new-york-policy

July 16, 2020

[54]:
```
import pandas as pd
import nltk
from collections import Counter

import matplotlib
import matplotlib.pyplot as plt

#C:\Users\elmsc\AppData\Roaming\nltk_data
#https://stackoverflow.com/questions/40206249/
 ↪count-of-most-popular-words-in-a-pandas-dataframe?rq=1
```

[55]:
```
policy = pd.read_csv('policy.csv')
policy.info()
policy.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4090 entries, 0 to 4089
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Date                    4088 non-null   object
 1   Location                4090 non-null   object
 2   Level                   4090 non-null   object
 3   Type                    4087 non-null   object
 4   Title/Description       4090 non-null   object
 5   Comments                1572 non-null   object
 6   Travel restrictions     13 non-null     float64
 7   State of Emergency      15 non-null     float64
 8   School Closure          21 non-null     float64
 9   Work Closure            9 non-null      float64
 10  Business closing        15 non-null     float64
 11  Gatherings              17 non-null     float64
 12  COVID/Antibody Testing  30 non-null     float64
 13  Safer at Home           12 non-null     float64
 14  Reopening               71 non-null     float64
 15  Relief/Funding/Grant    61 non-null     float64
 16  Masks                   8 non-null      float64
 17  Other                   144 non-null    float64
```

```
 18  Economics               0 non-null      float64
dtypes: float64(13), object(6)
memory usage: 607.2+ KB
```

[55]:

	Date	Location	Level	Type	Title/Description	Comments	Travel restrictions	State of Emergency	School Closure	Work Closure	Business closing	Gatherings	COVID/Antibody Testing	Safer at Home	Reopening	Relief/Funding/Grant	Masks	Other	Economics
0	3/6/2020	Alabama	State	Announcement	Formation of COVID-19 Task Force	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0	NaN
1	3/10/2020	Alabama	State	Announcement	State employees must notify of recent travel	if they have traveled in areas affected by cor…	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	3/13/2020	Alabama	State	Issued	State of Emergency	NaN	NaN	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	3/14/2020	Alabama	State	Ordered	Public school closure	for 2.5 weeks	NaN	NaN	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	3/15/2020	Alabama	State	Authorization	Changing work schedules	directors of all state agencies change schedul…	NaN	NaN	NaN	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

[56]:
```
policy.fillna('none',inplace=True)
policy.head()
```

[56]:

	Date	Location	Level	Type	Title/Description	Comments	Travel restrictions	State of Emergency	School Closure	Work Closure	Business closing	Gatherings	COVID/Antibody Testing	Safer at Home	Reopening	Relief/Funding/Grant	Masks	Other	Economics
0	3/6/2020	Alabama	State	Announcement	Formation of COVID-19 Task Force	none	none	none	none	none	none	none	none	none	none	none	none	1	none
1	3/10/2020	Alabama	State	Announcement	State employees must notify of recent travel	if they have traveled in areas affected by cor…	1	none	none	none	none	none	none	none	none	none	none	none	none
2	3/13/2020	Alabama	State	Issued	State of Emergency	none	none	1	none	none	none	none	none	none	none	none	none	none	none
3	3/14/2020	Alabama	State	Ordered	Public school closure	for 2.5 weeks	none	none	1	none	none	none	none	none	none	none	none	none	none
4	3/15/2020	Alabama	State	Authorization	Changing work schedules	directors of all state agencies change schedul…	none	none	none	1	none	none	none	none	none	none	none	none	none

[57]:
```
ny = policy[policy['Location'] == 'New York']
ny.head()
```

[57]:
```
          Date  Location  Level          Type  \
2321  3/2/2020  New York  State  Announcement
2322  3/2/2020  New York  State  Announcement
2323  3/2/2020  New York  State   Declaration
2324  3/3/2020  New York  State        Signed
2325  3/3/2020  New York  State  Announcement

                                    Title/Description   \
2321  School & Public Transportation cleaning protoc…
2322                      Expand Testing to 1000 per day
2323  Governer directs health insurers to waive cost…
2324  $40 million emergency funds for coronavirus re…
2325  Governer will amend Paid Sick Leave budget to …

                                              Comments Travel restrictions  \
2321                                              none                none
2322  Wadsworth Center provides hospitals instructio…                none
2323  The State Department will require health insur…                none
2324  Governer Cuomo signed an emergency management …                none
2325  Governer Cuomo announced he will amend Paid Si…                none

     State of Emergency School Closure  Work Closure Business closing  \
2321               none            none         none             none
2322               none            none         none             none
2323               none            none         none             none
2324               none            none         none             none
2325               none            none         none             none

     Gatherings COVID/Antibody Testing Safer at Home Reopening  \
2321       none                   none          none      none
2322       none                   none          none      none
2323       none                   none          none      none
2324       none                   none          none      none
2325       none                   none          none      none

     Relief/Funding/Grant Masks Other Economics
2321                 none  none  none      none
2322                 none  none  none      none
2323                 none  none  none      none
2324                 none  none  none      none
2325                 none  none  none      none
```

[59]:
```
top_N = 10

stopwords = nltk.corpus.stopwords.words('english')

print(' '.join(stopwords))
```

```
i me my myself we our ours ourselves you you're you've you'll you'd your yours
yourself yourselves he him his himself she she's her hers herself it it's its
itself they them their theirs themselves what which who whom this that that'll
these those am is are was were be been being have has had having do does did
doing a an the and but if or because as until while of at by for with about
against between into through during before after above below to from up down in
out on off over under again further then once here there when where why how all
any both each few more most other some such no nor not only own same so than too
very s t can will just don don't should should've now d ll m o re ve y ain aren
aren't couldn couldn't didn didn't doesn doesn't hadn hadn't hasn hasn't haven
haven't isn isn't ma mightn mightn't mustn mustn't needn needn't shan shan't
shouldn shouldn't wasn wasn't weren weren't won won't wouldn wouldn't
```

[60]:
```
# RegEx for stopwords
RE_stopwords = r'\b(?:{},;)\b'.format('|'.join(stopwords))

# replace '|'-->' ' and drop all stopwords
words = (ny.Comments
           .str.lower()
           .replace([r'\|', RE_stopwords], [' ', ''], regex=True)
           .str.cat(sep=' ')
           .split()
)
```

[61]:
```
# generate DF out of Counter
rslt = pd.DataFrame(Counter(words).most_common(top_N),
                    columns=['Word', 'Frequency']).set_index('Word')

print('all frequencies, not including stopwords: ')

print('=' * 60)
print(rslt)
print('=' * 60)
```

```
All frequencies, not including stopwords:
============================================================
           Frequency
Word
none             208
state             15
health            11
new               11
testing            8
cuomo              7
governer           6
emergency          6
hospital           6
individual         6
```

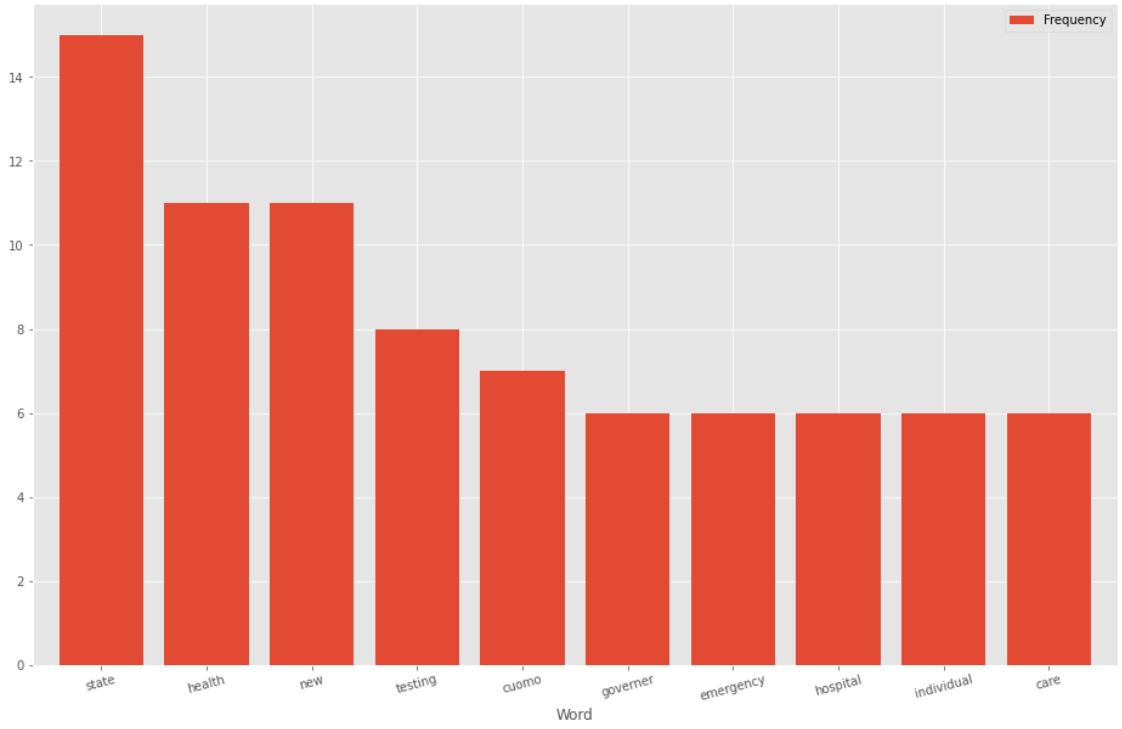
```
============================================================
```

[63]:
```
words = (ny[ny['Comments']!='none'].Comments
           .str.lower()
           .replace([r'\|', RE_stopwords], [' ', ''], regex=True)
           .str.cat(sep=' ')
           .split()
)

rslt = pd.DataFrame(Counter(words).most_common(top_N),
                    columns=['Word', 'Frequency']).set_index('Word')

print('filtered frequencies, not including stopwords: ')

print('=' * 60)
print(rslt)
print('=' * 60)
```

```
filtered frequencies, not including stopwords:
============================================================
           Frequency
Word
state             15
health            11
new               11
testing            8
cuomo              7
governer           6
emergency          6
hospital           6
individual         6
care               6
============================================================
```

[64]:
```
# plot
rslt.plot.bar(rot=15, figsize=(16,10), width=0.8)
```

[64]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2246a023508>
```



[ ]:
```
!jupyter nbconvert --output-dir='output/' --to pdf new-york-policy.ipynb
!jupyter nbconvert --output-dir='output/' --to markdown new-york-policy.ipynb
!jupyter nbconvert --output-dir='output/' --to html new-york-policy.ipynb
!jupyter nbconvert --output-dir='output/' --to python new-york-policy.ipynb
```