# Team Brainstorm - Bloomberg Challenge - TAMU Datathon 2022

Jacob Mashburn, Molly Mckinney, Moeka Ono, Erin Mok

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()  # for plot styling
import numpy as np
import pandas as pd
import json
```

# PART 1

## Guesses for each article (methodology to follow)

1. An article on recommended travel/tourist destinations.
2. Nice things happening in the pandemic: stimulus checks, Matthew Mcconaughey showing people how to make a mask out of a coffee filter, and people buying coffees for essential workers.
3. An article written during/about the COVID-19 pandemic.
4. The war on terror in the Middle East.
5. Long-term consequences of climate change, maybe with some remarks on what we can do about it now.

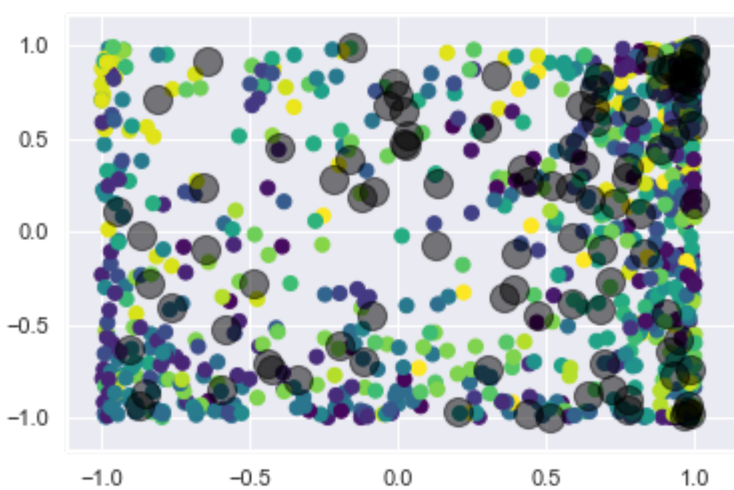
## Training dataset importing

In [2]:
```
cnn_samples0 = pd.read_csv('cnn_samples-54b19b96f3c0775b116bad527df8c7b5.cs
v')

# Wrangling the data from strings to NP arrays.
cnn_samples1 = np.fromstring((cnn_samples0.values[0,3]).replace('[','').rep
lace(']',''), sep=',').reshape(1,512)

# Rebuilding the DataFrame after this, with headline as index.
for i in np.arange(1,np.shape(cnn_samples0)[0]):
    temp = np.fromstring((cnn_samples0.values[i,3]).replace('[','').replace
(']',''), sep=',').reshape(1,512)
    cnn_samples1 = np.vstack([cnn_samples1, temp])
cnn_samples = pd.DataFrame(cnn_samples1, index = cnn_samples0['text'])

#Repeating the process for the challenge data.
gov_samples0 = pd.read_csv('federal_samples-a586d0681e005629453435bea5b173e
b.csv')
gov_samples1 = np.fromstring((gov_samples0.values[0,3]).replace('[','').rep
lace(']',''), sep=',').reshape(1,512)
for i in np.arange(1,np.shape(gov_samples0)[0]):
    temp = np.fromstring((gov_samples0.values[i,3]).replace('[','').replace
(']',''), sep=',').reshape(1,512)
    gov_samples1 = np.vstack([gov_samples1, temp])
gov_samples = pd.DataFrame(gov_samples1, index = gov_samples0['text'])

#also need to merge the two DataFrames
cnngov_samples = pd.concat([cnn_samples, gov_samples], axis = 'rows')

#Repeating the process for the challenge data.
challenge0 = pd.read_csv('challenge-ddec63cf66ea88f128e3c21e457f393a.csv')
challenge1 = np.fromstring((challenge0.values[0,1]).replace('[','').replace
(']',''), sep=',').reshape(1,512)
for i in np.arange(1,np.shape(challenge0)[0]):
    temp = np.fromstring((challenge0.values[i,1]).replace('[','').replace
(']',''), sep=',').reshape(1,512)
    challenge1 = np.vstack([challenge1, temp])
challenge = pd.DataFrame(challenge1, index = challenge0['id'])


#Finally, getting the mystery 6th article.
with open('mystery.json') as file:
    mystery0 = json.load(file)['embedding']

mystery = pd.DataFrame(np.array(mystery0).reshape(1,512), index = ['mystery
'], columns = np.arange(0,512)) #it's a dict

#also need to merge these two DataFrames
challenge = pd.concat([challenge, mystery], axis = 'rows')

#print(cnngov_samples.head()) #just double-checking
#print(challenge.head())
```

## Trying K-means (as a rough starting point with no labels whatsoever)

In [3]:
```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=100)
kmeans.fit(cnngov_samples)
y_kmeans = kmeans.predict(cnngov_samples) #in addition to coloring, we can
use these to make Boolean masks to view contents of each cluster KM predict
ed for the challenge articles.
print(type(y_kmeans))

plt.scatter(cnngov_samples.values[:, 0], cnngov_samples.values[:, 1], c=y_k
means, s=50, cmap='viridis')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);

#the moment of truth
challengeKM = kmeans.predict(challenge)
print(challengeKM)

#great, let's mask the original cnngov_sample DataSeries with Boolean array
s generated by y_kmeans
article1KM = y_kmeans == challengeKM[0]
article2KM = y_kmeans == challengeKM[1]
article3KM = y_kmeans == challengeKM[2]
article4KM = y_kmeans == challengeKM[3]
article5KM = y_kmeans == challengeKM[4]
article6KM = y_kmeans == challengeKM[5]

clusterArticle1KM = cnngov_samples[article1KM].index.values.tolist()
clusterArticle2KM = cnngov_samples[article2KM].index.values.tolist()
clusterArticle3KM = cnngov_samples[article3KM].index.values.tolist()
clusterArticle4KM = cnngov_samples[article4KM].index.values.tolist()
clusterArticle5KM = cnngov_samples[article5KM].index.values.tolist()
clusterArticle6KM = cnngov_samples[article6KM].index.values.tolist()
```

```
<class 'numpy.ndarray'>
[40 67 53  3 94 17]
```



## Trying K-Nearest Neighbor with a TF-IDF Vectorizer as Target Array (to count/weigh important words)

In [4]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

# TF-IDF word frequency counter (for headlines)
vec = TfidfVectorizer()
cnnX = vec.fit_transform(pd.concat([cnn_samples0['text'],gov_samples0['text
']]))
Y = pd.DataFrame(cnnX.toarray(), columns=vec.get_feature_names())
Y = Y.drop(columns = stopwords.words('english'), errors = 'ignore') #remove
filler words (Stopwords)

top_n = 12 #return the top N words from each source, except the WordCloud S
topwords.
Z = pd.DataFrame({n: Y.T[col].nlargest(top_n).index.tolist()
                  for n, col in enumerate(Y.T)}).T

#print(Z)

knnModel = KNeighborsClassifier(n_neighbors= 8, algorithm = 'auto', p = 2,
metric = 'cosine')
knnModel.fit(cnngov_samples, Z)

#Had to comment this out for space
print('Probability estimates for the test data:')
print(knnModel.predict_proba(challenge))

# the moment of truth...
challengeKNN = knnModel.kneighbors(challenge, n_neighbors = 8, return_dista
nce = True)
print('The challenge articles\' 5 nearest neighbors according to the KNN al
gorithm:')
print(challengeKNN[1])

article1KNN = (np.arange(0,len(cnngov_samples)) == challengeKNN[1][0,0]) +
(np.arange(0,len(cnngov_samples)) == challengeKNN[1][0,1]) + (np.arange(0,l
en(cnngov_samples)) == challengeKNN[1][0,2]) + (np.arange(0,len(cnngov_samp
les)) == challengeKNN[1][0,3]) + (np.arange(0,len(cnngov_samples)) == chall
engeKNN[1][0,4]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][0,
5]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][0,6]) + (np.aran
ge(0,len(cnngov_samples)) == challengeKNN[1][0,7])
article2KNN = (np.arange(0,len(cnngov_samples)) == challengeKNN[1][1,0]) +
(np.arange(0,len(cnngov_samples)) == challengeKNN[1][1,1]) + (np.arange(0,l
en(cnngov_samples)) == challengeKNN[1][1,2]) + (np.arange(0,len(cnngov_samp
les)) == challengeKNN[1][1,3]) + (np.arange(0,len(cnngov_samples)) == chall
engeKNN[1][1,4]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][1,
5]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][1,6]) + (np.aran
ge(0,len(cnngov_samples)) == challengeKNN[1][1,7])
article3KNN = (np.arange(0,len(cnngov_samples)) == challengeKNN[1][2,0]) +
(np.arange(0,len(cnngov_samples)) == challengeKNN[1][2,1]) + (np.arange(0,l
en(cnngov_samples)) == challengeKNN[1][2,2]) + (np.arange(0,len(cnngov_samp
les)) == challengeKNN[1][2,3]) + (np.arange(0,len(cnngov_samples)) == chall
engeKNN[1][2,4]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][2,
```

```
5]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][2,6]) + (np.aran
ge(0,len(cnngov_samples)) == challengeKNN[1][2,7])
article4KNN = (np.arange(0,len(cnngov_samples)) == challengeKNN[1][3,0]) +
(np.arange(0,len(cnngov_samples)) == challengeKNN[1][3,1]) + (np.arange(0,l
en(cnngov_samples)) == challengeKNN[1][3,2]) + (np.arange(0,len(cnngov_samp
les)) == challengeKNN[1][3,3]) + (np.arange(0,len(cnngov_samples)) == chall
engeKNN[1][3,4]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][3,
5]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][3,6]) + (np.aran
ge(0,len(cnngov_samples)) == challengeKNN[1][3,7])
article5KNN = (np.arange(0,len(cnngov_samples)) == challengeKNN[1][4,0]) +
(np.arange(0,len(cnngov_samples)) == challengeKNN[1][4,1]) + (np.arange(0,l
en(cnngov_samples)) == challengeKNN[1][4,2]) + (np.arange(0,len(cnngov_samp
les)) == challengeKNN[1][4,3]) + (np.arange(0,len(cnngov_samples)) == chall
engeKNN[1][4,4]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][4,
5]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][4,6]) + (np.aran
ge(0,len(cnngov_samples)) == challengeKNN[1][4,7])
article6KNN = (np.arange(0,len(cnngov_samples)) == challengeKNN[1][5,0]) +
(np.arange(0,len(cnngov_samples)) == challengeKNN[1][5,1]) + (np.arange(0,l
en(cnngov_samples)) == challengeKNN[1][5,2]) + (np.arange(0,len(cnngov_samp
les)) == challengeKNN[1][5,3]) + (np.arange(0,len(cnngov_samples)) == chall
engeKNN[1][5,4]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][5,
5]) + (np.arange(0,len(cnngov_samples)) == challengeKNN[1][5,6]) + (np.aran
ge(0,len(cnngov_samples)) == challengeKNN[1][5,7])

clusterArticle1KNN = cnngov_samples[article1KNN].index.values.tolist()
clusterArticle2KNN = cnngov_samples[article2KNN].index.values.tolist()
clusterArticle3KNN = cnngov_samples[article3KNN].index.values.tolist()
clusterArticle4KNN = cnngov_samples[article4KNN].index.values.tolist()
clusterArticle5KNN = cnngov_samples[article5KNN].index.values.tolist()
clusterArticle6KNN = cnngov_samples[article6KNN].index.values.tolist()


print('Each of those nearest neighbor\'s most frequent words:')
print(Z[article1KNN])
print(Z[article2KNN])
print(Z[article3KNN])
print(Z[article4KNN])
print(Z[article5KNN])
print(Z[article6KNN])
```

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

```
Probability estimates for the test data:
[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.   , 0.125],
       [0.   , 0.   , 0.   , ..., 0.   , 0.   , 0.   ]]), 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.]]), 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.]]), array([[0.125, 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.   ]]), 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.]]), 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.]]), 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.   , 0.125]]), 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.   , 0.125, 0.   , ..., 0.   , 0.   , 0.   ],
       [0.   , 0.   , 0.   , ..., 0.   , 0.   , 0.   ]]), array([[0.   , 0.
, 0.   , ..., 0.   , 0.   , 0.   ],
       [0.   , 0.   , 0.   , ..., 0.   , 0.   , 0.125],
       [0.   , 0.   , 0.   , ..., 0.   , 0.   , 0.   ],
       [0.   , 0.   , 0.   , ..., 0.   , 0.   , 0.   ],
       [0.   , 0.   , 0.   , ..., 0.   , 0.   , 0.   ],
       [0.   , 0.   , 0.   , ..., 0.   , 0.   , 0.   ]]), 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.   , 0.125, 0.   ],
       [0.   , 0.   , 0.   , ..., 0.   , 0.   , 0.   ]]), 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.]]), 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.   , 0.125],
       [0.   , 0.   , 0.   , ..., 0.   , 0.   , 0.   ]])]
The challenge articles' 5 nearest neighbors according to the KNN algorithm:
[[657  67 530 673 428 329 465 146]
 [133 542 105 499 496 124 431 388]
 [ 64  75  29 339  69 304  38 343]
 [659 480  89 223 567 314 380 714]
 [705 419 274 446 511 213 411  74]
 [144 261 286  35 469 589 578 455]]
Each of those nearest neighbor's most frequent words:
              0           1         2         3       4            5   \
67        dates       valid      2014    travel   cyber         book
146     hornets      humans    forage    hornet    food         wild
329     ukraine      rooney   england    sweden   cross         shot
428     airport      ciudad   spanish      land    real       sounds
465  paralympic  paralympics     medal      100m  london     mcfadden
530       space    training  garriott    flight     air        mouth
657      rahman        film     music  composer  indian      singing
673    legoland        park     water  malaysia    lego  attractions

             6          7            8           9            10
11
67    blackout   december     offering       hotel          rates         de
al
146     larvae      asian       insect       giant        insects      stin
gs
329     france       hart      larsson       hosts  spectacularly     minut
es
428  almodovar    ryanair       runway      bubble          perez    proper
ty
465   athletes   coverage       sprint  wheelchair         sports  disabili
ty
530     meters  preparing  acclimatize  adventures             gs  procedur
es
657     melody      movie     explained       theme      hollywood     jammi
ng
673      theme     bricks         raft      slides          hotel       reso
rt
                0          1        2            3           4            5
\
105  mcconaughey      alves    maybe       camila     planner  wonderfully
```

```
124      hortons        cups      tim        coffee          ctv     edmonton
133         bank         oct  billion          sept    announces           uk
388       samuel       moshe     baby        chabad        nanny        rabbi
431      letters    brencher     love  organization      writing       letter
496     protests   financial   police        sector      protest     thursday
499        lindt      tanner    franc     chocolate        brand      markets
542      dempsey       tully   coffee       seattle   bankruptcy    locations

          6               7         8          9           10           11
105    knot             wed  intimate   longtime       austin    brazilian
124  alberta            500  hospital    coffees    kavanaugh   robichaud
133   losses        quarter     banks   deposits     wachovia    citigroup
388   israel     holtzberg   heroism      siege       sandra  unconscious
431    write  encouragement deserving  nominated       people         find
496     city  demonstrators  zuccotti     batons   protesters         wall
499  billion       currency     swiss   emerging        sales      company
542     save          group      jobs     geurts        hunky     mcdreamy
              0             1            2         3            4   \
29   restaurant      measures      quintas      h1n1         soya
38   meningitis     princeton      vaccine       cdc     bacteria
64        leone        sierra   seychelles     ebola        match
69       dengue         fever          cdc  mosquito  hemorrhagic
75        saudi        arabia         hajj    health     khordany
304     disease  legionnaires        hotel   chicago       guests
339       polio      pakistan  vaccination      bara       khyber
343      dengue         fever        cases   florida   mosquitoes

               5           6           7         8             9           1
0  \
29   restaurants        silk   customers     virus         staff          ge
l
38      students  university   bacterial    school   dormitories   condition
s
64         virus      health      chetty   forfeit         cases    prevente
d
69        health    ministry     janeiro      said     brazilian         kha
n
75       rabeeah    pilgrims  pilgrimage       flu    guidelines           e
l
304          tub    fountain    marriott       flu         cases     symptom
s
339     campaign      tribal   kidnapped   aylward         cases       schoo
l
343       health    acquired   officials  declares  organization         dad
e

                11
29         waiters
38        graduate
64     immigration
69             rio
75   participating
304           pool
339        endemic
343          falco
               0         1           2           3           4          5   \
```

| | | | | | | |
|---|---|---|---|---|---|---|
| 89 | north | korea | council | rights | human | korean |
| 223 | afghan | parwan | agreement | detainees | bagram | facility |
| 314 | afghanistan | british | helicopter | helicopters | brown | troops |
| 380 | vahidi | jewish | morales | daia | bolivia | iranian |
| 480 | farc | santos | colombian | rebels | government | talks |
| 567 | tombs | mali | timbuktu | ansar | dine | shrines |
| 659 | flotilla | greek | israeli | hirschmann | klusmire | boats |
| 714 | ugandan | libyan | guards | mosque | gadhafi | uganda |

| | 6 | 7 | 8 | 9 | 10 \ |
|---|---|---|---|---|---|
| 89 | abuses | inquiry | darusman | disappearances | commission |
| 223 | detention | afghanistan | negotiations | base | troops |
| 314 | war | capacity | involvement | ainsworth | forces |
| 380 | bolivian | holocaust | argentina | mistake | president |
| 480 | isacson | sources | colombia | said | exploratory |
| 567 | islamists | islamist | militant | militants | mayor |
| 659 | activists | passengers | ship | crew | gaza |
| 714 | amin | museveni | kampala | dictator | fight |

| | 11 |
|---|---|
| 89 | systematic |
| 223 | partnership |
| 314 | cannot |
| 380 | community |
| 480 | group |
| 567 | area |
| 659 | boarded |
| 714 | pushed |

| | 0 | 1 | 2 | 3 | 4 | 5 \ |
|---|---|---|---|---|---|---|
| 74 | decrees | indigenous | peru | garcia | government | gass |
| 213 | edel | farm | plant | waste | loops | food |
| 274 | power | sandy | jersey | lipa | new | residents |
| 411 | zwally | ice | greenland | glaciers | nasa | sea |
| 419 | butterfly | woodland | conservation | fritillary | species | warren |
| 446 | flood | river | missouri | armstrong | feet | corps |
| 511 | domjan | carbon | global | scenario | kyoto | scenarios |
| 705 | wildfire | michigan | miles | 97 | structures | blaze |

| | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|
| 74 | laws | lands | simon | zapata | blockades | logging |
| 213 | fish | hydroponic | urban | produce | plants | zero |
| 274 | utility | dark | island | superstorm | york | christie |
| 411 | gigatons | luthcke | global | arctic | changing | warming |
| 419 | landowners | clearing | heath | forest | colonies | profit |
| 446 | bismarck | 1952 | dakota | flooding | cubic | dams |
| 511 | change | degrees | climate | temperatures | fire | 2100 |
| 705 | firefighters | fire | lake | campgrounds | firelines | motel |

| | 0 | 1 | 2 | 3 | 4 | 5 \ |
|---|---|---|---|---|---|---|
| 35 | mets | stadium | piazza | york | giants | inning |
| 144 | syrian | syria | moadamiyeh | opposition | humanitarian | said |
| 261 | lincoln | memorial | inauguration | obama | washington | bible |
| 286 | syrian | syria | syrians | lebanon | turkey | friday |
| 455 | sherpao | pakistan | mosque | blast | musharraf | attack |
| 469 | status | power | study | demeaning | roles | fast |
| 578 | xinjiang | uyghurs | chinese | police | authorities | uyghur |

```
589    syrian     daraa        avaaz    damascus    protesters  banias

            6           7            8          9             10          1
1
35      zeile        game       braves     franco          going         ne
w
144    aleppo        amos    civilians  outskirts           sana        hom
s
261     elect      family      visited      train          steps   presiden
t
286  activist    mourners     security     forces  demonstrators  processio
n
455  peshawar   elections      suicide  militants      political      bombe
r
469       low  management  researchers      times           bark      filth
y
578       han      people       region   violence          clash      unres
t
589   shaghri        city        syria       said       protests     amnest
```

# A quick look at the max and min of each column

In [5]:
```
# What could each column mean? Will help a lot in 2nd part.
columnMax = cnngov_samples.idxmax()

print(columnMax)
#print(Z[columnMax])

columnMin = cnngov_samples.idxmin()
print(columnMin)
#print(Z[columnMin])
```

```
0      (CNN) -- The oldest sitting member of Congress...
1      (CNN) -- Amid calls for greater openness at th...
2      (CNN) -- Civic-minded residents of Bismarck an...
3      TECUMSEH, Neb. - Following the death of a truc...
4      (CNN) -- Did you know that there exists an all...
5      (CNN) -- Fearing that flocks of unmanned aircr...
6      ATLANTA, Georgia (CNN) -- Steve Karas and Matt...
7      G7 Health Ministers condemn attacks on health ...
8      On Saturday, the President by Executive Order ...
9      The Substance Abuse and Mental Health Services...
10     ROME, Italy (CNN) -- The Vatican announced Thu...
11     (CNN) -- Economists might not like how quickly...
12     Washington D.C., Dec. 15, 2021 —\nThe Securiti...
13     (CNN) -- As a child, I thought growing up in a...
14     WASHINGTON (CNN) -- Pledging to take "the air ...
15     (CNN) -- On Sunday, Pope John Paul II and Pope...
16     Dallas (CNN) -- A federal judge has blocked ke...
17     (CNN) -- Authorities blocked U.S. Sen. Rand Pa...
18     Washington D.C., Dec. 15, 2021 —\nThe Securiti...
19     (CNN) -- For all the heavy baggage that she ca...
20     (CNN) -- Spain's football duopoly of Real Madr...
21     (CNN) -- Mexican authorities on Thursday conti...
22     (CNN) -- It was 1981, and Cindy Morgan was fil...
23     Today, the Office for Civil Rights (OCR) at th...
24     WASHINGTON — The Internal Revenue Service prov...
25     (CNN) -- A California company is voluntarily r...
26     Editor's Note: Alfred Liggins is chief executi...
27     WASHINGTON, DC – The U.S. Department of Labor’...
28     (CNN) -- As a child, I thought growing up in a...
29     Washington D.C., Oct. 5, 2020 —\nThe Securitie...
                             ...
482    (CNN) -- Just when it seems impossible for thi...
483    HUNTINGTON, W.V. - CSX Transportation, Inc. (C...
484    Protecting Americans’ health and well-being ha...
485    (CNN) -- It must have been a galling week for ...
486    San Francisco (CNN) -- On first glance, the ne...
487    (CNN) -- As a child, I thought growing up in a...
488    MADRID, Spain (CNN) -- Spain has won a major v...
489    WASHINGTON — The Internal Revenue Service issu...
490    (CNN) -- Boeing resumed testing Monday of its ...
491    DALLAS – The EEOC filed two lawsuits in Texas ...
492    LONDON, England (CNN) -- If climate change wer...
493    (CNN) -- The last South Korean employees left ...
494    Washington D.C., Feb. 17, 2022 —\nThe Securiti...
495    The Department of Justice today announced that...
496    HHS and DoD Statements on FDA Authorization of...
497    (CNN) -- Samsung Electronics announced plans W...
498    Washington D.C., Dec. 18, 2019 —\nFollowing a ...
499    Protecting Americans’ health and well-being ha...
500    The founder and principal operator of My Big C...
501    LONDON, England (CNN) -- If climate change wer...
502    (CNN) -- Between 1.5 trillion and 2 trillion t...
503    (CNN) -- The flow of undocumented immigrants i...
504    Port-au-Prince, Haiti (CNN) -- Raymond Thomas ...
505    (CNN) -- The scandal embroiling the empire of ...
506    Washington — The U.S. Department of Commerce’s...
```

```
507    Editor's Note: Alfred Liggins is chief executi...
508    Los Angeles (CNN) -- NASA announced Tuesday th...
509    (CNN) -- With a star-studded client list that ...
510    (CNN) -- The husband of a pregnant Texas woman...
511    (CNN) -- The bad news came via certified lette...
Length: 512, dtype: object
0      (CNN) -- This is no treasure hunt for a casket...
1      LONDON, England (CNN) -- Britons including Pri...
2      WASHINGTON – The competent authorities of the ...
3      (CNN) -- We said we would, and you said, "I do...
4      (CNN) -- The credit crisis has transformed the...
5      (CNN) -- The international draw of its star pl...
6      PHILADELPHIA – The nation's largest rail carri...
7      (CNN) -- Just when it seems impossible for thi...
8      Editor's note: Simon Johnson, a former Interna...
9      (CNN) -- Most Americans think of the Arctic as...
10     (CNN) -- Saudi Arabia has had its first death ...
11     Buchanan, New York (CNN) -- Stepping into the ...
12     (CNN) -- Boeing resumed testing Monday of its ...
13     LONDON, England (CNN) -- Getting "out of this ...
14     (CNN) -- It was a mother's worst nightmare. On...
15     HUNTINGTON, W.V. - CSX Transportation, Inc. (C...
16     LONDON, England (CNN) -- If climate change wer...
17     (CNN) -- "Grey's Anatomy" star Patrick Dempsey...
18     Today, the United States Patent and Trademark ...
19     (CNN) -- "Grey's Anatomy" star Patrick Dempsey...
20     WASHINGTON –The Internal Revenue Service today...
21     Boulder City, Nevada (CNN) -- Driving across t...
22     (CNN) -- 2014 is an election year. We know thi...
23     London (CNN) -- Never mind the traffic, car ow...
24     Statement by HHS Secretary Xavier Becerra on U...
25     (CNN) -- It hurts me to say this, but I bet wh...
26     (CNN) -- The credit crisis has transformed the...
27     (CNN) -- Fire up the griddle! Much of a huge c...
28     (CNN) -- Deadly explosions rocked parts of Dam...
29     (CNN) -- Pope Francis has made clear that "wea...
                              ...
482    MONTGOMERY, Ala. - American Family Care Inc. i...
483    (CNN) -- The first complete gorilla genome has...
484    (CNN) -- To resolve America's ongoing, bruisin...
485    WASHINGTON – As part of the Obama Administrati...
486    (CNN) -- Livestock producers suffering through...
487    (CNN) -- Spain's football duopoly of Real Madr...
488    ATLANTA - The U.S. Department of Labor's Occup...
489    (CNN) -- The Arab League's decision to impose ...
490    Washington D.C., Dec. 15, 2021 –\nThe Securiti...
491    WASHINGTON – The Internal Revenue Service has ...
492    WASHINGTON – The Internal Revenue Service toda...
493    Washington (CNN) -- The early favorite to win ...
494    (CNN) -- Mercedes-Benz will head up their own ...
495    WASHINGTON – The Internal Revenue Service prov...
496    MADRID, Spain (CNN) -- Spain has won a major v...
497    (CNN) -- A state fair's response to the uproar...
498    (CNN) -- More than 1 million cell phones in Ch...
499    (CNN) -- A Lionel Messi brace helped Barcelona...
500    (CNN) -- "She has an incredible legacy," Austr...
```

```
501    Today, the U.S. Department of Health and Human...
502    DALLAS – The EEOC filed two lawsuits in Texas ...
503    (CNN) -- "Grey's Anatomy" star Patrick Dempsey...
504    WASHINGTON – The U.S. Department of the Treasu...
505    (CNN) -- "Grey's Anatomy" star Patrick Dempsey...
506    San Diego, California (CNN)Political attacks h...
507    (CNN) -- American golfer Webb Simpson expects ...
508    (CNN) -- Miracles do happen. Like this week, w...
509    (CNN) -- Elizabeth Smart stormed out of the co...
510    WASHINGTON — The Internal Revenue Service issu...
511    On September 30, the United States Patent and ...
Length: 512, dtype: object
```

# PART 2: Embedding Classifier

## Setup and Custom Estimator Class Cell

In [6]:
```
from sklearn.base import BaseEstimator, ClassifierMixin
from sklearn.mixture import GaussianMixture
# Going to try a custom estimator based on Gaussian mixtures, based
# on promising results from a scrapped idea at the end of the notebook.

# Crucial note: Gaussian Mixtures is unsupervised, while Bayes is supervise
d.

class CustomBayesClassifier(BaseEstimator, ClassifierMixin):
    """Parameters
    ----------
    n : int
        Number of clusters total (a single label can have several cluster
s).
    covariance_type : str
        Controls the degrees of freedom in the shape of each cluster.
        Three common options: 'full' (default), 'diag', or 'spherical'.
    """
    def __init__(self, n, covariance_type = 'full', random_state=0):
        self.n = n
        self.covariance_type = covariance_type
        self.random_state = random_state

    def fit(self, X, y):
        self.classes_ = np.sort(np.unique(y))
        training_sets = [X[y == yi] for yi in self.classes_]
        self.models_ = [GaussianMixture(n=self.n, covariance_type=self.cova
riance_type, random_state=self.random_state).fit(Xi)
                        for Xi in training_sets]
        self.logpriors_ = [np.log(Xi.shape[0] / X.shape[0])
                           for Xi in training_sets]
        return self

    def predict_proba(self, X):
        logprobs = np.array([model.score_samples(X)
                             for model in self.models_]).T
        result = np.exp(logprobs + self.logpriors_)
        return result / result.sum(1, keepdims=True)

    def predict(self, X):
        return self.classes_[np.argmax(self.predict_proba(X), 1)]
```

# Proposed Methodology (in case I run out of time)

Cluster using custom Gaussian mixtures, because many bodies of text on news websites, blogs, etc can be fit in multiple clusters.

Gaussian mixtures with general elliptical "boundaries" (not hard boundaries, since these are based on Gaussian, i.e. normal curves) is one way of handling the points in the "gray area" by asking whether it more likely fit one cluster or another.

GM are generative, meaning under the hood, it views these points as having been generated by multi-dimensional Gaussians distributions.

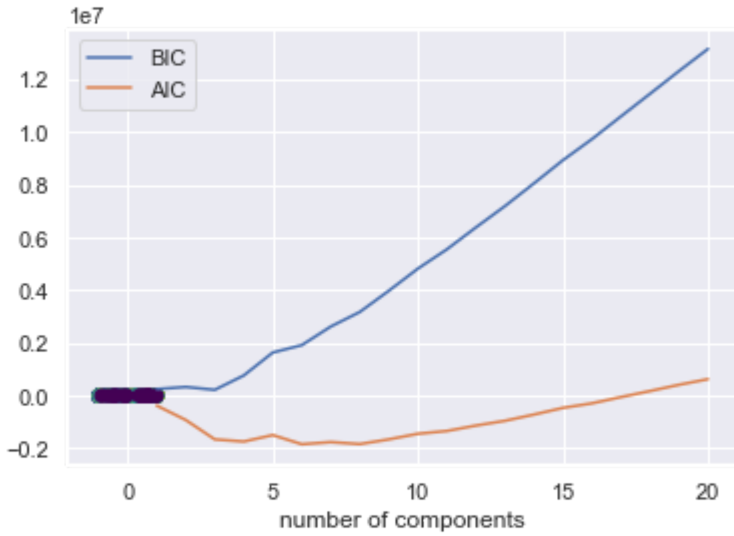
Bayesian methods require a target array, which I will generate based on the article text from the TF-IDF vector.

In [7]:
```
n_clusters = 20

customModel = CustomBayesClassifier(n= n_clusters) #use defaults for the re
st

# targets = generated target array
#customModel.fit(cnngov_samples, targets)
```

# Leftover from early in the project: Trying Gaussian Mixtures

In [8]:
```
from sklearn.mixture import GaussianMixture

gmModel = GaussianMixture(n_components=8, covariance_type="full").fit(cnngo
v_samples)
labels = gmModel.predict(cnngov_samples)
plt.scatter(cnngov_samples.values[:, 0], cnngov_samples.values[:, 1], c=lab
els, s=40, cmap='viridis')

#Giving each components' densities for this sample data
print(gmModel.predict_proba(challenge))


#Giving the Akaike and Bayesian information criteria for this model on the
sample data, to help decide the # of components.
#Try to pick # of components that minimizes either of these criteria.
n_components = np.arange(1, 21)
models = [GaussianMixture(n, covariance_type='full', random_state=0).fit(cn
ngov_samples)
          for n in n_components]

plt.plot(n_components, [m.bic(cnngov_samples) for m in models], label='BIC
')
plt.plot(n_components, [m.aic(cnngov_samples) for m in models], label='AIC
')
plt.legend(loc='best')
plt.xlabel('number of components');


# the moment of truth...
print('The most likely clusters for each article.')
challengeGM = gmModel.predict(challenge)
print(challengeGM)

# On a side note, I'm very curious to see if these generated embeddings cor
respond to anything...
# print(gmModel.sample(5))
```

```
[[0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 1. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0.]]
The most likely clusters for each article.
[1 1 4 7 3 0]
```




```
In [ ]:
```