In [1]:

```
import tweepy
import pprint
import pandas as pd
import datetime
import pytz
```

(30 points) Use Twitter data to create a social network diagram using NetworkX

for the College of Arts & Sciences (@GSUArtSci). a. This social network is three layers deep. First, select 5 friends of “GSUArtSci”. b. For each friend of “GSUArtSci”, select at most 3 friends. For example, if A is a friend of “GSUArtSci”, then select 3 friends of A. c. For each friend of friend of “GSUArtSci”, select at most 3 friends. For example, if B is a friend of A who is a friend of “GSUArtSci”, select at most 3 friends of B. d. There should be an edge between any two nodes that are friends. e. Create a network visualization of the social network using either Plotly or python- graphviz. f. Each node should include the screen name of the Twitter user.

In [2]:

```
auth = tweepy.OAuthHandler("W4wasE9VfZt9E35aqGMwidkOo",
                           "ySEJEo4DMlrD1Rkse6ybA669SdI1N8oeBwl3jasD5vl4PZc20L")
auth.set_access_token("1097579915845750784-6O7ErUlz1YNpUu5Xy0njE2FGCuEtH7",
                      "x0xIRW4q2uT90ICCEvJuZsGW9WWffOv97WnPcGGzvGd1P")

api = tweepy.API(auth)

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

edge_list = pd.DataFrame(columns = ["USER", "FRIEND"])
max_num_friends = 5
max_num_followers = 3
for friend in friends[0:min(len(friends), max_num_friends)]:
    # Create an edge for this connection and add it to the edge list.
    edge_list = edge_list.append({'USER' : user.screen_name,
                                  'FRIEND' : friend.screen_name} ,
                                 ignore_index=True)

    friends_of_friends = friend.friends()

for friend_of_friend in friends_of_friends[0:min(len(friends_of_friends), max_num_f
riends)]:
        edge_list = edge_list.append({'USER' : friend.screen_name,
                                      'FRIEND' : friend_of_friend.screen_name} ,
                                     ignore_index=True)

edge_list
```

Out[2]:

	USER	FRIEND
0	GSUArtSci	GSU_English
1	GSUArtSci	exlab_gsu
2	GSUArtSci	Georgia_Bio
3	GSUArtSci	dustandashco
4	GSUArtSci	AtlSciFestGSU
5	AtlSciFestGSU	cmii_gsu
6	AtlSciFestGSU	OHBM_BrainArt
7	AtlSciFestGSU	williamhu43
8	AtlSciFestGSU	HollyHolm
9	AtlSciFestGSU	GeorgiaStateU

In [3]:

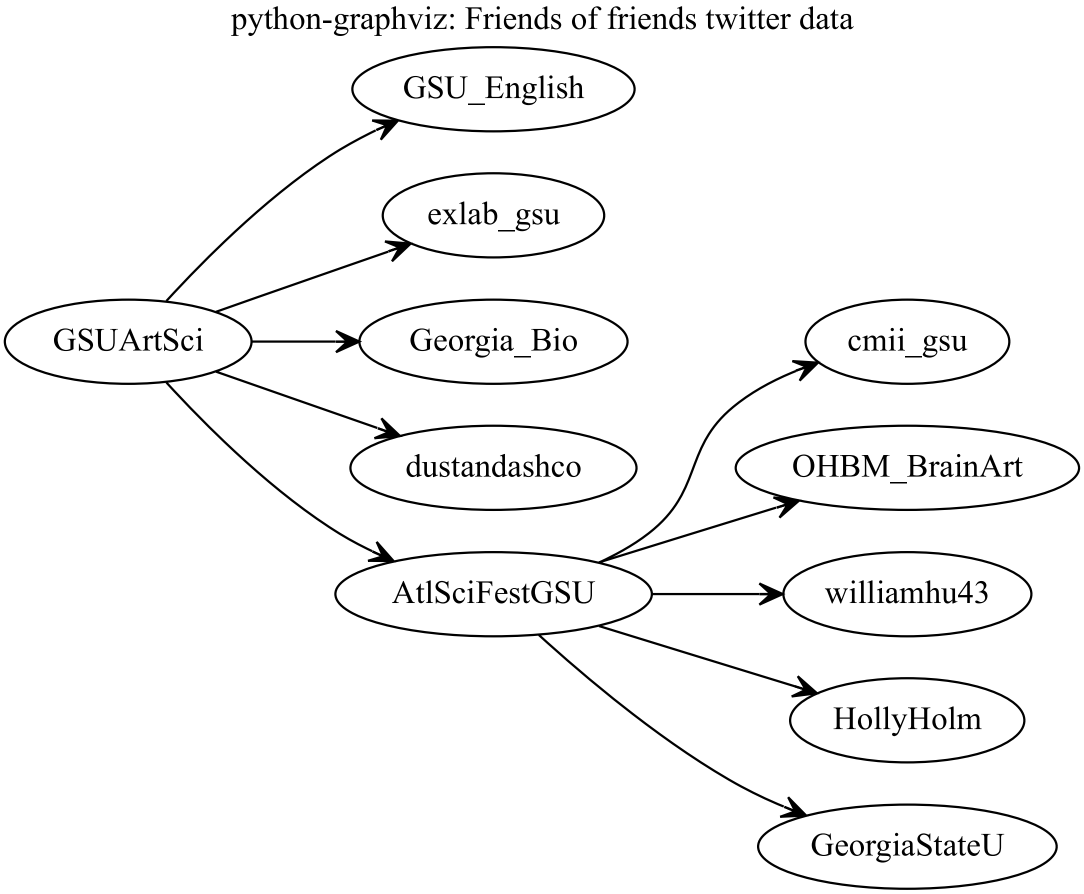
```python
import networkx as nx
import matplotlib.pyplot as plt

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

# Scale the current figure.
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)
```

```
C:\Users\Juney\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:579: Mat
plotlibDeprecationWarning:
The iterable function was deprecated in Matplotlib 3.1 and will be removed in 3.
3. Use np.iterable instead.
  if not cb.iterable(width):
```

In [4]:

```
import graphviz
import pandas as pd

GV = graphviz.Digraph(name = "social network",
                      filename='social_network.gv')

GV.attr("graph",
       rankdir = "LR",
       splines = "spline",
       label = "python-graphviz: Friends of friends twitter data",
       labelloc = "t", # Place the graph label on top
       layout = "dot")

for i in range(0, len(edge_list)):
    GV.edge(tail_name = edge_list.iloc[i]["USER"],
           head_name = edge_list.iloc[i]["FRIEND"],
          arrowhead = "vee")

GV
```

Out[4]:



(20 points) Retrieve the most recent tweets from CDC’s

Twitter account (@ CDCgov). Collect at least 100 tweets (or as many as you can) , excluding retweets. a. Conduct sentiment analysis of the tweets. Calculate the average sentiment index for each day of the last 7 days, ending with the day you write the code.

b. Based on your data, draw a bar plot with Plotly Express (or Plotly) showing the sentiment index for the last 7 days.

In [5]:

```
handle = "CDCemergency"
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)

df = pd.DataFrame(list_tweets)

for i in range(0, len(df)):
     df.loc[i, "datetime"] = datetime.datetime.strptime(df.loc[i, "created_at"],'%a %b %d %H:%M:%S +0000 %Y').replace(tzinfo=pytz.UTC)

df
```

Out[5]:

	created_at	id	id_str	text	truncated	entities	
0	Tue Apr 28 22:15:11 +0000 2020	1255259291176644609	1255259291176644609	Given the shortage of N95 respirators during t...	True	{'hashtags': [{'text': 'COVID19', 'indices': [...	h
1	Tue Apr 28 18:10:11 +0000 2020	1255197637994823680	1255197637994823680	Is your child care program staying open or reo...	True	{'hashtags': [{'text': 'COVID19', 'indices': [...	h
2	Tue Apr 28 15:10:12 +0000 2020	1255152340488695808	1255152340488695808	Protect yourself &amp; others when running ess...	True	{'hashtags': [], 'symbols': [], 'user_mentions...	h
3	Tue Apr 28 15:08:40 +0000 2020	1255151954310750211	1255151954310750211	RT @Surgeon_General: #Telehealth is a valuable...	False	{'hashtags': [{'text': 'Telehealth', 'indices'...	
4	Mon Apr 27 21:45:09 +0000 2020	1254889345254920192	1254889345254920192	Meat and poultry processing facilities face un...	True	{'hashtags': [{'text': 'COVID19', 'indices': [...	h
...	...	...	...	...	...	...	
114	Mon Apr 13 18:00:40 +0000 2020	1249759424157229058	1249759424157229058	Reduce spread of #COVID19. In public, wear a c...	True	{'hashtags': [{'text': 'COVID19', 'indices': [...	h
115	Mon Apr 13 16:04:56 +0000 2020	1249730298947960832	1249730298947960832	RT @CDCDirector: Reopening the US will be a ca...	False	{'hashtags': [], 'symbols': [], 'user_mentions...	
116	Mon Apr 13 16:04:44 +0000 2020	1249730248192798723	1249730248192798723	RT @CDCgov: Ask CDC: Can you get COVID-19 thro...	False	{'hashtags': [], 'symbols': [], 'user_mentions...	
117	Mon Apr 13 16:00:57 +0000 2020	1249729296610078720	1249729296610078720	Be sure to #takebreaks from news and social me...	True	{'hashtags': [{'text': 'takebreaks', 'indices'...	h
118	Mon Apr 13 13:15:22 +0000 2020	1249687624773783556	1249687624773783556	RT @CDCEnvironment: Be prepared for unpredicta...	False	{'hashtags': [{'text': 'tornadoes', 'indices':...	

119 rows × 30 columns

In [6]:

```
from cleantext import clean
from textblob import TextBlob

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(str(len(tweet_text)) + " tweets")

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

# Get sentiment values "polarity"
sentiment_values = [[tweet.sentiment.polarity,
                     str(tweet)] for tweet in sentiment_objects]

sentiment_df = pd.DataFrame(sentiment_values, columns=["polarity", "tweet"])
```

```
Since the GPL-licensed package `unidecode` is not installed, using Python's `uni
codedata` package which yields worse results.

119 tweets
```

In [7]:

```
sentiment_df
```

Out[7]:

	polarity	tweet
0	0.200000	given the shortage of n00 respirators during t...
1	0.183333	is your child care program staying open or reo...
2	0.000000	protect yourself others when running essential...
3	0.500000	rt surgeongeneral telehealth is a valuable too...
4	0.375000	meat and poultry processing facilities face un...
...	...	...
114	0.033333	reduce spread of covid00 in public wear a clot...
115	-0.050000	rt cdcdirector reopening the us will be a care...
116	0.000000	rt cdcgov ask cdc can you get covid through ex...
117	0.244444	be sure to takebreaks from news and social med...
118	0.166667	rt cdcenvironment be prepared for unpredictabl...

119 rows × 2 columns

In [8]:

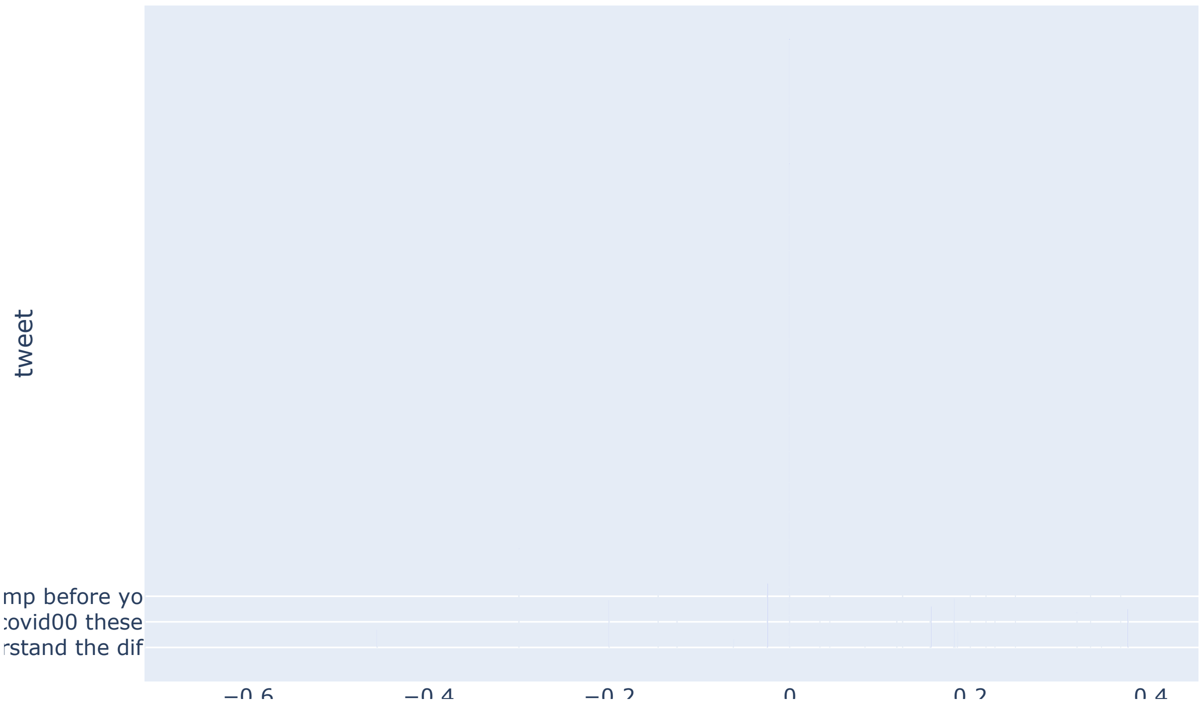
```
sentiment_df = sentiment_df.head(58)
sentiment_df
```

Out[8]:

	polarity	tweet
0	0.200000	given the shortage of n00 respirators during t...
1	0.183333	is your child care program staying open or reo...
2	0.000000	protect yourself others when running essential...
3	0.500000	rt surgeongeneral telehealth is a valuable too...
4	0.375000	meat and poultry processing facilities face un...
5	0.250000	if you have symptoms of covid00 and want to ge...
6	0.000000	rt cdcgov household cleaners and disinfectants...
7	-0.062500	thinking about a trip to the store before maki...
8	0.000000	getting takeout while slowing the spread of co...
9	0.000000	wearing a cloth face covering correctly can he...
10	0.000000	rt cdcgov covid00surge is a spreadsheetbased t...
11	-0.125000	if you have diabetes you are at higher risk fo...
12	0.000000	household cleaners and disinfectants can cause...
13	0.185714	a upmc microbiology laboratory recently receiv...
14	-0.457143	feeling sick answer a few questions about your...
15	0.345455	rt cdcgov the latest cdc covidview report with...
16	0.083333	rt cdcgov if you have covid00 symptoms want to...
17	0.000000	injuries illnesses that are not covid00 still ...
18	-0.125000	take action to slow the spread of covid00 by w...
19	0.333333	rt cdcgov the latest covidview report shows th...
20	0.000000	rt cdcdirector cdcgov to award <cur> million t...
21	0.200000	rt cdcdirector cdcgov is really focused on fla...
22	0.000000	protect yourself others when picking up prescr...
23	0.000000	rt cdcgov were still learning about how covid0...
24	0.000000	rt hhsgov as surgeongeneral explains telemedic...
25	0.155556	rt cdcgov looking for cdc covid00 resources fo...
26	0.200000	given the shortage of n00 respirators during t...
27	0.375000	rt cdcgov adults age + and those with underlyi...
28	0.318182	rt cdcgov new covidnet data reported more than...
29	0.000000	rt cdcenvironment a public tornado shelter can...
30	0.000000	cover your cough covid00 spreads through respi...
31	-0.300000	rt usfda people who have fully recovered from ...
32	0.000000	stayathome means do not leave home unless it i...
33	0.156250	leadbyexample make a habit of practicing your ...
34	0.118750	planahead create an emergency action plan that...
35	0.000000	stayinformed know where to find timely reliabl...
36	0.000000	dont wait until severe weather is in the forec...
37	0.227273	rt cdcgov on march a homeless shelter resident...
38	-0.200000	rt cdcgov you can help slow the spread of covi...
39	0.000000	rt cdcenvironment a public tornado shelter can...

In [9]:

```
import plotly.express as px

data = sentiment_df
fig = px.bar(data, x='polarity', y='tweet')
fig.show()
```



(20 points) Retrieve at least 100 (or as many as you can) tweets that contain

COVID19

and conduct the following data analysis and visualization. a. Clean the text to remove all the URLs, email, number, etc. Remove all the stop words. Convert all words to lower case letters. See my lecture notes for an example. 2 b. Create a histogram plot using Plotly Express (or Plotly) to show the most frequently used words and their frequencies.

In [10]:

```
import nltk
from nltk.corpus import stopwords
keyword = "COVID19" + " -filter:retweets"
since_when = "2020-04-29"
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)):
    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]
print(words)
```

```
['if', 'you', 'already', 'got', 'your', 'stimulus', 'check', 'and', 'need', 'add
itional', 'assistance', 'fema', 'is', 'giving', 'us', 'another', 'one', 'time',
'payment', 'of', 'common', 'denominators', 'for', 'innovation', 'soft', '+', 'ha
rd', 'skills', '+', 'perseverance', '👍🌎', 'digitaltransformation', 'futureofwo
rk', 'the', 'its', 'covid00omo', 'yes', 'indeed', '😂😂😂😂', 'a', 'good', 'rep
ort', 'by', 'ayshahtull', 'on', 'how', 'the', 'covid00', 'epidemic', 'is', 'disp
roportionately', 'affecting', 'the', 'poor', 'however', 'i', 'dont', 'kishanpate
lfit', 'rightangledltd', 'ask', 'him', 'who', 'the', 'lab', 'is', 'he', 'uses',
'for', 'covid00', 'ask', 'him', 'for', 'proof', 'then', 'look', 'that', 'company
', 'up', 'on', 'companies', 'house', 'more', 'days', 'is', 'mamas', 'birthday',
'woah', 'dont', 'know', 'what', 'to', 'do', 'since', 'its', 'covid00', 'heres',
'how', 'lausd', 'can', 'better', 'serve', 'kids', 'with', 'disabilities', 'durin
g', 'and', 'after', 'the', 'pandemic', 'speak', 'up', 'specialeducation', 'nativ
e', 'producers', 'community', 'grocers', 'food', 'hubs', 'cooperatives', 'food',
'businesses', 'and', 'tribalcommunity', 'leaders', 'quickly', 'deploy', 'manage
', 'and', 'monitor', 'your', 'network', 'through', 'a', 'single', 'pane', 'of',
'glass', 'stay', 'connected', 'anywhere', 'dont', 'all', 'covid', 'really', 'sta
rting', 'to', 'bring', 'some', 'strange', 'folk', 'out', 'the', 'woodworks', 'li
ke', 'the', 'random', 'rv', 'parked', 'in', 'front', 'of', 'my', 'house', 'the',
'event', 'which', 'was', 'to', 'go', 'july', 'august', 'is', 'being', 'postponed
', 'because', 'of', 'covid00', 'read', 'more', 'at', 'we', 'have', 'updated', 'o
ur', 'site', 'examining', 'the', 'impact', 'of', 'covid00', 'on', 'specific', 'c
rime', 'trends', 'while', 'overall', 'crime', 'has', 'decrea', 'dixie', 'in', 't
he', 'crosshairs', 'the', 'south', 'is', 'likely', 'to', 'have', 'americas', 'hi
ghest', 'death', 'rate', 'from', 'covid', '|', '\u2066theeconomist\u2069', 'than
k', 'you', 'themotivatur', 'things', 'that', 'protect', 'my', 'mentalhealth', 'd
uring', 'covid00', 'include', 'awakening', 'to', 'daily', 'pray', 'great', 'list
ening', 'to', 'jelani0', 'education', 'should', 'be', 'liberatory', 'allowing',
'ss', 'to', 'name', 'their', 'oppression', 'the', 'ongoing', 'bostonstrongb', 'c
aptrwrpnts', 'actionp00', 'acjjustice', 'dcooperresists', 'bjcreigh', 'avestige0
', 'cannabizlawyr', 'brat0000', 'robtregaskes', 'matthewsgould', 'nhsx', 'apple
', 'google', 'surely', 'contact', 'tracing', 'has', 'to', 'be', 'precursor', 'to
', 'getting', 'covid00', 'te', 'magats', 'are', 'torn', 'between', 'the', 'new',
'world', 'order', 'depopulation', 'conspiracy', 'and', 'the', 'covid00', 'is', '
no', 'big', 'deal', 'lie', 'it', 'cant', 'be', 'both', 'inners', 'mayoclinic', '
vp', 'govtimwalz', 'ive', 'just', 'read', 'all', 'the', 'comments', 'and', 'agre
e', 'with', 'the', 'popular', 'opinion', 'vp', 'is', 'supposed', 'to', 'gtconway
0d', 'pence', 'is', 'here', 'inside', 'a', 'clinic', 'with', 'covid00', 'patient
s', 'no', 'mask', 'good', 'luck', 'with', 'getting', 'there', 'pence', 'tedlieu
', 'just', 'got', 'schooled', 'by', 'tuckercarlson', 'covid00', 'celebrating', '
our', 'doctors', 'and', 'nurses', 'around', 'the', '🌎world🌎', 'during', 'the',
'covid00☣', 'pandemic', 'stayhome', 'thomsonreuters', 'answerson', 'compiled',
'a', 'list', 'of', 'covid00', 'costcutting', 'measures', 'at', 'us', 'law', 'fir
ms', 'pay', 'cuts', 'esp', 'for', 'high', 'thanks', 'to', 'our', 'partner', 'who
', 'compiled', 'the', 'list', 'covid00', 'cuhlmann', '0newsaus', 'google', 'blac
ktown', 'childcare', 'covid00', 'these', 'rock', 'strength', 'stones', 'outside
', 'the', 'staff', 'entrance', '❤', 'covid00', 'inthistogether', 'almasthela', '
purviparwani', 'iamritu', 'mariovar00', 'oncocardiology', 'maecocardio', 'garcia
edinson00', 'tavoave', 'wikimagen', 'speakoutapril', 'if', 'there', 'is', 'nothi
ng', 'to', 'it', 'then', 'why', 'in', 'the', 'hell', 'does', 'google', 'own', 't
he', 'patent', 'for', 'it', 'in', 'just', 'months', 'the', 'coronavirus', 'has',
'killed', 'more', 'americans', 'than', 'years', 'of', 'vietnamwar', 'it', 'took
', '<phone>', 'for', 'the', 'u', 'fredtjoseph', 'single', 'mom', 'been', 'needin
g', 'help', 'since', 'day', 'od', 'rentrelief', 'and', 'still', 'no', 'help', 'w
e', 'have', 'no', 'food', 'and', 'i', 'have', 'starting', 'tomorrow', 'la', 'con
struction', 'workers', 'are', 'eligible', 'with', 'or', 'without', 'symptoms', '
can', 'get', 'tested', 'for', 'covid00', 'coroanvirus', 'says', 'garcetti', 'say
sdana', 'justinamash', 'joebiden', 'he', 'wants', 'to', 'pick', 'up', 'all', 'th
e', 'gopers', 'who', 'lost', 'a', 'grandma', 'to', 'covid00', 'and', 'are', 'but
thurt', 'if', 'the', 'ratios', 'hold', 'up', 'the', 'mortality', 'rate', 'for',
'reported', 'covid00', 'cases', '=', '00k', '00k', '+', '000k', '=', 'but', 'man
y', 'perh', 'lets', 'start', 'virtual', 'learning', 'book', 'a', 'demo', 'now',
'virtualclassroom', 'eduserv', 'lms', 'onlinelearning', 'hllfrezenovr', 'sorry',
'to', 'hear', 'the', 'justice', 'system', 'in', 'canada', 'is', 'not', 'set', 'u
```

In [11]:

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

```
['already', 'got', 'stimulus', 'check', 'need', 'additional', 'assistance', 'fem
a', 'giving', 'us', 'another', 'one', 'time', 'payment', 'common', 'denominators
', 'innovation', 'soft', '+', 'hard', 'skills', '+', 'perseverance', '👍🌎', 'di
gitaltransformation', 'futureofwork', 'covid00omo', 'yes', 'indeed', '😂😂😂😂
', 'good', 'report', 'ayshahtull', 'covid00', 'epidemic', 'disproportionately',
'affecting', 'poor', 'however', 'dont', 'kishanpatelfit', 'rightangledltd', 'ask
', 'lab', 'uses', 'covid00', 'ask', 'proof', 'look', 'company', 'companies', 'ho
use', 'days', 'mamas', 'birthday', 'woah', 'dont', 'know', 'since', 'covid00', '
heres', 'lausd', 'better', 'serve', 'kids', 'disabilities', 'pandemic', 'speak',
'specialeducation', 'native', 'producers', 'community', 'grocers', 'food', 'hubs
', 'cooperatives', 'food', 'businesses', 'tribalcommunity', 'leaders', 'quickly
', 'deploy', 'manage', 'monitor', 'network', 'single', 'pane', 'glass', 'stay',
'connected', 'anywhere', 'dont', 'covid', 'really', 'starting', 'bring', 'strang
e', 'folk', 'woodworks', 'like', 'random', 'rv', 'parked', 'front', 'house', 'ev
ent', 'go', 'july', 'august', 'postponed', 'covid00', 'read', 'updated', 'site',
'examining', 'impact', 'covid00', 'specific', 'crime', 'trends', 'overall', 'cri
me', 'decrea', 'dixie', 'crosshairs', 'south', 'likely', 'americas', 'highest',
'death', 'rate', 'covid', '|', '\u2066theeconomist\u2069', 'thank', 'themotivatu
r', 'things', 'protect', 'mentalhealth', 'covid00', 'include', 'awakening', 'dai
ly', 'pray', 'great', 'listening', 'jelani0', 'education', 'liberatory', 'allowi
ng', 'ss', 'name', 'oppression', 'ongoing', 'bostonstrongb', 'captrwrpnts', 'act
ionp00', 'acjjustice', 'dcooperresists', 'bjcreigh', 'avestige0', 'cannabizlawyr
', 'brat0000', 'robtregaskes', 'matthewsgould', 'nhsx', 'apple', 'google', 'sure
ly', 'contact', 'tracing', 'precursor', 'getting', 'covid00', 'te', 'magats', 't
orn', 'new', 'world', 'order', 'depopulation', 'conspiracy', 'covid00', 'big', '
deal', 'lie', 'cant', 'inners', 'mayoclinic', 'vp', 'govtimwalz', 'ive', 'read',
'comments', 'agree', 'popular', 'opinion', 'vp', 'supposed', 'gtconway0d', 'penc
e', 'inside', 'clinic', 'covid00', 'patients', 'mask', 'good', 'luck', 'getting
', 'pence', 'tedlieu', 'got', 'schooled', 'tuckercarlson', 'covid00', 'celebrati
ng', 'doctors', 'nurses', 'around', '🌎world🌎', 'covid00☣', 'pandemic', 'stayh
ome', 'thomsonreuters', 'answerson', 'compiled', 'list', 'covid00', 'costcutting
', 'measures', 'us', 'law', 'firms', 'pay', 'cuts', 'esp', 'high', 'thanks', 'pa
rtner', 'compiled', 'list', 'covid00', 'cuhlmann', '0newsaus', 'google', 'blackt
own', 'childcare', 'covid00', 'rock', 'strength', 'stones', 'outside', 'staff',
'entrance', '🖤', 'covid00', 'inthistogether', 'almasthela', 'purviparwani', 'iam
ritu', 'mariovar00', 'oncocardiology', 'maecocardio', 'garciaedinson00', 'tavoav
e', 'wikimagen', 'speakoutapril', 'nothing', 'hell', 'google', 'patent', 'months
', 'coronavirus', 'killed', 'americans', 'years', 'vietnamwar', 'took', '<phone>
', 'u', 'fredtjoseph', 'single', 'mom', 'needing', 'help', 'since', 'day', 'od',
'rentrelief', 'still', 'help', 'food', 'starting', 'tomorrow', 'la', 'constructi
on', 'workers', 'eligible', 'without', 'symptoms', 'get', 'tested', 'covid00', '
coroanvirus', 'says', 'garcetti', 'saysdana', 'justinamash', 'joebiden', 'wants
', 'pick', 'gopers', 'lost', 'grandma', 'covid00', 'butthurt', 'ratios', 'hold',
'mortality', 'rate', 'reported', 'covid00', 'cases', '=', '00k', '00k', '+', '00
0k', '=', 'many', 'perh', 'lets', 'start', 'virtual', 'learning', 'book', 'demo
', 'virtualclassroom', 'eduserv', 'lms', 'onlinelearning', 'hllfrezenovr', 'sorr
y', 'hear', 'justice', 'system', 'canada', 'set', 'keep', 'criminals', 'locked',
'marco', 'muzzo', 'wa', 'tested', 'gray', 'coverage', 'sharpie', 'actually', 'wo
rked', 'sure', 'permanent', 'husba', 'friend', 'shared', 'today', 'sad', 'every
', 'covid00', 'death', 'story', 'case', 'dear', 'physician', 'e', 'covid00', 'up
date', 'mayorofla', 'says', 'hopes', 'open', 'testing', 'asymptomatic', 'angelen
os', 'coming', 'weeks', 'knx0000', 'consequences', 'covid00', 'economic', 'fallo
ut', 'means', 'equal', 'queensland', 'executive', 'chair', 'ericmoranfilms', 'di
abeetuscat', 'joeysalads', 'keep', 'pushing', 'next', 'thing', 'know', 'problems
', 'covi', 'trump', 'flooding', 'reddit', 'slanderous', 'ridiculous', 'attack',
'ads', 'biden', 'honestly', 'hope', 'corona', 'gets', 'trump', 'feeling', 'covid
00', 'craziness', 'trying', 'help', 'parents', 'lost', 'house', 'tornad', 'hereb
eproof', 'kean0s', 'jorichardskent', 'weneedeu', 'abcpoppins', 'brexitbin', 'roa
dwarrior00', 'sillyshib', 'bbc', 'mad', 'strange', 'times', 'thought', 'hoax', '
promises', 'game', 'changer', 'anyone', 'else', 'increasingly', 'con', 'corona',
'finally', 'revealed', 'redtabletalk', 'redtabletalk', 'marcglovercomedy', 'jada
', 'willow', 'adrianne', 'gammy', 'comedy', 'states', 'back', 'business', 'covid
00', 'coronavirus', 'backtobusiness', 'covidiots', 'covid—00', 'great', 'post',
'mikeatalla', 'racial', 'segregation', 'disparities', 'large', 'cities', 'americ
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Juney\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [12]:

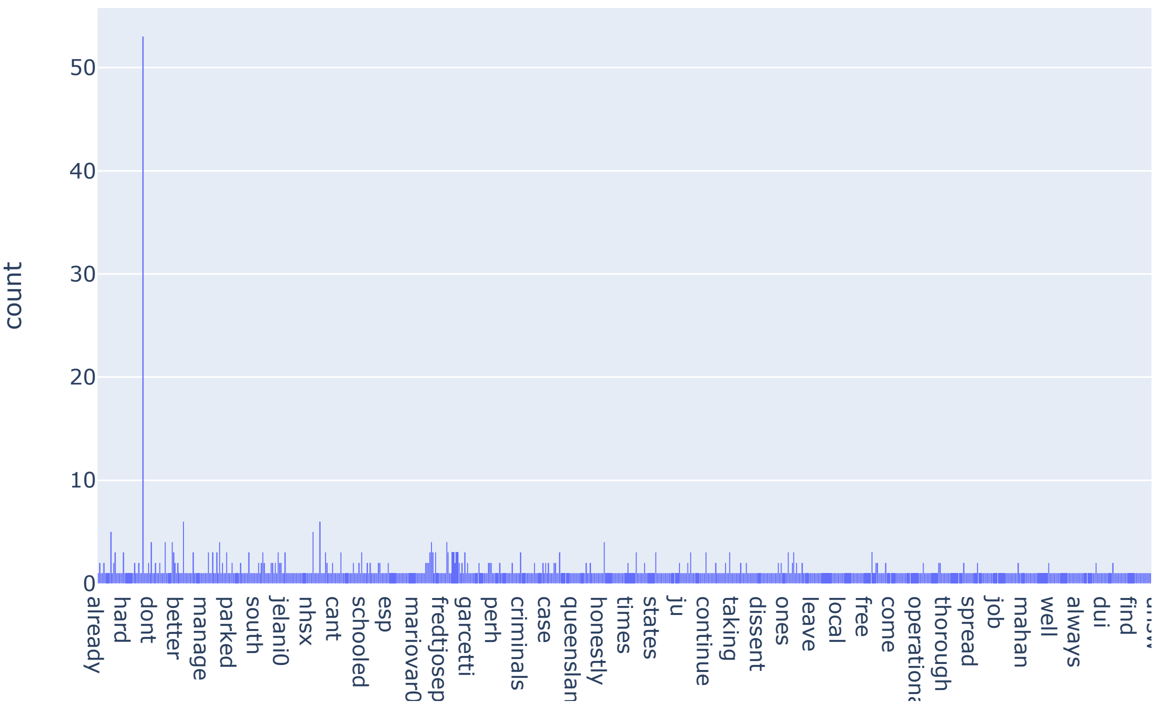
```
df = pd.DataFrame(words, columns=["word"])

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

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

word_frequency
```

Out[12]:

	word	frequency
0	covid00	53
1	pandemic	6
2	getting	6
3	google	5
4	us	5
...	...	...
782	yeast	1
783	layoffs	1
784	school	1
785	bullshit	1
786	councilmember	1

787 rows × 2 columns

In [13]:
```
fig = px.histogram(df, x="word")
fig.show()
```



Retrieve captions from the following YouTube videos, conduct sentiment

analysis , and draw a line plot showing the sentiment index over time using Plotly Express (or Plotly). a. (15 points) Create a sentiment timeline for this video: https://www.youtube.com/watch?v=6Af6b_wyiwI (https://www.youtube.com/watch?v=6Af6b_wyiwI) b. (15 points) Create a sentiment timeline for a YouTube video of your choice.

In [14]:
```
from pytube import YouTube
```

In [15]:

```
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_3d.append(stream_info["is_3d"])

    caption_lang = []

    for caption in yt.captions.all():
        caption_info = caption.__dict__
        caption_lang.append(caption_info["name"])

    print("title: " + yt.title)
    print("author: " + yt.author)
    print("length: " + str(yt.length/60) + " minutes")
    print("views: " + str(yt.views))
    print("rating: " + str(yt.rating))

    # Convert a list to a set to remove duplicates.
    print("mime_type: " + str(set(mime_type)))
    print("type: " + str(set(stream_type)))
    print("fps: " + str(set(fps)))
    print("resolution: " + str(set(resolution)))
    print("is_live: " + str(set(is_live)))
    print("is_3d: " + str(set(is_3d)))
    print("caption languages: " + str(set(caption_lang)))

    print("-------------------------")
    print("description" + "(max " + str(num_chars) + " characters): " + yt.descript
ion[:(min(num_chars, len(yt.description)))])

get_youtube_info(yt, 500)
```

```
title: The next outbreak? We’re not ready | Bill Gates
author: TED
length: 8.616666666666667 minutes
views: 27641117
rating: 4.8341508
mime_type: {'video/mp4', 'audio/mp4', 'audio/webm', 'video/webm'}
type: {'video', 'audio'}
fps: {30}
resolution: {'360p', '144p', None, '240p', '480p', '720p'}
is_live: {False}
is_3d: {False}
caption languages: {'Dutch', 'French (Canada)', 'Czech', 'Galician', 'Persian',
'Italian', 'Vietnamese', 'Korean', 'Serbian', 'Chinese (China)', 'Portuguese (Po
rtugal)', 'Turkish', 'Croatian', 'Slovak', 'Russian', 'Uzbek', 'French', 'Portug
uese (Brazil)', 'Latvian', 'Ukrainian', 'Polish', 'Chinese (Hong Kong)', 'Hebrew
', 'Hungarian', 'Bulgarian', 'Chinese (Taiwan)', 'Danish', 'Indonesian', 'Englis
h', 'Macedonian', 'Romanian', 'Spanish', 'Arabic', 'Lithuanian', 'German', 'Burm
ese', 'Greek', 'Japanese', 'Mongolian', 'Thai', 'Swedish'}
-------------------------
description(max 500 characters): Visit http://TED.com to get our entire library
of TED Talks, transcripts, translations, personalized talk recommendations and m
ore.

In 2014, the world avoided a horrific global outbreak of Ebola, thanks to thousa
nds of selfless health workers -- plus, frankly, thanks to some very good luck.
In hindsight, we know what we should have done better. So, now's the time, Bill
Gates suggests, to put all our good ideas into practice, from scenario planning
to vaccine research to health worker training.
```

```
C:\Users\Juney\Anaconda3\lib\site-packages\ipykernel_launcher.py:12: Deprecation
Warning:

Call to deprecated function all (This object can be treated as a list, all() is
useless).

C:\Users\Juney\Anaconda3\lib\site-packages\ipykernel_launcher.py:23: Deprecation
Warning:

Call to deprecated function all (This object can be treated as a dictionary).
```

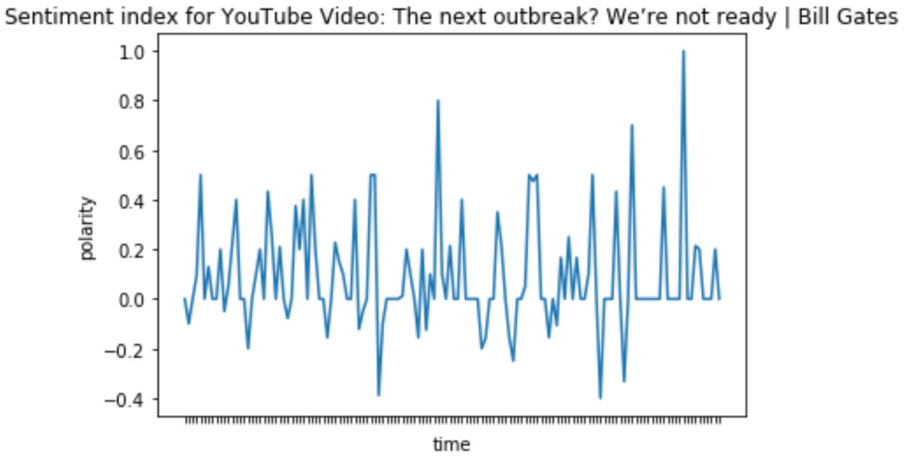
In [16]:

```
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_lines_per_ite
m):
        nested.append(caption_lines[ix:ix + num_lines_per_item])
        caption_df = pd.DataFrame(nested, columns = ["index", "time", "text", "line
_break"])
        caption_df = caption_df.drop(columns = ["line_break"])
        caption_df
```

```
C:\Users\Juney\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: DeprecationW
arning:

Call to deprecated function get_by_language_code (This object can be treated as
a dictionary, i.e. captions['en']).
```

```
In [17]: if caption_df.empty != True:
             sentiment_objects = [TextBlob(caption) for caption in caption_df["text"]]
             sentiment_values = [[sentiment_obj.sentiment.polarity,
                                  str(sentiment_obj)] for sentiment_obj in sentiment_object
         s]
             caption_df["polarity"] = [sentiment_obj.sentiment.polarity for sentiment_obj in
         sentiment_objects]

             caption_df
```

```
In [18]: import seaborn as sns
         if caption_df.empty != True:
             fig = sns.lineplot(x = "index", y="polarity", data = caption_df)
             # Remove the X tick labels because it's too crowded.
             fig.set_xticklabels(labels = "")
             fig.set_xlabel("time")
             fig.set_title("Sentiment index for YouTube Video: " + yt.title)
```



My random video to do a timeline

In [19]:

```
youTubeURL = "https://www.youtube.com/watch?v=eZUKSxE2UZg"

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 = []

    for caption in yt.captions.all():
        caption_info = caption.__dict__
        caption_lang.append(caption_info["name"])

    print("title: " + yt.title)
    print("author: " + yt.author)
    print("length: " + str(yt.length/60) + " minutes")
    print("views: " + str(yt.views))
    print("rating: " + str(yt.rating))

    # Convert a list to a set to remove duplicates.
    print("mime_type: " + str(set(mime_type)))
    print("type: " + str(set(stream_type)))
    print("fps: " + str(set(fps)))
    print("resolution: " + str(set(resolution)))
    print("is_live: " + str(set(is_live)))
    print("is_3d: " + str(set(is_3d)))
    print("caption languages: " + str(set(caption_lang)))

    print("-------------------------")
    print("description" + "(max " + str(num_chars) + " characters): " + yt.descript
ion[:(min(num_chars, len(yt.description)))])

get_youtube_info(yt, 500)
```

```
title: Quarantine Stereotypes
author: Dude Perfect
length: 9.716666666666667 minutes
views: 7235161
rating: 4.9545255
mime_type: {'video/mp4', 'audio/mp4', 'audio/webm', 'video/webm'}
type: {'video', 'audio'}
fps: {30}
resolution: {'1080p', '360p', '144p', '240p', None, '480p', '720p'}
is_live: {False}
is_3d: {False}
caption languages: {'English'}
-------------------------
description(max 500 characters): Quarantine Stereotypes. Love 'em or hate 'em, w
e all know 'em.
We dedicate this video to all our heroes on the front lines!
Please use the donate button to give to Feeding America!

The Dude Perfect Documentary comes out May 11 for FREE!
▶ Watch the Trailer - https://youtu.be/jf9Iue_Fwhs

COMMENT which Stereotype character is in your family for a chance to receive a #
DudePerfectDoc Quarantine Kit full of fun stuff!

▶ Thanks for subscribing! - http://bit.ly/SubDudePerfect
Thanks for staying hom

C:\Users\Juney\Anaconda3\lib\site-packages\ipykernel_launcher.py:12: Deprecation
Warning:

Call to deprecated function all (This object can be treated as a list, all() is
useless).

C:\Users\Juney\Anaconda3\lib\site-packages\ipykernel_launcher.py:23: Deprecation
Warning:

Call to deprecated function all (This object can be treated as a dictionary).
```

In [20]:

```
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_lines_per_ite
m):
        nested.append(caption_lines[ix:ix + num_lines_per_item])
        caption_df = pd.DataFrame(nested, columns = ["index", "time", "text", "line
_break"])
        caption_df = caption_df.drop(columns = ["line_break"])
        caption_df
```

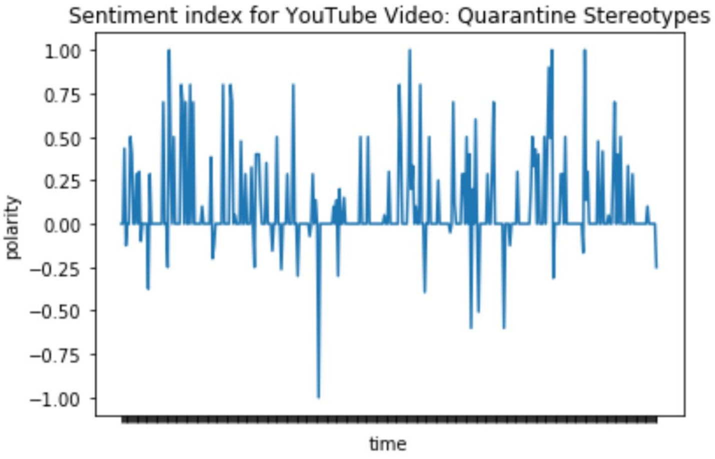
```
C:\Users\Juney\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: DeprecationW
arning:

Call to deprecated function get_by_language_code (This object can be treated as
a dictionary, i.e. captions['en']).
```

In [21]:

```
if caption_df.empty != True:
    sentiment_objects = [TextBlob(caption) for caption in caption_df["text"]]
    sentiment_values = [[sentiment_obj.sentiment.polarity,
                         str(sentiment_obj)] for sentiment_obj in sentiment_object
s]
    caption_df["polarity"] = [sentiment_obj.sentiment.polarity for sentiment_obj in
sentiment_objects]

    caption_df
```

In [22]:

```
import seaborn as sns
if caption_df.empty != True:
    fig = sns.lineplot(x = "index", y="polarity", data = caption_df)
    # Remove the X tick labels because it's too crowded.
    fig.set_xticklabels(labels = "")
    fig.set_xlabel("time")
    fig.set_title("Sentiment index for YouTube Video: " + yt.title)
```



In [ ]: