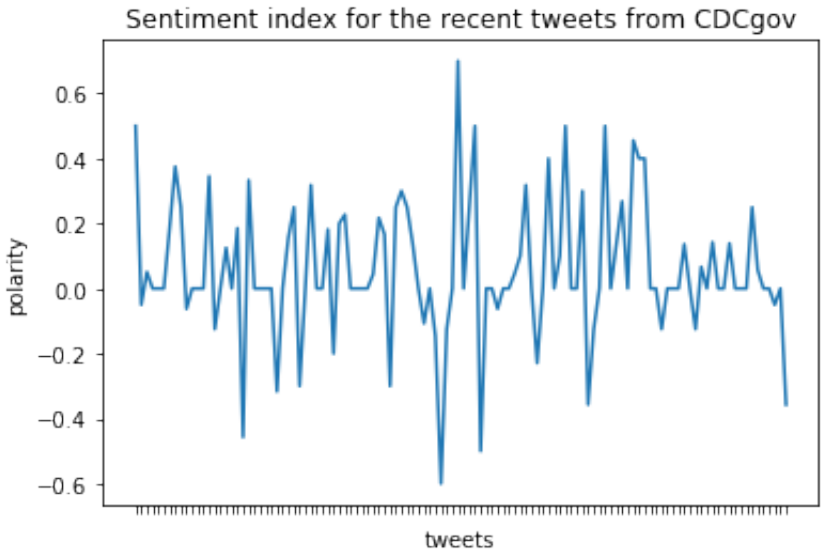
Question 2

In [6]:
```
import pandas as pd
from cleantext import clean
from textblob import TextBlob
import seaborn as sns

handle = "CDCgov"
user = api.get_user(handle)

list_tweets = []
max_num_pages = 6
for i in range(1, max_num_pages+1):
    tweets = api.user_timeline(handle, page = i)
    for tweet in tweets:
        list_tweets.append(tweet._json)

tweet_text = [tweet["text"] for tweet in list_tweets]

for i in range(len(tweet_text)):
    # Clean text with "cleantext"
    tweet_text[i] = clean(tweet_text[i],
                          fix_unicode = True,
                          to_ascii = True,
                          lower = True,
                          no_line_breaks = True,
                          no_urls=True,
                          no_emails=True,
                          no_numbers=True,
                          no_digits = True,
                          no_phone_numbers=True,
                          no_currency_symbols=True,
                          no_punct=True,
                          replace_with_url="",
                          replace_with_number="",
                          lang="en")

print("We get " + str(len(tweet_text)) + " tweets")

sentiment_objects = [TextBlob(tweet) for tweet in tweet_text]
```

```
sentiment_values = [[tweet.sentiment.polarity,
                     str(tweet)] for tweet in sentiment_objects]

sentiment_df = pd.DataFrame(sentiment_values, columns=["polarity", "tw

fig = sns.lineplot(x = range(len(sentiment_df)), y="polarity", data =
fig.set_xticks(range(len(sentiment_df)))
fig.set_xlabel("tweets")
fig.set_xticklabels(labels = "")
fig.set_title("Sentiment index for the recent tweets from " + handle)
```

We get 116 tweets

Out[6]: Text(0.5, 1.0, 'Sentiment index for the recent tweets from CDCgov')



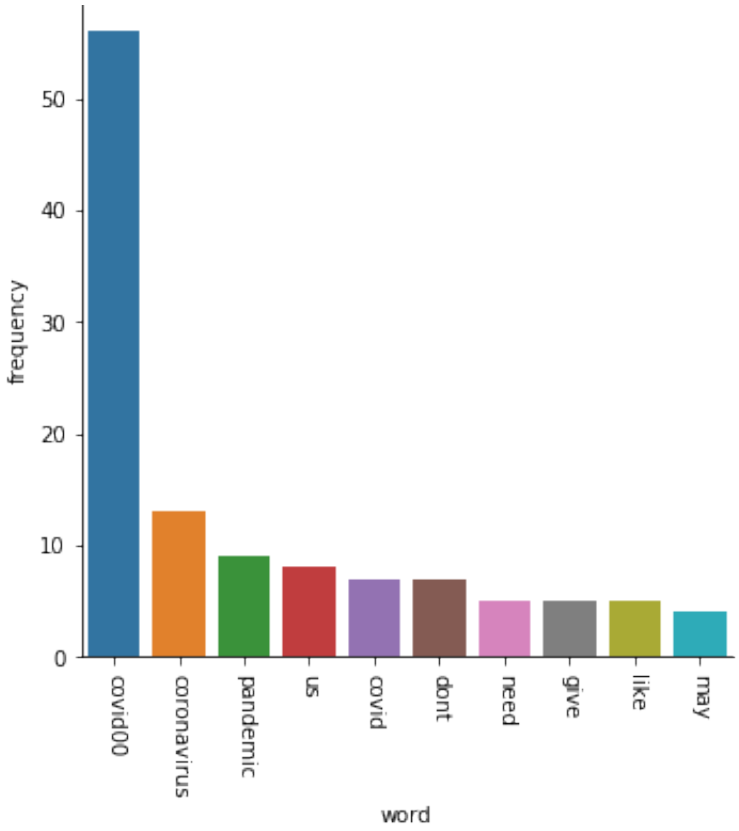
Question 3

In [7]:
```
from cleantext import clean

import nltk
from nltk.corpus import stopwords

keyword = "#COVID19" + " -filter:retweets"
since_when = "2020-04-23"

tweets = tweepy.Cursor(api.search, q = keyword,
                       lang="en", since = since_when).items(100)

tweet_text = [tweet.text for tweet in tweets]

words = []

for i in range(len(tweet_text)):
```

```
for i in range(len(tweet_text)):
    tweet_text[i] = clean(tweet_text[i],
                          fix_unicode = True,
                          to_ascii = True,
                          lower = True,
                          no_line_breaks = True,
                          no_urls=True,
                          no_emails=True,
                          no_numbers=True,
                          no_digits = True,
                          no_phone_numbers=True,
                          no_currency_symbols=True,
                          no_punct=True,
                          replace_with_url="",
                          replace_with_number="",
                          lang="en")

    words.append(tweet_text[i].split())

words = [y for x in words for y in x]
nltk.download("stopwords")
stop_words = set(stopwords.words('english'))
words = [w for w in words if not w in stop_words]

import pandas as pd

df = pd.DataFrame(words, columns=["word"])

frequency = df["word"].value_counts()

word_frequency = pd.DataFrame({"word": frequency.index.tolist(),
                               "frequency": frequency.tolist()})


import seaborn as sns

num_words = 10

most_frequent_words = word_frequency.iloc[:num_words]

g4 = sns.catplot(x = "word", y = "frequency", kind = "bar",
                 data = most_frequent_words)

g4.set_xticklabels(rotation=-90)
```

```
[nltk_data] Downloading package stopwords to /Users/shilp/nltk_data..
.
[nltk_data]   Package stopwords is already up-to-date!
```

Out[7]: <seaborn.axisgrid.FacetGrid at 0x1a205f6210>



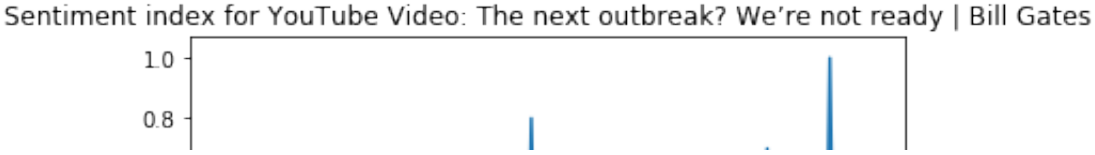
Question 4

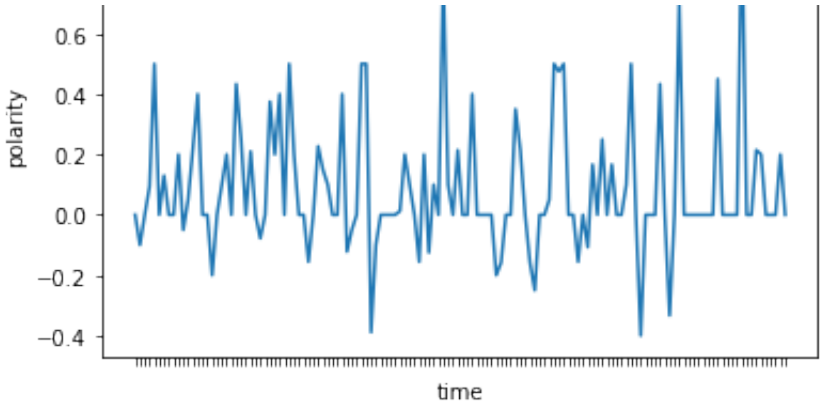
Given Link

In [8]:
```
from pytube import YouTube

youTubeURL = "https://www.youtube.com/watch?v=6Af6b_wyiwI"

yt = YouTube(youTubeURL)

def get_youtube_info(yt, num_chars = 300):
    mime_type = []
    stream_type = []
    fps = []
    resolution = []
    is_live = []
    is_3d = []

    for stream in yt.streams.all():
        stream_info = stream.__dict__
        mime_type.append(stream_info["mime_type"])
        stream_type.append(stream_info["type"])
        fps.append(stream_info["fps"])
        resolution.append(stream_info["resolution"])
        is_live.append(stream_info["is_live"])
```

```
        is_live.append(stream_info["is_live"])
        is_3d.append(stream_info["is_3d"])

    caption_lang = []

import pandas as pd

caption = yt.captions.get_by_language_code("en")


if(caption != None):

    caption_srt = caption.generate_srt_captions()

    caption_lines = caption_srt.splitlines()

    nested = []
    num_lines_per_item = 4
    for ix in range(0, len(caption_lines) - num_lines_per_item, num_li
        nested.append(caption_lines[ix:ix + num_lines_per_item])

    caption_df = pd.DataFrame(nested, columns = ["index", "time", "tex

    caption_df = caption_df.drop(columns = ["line_break"])

from textblob import TextBlob

if caption_df.empty != True:
    sentiment_objects = [TextBlob(caption) for caption in caption_df['

    sentiment_values = [[sentiment_obj.sentiment.polarity,
                         str(sentiment_obj)] for sentiment_obj in sent

    caption_df["polarity"] = [sentiment_obj.sentiment.polarity for ser

import seaborn as sns

if caption_df.empty != True:
    fig = sns.lineplot(x = "index", y="polarity", data = caption_df)
    fig.set_xticklabels(labels = "")
    fig.set_xlabel("time")
    fig.set_title("Sentiment index for YouTube Video: " + yt.title)
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:28:
DeprecationWarning: Call to deprecated function get_by_language_code
(This object can be treated as a dictionary, i.e. captions['en']).
```



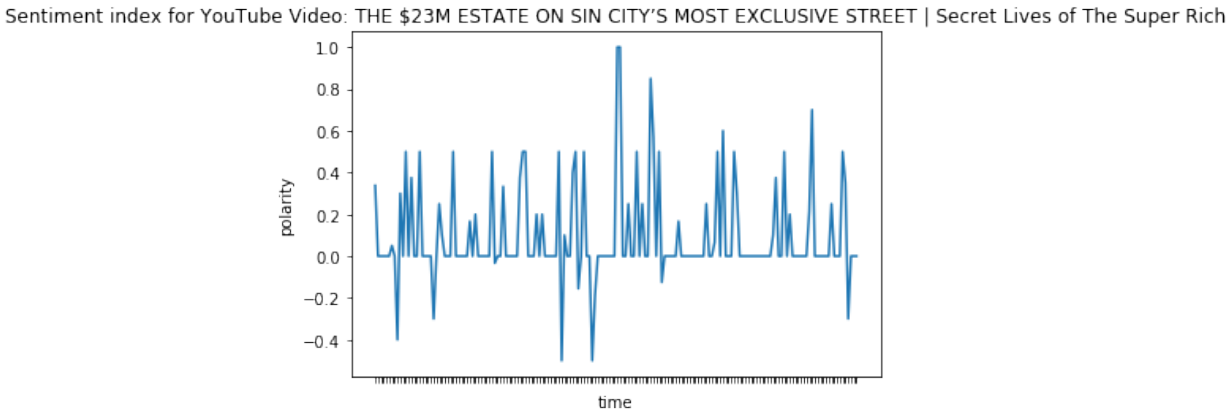


Own Link

In [9]:
```
from pytube import YouTube

youTubeURL = "https://www.youtube.com/watch?v=k3jzePlIowk"

yt = YouTube(youTubeURL)

def get_youtube_info(yt, num_chars = 300):
    mime_type = []
    stream_type = []
    fps = []
    resolution = []
    is_live = []
    is_3d = []

    for stream in yt.streams.all():
        stream_info = stream.__dict__
        mime_type.append(stream_info["mime_type"])
        stream_type.append(stream_info["type"])
        fps.append(stream_info["fps"])
        resolution.append(stream_info["resolution"])
        is_live.append(stream_info["is_live"])
        is_3d.append(stream_info["is_3d"])

    caption_lang = []

import pandas as pd

caption = yt.captions.get_by_language_code("en")


if(caption != None):

    caption_srt = caption.generate_srt_captions()

    caption_lines = caption_srt.splitlines()
```

```
    nested = []
    num_lines_per_item = 4
    for ix in range(0, len(caption_lines) - num_lines_per_item, num_li
        nested.append(caption_lines[ix:ix + num_lines_per_item])

    caption_df = pd.DataFrame(nested, columns = ["index", "time", "tex

    caption_df = caption_df.drop(columns = ["line_break"])

    caption_df

from textblob import TextBlob

if caption_df.empty != True:
    sentiment_objects = [TextBlob(caption) for caption in caption_df['

    sentiment_values = [[sentiment_obj.sentiment.polarity,
                         str(sentiment_obj)] for sentiment_obj in sent

    caption_df["polarity"] = [sentiment_obj.sentiment.polarity for ser

import seaborn as sns

if caption_df.empty != True:
    fig = sns.lineplot(x = "index", y="polarity", data = caption_df)
    fig.set_xticklabels(labels = "")
    fig.set_xlabel("time")
    fig.set_title("Sentiment index for YouTube Video: " + yt.title)
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:28:
DeprecationWarning: Call to deprecated function get_by_language_code
(This object can be treated as a dictionary, i.e. captions['en']).
```



In [ ]: